

THE EVOLUTION OF POPULATION III STARS:
A STUDY OF SELECTED MODELS WITH REGARD TO
THE EARLY ENRICHMENT OF INTERSTELLAR
MEDIUM

Peter D. Thacker

A Thesis Submitted for the Degree of MPhil
at the
University of St Andrews



1996

Full metadata for this item is available in
St Andrews Research Repository
at:
<http://research-repository.st-andrews.ac.uk/>

Please use this identifier to cite or link to this item:
<http://hdl.handle.net/10023/14308>

This item is protected by original copyright

The Evolution of Population III Stars

A Study of Selected Models With Regard to the Early
Enrichment of the Interstellar Medium

by
Peter D. Thacker



A Thesis submitted for the Degree of Master of Philosophy
at the University of St. Andrews.

June, 1995

ProQuest Number: 10166971

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10166971

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Th
B824

Abstract

Modelling of the evolution of $15M_{\text{sun}}$, $10M_{\text{sun}}$, $5M_{\text{sun}}$ and $2M_{\text{sun}}$ stars has been carried out at three different initial compositions that could be thought of as 'population III' - $Z=10^{-10}$, 10^{-12} and zero. The effects of mass loss due to a stellar wind have been included in the modelling, the mass loss rates taken from an empirical formula by Nieuwenhuijzen and de Jager (1990). An original and amended form of a FORTRAN 77 quasi-hydrostatic evolutionary code, written by Dr. T. R. Carson and modified by the author of this work, were used to create and evolve the models. The aims of the evolutionary modelling were to confirm that the sensitivity of stellar parameters to the initial metallicity of a star continues to a lower level of Z than has previously been thought, and to provide some evidence that nucleosynthesis and subsequent mass loss in a population III is a plausible mechanism for the prompt enrichment of the early interstellar medium.

We find that there is sensitivity of stellar parameters to the initial metallicity Z in the range $0 \leq Z \leq 10^{-10}$, but this is less apparent in models of intermediate or low masses. During the evolution of the models, no significant loss of enriched material occurs, due to low rates of mass loss and the absence of any 'dredge-up' of enriched material. The results of the modelling were connected with bare-core studies of helium stars to determine the amount and composition of the material that would be returned to the interstellar medium during the endstates of the models. Functions for the rates of enrichment due to a population III and the distribution of mass in a population III were determined and their implications examined.

From our results, we find that the enrichment of the interstellar medium due to population III stars must have come about entirely due to rapid mass loss during endstates, such as supernovae, as opposed to a slower rate of mass loss during the lifetime of the stars. We also find that our results indicate that the level of helium enrichment from a population III would be negligible compared to that due to cosmological mechanisms. Our distribution functions for a population III indicate that massive stars are very much fewer by number that would be expected for stars in our epoch. Overall, we can consider that a primordial population III can be considered a plausible mechanism for a prompt enrichment of metals in the early universe, but not for the prompt enrichment of helium.

To Ellen, for being faithful
and for being my friend.

I would like to express my gratitude to Dr. T. R. Carson for the time and guidance he has given me (and for putting up with my approach to life). The numerous discussions we have had in the process of completing this work have increased my understanding of the subject as a whole, and I am indebted. I would also like to thank the Department of Physics and Astronomy and the University of St. Andrews as a whole for the support they have shown over the whole of my academic career to date.

Certificate

I hereby certify that the candidate has fulfilled the conditions and regulations appropriate to the Degree of Master of Philosophy (*M.Phil.*) of the University of St. Andrews and that he is qualified to submit this thesis in application for that degree.

T. R. Carson
(Supervisor)

Declaration

I, Peter D. Thacker, hereby certify that this thesis, which is approximately 40,000 words in length, has been written by me, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student and candidate for the degree of M.Phil. in October 1994; the higher study of which this is a record was carried out at the University of St. Andrews between October 1994 and June 1995.

Peter D. Thacker

In submitting this thesis to the University of St. Andrews I understand that I am giving permission for it to be made available for use in accordance with regulations of the University Library in force at the time; subject to any copyright vested in the work not being affected thereby. I also understand that the title and abstract will be published and that a copy of the work may be made and supplied to any *bona fide* library or research worker.

Peter D. Thacker

Contents

Chapter One - Introduction

1.1 What Is Population III?.....	2
1.2 So Why A Population III?.....	2
1.3 The Properties of Population III Stars.....	3
1.4 The Formation of Population III Stars.....	5
1.5 Evolutionary Studies of Population III Stars.....	6
1.6 Mass Loss in Population III Stars.....	6
1.7 A Note on Computational Physics.....	7

Chapter Two - The Theory of Stellar Structure and Evolution

2.1 The Equations of Stellar Structure.....	9
2.2 The Temperature Gradient ∇	11
2.3 Thermodynamic Properties.....	12
2.4 The Equation of State.....	12
2.5 The Opacity κ	15
2.6 Energy Generation and Reaction Rates.....	19
2.7 The Hydrostatic Equation.....	22
2.8 Boundary Conditions.....	23
2.9 The Central Behaviour of the Functions.....	25

Chapter Three - The Method of Solution

3.1 Shooting and Fitting Methods.....	28
3.2 The Henyey Method.....	28
3.3 A Sample Linearisation.....	32

3.4 The Solution of the Linearised Equations.....	33
3.5 Some Comments on the Henyey Method.....	39

Chapter Four - The TRC Evolutionary Code

4.1 An Introduction to the Code.....	42
4.2 The Variables Used in Implementing the Henyey Method.....	42
4.3 The Equation of State.....	43
4.4 Thermonuclear Reactions.....	44
4.5 Opacities.....	48
4.6 Convection.....	49
4.7 The Use of the Hydrostatic Equation.....	49
4.8 Zoning and Mass Intervals.....	50
4.9 The Calculation of Mass Loss Rates.....	53
4.10 Composition Changes Due to Mass Loss.....	54
4.11 The Timestep for Evolution.....	55
4.12 Limits on the Timestep.....	56
4.13 Input Requirements of the Code.....	57
4.14 Output From the Code.....	61
4.15 Sample Output for a Single Timestep.....	65
4.16 The Structure of the Code.....	65
4.17 The Amended Version of the Code.....	70

Chapter Five - Checking the TRC Code

5.1 The Consistency of the Code.....	72
5.2 Errors Introduced Due to the Choice of Timestep.....	72
5.3 The Choice of Temperature Gradient, ∇ , ∇_{ad} or ∇_{rad}	73

5.4 The Behaviour of the Rate of Mass Loss With Metallicity.....	74
--	----

Chapter Six - Results

6.1 The Aims of the Work.....	77
6.2 The Choice of Initial Mass and Metallicity.....	77
6.3 Other Parameters.....	78
6.4 Tables of Properties for Zero-Age and Evolved Models.....	78
6.5 Differences in Models Due to the Amendment of the Code.....	83
6.6 Differences in Models With Initial Metallicity.....	92
6.7 Other Results.....	117
6.8 Discussion and Conclusions.....	117

Chapter Seven - Chemical Evolution by a Population III

7.1 The Processing of Material by a Population III.....	123
7.2 The Role of Helium Star Studies.....	123
7.3 Obtaining the 'Final' Core Masses for the Lower Mass Models.....	123
7.4 Core Data for the Lower Mass Models.....	124
7.5 Stellar Remnants.....	125
7.6 Supernova Events and Endstates.....	126
7.7 The Composition of the Ejecta by Mass and Initial Metallicity.....	127
7.8 Functions for Z_{ej} and Y_{ej} With Initial Mass.....	128
7.9 The Rates of Increase of Y and Z From Nuclear Processing.....	131
7.10 The Rate at Which Material Is Deposited Into Remnants.....	133
7.11 The Distribution and Initial Mass Functions for a Population III.....	134
7.12 Helium Enrichment Due to a Population III.....	143
7.13 The Mass Left As Remnants by a Population III.....	144
7.14 Discussion and Conclusions.....	147

Chapter Eight - Summary

8.1 Summary.....	151
------------------	-----

Appendix A - A Discussion of the Functions Presented in Chapter Seven

A.1 A Consideration of the Distribution Functions.....	153
A.2 The Distribution of Stars with Initial Mass.....	153
A.3 Describing the Nucleosynthesis Rates with Mass.....	155
A.4 The Fraction of Material Ejected.....	158
A.5 The Fraction of Material Forming Stars.....	159
A.6 The Average Rates of Nucleosynthesis.....	159

Appendix B - Composite Diagrams of Evolution for the Amended Code

B.1 H-R Diagrams for the Amended Code.....	162
B.2 Convective Core Data for the Amended Code.....	165
B.3 Hydrogen-Depleted Core Data for the Amended Code.....	167

Appendix C - Diagrams of Evolution for All Models

C.1 H-R Diagrams, Initial Code.....	172
C.2 H-R Diagrams, Amended Code.....	178
C.3 Convective Core Data, Initial Code.....	184
C.4 Convective Core Data, Amended Code.....	190
C.5 Hydrogen-Depleted Core Data, Initial Code.....	196
C.6 Hydrogen-Depleted Core Data, Amended Code.....	201
C.7 Mass Loss, Luminosity and Temperature Data, Initial Code.....	207
C.8 Mass Loss, Luminosity and Temperature Data, Amended Code.....	211

Appendix D - Listings of the TRC Code and Others

D.1 The TRC Code.....	217
-----------------------	-----

D.2 Other Codes.....297

References

Chapter One

Introduction

1.1 What Is Population III?

It is well known that all observed stars can be divided into two broad groups according to the metallicity of these stars. These groups are known as population I and population II, with the latter group having lower metallicities and being assumed to be older. So what exactly constitutes a population III star? This is varyingly defined, depending on the source. From an observational viewpoint, various authors have classed a population III star as typically exhibiting $[\text{Fe}/\text{H}] < -3$. For the purposes of this work, however, we will use the more cosmologically relevant definition of a star with a zero metallicity, or a near-zero metallicity. The near-zero metallicities used in this work reflect the proportion of metals thought to have been created by cosmological mechanisms, before the formation of the first stars. We give metals as having a proportion $Z = 10^{-10} - 10^{-12}$ of the elements synthesised, the rest being hydrogen and helium in the approximate proportions of $X = 0.75$ and $Y = 0.25$ respectively. The reader may like to refer to the work by Applegate, Hogan and Scherrer (1988), who give an upper limit of $Z = 10^{-10}$ for the metals created by cosmological processes.

There is certainly little or no evidence for stars of such low metallicities to exist today, although some observations claim values of $[\text{Fe}/\text{H}] = -4$ or even lower. For an example of such a star, see Carney and Peterson (1981). In fact, it has been observed for some time that there appears to be a lack of low-metal stars in our galaxy, even for $[\text{Fe}/\text{H}] < -1$. This is known as the 'G-dwarf problem', even though more low-metal stars are being discovered as observational technology improves our ability to search for them. See, for example, Bond (1981) for a discussion of the 'G-dwarf problem'. In addition, see Zinn (1980), Harris and Canterna (1979) or Carney (1979) for observational results concentrating on the values of $[\text{Fe}/\text{H}]$ in metal-poor stars, both in the galactic halo and local subdwarfs.

1.2 So Why A Population III?

So why do we need to consider a population III? A very simple model of stellar chemical evolution, as demonstrated by Bond (1981), can be used to answer this question. We start with a closed region of gas, of zero metallicity, and allow stars to form and evolve. We will find that the distribution of metallicities in these stars won't even approximately match the metallicities observed in our epoch. However, if we use the same model, but allow a small but non-zero initial metallicity, we find that the final distribution matches well with the observed distribution. This would suggest that the oldest population II stars were forming in an environment with a low but significant metallicity. This level of metallicity cannot be generated by cosmological mechanisms under any of the currently accepted theories, and so another mechanism is needed. The mechanism considered in this work is stellar nucleosynthesis, and later

mass loss of enriched material, in a population III. This population III is presumed to have enriched the interstellar medium prior to the formation of a population II.

It is worth noting that in addition, the existence of a population III has been variously invoked by many authors to explain the microwave excess in the cosmic background radiation, the background radiation as a whole, the 'missing mass' in part or whole, early dust formation in the universe and as the mechanism whereby regions of denser gas were created for the formation of population II - leading to the present structure of the universe. A discussion of these theories would not be appropriate for this work, however, and the interested reader should examine the excellent paper by Carr, Bond and Arnett (1984). Further discussions of these topics can be found in work by Bertotti, Carr and Rees (1983), Hogan (1983), Karimabadi and Blitz (1984), Kolesnik (1978), Lacey and Field (1988), Matsumoto, et al (1988), Rees (1978), and Vainer (1990). The properties and evolution of population III stars are of great interest if we are to justify their existence on the basis of an enrichment of the interstellar medium, or if we are to draw any conclusions as to a possible mass function. In addition, we must consider what mechanism was largely responsible for the loss of enriched material to the interstellar medium.

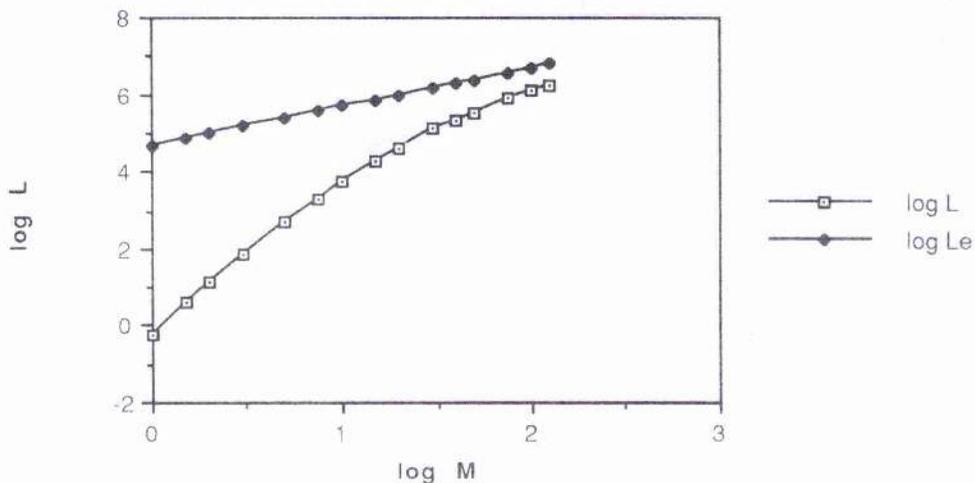
1.3 The Properties of Population III Stars

It can be considered that there are four classes of star, grouped loosely by mass; supermassive objects with masses $M > 10^5 M_{\text{sun}}$, very massive objects with a mass range of $10^2 M_{\text{sun}} < M < 10^5 M_{\text{sun}}$, massive objects with a mass range of $4 M_{\text{sun}} < M < 10^2 M_{\text{sun}}$, and low mass objects, with masses $M < 4 M_{\text{sun}}$. These are the definitions proposed by Carr, Bond and Arnett (1984). The former two classes are not thought to exist in the present epoch, while the latter two classes are the supergiants, giants, dwarfs and subdwarfs that make up populations I and II. It is, however, worth considering the observations of 30 Doradus by Cassinelli, Mathis and Savage (1981), and of η Carinae by Andriessse, Donn and Viotti (1978) in this respect. Both stars could be very massive objects. The supermassive and very massive objects have been invoked in past works, Carr, Bond and Arnett (1984) for example, to class the mass functions which some authors have proposed for population III stars, in most cases to ensure that the desired endstates are produced in a very short time. It is desirable in most theories requiring early metal enrichment of the interstellar medium, that the enrichment occurred on a short timescale, compared to the lifetimes of the population II stars that formed from the enriched medium. It should be noted that the oldest observed population II stars have estimated lifetimes comparable to the age of the universe. Larger population III stars are needed in any such theory because they will evolve faster, and enrich the interstellar medium more quickly. However, these larger objects present many difficulties when modelling is attempted,

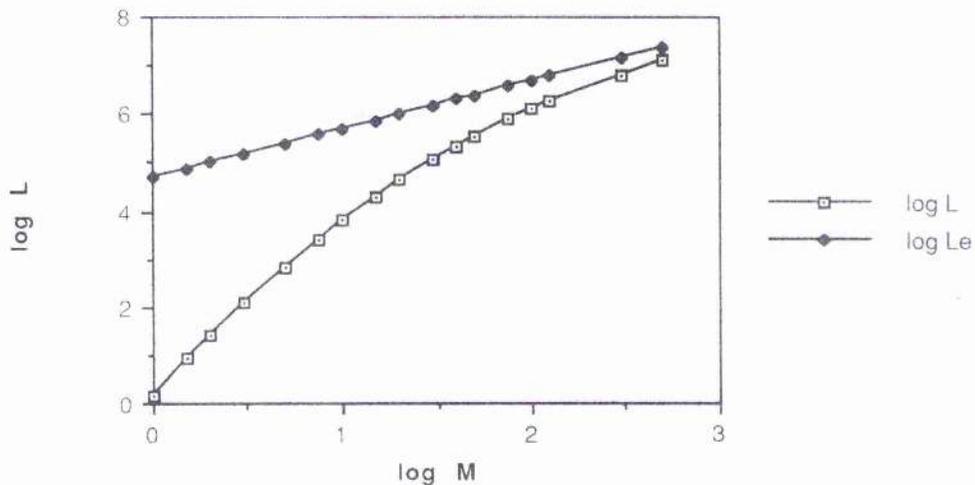
including dynamical and relativistic instability. In this work, only low mass and massive population III stars will be examined.

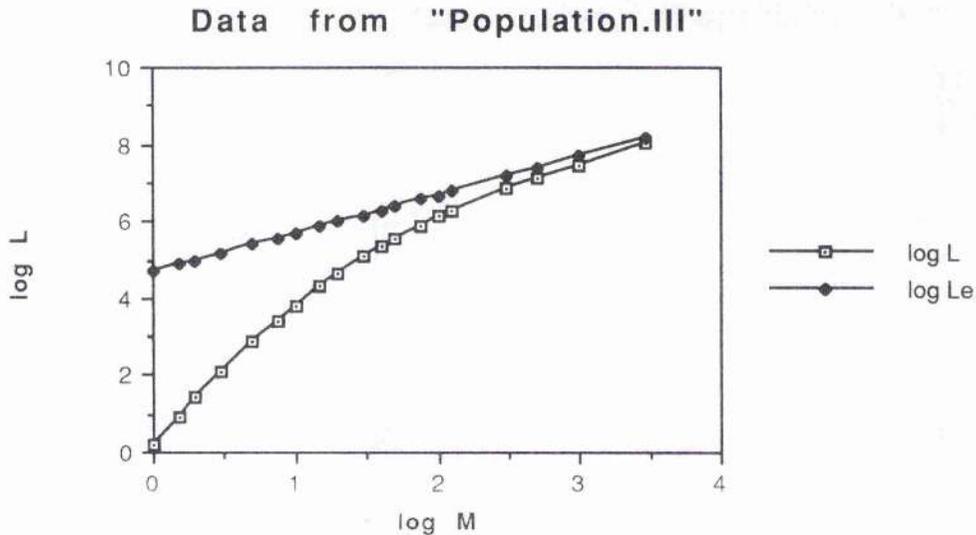
The properties of main sequence population III stars, in comparison to main sequence population I or II stars, has been examined in unpublished work by the author of this work and by Milligan and Carson (1992). In addition, other authors have published studies of main-sequence population III stars, such as those by Klapp (1983), Ober, El Eid and Fricke (1982) and Wagner (1974). We find that compared to population I or II stars at a given mass, population III stars are more luminous, and approach more closely their Eddington limit for luminosity. This can be seen in the diagrams below, reproduced from the authors undergraduate dissertation (1993), which show luminosities for main sequence stars at varying masses. The higher line in each diagram represents the Eddington luminosities at these masses.

Data from "Population.I"



Data from "Population.II"





The characteristic values of Z for the stars represented in the diagrams given in this section are $Z_{\text{popI}}=2 \times 10^{-2}$, $Z_{\text{popII}}=2 \times 10^{-4}$ and $Z_{\text{popIII}}=0$.

1.4 The Formation of Population III Stars

The formation of supermassive or very massive objects is still open to debate. The characteristic mass of protostars formed from a cloud of gas collapsing under self-gravitation depends on the cooling mechanisms in the gas, and on the opacity of the forming protostars. The theory behind this is known as fragmentation theory, and is still fairly unrefined. See, for example, Silk (1977) for a basic examination of the theory. The more efficient the cooling processes, the larger the characteristic mass of the stars formed, as cooling reduces the pressure inside the collapsing cloud, and thus raises the mass a protostar can attain without itself fragmenting further. As opacity assists the generation of pressure, then the sooner opacity becomes appreciable, and the greater it is, the smaller the characteristic mass of the protostars. Important cooling mechanisms, in order of increasing efficiency, include Compton scattering of hot electrons by colder photons, hydrogen and helium cooling (especially cooling via Lyman- α excitations), heavy atom cooling, and dust grain cooling. The latter two mechanisms are not significant in material of population III composition, which thus has a less efficient selection of cooling mechanisms. However, this material will have a lower opacity than population II or I compositions, due to the lack of metals. In past works invoking a very massive or supermassive characteristic mass for population III, it has been assumed that the effects of lowered opacity outweigh those due to less efficient cooling in determining the characteristic mass of population III - thus the protostars will have a higher characteristic mass than those of population I or II composition.

For a more detailed discussion of the formation of population III stars and their mass function, see Lahav (1986), Silk (1977), Stahler, Palla

and Salpeter (1986), or Vainer, Chuvenkov and Shchekinov (1986). It should be noted that these authors come to somewhat different conclusions, depending on their approaches to the problem.

1.5 Evolutionary Studies of Population III Stars

The evolutionary behaviour of population III stars has been examined by numerous authors. Recently, for example, Cassisi and Castellani (1993) modelled the evolution of population III stars at numerous masses. They took an initial metallicity of $Z=10^{-10}$ and followed the evolution of these stars up to either the helium flash, carbon flash or the collapse of the carbon-oxygen core. D'Antona (1982) briefly described the evolution of a solar-mass zero-metal population III star as far as the helium flash. Guenther and Demarque (1983) evolved a similar zero-metal star. Castellani, Chieffi and Tornambé (1983) and Chieffi and Tornambé (1984) have studied the evolution of several zero-metal stars between them. In general, these authors and others have concluded that stars of population III composition exhibit significantly different evolutionary behaviour in comparison to stars of population I or II. For example, population III stars begin helium-burning during the core hydrogen-burning stage. Population I and II stars, of course, do not. Population III stars have also been noted to have smaller convective cores, and smaller hydrogen-depleted cores than population I or II stars. Chieffi and Tornambé (1984) found that the first 'dredge-up' of enriched material to the surface of a star during helium-burning does not occur for stars of population III composition.

It has been thought that the sensitivity to changes in the metallicity of a star vanished for $Z < 10^{-5}$ - 10^{-6} . See, for example, comments on this by Wagner (1974). In fact, Cassisi and Castellani (1993) have shown that some stellar parameters, and the internal structure of stars, remain sensitive to changes in metallicity at least as low as $Z=10^{-10}$.

1.6 Mass Loss In Population III Stars

The problem of mass loss seems not to have been addressed by any previous authors in connection with the evolutionary development of population III stars. In many ways this is not surprising - there are few, if any, comprehensive theories of mass loss available to apply to any model that is not related in some way to observed stars. However, consideration of mass loss would be desirable if we are interested in the way that population III stars would have enriched the interstellar medium. For a general overview of the effects of mass loss upon stellar evolution, the reader may like to refer to Chiosi (1980).

There are several ways in which a star can lose a significant proportion of its mass - due to its stellar wind over its entire lifetime, overflowing its Roche lobe in a binary or multiple system, or a supernova event late in its life, to name the most common mechanisms. Much has

been written on the cataclysmic endstates of population III stars, and this work will not go into this area in great detail. Similarly, this work will not consider binary systems, as we are more interested in the enrichment of the interstellar medium, as opposed to mass transfer between stars. In general, the supermassive or very massive population III stars, if they do not fragment, will either collapse to form black holes, or become supernovae, and there is some uncertainty in any description of their evolution to date. Smaller population III stars will have much the same evolutionary behaviour as comparable stars of population I or II, and it is these smaller stars this work will concentrate on. The mechanism for mass loss, and the enrichment of the interstellar medium, that will initially be considered in this work is that provided by a radiation-driven stellar wind.

1.7 A Note on Computational Physics

When modelling aspects of stellar physics, we should remember that for realistic physical situations, described by differential equations, it is usually the case that no analytical or exact solution of the equations is possible. Solutions thus depend upon numerical methods. Fortunately, these numerical methods have become ever more refined and easy to carry out in practice as faster computers have been developed alongside more sophisticated procedures. In this work, the most generally utilised numerical procedure is used, that first put forward by L. G. Henyey. The reader may like to refer to Henyey, Vardya and Bodenheimer (1965) for an account of this method.

Chapter Two

The Theory of Stellar Structure and Evolution

2.1 The Equations of Stellar Structure

Before proceeding with a description of the code used in this work to model the evolution of population III stars, it would first be appropriate to describe the theory underlying the study of stellar structure and evolution. This chapter will discuss the equations and boundary conditions describing the problem, while chapter three will outline the numerical methods used to solve these equations. Also in the next two chapters the reader will also find the physics specific to this work that might not be found in a more general overview of the theory of stellar evolution. The reader might like to refer to Kippenhahn and Weigert (1990), or to Kippenhahn, Weigert and Hofmeister (1967).

In this work we are assuming that stars are spherically symmetric in structure, so that there is only one independent variable describing position in the star. This spherical symmetry is assumed to extend to any convection considered. While convection normally occurs in a very asymmetric manner, we assume that the effects of convection average out over time. In other words, the timescale associated with any non-radial structure of convection zones is much smaller than the timescale associated with its radial structure, or timescale associated with the evolution of the star. So, under these assumptions, we can consider convection to be occurring in spherical shells in a star.

The equations of stellar structure are well known, and for a spherically symmetric star can be expressed as the follows. Firstly, the equation of continuity given by

$$\frac{\partial r}{\partial m} = \frac{1}{4\pi r^2 \rho'} \quad (2.1-1)$$

where m is the mass contained within radius r , and ρ is the density at radius r . Secondly, the hydrodynamic equation given by

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4} - \frac{1}{4\pi r^2} \frac{\partial^2 r}{\partial t^2} \quad , \quad (2.1-2)$$

where P is the pressure, G the gravitational constant and t the time. Thirdly, we have the equation of energy generation, or thermal balance

$$\frac{\partial l}{\partial m} = \epsilon_n - \epsilon_\nu - c_p \frac{\partial T}{\partial t} + \frac{\delta}{\rho} \frac{\partial P}{\partial t} \quad , \quad (2.1-3)$$

where ϵ_n and ϵ_ν are the energy production rate and rate of energy 'loss' due to neutrinos. c_p is the specific heat capacity at constant pressure, T the

temperature and P the pressure. δ is a thermodynamic property defined later in this chapter. Next, the equation of energy transport

$$\frac{\partial T}{\partial m} = -\frac{GmT}{4\pi r^4 P} \nabla \quad , \quad (2.1-4)$$

where ∇ is the temperature gradient. Lastly, we have the equations representing the change in mass fraction of the elements present, in terms of the reaction rates involved :

$$\frac{\partial X_i}{\partial t} = \frac{m_i}{\rho} \left[\sum_j r_{ji} - \sum_k r_{ik} \right] \quad , \quad i=1, \dots, I, \quad (2.1-5)$$

Here, (2.1-5) consists of a set of I equations for the change of the mass fractions of elements X_i , each having an atomic mass of m_i , where I denotes the number of elements considered to be present in the star. r_{ij} are the nuclear reaction rates for elements i and j . In equation (2.1-3), $\delta = -(\partial \ln \rho / \partial \ln T)_P$, and the code used can consider either of the two alternative forms for (2.1-3), given by equations (4.13-1, 2) in chapter four. These equations contain more easily calculated thermodynamic properties, and are thus more convenient to use. It should be noted that equation (2.1-5) can only be considered to be true if there is no convection.

Equations (2.1-1, 5) contain functions describing the properties of the matter within a star, such as ρ , ϵ_n , ϵ_v , δ , κ , c_p , ∇ and the reaction rates r_{ij} . These are all known functions of P , T , and the chemical composition described by the functions X_i . So we have an equation of state

$$\rho = \rho(P, T, X_i) \quad , \quad (2.1-6)$$

and equations for the other thermodynamic properties of the stellar matter

$$c_p = c_p(P, T, X_i) \quad , \quad (2.1-7)$$

$$\delta = \delta(P, T, X_i) \quad , \quad (2.1-8)$$

$$\nabla = \nabla(P, T, X_i) \quad , \quad (2.1-9)$$

in addition, the Rosseland mean opacity, needed to calculate ∇ ,

$$\kappa = \kappa(P, T, X_i) \quad . \quad (2.1-10)$$

Lastly, we have the nuclear reaction rates, energy production and energy loss via neutrinos :

$$r_{jk} = r_{jk}(P, T, X_i) \quad , \quad (2.1-11)$$

$$\varepsilon_n = \varepsilon_n(P, T, X_i) \quad , \quad (2.1-12)$$

$$\varepsilon_\nu = \varepsilon_\nu(P, T, X_i) \quad , \quad (2.1-13)$$

The above general equations are described in more detail in the remaining sections of this chapter.

2.2 The Temperature Gradient ∇

In equation (2.1-4), the value of the temperature gradient, ∇ , depends on which physical approximations we are making. If we assume the temperature gradient to be entirely adiabatic, then ∇ becomes ∇_{ad} , given by :

$$\nabla_{ad} = \left(\frac{\partial \ln P}{\partial \ln T} \right)_s = \frac{P}{T} \left(\frac{\partial P}{\partial T} \right) = \frac{P\delta}{T\rho c_p} \quad , \quad (2.2-1)$$

If energy transport is due to radiation and conduction only, however, then ∇ is replaced by the radiative temperature gradient ∇_{rad} , which is given by :

$$\nabla_{rad} = \frac{3}{16\pi acG} \frac{\kappa_l P}{mT^4} \quad , \quad (2.2-2)$$

where a is the radiation-density constant, and c is the speed of light in a vacuum. If we are considering radiative and convective energy transport, then we calculate this value for ∇ in the normal way, by obtaining the single real solution of the cubic equation for the parameter ξ ;

$$(\xi - U)^3 + 8U(\xi^2 - U^2 - W) = 0 \quad , \quad (2.2-3)$$

where the parameter U is given by ;

$$U = \frac{3acT^3}{c_p \rho^2 \kappa_m^2} \sqrt{\frac{8H_p}{g\delta}} \quad , \quad (2.2-4)$$

and the parameter W by ;

$$W = \nabla_{\text{rad}} - \nabla_{\text{ad}} \quad (2.2-5)$$

Once equation (2.2-3) has been solved for the parameter ξ , then a value for ∇ can be found from ;

$$\xi^2 = \nabla - \nabla_{\text{ad}} + U^2 \quad , \quad (2.2-6)$$

2.3 Thermodynamic Properties

We may take the standard thermodynamic definition for δ given below :

$$\delta = - \left(\frac{\partial \ln \rho}{\partial \ln T} \right)_P \quad , \quad (2.3-1)$$

a logarithmic derivative. The form taken by c_P is equally straightforward, and can be seen from the basic relation

$$c_P = c_v + \frac{P\delta^2}{\rho T \alpha} \quad , \quad (2.3-2)$$

where c_v , the specific heat at constant volume is given by

$$c_v = \frac{u}{T} \left(\frac{\partial \ln u}{\partial \ln T} \right)_V \quad . \quad (2.3-3)$$

Here, u is the internal energy per unit mass. The thermodynamic parameter α is given by

$$\alpha = \left(\frac{\partial \ln \rho}{\partial \ln P} \right)_T \quad . \quad (2.3-4)$$

2.4 The Equation of State

The equation of state is perhaps the most complicated of the general relations (2.1-6, 13) to define. The total pressure P in any stellar material can be taken to be the sum of the radiation pressure P_{rad} , due to photons, and the gas pressure P_{gas} , due to material particles. The latter can be given as the sum of the electron pressure P_e and the ion pressure P_i . So we have that

$$P = P_{\text{gas}} + P_{\text{rad}} = P_e + P_i + P_{\text{rad}} \quad . \quad (2.4-1)$$

If we assume that the radiation in a star is approximately that of a black body, then we can simply give the radiation pressure P_{rad} as being

$$P_{\text{rad}} = \frac{1}{3} U_{\text{rad}} = \frac{a}{3} T^4 \quad , \quad (2.4-2)$$

where U_{rad} is the radiation energy density. The pressure P_i due to the ion gas is equally straightforward to define. If we assume the ion gas to be ideal in nature, then we can use the ideal gas equation

$$P_i = nkT = \frac{\mathfrak{R}}{\mu_0} \rho T \quad , \quad (2.4-3)$$

where \mathfrak{R} is the universal gas constant, and μ_0 is the mean molecular weight for the material.

It is the electron pressure that proves to be time-consuming to calculate, as it is influenced by both the degree of ionisation in the material, and by degeneracy effects. However, the physics describing these effects is well known, and we will not dwell upon it overmuch. In the general case, then, we have elements i with mass fractions X_i , charge numbers Z_i and molecular weight μ_i . We can define the number of atoms of type i in ionisation state r , in units of the total number of atoms of type i , as x_i^r . If n_i/n is the relative number of atoms of type i , then the number of electrons E freed per atom by ionisation is given by

$$E = \sum_i \frac{n_i}{n} \sum_{r=0}^{Z_i} x_i^r = \sum_i \frac{\mu_0}{\mu_i} X_i \sum_{r=0}^{Z_i} x_i^r r \quad . \quad (2.4-4)$$

The degrees of ionisation can be obtained from the set of Saha equations

$$\frac{x_i^{r+1}}{x_i^r} \frac{E}{E+1} = K_i^r \quad , \quad r=0, \dots, Z_i \quad , \quad (2.4-5)$$

and the relations

$$\sum_{r=0}^{Z_i} x_i^r = 1 \quad . \quad (2.4-6)$$

We can define the quantity K_i^r from equation (2.4-5) as follows. For an atom of type i with the ionisation potential, in ionisation state r , χ_i^r , we have that

$$K_i^r = \frac{z_{r+1}}{z_r} \frac{2}{P_c} \frac{(2\pi m_e)^{3/2} (kT)^{5/2}}{h^3} e^{-\chi_i^r/kT}, \quad (2.4-7)$$

where z_{r+1}/z_r is the ratio of the partition functions for the two ionisation states $r+1$ and r . These partition functions can be obtained from a consideration of the Boltzmann formula. We may write this for ions in ionisation state r and excitation state s as

$$\frac{n_{r,s}}{n_{r,0}} = \frac{g_{r,s}}{g_{r,0}} e^{-\psi_{r,s}/kT}, \quad (2.4-8)$$

where $g_{r,s}$ is the statistical weight of this state, and $\psi_{r,s}$ is the excitation energy of this state. The partition function z_r is then given by

$$z_r = g_{r,0} \sum_s \frac{n_{r,s}}{n_{r,0}} = \sum_s g_{r,s} e^{-\psi_{r,s}/kT}. \quad (2.4-9)$$

If calculated numerically, the summation (2.4-9) can be truncated if it tends towards a single value. Alternatively, the 'nearest neighbour' approximation could be used. In this case, the summation is truncated at a value of s at which the orbit of the outermost bound electron approaches another atom. The partition function defined in equation (2.4-9) should be used only for single particles or nuclei that can be considered as single particles. If we are considering the presence of molecules, then their partition functions are more complex. On the simplest level, we can assume that the electronic, vibrational and rotational energy levels of a molecule are independent of each other. In this case a molecular partition function may be written as a product of partition functions for the electronic, vibrational and rotational states

$$Z = z_e z_v z_{rot}, \quad (2.4-10)$$

functions for which can be readily found in the literature. See, for example, Carson (1992).

We should note that the Saha formula is derived in an assumption of thermodynamic equilibrium, or local thermal equilibrium, LTE. It may therefore not hold in stellar corona, for example, or other non-LTE cases. We can solve the Saha equations for E and for the n_i iteratively, by taking a trial value of E in equations (2.4-5), and solving for the n_i . This value of n_i can be used in equation (2.4-4) to determine a new trial value for E . This process can be continued until the values of E and n_i are consistent. The Saha equations (2.4-5) do not hold in the core regions of stars, as they fail to take into account the decrease of ionisation energy with increasing density. The degrees of ionisation predicted by the Saha equations will

therefore be too low. However, we can use the assumption that elements in a stellar core will be completely ionised due to pressure ionisation. We therefore assume complete ionisation at any depth below which the Saha equations give complete ionisation.

Assuming no degeneracy, then the electron pressure can now be given by

$$P_e = \frac{\mathcal{R}E}{\mu_0} \rho T \quad , \quad (2.4-11)$$

in a similar manner to equation (2.4-3).

At high densities in a stellar interior, then the perfect gas law does not hold, as quantum mechanical effects must be taken into account when considering the behaviour of the stellar material. In this case, when the electron gas is degenerate, the density ρ and electron pressure P_e must be found from the integrals

$$\frac{\rho}{\mu_e} = \frac{4\pi m_u}{h^3} (2m_e)^{3/2} \int_0^\infty \frac{u^{1/2} du}{e^{u/kT-\Lambda} + 1} \quad , \quad (2.4-12)$$

and

$$P_e = \frac{8\pi}{3h^3} \int_0^\infty \frac{p^3 (\partial u / \partial p) dp}{e^{u/kT-\Lambda} + 1} \quad , \quad (2.4-13)$$

where we are assuming complete pressure ionisation. In both equations, u is the kinetic energy of the electrons, while p is their momentum. Both the electron pressure P_e and density ρ are given in terms of a degeneracy parameter Λ . For a given P , T and chemical composition (leading to μ_0) we can determine Λ and then ρ from the above equations (2.4-12, 13).

2.5 The Opacity κ

When solving the stellar structure equations (2.1-1, 5), the opacity is normally determined from numerical tables. Values for function (2.1-10) are determined from these tables at different compositions, each table giving $\kappa(\rho, T)$ for the range of density ρ and temperature T . Therefore, here we shall restrict ourselves to describing the major contributions to opacity. We can consider opacity to arise from electron scattering, absorption due to free-free transitions, bound-free transitions and bound-bound transitions. Of the bound-free transitions, negative hydrogen ions usually contribute greatly to the opacity. Finally, conduction effects may reduce the opacity, if significant.

When an electromagnetic wave interacts with an electron, the electric field of the wave causes the electron to oscillate. The electron then represents a classical dipole, which will radiate in other directions - in other words, it will scatter a fraction of the energy of the wave. This scattering of the radiation is equivalent to its weakening by absorption, and we may calculate an opacity resulting from this effect. Classically, we may obtain the opacity due to 'Thomson scattering', κ_{sc} . This neglects any exchange of momentum between the radiation and an electron, and has the form

$$\kappa_{sc} = \frac{8\pi}{3} \frac{r_e^2}{\mu_e m_u} = 0.2(1+X)\text{cm}^2\text{g}^{-1}. \quad (2.5-1)$$

Here, X is the mass fraction of hydrogen, and r_e is the classical electron radius. We arrive at the second expression above via

$$\mu_e = \left(X + \frac{1}{2}Y + \frac{1}{2}(1-X-Y) \right)^{-1} = \frac{2}{1+X}. \quad (2.5-2)$$

If the exchange of momentum between radiation and electrons becomes significant, however, then the value of κ_{sc} will be lower than that given by equation (2.5-1). In fact, this only occurs at temperatures where relativistic effects make equation (2.5-1) a bad approximation. At these temperatures, we consider relativistic corrections coming from 'Compton scattering'.

Absorption from free-free transitions occur when a free electron is sufficiently close to an ion or molecule to form a system which can absorb and emit radiation. Classically, we can calculate a Kramers opacity due to this process in a fully ionised medium to have an absorption coefficient of the order

$$\kappa_{\nu} \sim Z^2 \rho T^{-1/2} \nu^{-3}, \quad (2.5-3)$$

where Z is some mean charge number and ν is the frequency. The Rosseland mean of the opacity, first expounded by Rosseland (1924), shows that a factor ν^{-3} in κ_{ν} leads to a factor T^{-3} in κ . So now have that the Kramers opacity is given by

$$\kappa_{ff} \sim \rho T^{-7/2}. \quad (2.5-4)$$

The factor Z^2 is normally incorporated into the constant of proportionality as a weighted sum. Of course, equation (2.5-4) is still a classical approximation. It is normally multiplied by the 'Gaunt factor', g , to correct for quantum-mechanical effects.

Absorption of radiation due to the ionisation of atoms or molecules make up the opacity effects from bound-free transitions. The opacity can again be calculated classically, and corrected by a Gaunt factor for quantum mechanical effects. However, we find that the absorption coefficient a_ν for a single ion or molecule

$$a_\nu = \frac{\kappa_\nu \rho}{n_{\text{ion}}} \quad , \quad (2.5-5)$$

where n_{ion} is the number density of ions or molecules, varies strongly with the excitation state of that ion or molecule. So the absorption coefficient for a mixture of ions or molecules in different excitation and ionisation states will be a weighted superposition of many different a_ν . So to calculate an opacity we would need to determine the relative numbers of ions in varying states from the Boltzmann and Saha equations. For example, an important source of opacity are the bound-free transitions of neutral hydrogen atoms, where the opacity is proportional to the number of such atoms. Here, the Rosseland mean opacity can be written as

$$\kappa_{\text{bf}} = X(1-x)\tilde{\kappa}(T) \quad , \quad (2.5-6)$$

where X is again the mass fraction of hydrogen, x is the fraction of hydrogen atoms that are ionised, and the value of the absorption coefficient $\tilde{\kappa}(T)$ is obtained by Rosseland integration over weighted sums of the function a_ν for different stages of excitation.

If radiation absorption by a bound electron does not cause ionisation, only a change of excitation state, then the transition is bound-bound. The energy absorbed will later be released in an arbitrary direction, thus weakening the radiation beam in much the same way as electron scattering. This mechanism is only effective at certain frequencies, those of absorption lines. However, lines in stars are strongly broadened by collisions, and can occupy considerable portions of the stellar spectrum, making bound-bound transitions important in the calculation of the overall opacity. The effects of bound-bound transitions are far greater at lower temperatures, at $T < 10^6$ K, for example.

Negative hydrogen ions have a low ionisation energy - photons with energies $h\nu > 0.75$ eV can cause a bound-free transition. The numbers of negative hydrogen ions, assuming thermal equilibrium, can be given by the Saha equation as

$$\frac{n_0}{n_{-1}} P_e = 4 \frac{(2\pi m_e)^{3/2} (kT)^{5/2}}{h^3} e^{-\chi/kT} \quad , \quad (2.5-7)$$

where χ is the ionisation potential of the second electron, P_e is the electron pressure, n_0 and n_{-1} are the numbers of neutral hydrogen atoms and negative hydrogen ions, respectively. If we now take

$$n_0 = \frac{(1-x)\rho X}{m_u} \quad , \quad (2.5-8)$$

with x and X defined as for equation (2.5-6), then we have that

$$n_{-1} = \frac{1}{4} \frac{h^3}{(2\pi m_e)^{3/2} (kT)^{5/2} m_u} P_e (1-x) X \rho e^{\chi/kT} \quad . \quad (2.5-9)$$

For an absorption coefficient a_ν per ion, we can determine the form of κ_ν from equations (2.5-5) and (2.5-9). Hence we can find the Rosseland mean opacity

$$\kappa_{H^-} = \frac{1}{4} \frac{h^3}{(2\pi m_e)^{3/2} (kT)^{5/2} m_u} P_e (1-x) X a(T) e^{\chi/kT} \quad . \quad (2.5-10)$$

Here, $a(T)$ is obtained from a_ν by Rosseland integration. We can see from the above that

$$\kappa_{H^-} \sim n_{-1} \sim n_0 n_e \quad , \quad (2.5-11)$$

or the opacity is proportional to the electron density.

In stellar material, heavy elements provide electrons at relatively low temperatures due to their low ionisation potentials. Even if there is a small mass fraction of metals, they determine the electron density at low temperatures. When all metals are singly ionised, in the temperature range 3000 K - 5000 K, we have

$$n_e = \frac{\rho [xX + (1-X-Y/A)]}{m_u} \quad , \quad (2.5-12)$$

where Y is the mass fraction of helium, and A is some mean mass number for all elements heavier than helium. Since the opacity due to negative hydrogen ions is proportional to the electron density - from equation (2.5-11) - the metal content can determine opacity at lower temperatures. Effectively, this occurs when

$$(1-X-Y)/A > xX \quad , \quad (2.5-13)$$

as x , the fraction of hydrogen that is ionised, becomes very small at low temperatures.

Electrons can transport heat by conduction. Normally this contribution to the total transport of energy can be neglected, as it is small compared to the transport of energy by photons. The conductivity is proportional to the mean free path λ , and the mean free path for a photon in a star is far longer than that for an electron. However, in degenerate material free electrons have more difficulty in exchanging momentum with other particles, as quantum states are filled. So we can say that encounters between electrons and other particles are less frequent, and thus the mean free path λ is longer. When conduction is significant, we can take account of it by defining a 'conductive opacity' κ_c . If κ_{rad} is the radiative opacity, then the total opacity can be defined from

$$\frac{1}{\kappa} = \frac{1}{\kappa_{\text{rad}}} + \frac{1}{\kappa_c} \quad (2.5-14)$$

Therefore increasing conduction reduces the total opacity κ as energy finds a non-radiative means of transport.

2.6 Energy Generation and Reaction Rates

If we consider a reaction of nucleus X with the particle a , which forms nucleus Y and particle b



we may represent this with the notation $X(a, b)Y$. If we denote the type of reacting particles by indices j for X and k for a , then the number of reactions per unit volume and time can be given by

$$\tilde{r}_{jk} = \frac{1}{1 + \delta_{jk}} n_j n_k \sigma v \quad , \delta_{jk} = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases} \quad (2.6-2)$$

Here, n_j particles of type j move with velocity v relative to n_k particles of type k . Each particle of type j has a reaction cross-section σ and sweeps out a volume σv per unit time. To take into account the differences in velocity between individual particles, we can assume that both types j and k have a Maxwell-Boltzmann velocity distribution. This is a good approximation except for extreme densities, such as those found in neutron stars for example. If both velocity distributions are Maxwellian, then the distribution of relative velocities will also be Maxwellian in form. The energy associated with the velocity of a particle is

$$E = \frac{1}{2} m v^2 , \quad (2.6-3)$$

and using the normal definition of the reduced mass of a two particle system

$$m = \frac{m_j m_k}{(m_j + m_k)} , \quad (2.6-4)$$

then we can show that the fraction of all j, k , particle pairs contained within an energy interval $[E, E + dE]$ is given by

$$f(E) dE = \frac{2}{\sqrt{\pi}} \frac{E^{1/2}}{(kT)^{3/2}} e^{-E/kT} dE . \quad (2.6-5)$$

This fraction of particle pairs will then contribute an amount

$$dr_{jk} = \tilde{r}_{jk} f(E) dE , \quad (2.6-6)$$

to the total reaction rate r_{jk} per unit time and volume. We can find this by integrating over all energies, so that

$$r_{jk} = \frac{1}{1 + \delta_{jk}} n_j n_k \overline{\sigma v} , \quad (2.6-7)$$

where we have the averaged probability

$$\overline{\sigma v} = \int_0^{\infty} \sigma(E) v f(E) dE . \quad (2.6-8)$$

We can then replace the numbers of particles per unit volume n_j, n_k , by the mass fractions X_j and X_k in equation (2.6-7) to give the reaction rate per unit time

$$r_{jk} = \frac{1}{(1 + \delta_{jk})(m_j m_k)} \rho^2 X_j X_k \overline{\sigma v} . \quad (2.6-9)$$

If an energy Q is released by each reaction of particles j and k , then the energy generation rate for this reaction per unit time and mass is simply

$$\varepsilon_{jk} = \frac{Qr_{jk}}{\rho} \quad (2.6-10)$$

All thermonuclear reactions are able to occur because of the quantum-mechanical effect of 'tunnelling'. It is possible for particles with energies less than that of the Coulomb potential barrier surrounding a nucleus to 'tunnel' through the barrier. The probability p of a particle 'tunnelling' through the Coulomb potential barrier, and thus a reaction occurring, varies as

$$p \sim E^{1/2} e^{-2\pi\eta} \quad (2.6-11)$$

where

$$\eta = \left(\frac{m}{2}\right)^{1/2} \frac{Z_1 Z_2 e^2}{\hbar E^{1/2}} \quad (2.6-12)$$

Here, m is the reduced mass of the particles, and Z_1 and Z_2 their respective nuclear charges. We can see that the probability of a reaction increases with increasing energy. However, we find that at certain values for E the two reacting particles find a 'resonance', and at such resonance energies the probability of a reaction occurring is much higher. These resonance energies actually represent energies at which the nucleus formed by the reacting particles has a quasi-stationary energy level. Of course, the probability of a reaction occurring determines the reaction cross-section, σ . We may write the cross-section, dependent on the energy E , in the form

$$\sigma(E) = S(E) E^{-1} e^{-2\pi\eta} \quad (2.6-13)$$

The function $S(E)$, for astrophysical applications, may be taken as a constant, S_0 , providing we are considering only non-resonant energies. From equation (2.6-8) we can now write

$$\overline{\sigma v} = \frac{2^{3/2}}{(m\pi)^{1/2}} \frac{1}{(kT)^{3/2}} \int_0^{\infty} S(E) e^{-E/kT - \bar{\eta}/E^{1/2}} dE \quad (2.6-14)$$

where

$$\bar{\eta} = 2\pi\eta E^{1/2} = \pi(2m)^{1/2} \frac{Z_j Z_k e^2}{\hbar} \quad (2.6-15)$$

If we take the approximation that $S(E) = S_0$, so that we are considering all reactions to be non-resonant, it is usual to take values for

S_0 and the energy released per reaction Q from the literature. Such rates will either be calculated, or experimentally established in the laboratory. It is to be noted that the values for Q exclude the energy carried away by neutrinos. It should also be noted that the triple- α reaction, described in chapter four, is unusual in being both a three-body reaction and a resonant reaction via an intermediate state of ^{12}C . However, it is still possible to take values for $S(E)$ from the literature.

We find that shielding by electrons plays a role in determining nuclear reaction rates, due to the modification of the nuclear potential of a nucleus by the presence of nearby free electrons. The positive charge of the nucleus will tend to attract the electrons and repel any other nearby positive ions. This local polarisation effectively lowers the potential barrier and increases the reaction rate. This increase is larger for increasing density ρ and decreasing temperature T and electron shielding is the dominant factor in the high-density, low-temperature regime.

2.7 The Hydrostatic Equation

We can make some simplifications to the system described in equations (2.1-1, 5). These are dependant on the relation of various timescales characteristic to the time derivatives in equations (2.1-1, 5). From equation (2.1-2), we can derive τ_{hydr} , the hydrodynamical timescale;

$$\tau_{\text{hydr}} \approx \sqrt{\frac{R^3}{GM}} \quad , \quad (2.7-1)$$

where R is the radius and M the mass of the star. This gives the typical time for a slightly perturbed star to reach hydrostatic equilibrium. The value for any given star typically varies from the order of seconds to the order of days. From equation (2.1-3), we can derive τ_{KH} , the Kelvin-Helmholtz timescale ;

$$\tau_{\text{KH}} \approx \frac{GM^2}{2RL} \quad , \quad (2.7-2)$$

which gives the time for a star to cool and collapse, assuming its only source of energy is gravitational. We can derive a nuclear, or chemical, timescale τ_n from equations (2.1-5). This can best be represented by :

$$\tau_n \approx \frac{E_n}{L} \quad , \quad (2.7-3)$$

where E_n is the total reservoir of energy that the star has access to, and thus τ_n represents the lifetime of the star.

In general, for most stars, and certainly most of the population III stars considered in this work, $\tau_n \gg \tau_{kh} \gg \tau_{hydr}$, throughout most of their lifetimes. In the case where both $\tau_n \gg \tau_{hydr}$ and $\tau_{kh} \gg \tau_{hydr}$, the evolution of the star is governed by thermal adjustment or nuclear reactions, and then we can neglect the inertial term in equation (2.1-2). This then becomes the equation of hydrostatic equilibrium

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4}, \quad (2.7-4)$$

and the star can be considered to evolve along a sequence of states of hydrostatic equilibrium. Now, due to the lesser number of variables in the equations of evolution, we have to specify the initial values of $P(m)$, $T(m)$, and $X_i(m)$ in any calculation when using this approximation.

In the case where $\tau_n \gg \tau_{kh}$, the time derivatives in equation (2.1-3) become small and can be neglected, so the equation then becomes ;

$$\frac{\partial l}{\partial m} = \epsilon_n - \epsilon_v, \quad (2.7-6)$$

and the star can then be said to be in complete (mechanical and thermal) equilibrium. This is, however, only a good approximation for main sequence stars, and it is not used for the purposes of this work.

2.8 Boundary Conditions

When solving the equations of stellar evolution, an important part of the problem are the boundary conditions. These cannot be applied at just one end of the mass interval $[0, M]$, but are split between the 'surface' and the centre of the star. The central boundary conditions are obviously given by ;

$$m_{r=0} = 0, \quad r = 0, \quad l_{r=0} = 0. \quad (2.8-1)$$

We have that $m=0$, and since the density ρ must go to a finite and non-vanishing number, we must have $r=0$ at the centre. As energy sources must also remain finite, we must have that the luminosity $l=0$ at the centre as well. Unfortunately, we cannot define the behaviour of the pressure, P_c , and temperature T_c at the centre.

The upper boundary conditions can be defined at the photosphere, by the two equations given below ;

$$P = \frac{2GM}{3R^2\kappa'} \quad (2.8-2)$$

and ;

$$L = 4\pi R^2 \sigma T^4, \quad (2.8-3)$$

which together can be used to relate the photospheric values of P , T , r and l . So the boundary conditions for the solution of the equations of stellar evolution are given by the equations (2.8-1, 3).

The photosphere was chosen for the upper boundary conditions of the problem, as it makes the method of solution slightly easier to set up (see chapter three). In addition, should we wish to consider the stellar atmosphere exterior to the photosphere, then it is comparatively easy to compute a solution for values at the upper limit of this stellar atmosphere, using the known photospheric conditions as a starting point. Firstly, we must define what constitutes the 'surface' of the atmosphere. From Eddington's approximation for radiative transport, given by

$$T^4 = \frac{3}{4} \left(\frac{L}{4\pi\sigma R^2} \right) \left(\tau + \frac{2}{3} \right) \quad , \quad (2.8-4)$$

we can see that at the photosphere, where equation (2.8-3) holds true, the optical depth $\tau=2/3$. So we can define the surface of the atmosphere as $\tau=0$, and use τ as the independent variable in any integration. We choose the optical depth τ over the mass m or radius r , as it changes by a far larger amount over the stellar atmosphere. From the theory of radiative transport in stellar atmospheres, we know that

$$\frac{d\tau}{dr} = -\kappa\rho \quad , \quad (2.8-5)$$

and then using the above and the equation of continuity, (2.1-1), we can show that

$$\frac{\partial M_r}{\partial \tau} = -\frac{4\pi R^2}{\kappa} \quad . \quad (2.8-6)$$

With the equation of hydrostatic equilibrium (2.7-4), this yields

$$\frac{\partial P}{\partial \tau} = \frac{GM}{\kappa R^2} \quad . \quad (2.8-7)$$

The upper boundary condition for this last atmospheric equation is that the gas pressure $P_g = 0$ at $\tau=0$.

In the stellar atmosphere, we can take the approximation that the luminosity is constant at the photospheric value, so $l = L$. In addition, the energy generation rates in the atmosphere are zero, so $\epsilon = 0$. We can then integrate numerically inwards from the exterior boundary condition to the known photospheric values, and thus gain values for the other parameters at the upper boundary of the stellar atmosphere, at $\tau=0$.

2.9 The Central Behaviour of the Functions

In addition to the boundary conditions, we will need to know the behaviour of the functions describing P , T , r and l in as $m \rightarrow 0$, close to the centre of the star. This is due to the two singularities at $m=0$, one in $\ln r$ and the other in the equation of continuity, (2.1-1). By using expansions around $m=0$ we can avoid these singularities.

The equation of continuity (2.1-1) may be written as

$$d(r^3) = \frac{3}{4\pi\rho} dm, \quad (2.9-1)$$

and then integrated at constant $\rho = \rho_c$ and at some time $t = t_0$, provided we assume the values of m and r are small enough, to give :

$$r = \left(\frac{3}{4\pi\rho_c} \right)^{1/3} m^{1/3}, \quad (2.9-2)$$

where ρ_c is the central density. This can be considered the first term of a series expansion of r around $m=0$. If we now perform a corresponding integration for the equation of energy generation (2.1-3), we find

$$l = \left(\epsilon_n - \epsilon_v - c_p \frac{\partial T}{\partial t} + \frac{\delta}{\rho} \frac{\partial P}{\partial t} \right)_c m, \quad (2.9-3)$$

where the subscript 'c' denotes values at the centre of the star. In both of these cases the central boundary conditions given by (2.8-1) imply that the constants of integration are zero. If we use equation (2.9-1) to eliminate r from the hydrostatic equation (2.7-4), for small values of m , then we obtain :

$$\frac{dP}{dm} = -\frac{G}{4\pi} \left(\frac{4\pi\rho_c}{3} \right)^{4/3} m^{-1/3}, \quad (2.9-4)$$

which can be integrated, as before, to form

$$P - P_c = -\frac{3G}{8\pi} \left(\frac{4\pi}{3} \rho_c \right)^{4/3} m^{2/3} \quad , \quad (2.9-5)$$

It should be noted that the pressure gradient will, of course, vanish at the centre. This can be seen from writing the hydrostatic equation (2.7-4) in the form :

$$\frac{dP}{dr} \sim \frac{m}{r^2} \sim \frac{r^3}{r^2} \rightarrow 0 \quad , \quad (2.9-6)$$

for $r \rightarrow 0$.

For the variation of temperature, we could consider either the radiative or adiabatic gradients, or the value for the temperature gradient derived from the cubic equations (2.2-3, 6). However, for the purposes of this work, the adiabatic temperature gradient provides the best approximation of the behaviour close to the centre of the star. Starting from

$$\frac{dT}{dm} = -\frac{T}{P} \frac{Gm}{4\pi r^4} \left(\frac{P\delta}{T\rho c_p} \right) \quad , \quad (2.9-7)$$

if we then use (2.7-1) to replace r in the above equation, an integration for small values of m gives :

$$\ln T - \ln T_c = -\left(\frac{\pi}{6} \right)^{1/3} G \frac{\nabla_{ad,c} \rho_c^{4/3}}{P_c} m^{2/3} \quad . \quad (2.9-8)$$

The equations (2.9-2, 3), (2.9-5) and (2.9-8) now provide a description of the behaviour of the four dependant variables P , T , r and l in the centre of the star as $m \rightarrow 0$.

Now that the theory of the problem has been discussed, we can move on to the method of solution.

Chapter Three

The Method of Solution

3.1 Shooting and Fitting Methods

We must now examine the methods which could be used to solve the problem we have presented ourselves with. This problem may be considered to be to find solutions to the equations (2.1-1, 4) at all points in the mass range $[0, M]$ for a given mass and chemical composition (i.e. M is known, as are the X_i at all points in this range). If we are assuming a state of hydrostatic equilibrium, the problem is then to find solutions to equations (2.1-1), (2.7-4), (2.1-3, 4) at all points in the same mass range. Because of the use of the equation of hydrostatic equilibrium (2.7-4) in place of equation (2.1-2), we must include values of P and T , across the mass range, in the initial known conditions. For the purposes of this work, we define a model as being a full set of solutions to these equations, at all points across a given mass range.

The simplest attempt to achieve a solution would be to make use of the 'shooting method'. To do this, we would take trial values of P_c, T_c at the centre of the model and integrate outwards - but this has little chance of meeting the correct surface conditions. We would find that small changes in the trial values can lead to large variations when approaching the surface, due to the T^{-4} term arriving from equation (2.1-4). Similarly if we took trial values of R, L at the surface and integrated inwards, the same would happen due to the r^{-4} term arriving from equation (2.7-4).

An alternative is to use a compromise between the two directions of integration, or a 'fitting method'. Inward and outward integrations, as described above, are carried out to an intermediate 'fitting point'. At this point, the variables r, P, T and l from both integrations are matched by a variation of the four trial parameters P_c, T_c, R , and L . It is usually preferable to choose the fitting point to be at the interface of two physically different regions of the model, between a convective core and radiative envelope, for example. However, the fitting method has been surpassed by other methods for the calculation of complicated models, and is only used in certain situations - for example, finding all possible solutions for a given set of initial parameters.

3.2 The Henyey Method

The method used by the TRC code is an implementation of the Henyey procedure put forward by Kippenhahn, Weigert and Hofmeister (1967), with a few adaptations. This method is very practical for the solution of problems in which the boundary conditions are split between the ends of an interval. In general, then, a trial solution for the whole model is taken, and improved upon by successive corrections in an iterative procedure. In each iteration, the corrections to all variables at all points are evaluated such that their effect on the whole solution is taken into account. We would suggest viewing the original paper by

Kippenhahn, et al, by way of comparison. In particular the difference in the choice of the outer boundary of the model is to be noted.

The corrections may be obtained from linearised algebraic equations. Firstly, we can write the differential equations (2.1-1) ,(2.7-4), (2.1-3, 4) differently, as :

$$\frac{\partial y_i}{\partial m} = f_i(y_1, \dots, y_4) \quad , \quad i = 1, \dots, 4 \quad , \quad (3.2-1)$$

where $y_1 = r$, $y_2 = l$, $y_3 = P$ and $y_4 = T$ are the relevant abbreviations. Now we can describe the difference equations corresponding to these differential equations. These difference equations will apply to a finite mass interval $[m^j, m^{j+1}]$, corresponding to a zone in a model, and we denote the functions at either end of this interval by upper indices. e.g. $f_1^j, f_1^{j+1}, \dots, f_4^j, f_4^{j+1}$. The functions f_i in equation (3.2-1) can then be taken at some average value for a mass interval, which we will label as $f_i^{j+1/2}$, which is some combination of f_i^j , and f_i^{j+1} . This combination can be either the arithmetic mean :

$$f_i^{j+1/2} = \frac{1}{2}(f_i^j + f_i^{j+1}) \quad , \quad (3.2-2)$$

or the geometric mean :

$$f_i^{j+1/2} = \sqrt{\frac{1}{f_i^j f_i^{j+1}}} \quad . \quad (3.2-3)$$

In this work, we use the arithmetic mean (3.2-2). We can now define the four functions

$$A_i^j = \frac{y_i^j - y_i^{j+1}}{m^j - m^{j+1}} - f_i^{j+1/2}(y_1^j, \dots, y_4^{j+1}) \quad , \quad i = 1, \dots, 4 \quad (3.2-4)$$

and then the difference equations replacing (3.2-1) for the mass interval, or zone, between m^j and m^{j+1} are

$$A_i^j = 0 \quad , \quad i = 1, \dots, 4 \quad . \quad (3.2-5)$$

If we number the zone boundaries from $j=1$ at the outermost interval, to $j = K$ at the centre of the model, then in the innermost interval of m , between m^{K-1} and m^K (where $m^K=0$, of course) we must apply series expansions for all four variables, given by equations (2.9-2, 3), (2.9-5) and (2.9-8) . These four equations are given by ;

$$C_i(y_1^{K-1}, \dots, y_4^{K-1}, y_2^K, y_3^K) = 0 \quad , \quad i=1, \dots, 4 \quad (3.2-6)$$

which already include the central boundary conditions that $y_1^K = y_4^K = 0$, or $r = l = 0$ at the centre of the model.

Previous authors have recommended that a fitting mass be chosen, generally below the hydrogen ionisation zone, and that the normal equations above this be integrated step by step as by the shooting method. This is done to remove the more time-consuming computations from the procedure, and was a choice forced upon these authors due to the comparatively limited computing power available to them. However, we choose not to use this approximation, and the difference equations are formed as above from the centre of the model to the photosphere, at which the outer boundary conditions are given.

We can now consider the whole interval of m , from $m^K=0$ to the photosphere m^1 , to be divided into $(K - 1)$ zones or intervals by K points, shown below :



These points will usually not be equidistant, and are in fact usually clustered towards the core and outer layers of a model, as explained later. At these K dividing points, we will have $(4K - 2)$ unknown variables, as $y_1^K = y_4^K = 0$. In order to have a solution, these unknowns must fulfil equation (3.2-6) for the central boundary, and equation (3.2-5) for the intervals between, except for the last one (so $j= 1, \dots, K - 2$). So thus we have $(4K - 4)$ equations. We need, however, $(4K - 2)$ equations, and so we can use the photospheric boundary conditions from equations (2.8-2, 3) to form the final two difference equations we need in the outermost interval, labelling them B_1 and B_2 . These equations take much the same form as the earlier difference equations C_i for the innermost mass interval. Thus we now have the required number of equations to solve the problem, which we can write as

$$B_i = 0 \quad , \quad i = 1, 2 \quad , \quad (3.2-14)$$

$$A_i^j = 0 \quad , \quad i = 1, \dots, 4, \quad j = 1, \dots, K-2, \quad (3.2-15)$$

$$C_i = 0 \quad , \quad i = 1, \dots, 4 \quad . \quad (3.2-16)$$

We are looking for a solution with given values of M , $X_i(m)$, and values of $P(m)$ and $T(m)$ from a previous model (at an earlier time in an

evolutionary sequence, for example). These last two values can form part of a first approximation to this solution, which is given by the values $(y_i^j)_1$ with $i = 1, \dots, 4$, and $j = 1, \dots, K$. This is typically an extrapolation of a former solution, or a solution for similar parameters. During evolutionary calculations, it would be easiest to take the solution from the previous time step in the calculation as the first approximation for the new solution.

Since the $(y_i^j)_1$ are only an approximation, they will not fulfil equations (3.2-14, 16) - when they are used as arguments in these equations, we find that

$$B_i(1) \neq 0, \quad A_i^j(1) \neq 0, \quad C_i(1) \neq 0, \quad (3.2-17)$$

where (1) indicates that the first approximation is used as arguments. We now look to apply corrections δy_i^j to all variables at all dividing points, such that the second approximation

$$(y_i^j)_2 = (y_i^j)_1 + \delta y_i^j \quad (3.2-18)$$

of the arguments fulfils equations (3.2-14, 16), i.e. it will make the B_i , A_i^j and C_i vanish. The corrections δy_i^j of the arguments will produce the changes δB_i , δA_i^j and δC_i in these functions, and we require that

$$B_i(1) + \delta B_i = 0 \quad (3.2-19)$$

$$A_i^j(1) + \delta A_i^j = 0 \quad (3.2-20)$$

$$C_i(1) + \delta C_i = 0 \quad (3.2-21)$$

If the corrections are small enough, we can expand the δB_i , δA_i^j and δC_i in terms of increasing powers of the corrections δy_i^j , keeping only the linear terms of the expansions. For example :

$$\delta B_i \approx \frac{\partial B_i}{\partial y_1^1} \delta y_1^1 + \frac{\partial B_i}{\partial y_2^1} \delta y_2^1 + \frac{\partial B_i}{\partial y_3^1} \delta y_3^1 + \frac{\partial B_i}{\partial y_4^1} \delta y_4^1 \quad (3.2-22)$$

Using this linearisation, equations (3.2-19, 21) can be written as :

$$\frac{\partial B_i}{\partial y_1^1} \delta y_1^1 + \dots + \frac{\partial B_i}{\partial y_4^1} \delta y_4^1 = -B_i \quad (3.2-23)$$

$$\frac{\partial A_i^j}{\partial y_1^j} \delta y_1^j + \dots + \frac{\partial A_i^j}{\partial y_4^j} \delta y_4^j + \frac{\partial A_i^j}{\partial y_1^{j+1}} \delta y_1^{j+1} + \dots + \frac{\partial A_i^j}{\partial y_4^{j+1}} \delta y_4^{j+1} = -A_i^j,$$

$$i = 1, \dots, A, \quad j = 1, \dots, K-2, \quad (3.2-24)$$

$$\frac{\partial C_i}{\partial y_1^{K-1}} \delta y_1^{K-1} + \dots + \frac{\partial C_i}{\partial y_4^{K-1}} \delta y_4^{K-1} + \frac{\partial C_i}{\partial y_2^K} \delta y_2^K + \frac{\partial C_i}{\partial y_3^K} \delta y_3^K = -C_i,$$

$$i = 1, \dots, A, \quad (3.2-25)$$

where the B_i , A_i^j , C_i and all derivatives are evaluated using the first approximation arguments $(y_i^j)_1$. This is now a system of $(4K - 2)$ linear, inhomogenous equations for the $(4K - 2)$ unknown corrections δy_i^j , where $i = 1, \dots, A$, $j = 1, \dots, K$, but $\delta y_1^K = \delta y_4^K = 0$ because of the central boundary conditions.

3.3 A Sample Linearisation

We can give, as an example, the linearisation of equation (2.7-4), the equation of hydrostatic equilibrium, at some central mass interval between dividing points j and $j+1$. For this one equation, we have that equation (3.2-4) becomes

$$A_2^j = \frac{P^j - P^{j+1}}{m^j - m^{j+1}} - \frac{Gm^{j+1/2}}{4\pi(r^{j+1/2})^4} \quad (3.3-1)$$

When we expand equation (3.2-20), keeping only the linear terms, to form equation (3.2-24), we need to know the differentials of A_2^j with respect to the variables P , T , l and r . These are given by

$$\frac{\partial A_2^j}{\partial P^{j,j+1}} = \frac{\pm 1}{m^j - m^{j+1}}, \quad \frac{\partial A_2^j}{\partial T^k} = 0, \quad \frac{\partial A_2^j}{\partial l^k} = 0, \quad (3.3-2)$$

and

$$\frac{\partial A_2^j}{\partial r^{j,j+1}} = \frac{Gm^{j+1/2}}{\pi(r^{j+1/2})^5}, \quad (3.3-3)$$

as we have defined that

$$\frac{m^{j+1/2}}{(r^{j+1/2})^n} = \frac{1}{2} \left[\frac{m^j}{(r^j)^n} + \frac{m^{j+1}}{(r^{j+1})^n} \right] \quad (3.3-4)$$

So equation (3.2-24) becomes

$$\begin{aligned} \frac{\delta P^j - \delta P^{j+1}}{m^j - m^{j+1}} + \frac{Gm^j}{\pi(r^j)^5} \delta r^j + \frac{Gm^{j+1}}{\pi(r^{j+1})^5} \delta r^{j+1} \\ = \frac{Gm^{j+1}}{4\pi(r^{j+1})^4} + \frac{P^{j+1} - P^j}{m^j - m^{j+1}}. \end{aligned} \quad (3.3-5)$$

If we look at the hydrodynamic equation (2.1-2) instead, we have that equation (3.2-4) becomes, with the extra term;

$$A_2^j = \frac{P^j - P^{j+1}}{m^j - m^{j+1}} - \frac{Gm^{j+1/2}}{4\pi(r^{j+1/2})^4} - \frac{1}{4\pi(r^{j+1/2})^2} \frac{\partial^2 r^{j+1/2}}{\partial t^2}. \quad (3.3-6)$$

The differentials with respect to the variables P , T and l remain the same, but we have that

$$\frac{\partial A_2^j}{\partial r} = \frac{Gm^{j+1/2}}{\pi(r^{j+1/2})^5} + \frac{1}{2\pi(r^{j+1/2})^3} \frac{\partial^2 r^{j+1/2}}{\partial t^2}, \quad (3.3-7)$$

so equation (3.2-24), in this case, will have several extra terms.

We must now look at the method used in solving the equations (3.2-23, 25) to find the corrections δy_j^j .

3.4 The Solution of the Linearised Equations

We can write equations (3.2-23, 25) in matrix form as :

$$H \begin{pmatrix} \delta y_1^1 \\ \cdot \\ \cdot \\ \cdot \\ \delta y_3^K \end{pmatrix} = - \begin{pmatrix} B_1 \\ \cdot \\ \cdot \\ \cdot \\ C_4 \end{pmatrix} \quad (3.4-1)$$

The matrix H of the coefficients is known as the 'Henye matrix', and its elements are given by the derivatives on the left-hand sides of equations (3.2-23, 25). These non-trivial elements form overlapping blocks about the diagonal axis of the Henye matrix, and we can solve one block at a time to obtain the corrections δy_j at each dividing point.

The diagram below gives the element blocks in the upper left hand corner of the Henye matrix - the rows correspond to the equations (3.2-23, 24), while the columns correspond to the corrections δy_j to the variables to be determined. Thus these blocks represent six inhomogeneous equations in the eight unknown corrections $\delta r_1, \delta l_1, \delta P_1, \delta T_1, \delta r_2, \delta l_2, \delta P_2$ and δT_2 . In both of these blocks, $j=1$. In the block below in the Henye matrix, which has the same form as the second block, containing the 'A' differentials, we would have that $j=2$, and so on.

δr_1	δl_1	δP_1	δT_1	δr_2	δl_2	δP_2	δT_2	$\delta r_3 \dots$
$\frac{\partial B_1}{\partial r_j}$	$\frac{\partial B_1}{\partial l_j}$	$\frac{\partial B_1}{\partial P_j}$	$\frac{\partial B_1}{\partial T_j}$					
$\frac{\partial B_2}{\partial r_j}$	$\frac{\partial B_2}{\partial l_j}$	$\frac{\partial B_2}{\partial P_j}$	$\frac{\partial B_2}{\partial T_j}$					
$\frac{\partial A_1}{\partial r_j}$	$\frac{\partial A_1}{\partial l_j}$	$\frac{\partial A_1}{\partial P_j}$	$\frac{\partial A_1}{\partial T_j}$	$\frac{\partial A_1}{\partial r_{j+1}}$	$\frac{\partial A_1}{\partial l_{j+1}}$	$\frac{\partial A_1}{\partial P_{j+1}}$	$\frac{\partial A_1}{\partial T_{j+1}}$	
$\frac{\partial A_2}{\partial r_j}$	$\frac{\partial A_2}{\partial l_j}$	$\frac{\partial A_2}{\partial P_j}$	$\frac{\partial A_2}{\partial T_j}$	$\frac{\partial A_2}{\partial r_{j+1}}$	$\frac{\partial A_2}{\partial l_{j+1}}$	$\frac{\partial A_2}{\partial P_{j+1}}$	$\frac{\partial A_2}{\partial T_{j+1}}$	
$\frac{\partial A_3}{\partial r_j}$	$\frac{\partial A_3}{\partial l_j}$	$\frac{\partial A_3}{\partial P_j}$	$\frac{\partial A_3}{\partial T_j}$	$\frac{\partial A_3}{\partial r_{j+1}}$	$\frac{\partial A_3}{\partial l_{j+1}}$	$\frac{\partial A_3}{\partial P_{j+1}}$	$\frac{\partial A_3}{\partial T_{j+1}}$	
$\frac{\partial A_4}{\partial r_j}$	$\frac{\partial A_4}{\partial l_j}$	$\frac{\partial A_4}{\partial P_j}$	$\frac{\partial A_4}{\partial T_j}$	$\frac{\partial A_4}{\partial r_{j+1}}$	$\frac{\partial A_4}{\partial l_{j+1}}$	$\frac{\partial A_4}{\partial P_{j+1}}$	$\frac{\partial A_4}{\partial T_{j+1}}$	

The following diagram illustrates the form of the last block in the Henye matrix, in its lower right hand corner, corresponding to equations (3.2-25), and thus representing four equations in six variables.

$$\begin{array}{cccccc}
 \dots \delta r_{K-1} & \delta l_{K-1} & \delta P_{K-1} & \delta T_{K-1} & \delta P_K & \delta T_K \\
 & \cdot & & & & \\
 & \cdot & & & & \\
 & \cdot & & & & \\
 \frac{\partial C_1}{\partial r_{K-1}} & \frac{\partial C_1}{\partial l_{K-1}} & \frac{\partial C_1}{\partial P_{K-1}} & \frac{\partial C_1}{\partial T_{K-1}} & \frac{\partial C_1}{\partial P_K} & \frac{\partial C_1}{\partial T_K} \\
 \frac{\partial C_2}{\partial r_{K-1}} & \frac{\partial C_2}{\partial l_{K-1}} & \frac{\partial C_2}{\partial P_{K-1}} & \frac{\partial C_2}{\partial T_{K-1}} & \frac{\partial C_2}{\partial P_K} & \frac{\partial C_2}{\partial T_K} \\
 \frac{\partial C_3}{\partial r_{K-1}} & \frac{\partial C_3}{\partial l_{K-1}} & \frac{\partial C_3}{\partial P_{K-1}} & \frac{\partial C_3}{\partial T_{K-1}} & \frac{\partial C_3}{\partial P_K} & \frac{\partial C_3}{\partial T_K} \\
 \frac{\partial C_4}{\partial r_{K-1}} & \frac{\partial C_4}{\partial l_{K-1}} & \frac{\partial C_4}{\partial P_{K-1}} & \frac{\partial C_4}{\partial T_{K-1}} & \frac{\partial C_4}{\partial P_K} & \frac{\partial C_4}{\partial T_K} \\
 \frac{\partial r_{K-1}}{\partial C_1} & \frac{\partial l_{K-1}}{\partial C_1} & \frac{\partial P_{K-1}}{\partial C_1} & \frac{\partial T_{K-1}}{\partial C_1} & \frac{\partial P_K}{\partial C_1} & \frac{\partial T_K}{\partial C_1} \\
 \frac{\partial r_{K-1}}{\partial C_2} & \frac{\partial l_{K-1}}{\partial C_2} & \frac{\partial P_{K-1}}{\partial C_2} & \frac{\partial T_{K-1}}{\partial C_2} & \frac{\partial P_K}{\partial C_2} & \frac{\partial T_K}{\partial C_2} \\
 \frac{\partial r_{K-1}}{\partial C_3} & \frac{\partial l_{K-1}}{\partial C_3} & \frac{\partial P_{K-1}}{\partial C_3} & \frac{\partial T_{K-1}}{\partial C_3} & \frac{\partial P_K}{\partial C_3} & \frac{\partial T_K}{\partial C_3} \\
 \frac{\partial r_{K-1}}{\partial C_4} & \frac{\partial l_{K-1}}{\partial C_4} & \frac{\partial P_{K-1}}{\partial C_4} & \frac{\partial T_{K-1}}{\partial C_4} & \frac{\partial P_K}{\partial C_4} & \frac{\partial T_K}{\partial C_4}
 \end{array}$$

Here of course the indices j have been replaced by $K-1$ and K . The blocks in between the first and last shown above, along the diagonal of the Henyey matrix, all represent four equations in eight variables.

To start with we take the first block, six equations for eight unknown variables, and calculate the coefficients and inhomogenous terms for these. The last two unknowns, δP_2 and δT_2 , are eliminated and the other six variables of the block expressed in linear terms of δP_2 and δT_2 , such that

$$\begin{aligned}
 \delta r_1 &= U_1 \delta P_2 + V_1 \delta T_2 + W_1 \\
 \delta l_1 &= U_2 \delta P_2 + V_2 \delta T_2 + W_2 \\
 \delta P_1 &= U_3 \delta P_2 + V_3 \delta T_2 + W_3 \\
 \delta T_1 &= U_4 \delta P_2 + V_4 \delta T_2 + W_4 \\
 \delta r_2 &= U_5 \delta P_2 + V_5 \delta T_2 + W_5 \\
 \delta l_2 &= U_6 \delta P_2 + V_6 \delta T_2 + W_6
 \end{aligned} \tag{3.4-2}$$

The constants U_i , V_i and W_i ($i = 1, \dots, 6$) are calculated as the solution of six equations in six unknowns, where δP_2 and δT_2 are treated as parameters whose values will be calculated later. We may find U_i , V_i and W_i by solving the following matrix equation, where the first matrix on the LHS is six by six entries, and the product matrix is three by six entries;

$$\begin{aligned}
 & \begin{bmatrix} \frac{\partial B_1}{\partial r_1} & \dots & \frac{\partial B_1}{\partial T_1} & 0 & 0 \\ \frac{\partial B_2}{\partial r_1} & \dots & \frac{\partial B_2}{\partial T_1} & 0 & 0 \\ \frac{\partial A_1}{\partial r_1} & \dots & \cdot & \cdot & \frac{\partial A_1}{\partial l_1} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{\partial A_4}{\partial r_1} & \dots & \cdot & \cdot & \frac{\partial A_4}{\partial l_1} \end{bmatrix} \begin{bmatrix} U_1 & V_1 & W_1 \\ U_2 & V_2 & W_2 \\ U_3 & V_3 & W_3 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ U_6 & V_6 & W_6 \end{bmatrix} \\
 & = \begin{bmatrix} 0 & 0 & -B_1 \\ 0 & 0 & -B_2 \\ -\frac{\partial A_1}{\partial P_2} & -\frac{\partial A_1}{\partial T_2} & -A_1 \\ \vdots & \vdots & \vdots \\ -\frac{\partial A_4}{\partial P_2} & -\frac{\partial A_4}{\partial T_2} & -A_4 \end{bmatrix}, \quad (3.4-3)
 \end{aligned}$$

where obviously $j=1$. Noting the values taken by the constants U_i , V_i and W_i , we may now proceed to the intermediate blocks. We may number these blocks $n=2, \dots, K-2$, and so can generalise the following treatment of the second block by the use of appropriate subscripts.

For the second block, we may eliminate δr_2 and δl_2 by using the last two of equations (3.4-2). Then we can use δP_3 and δT_3 as parameters in much the same way as δP_2 and δT_2 were used for the first block. This gives the set of linear equations corresponding to the matrix equation

$$\begin{bmatrix} \alpha_1 & \beta_1 & \frac{\partial A_1}{\partial r_{j+1}} & \frac{\partial A_1}{\partial l_{j+1}} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \alpha_4 & \beta_4 & \frac{\partial A_4}{\partial r_{j+1}} & \frac{\partial A_4}{\partial l_{j+1}} \end{bmatrix} \begin{bmatrix} \delta p_n \\ \delta T_n \\ \delta r_{n+1} \\ \delta l_{n+1} \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \cdot \\ \cdot \\ \gamma_4 \end{bmatrix}, \quad (3.4-4)$$

with $j=n$. For the second block, we have that $n=2$. The α_i , β_i and γ_i represent the terms which are affected by the elimination of δr_n , δl_n , δP_{n+1} and δT_{n+1} , and are given by

$$\alpha_i = \frac{\partial A_i}{\partial P_j} + \frac{\partial A_i}{\partial r_j} U_{4n-3} + \frac{\partial A_i}{\partial l_j} U_{4n-2} , \quad (3.4-5)$$

$$\beta_i = \frac{\partial A_i}{\partial T_j} + \frac{\partial A_i}{\partial r_j} V_{4n-3} + \frac{\partial A_i}{\partial l_j} V_{4n-2} , \quad (3.4-6)$$

and

$$\gamma_i = -A_i - \frac{\partial A_i}{\partial r_j} W_{4n-3} - \frac{\partial A_i}{\partial l_j} W_{4n-2} - \frac{\partial A_i}{\partial P_{j+1}} \delta P_{n+1} - \frac{\partial A_i}{\partial T_{j+1}} \delta T_{n+1} . \quad (3.4-7)$$

We now know or have calculated the coefficients of equations (3.4-4, 7), apart from the parameters δP_{n+1} and δT_{n+1} so the block may be solved to give

$$\begin{aligned} \delta P_n &= U_{4n-1} \delta P_{n+1} + V_{4n-1} \delta T_{n+1} + W_{4n-1} \\ \delta T_n &= U_{4n} \delta P_{n+1} + V_{4n} \delta T_{n+1} + W_{4n} \\ \delta r_{n+1} &= U_{4n+1} \delta P_{n+1} + V_{4n+1} \delta T_{n+1} + W_{4n+1} \\ \delta_{n+1} &= U_{4n+2} \delta P_{n+1} + V_{4n+2} \delta T_{n+1} + W_{4n+2} \end{aligned} , \quad (3.4-8)$$

In a similar manner to the first block, we may calculate the constants U_i , V_i and W_i by solving the matrix equation

$$\begin{aligned}
 & \begin{bmatrix} \alpha_1 & \beta_1 & \frac{\partial A_1}{\partial r_{j+1}} & \frac{\partial A_1}{\partial l_{j+1}} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \alpha_4 & \beta_4 & \frac{\partial A_4}{\partial r_{j+1}} & \frac{\partial A_4}{\partial l_{j+1}} \end{bmatrix} \begin{bmatrix} U_{4n-1} & V_{4n-1} & W_{4n-1} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ U_{4n+2} & V_{4n+2} & W_{4n+2} \end{bmatrix} \\
 & = \begin{bmatrix} -\frac{\partial A_1}{\partial P_{j+1}} & -\frac{\partial A_1}{\partial T_{j+1}} & -A_1 - \frac{\partial A_1}{\partial r_j} W_{4n-3} & -\frac{\partial A_1}{\partial l_j} W_{4n-2} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ -\frac{\partial A_4}{\partial P_{j+1}} & -\frac{\partial A_4}{\partial T_{j+1}} & -A_4 - \frac{\partial A_4}{\partial r_j} W_{4n-3} & -\frac{\partial A_4}{\partial l_j} W_{4n-2} \end{bmatrix}, \quad (3.4-9)
 \end{aligned}$$

where $j=n$. We note the values of the constants U_i, V_i, W_i and proceed to the next block. Here, the last two of equations (3.4-8) are used for elimination in this block, and we may proceed through equations (3.4-4, 9) for this block, with the value of n increased by one.

We continue this procedure through to, and including, the block labelled $n=K-2$. The last block in the matrix, as noted above, has four equations in six variables. The last two of equations (3.4-8) for the penultimate block are applied to the last block, eliminating δr_{K-1} and δl_{K-1} . This leaves the system which can be represented by the matrix equation

$$\begin{bmatrix} \alpha_1 & \beta_1 & \frac{\partial C_1}{\partial P_K} & \frac{\partial C_1}{\partial T_K} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \alpha_4 & \beta_4 & \frac{\partial C_4}{\partial P_K} & \frac{\partial C_4}{\partial T_K} \end{bmatrix} \begin{bmatrix} \delta P_{K-1} \\ \delta T_{K-1} \\ \delta P_K \\ \delta T_K \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \cdot \\ \cdot \\ \gamma_4 \end{bmatrix}, \quad (3.4-10)$$

where we have that

$$\begin{aligned}
 \alpha_i &= \frac{\partial C_i}{\partial P_{K-1}} + \frac{\partial C_i}{\partial r_{K-1}} U_{4K-7} + \frac{\partial C_i}{\partial l_{K-1}} U_{4K-6} \\
 \beta_i &= \frac{\partial C_i}{\partial T_{K-1}} + \frac{\partial C_i}{\partial r_{K-1}} V_{4K-7} + \frac{\partial C_i}{\partial l_{K-1}} V_{4K-6} \quad , i = 1, \dots, 4. \quad (3.4-11) \\
 \gamma_i &= -C_i - \frac{\partial C_i}{\partial r_{K-1}} W_{4K-7} - \frac{\partial C_i}{\partial l_{K-1}} W_{4K-6}
 \end{aligned}$$

This system must then be solved. All of these coefficients and inhomogenous terms can be calculated, so that equations (3.4-10) lead to the values of δP_{K-1} , δT_{K-1} , δP_K and δT_K . The values of δP_{K-1} and δT_{K-1} can then be used in equations (3.4-8) for the previous block, labelled as $n=K-2$, to give the values for δP_{K-2} , δT_{K-2} , δr_{K-1} and δl_{K-1} . We can then repeat the process for the block labelled as $n=K-3$ using δP_{K-2} and δT_{K-2} , and so on. Eventually we will reach the first block and equations (3.4-2) will yield the last of the unknown variables, δr_1 , δl_1 , δP_1 , δT_1 , δr_2 , and δl_2 .

3.5 Some Comments on the Henyey Method

It should be noted that when we determine the new X_i from equations (2.1-5) and (2.1-11) during an evolutionary sequence of iterations, we are effectively assuming that the values of P , T and the old X_i are remaining constant throughout the time step. This approximation is therefore only good for comparatively small changes in element abundances during any given time step. This is commented upon in chapter four. Ideally, we would not wish to alter a model after it has been calculated, but before the next model in an evolutionary sequence is calculated. If we change a model too much, then it may provide bad trial values for the next calculation. For practical purposes, however, this is the best time to apply changes to a model that would otherwise require complicated calculations to apply during an iterative procedure. The most obvious changes that might be applied to a model are convective mixing, or mass loss, both of which involve changes in the mass fractions of elements in large portions of a model. In fact, such changes should not harm an evolutionary sequence of calculations, provided they are small compared to the changes from model to model in the sequence. After all, each one may be considered separately as an example of the procedure outlined in this chapter, using the previous model as a trial set of values.

We may still find that at times the Henyey method breaks down and the corrections fail to converge. There may be several reasons for this, but in general it is usually because either the trial values are too different to the solution, or because one or more of the approximations we have chosen to use is becoming invalid. The former case occurs should we select too large a time step between models or change a model too much

before embarking upon the next time step. The correction functions δB_i , δA_{ij} and δC_i become so large that we cannot ignore the second-order terms when forming equation (3.2-24). It is fairly easy to avoid this problem by setting criteria for a maximum time step when entering an evolutionary calculation, or by avoiding large changes to a model in between time steps. In the second case, when an approximation is becoming invalid, there is little to do but go back and change the approximation or remove it entirely. For example, the approximation of hydrostatic equilibrium begins to break down when the inertial term in equation (2.1-2) becomes too large to be neglected at certain stages of stellar evolution. In these cases we would have to use a form of the hydrodynamic equation instead of the hydrostatic equation (2.7-4).

Chapter Four

The TRC Evolutionary Code

4.1 An Introduction to the Code

The code which was used to model the evolution of population III stars for this work, the TRC code, was written by Dr. T. R. Carson over the course of some fifteen years and modified by the author during time spent at the University of St. Andrews. It uses an implementation of the numerical method detailed in the previous chapter in order to solve the equations of stellar structure (2.1-1, 5) for a model of a given mass and chemical composition. The TRC code was compiled and used in a SunOS (UNIX-based) environment using the standard Sun FORTRAN operating set-up. The code is written using FORTRAN 4, with most of the later modifications making use of additions from FORTRAN 77. It is hierarchical in nature, consisting of a large initial routine, and slaved subroutines in several tiers. A description of the program structure will follow an introduction to its use, as one of the aims of this work is to provide a 'user-friendly' documentation for the use of the TRC code in evolutionary work. It is recommended that the reader should become familiar with the theory and numerical methods underlying the use of the code before attempting to use it.

An initial and amended version of the code will be used in this work. The differences between the two are comparatively minor, and involve the treatment of nitrogen in the equation of state and associated parts of the code. The author will detail the initial version in full in this chapter, and include a later section to detail the changes made for the amended version of the code.

4.2 The Variables Used in Implementing the Henyey Method

Throughout the code, transformed variables replace the normal variables of mass, radius, luminosity, effective temperature, pressure, and energy generation rate in each interval. The equations defining the transformed variables are as follows :

$$\xi_n = \ln \left[1 - \frac{m_n}{M^*(1 + \eta)} \right] , \quad (4.2-1)$$

$$\Lambda_n = \ln \left[1 + \frac{l_n}{L^*} \right] , \quad (4.2-2)$$

$$x_n = \ln r_n , \quad (4.2-3)$$

$$t_n = \ln T_n , \quad (4.2-4)$$

$$p_n = \ln P_n , \quad (4.2-5)$$

$$e_n = \ln \varepsilon_n, \quad (4.2-6)$$

Here, ε_n is identical to $(\varepsilon_n - \varepsilon_V)$ given in equation (2.1-3). If this quantity is negative, as occurs during late stages of evolution, then e_n in our code is set to some large negative number. The new mass variable ξ is used in order to narrow the range of the variable at its lower limit, while the others are used to prevent the values of the variables varying through (in some cases) several orders of magnitude. The form of equation (4.2-1) is used to avoid a singularity at the stellar surface, while the form of (4.2-2) is used to avoid the problems caused by negative values of l_n - provided of course that $l_n > -L^*$. The actual values taken by L^* and M^* are given in equations (4.13-3, 4). These changes in variable are of course those taken by Kippenhahn, Weigert and Hofmeister (1967) in their implementation of the Henyey method.

The changes in variable used in the code have no real effect on the form of the equations given in the previous chapter. By means of an example, if we label the alternative variables above by $z_1 = x$, $z_2 = p$, $z_3 = t$ and $z_4 = \Lambda$, then equation (3.2-4) would become

$$A_i^j = \frac{z_i^j - z_i^{j+1}}{\xi^j - \xi^{j+1}} - f_i(y_1^{j+1/2}, \dots, y_4^{j+1/2}) \frac{(dz_i / dy_i)^{j+1/2}}{(d\xi / dM)^{j+1/2}}, \quad (4.2-7)$$

assuming that the transformations between variables y_i and z_i , $i = 1, \dots, 4$, are well behaved. Writing these equations in the above form would not, however, have helped the clarity of the argument, and the fact that these transformed equations exist in the code is noted only in passing.

4.3 The Equation of State

The form taken by the equation of state when modelling a star can be seen to be one of the more important factors in determining the characteristics of that model. See for example the comments made by Carson, Luo and Sharp (1992). So it is important to state the form used and assumptions made in the code we use. The equation of state is defined as shown in chapter two, including the effects of degeneracy of the electron gas, and the various assumptions mentioned. The only atoms and ions considered for the purposes of the equation of state are hydrogen ^1H , helium ^4He , carbon ^{12}C , nitrogen ^{14}N and oxygen ^{16}O . Rather than unnecessarily complicating the calculation of the equation of state, we take the simplification that the total mass fraction of metals in the model is made up of ^{12}C , ^{14}N and ^{16}O , in the proportions 0.3, 0.1 and 0.6 by mass. It should be noted that as we are considering no thermonuclear reactions which change the abundance of ^{14}N , we need not keep track of its mass fraction.

The only molecules considered in the equation of state are H₂, CO and N₂. Providing that CNO abundances are unconstrained and there is more O than C, so that CH does not become important, we can neglect the presence of other molecules due to their low number densities when H₂, CO and N₂ are present in significant amounts. We find that as the numbers of H₂, CO and N₂ molecules are independent of each other - which should be obvious from their component atoms - we can obtain their number densities n easily from their partition functions Z , summed over all states. For example, for carbon monoxide, CO, we have

$$\frac{n_C n_O}{n_{CO}} = \frac{z_C z_O}{z_{CO}} \left(\frac{2\pi m k T}{h^2} \right)^{3/2} e^{-D/kT} \quad , \quad (4.3-1)$$

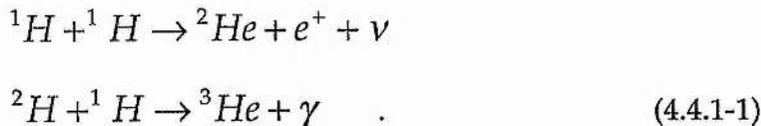
where m is the reduced mass and D the dissociation energy of the molecule. We should bear in mind that the partition functions for molecules are more complex than those of atoms or ions, as touched upon in chapter two.

4.4 Thermonuclear Reactions

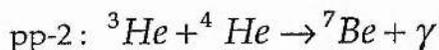
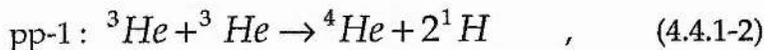
Many different thermonuclear reactions and chains of reactions occur in stellar material, but most of the energy released by a star comes from the nuclear fusion of hydrogen or helium. In the code we also consider carbon burning and oxygen burning. The heaviest element considered is sulphur ³²S, beyond which burning is assumed to cease.

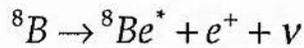
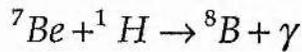
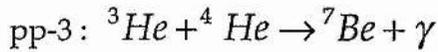
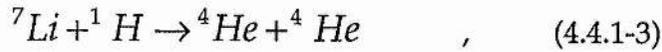
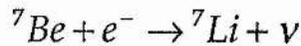
4.4.1 Hydrogen Burning - the pp-chain and CNO Cycle

If we consider the nuclear fusion of hydrogen as a whole, the net result is the fusion of four hydrogen nuclei ¹H into one helium nucleus ⁴He. The two main chains of reactions in stellar matter, which do just this, are known as the proton-proton chain and the CNO cycle. The proton-proton chain begins when two protons fuse to form a deuterium nucleus ²H, which then fuses with another proton to form ³He.



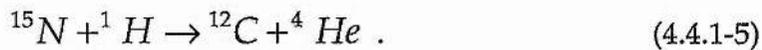
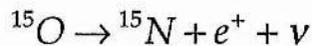
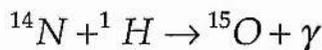
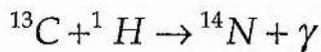
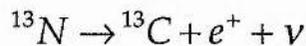
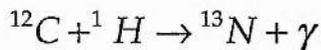
There are then three possible routes for completion of the pp-chain, which we can label pp-1, pp-2 and pp-3, all starting from ³He. They are :



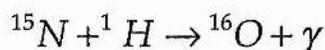


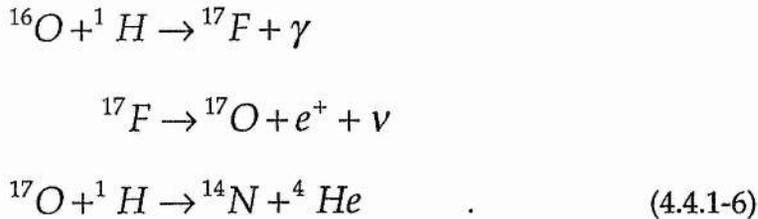
Due to the different energies of the neutrinos in the reactions above, the Q value for the energy generated differs for the three pp-chains, and the relative frequency of the chains depends on the local conditions in the stellar matter. We find that for increasing temperature T , pp-2 and pp-3 will dominate more over pp-1, providing that ${}^4\text{He}$ is present. This is due to the greater temperature sensitivity of the ${}^3\text{He} - {}^4\text{He}$ reaction, relative to the ${}^3\text{He} - {}^1\text{H}$ reaction. With further increases in T , we find that pp-3 will dominate over pp-2, as proton capture becomes more favourable for ${}^7\text{Be}$.

The CNO cycle requires the presence of various isotopes of carbon, nitrogen and oxygen, and the sequence of reactions can be shown as below :



The ${}^{12}\text{C}$ produced in the last reaction above, can obviously be recycled back to the first reaction in the series. However, ${}^{15}\text{N}$ and ${}^1\text{H}$ may react differently to continue the cycle :





The ${}^{14}\text{N}$ produced in this last reaction can then be cycled back to the fourth reaction in the first set (4.4.1-5) above.

For the code, the reaction rates for both the pp-chain and the CNO cycle are determined from the reaction rate of the slowest reaction in each chain. For the pp-chain this is the first, ${}^1\text{H} + {}^1\text{H}$, reaction, while for the CNO cycle we consider it to be the ${}^{14}\text{N} + {}^1\text{H}$ reaction. In taking this approximation for the CNO cycle, we are assuming that the nuclei involved have reached equilibrium in their abundances. In essence, we then have two reaction rates for the conversion



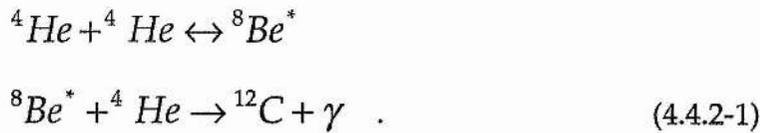
with two associated Q values for the energy produced. For the pp-chain, the Q value is taken as half the Q value of pp-1, as reactions (4.4.1-1) have to be performed twice to produce the two ${}^3\text{He}$ nuclei to react in equation (4.4.1-2). This is a crude approximation, in which we choose to neglect the other pp reactions, but removes the necessity of calculating a correction factor for the Q value. This correction factor would have to depend on the temperature T and the mass fraction of ${}^4\text{He}$, to reflect the increasing importance of the pp-2 and pp-3 reactions given in equations (4.4.1-3, 4).

For the CNO cycle to take place, carbon at least must be present. However, the code does not compute the equilibrium abundances of the CNO elements arising from the simultaneous CNO and triple- α reactions. Instead, the ${}^{14}\text{N}$ abundance is given by $X_{\text{N}}=0.1Z$. In addition, we are assuming that the reaction rate for the CNO cycle will not be affected by the lowering of the initial mass fraction of ${}^{12}\text{C}$, due to carbon burning.

4.4.2 Helium Burning - the 3α Reaction

Helium burning in stellar material consists of reactions fusing ${}^4\text{He}$ nuclei, or α -particles, into ${}^{12}\text{C}$, ${}^{16}\text{O}$, and so forth. We see α -capture reactions stepping up the mass of the product nuclei. Helium burning becomes important at temperatures $T > 10^8\text{K}$. In the code the largest nucleus considered to be produced by this fusion process is ${}^{32}\text{S}$. The first reaction in this step-up sequence is the triple alpha reaction, 3α , in which

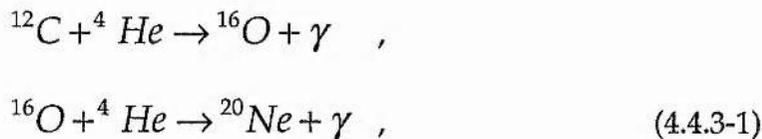
^{12}C is formed from three ^4He nuclei. This actually occurs in two steps, as the probability of a triple encounter is vanishingly small.



The product of the first reaction, $^8\text{Be}^*$, is very short-lived, and decays back to two α -particles if the second reaction doesn't occur.

4.4.3 α -capture Reactions, Carbon and Oxygen Burning

Further reactions in the step-up sequence of α -captures yield the nuclei ^{12}C , ^{16}O , ^{20}Ne and so on up to ^{32}S . For example, we have



and so forth. We can take reaction rates and Q values for each of these reactions separately. Carbon burning will set in at higher temperatures than helium burning - at perhaps $T > 10^9\text{K}$. The reaction can be conveniently expressed, ignoring side-reactions and many other possible products, as



and we can assign this a reaction rate and Q value. The ^{24}Mg nucleus is excited and has several decay modes which can cause complications due to the particles produced by this decay. We neglect any possible decay, however. In addition to carbon burning, oxygen burning may set in at temperatures $T > 10^9\text{K}$. We have the reaction



Again, we can assign a reaction rate and Q value to this reaction and again (4.4.3-3) is a convenient approximation to a series of reactions with a variety of products. Any immediate decay is again neglected. In addition to the immediate decays associated with the reactions (4.4.3-2, 3), we are also neglecting any photodisintegration of the larger nuclei, at temperatures comparable to those needed for carbon or oxygen burning. For example, we have the photodisintegration of neon :



This reaction, and other photodisintegration reactions, are not considered in the code.

4.4.4 Calculation of Reaction Rates

The code calculates the relevant reaction rates from equations (2.6-14) assuming all reactions, other than the triple- α reaction to be non-resonant in nature. The necessary values for the astrophysical correction factor S_0 , or $S(E)$ for the triple- α reaction, are taken from the papers by Fowler, Caughlan and Zimmerman (1967, 1975) and Harris, Fowler, Caughlan and Zimmerman (1983). The Q values for the reactions are taken from the same sources. We should note that in cases where only an upper value has been established for a specific reaction rate, Fowler, Caughlan and Zimmerman have included (0, 1) terms in their reaction rates. In all cases in the code, we have taken these to be equal to 0.1.

4.5 Opacities

The code has the capacity to use numerous tables or fits for the opacity and two were used to generate values for the models in this work. These were the OPAL tables computed by Rogers and Iglesias (1992), and the molecular opacity tables by Carson and Sharp (1991). These molecular opacities are only included at temperatures $T < 8 \times 10^3 \text{K}$. Other methods or tables that could have been considered include the use of the fitting method from work by Christy (1966). In most cases interpolation is needed to find the opacity values at a specific composition and values of T and ρ . The OPAL tables give values for the opacity at varying temperatures and densities for $X=0.00, 0.35, 0.70$ and ten values of Z from zero to 3×10^{-2} . Interpolations between values of temperature and density take the quadratic form

$$\begin{aligned} \log \kappa &= \alpha + \beta \log T + \gamma (\log T)^2 \\ \log \kappa &= \alpha + \beta \log \rho + \gamma (\log \rho)^2 \end{aligned} \quad . \quad (4.5-1)$$

Interpolations for values of X are linear for $X > 0.35$, but otherwise are quadratic. The interpolation of Z is linear, of the form

$$\kappa = \alpha + \beta Z \quad . \quad (4.5-2)$$

The OPAL values were calculated assuming only radiative processes were contributing to the opacity, electron conduction being neglected. It is worth noting that in calculating their opacities, Rogers and Iglesias have

taken the metals present to be in the proportions given by Anders and Grevesse (1989). These proportions are given as

$$C = 0.22690 \quad N = 0.07012 \quad O = 0.53192 \quad \text{Ne - Ni} = 0.17106$$

which we can see is not too far removed from the simple assumption of a 0.3, 0.1, 0.6 split between carbon, nitrogen and oxygen used in the code. The Anders-Grevesse figures were derived from observations of the solar photosphere.

4.6 Convection

The code uses a simple consideration of the Schwarzschild criterion for dynamic stability to determine where convection occurs in models. That is, convection can be considered to occur in zones of the model where the following inequality is true at both ends of each zone;

$$\nabla_{\text{rad}} > \nabla_{\text{ad}} \quad . \quad (4.6-1)$$

Here, of course, ∇_{rad} and ∇_{ad} are the radiative and adiabatic temperature gradients respectively. As mentioned in chapter two, we are assuming that the timescale associated with any non-radial structure of convective zones is much smaller than the timescale associated with the radial structure of these zones. Therefore, convection can be considered to occur in a spherically symmetric manner. When convection occurs across two or more neighbouring zones in a model, then the material within these zones is assumed to become fully mixed during a timestep. The mass fractions, X_i , for these zones are averaged. Neither semiconvection nor convective overshoot are considered in the code.

4.7 The Use of the Hydrostatic Equation

The code assumes that the stars to be modelled are in hydrostatic equilibrium, and so equation (2.1-2) is replaced with (2.7-4), as described in chapter two. The formulation of the numerical method of solution also proceeds under this assumption. However, it is still possible for us to consider a 'patch-on' inertial term in place of that in equation (2.1-2) without changing the way in which we solve the overall problem, as outlined in the previous chapter.

We can arrive at an estimate of the velocity at which any mass interval is moving, at step i when the time difference between step i and step $(i - 1)$ is Δt_i as follows :

$$v_i = \frac{\partial r}{\partial t} = (r_i - r_{i-1}) / \Delta t_i \quad . \quad (4.7-1)$$

We can then estimate an acceleration for that step from

$$a_i = \frac{\partial^2 r}{\partial t^2} = (v_i - v_{i-1}) / \Delta t_i \quad (4.7-2)$$

This calculated value can then be used to form an inertial term to add to equation (2.7-4), forming

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4} - \frac{a(r)}{4\pi r^2} \quad (4.7-3)$$

where $a(r)$ is the acceleration as a function of the radius, given by equation (4.7-2). This can then be used in the numerical method given in the previous chapter without changing any of the assumptions involving hydrostatic equilibrium used in chapters two and three.

It should be noted that for the above 'patch-on' acceleration,

$$\frac{\partial a(r)}{\partial r} \neq 0 \quad (4.7-4)$$

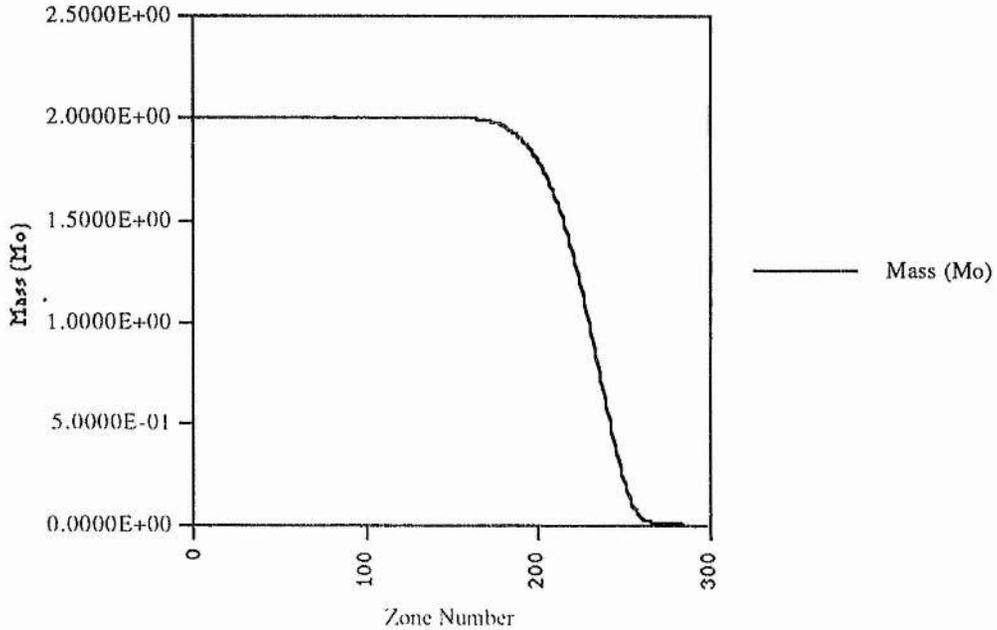
so while the example of linearisation for the hydrodynamic equation given in the previous chapter holds for (4.7-3) above, there will be additional terms in equation (3.3-6). What we hope to achieve by the use of this 'patch-on' term is to help the code to evolve models past points at which the assumption of hydrostatic equilibrium begins to break down.

4.8 Zoning and Mass Intervals

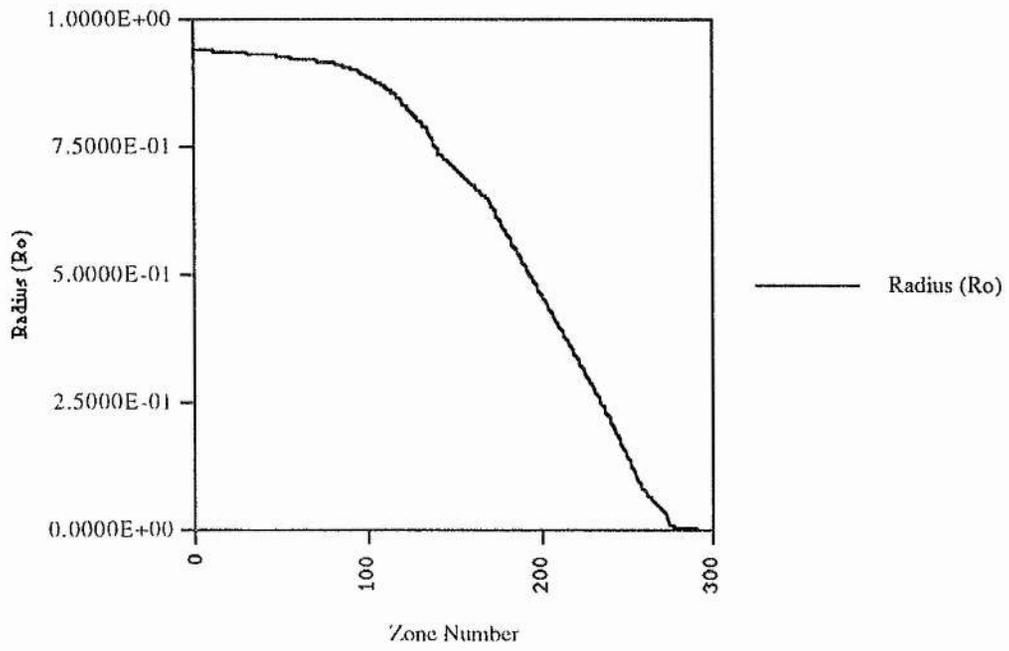
The largest number of mass intervals, or zones, that the code can consider for any model is 512, and the criterion for the placement of zones are given in equations (4.13-5, 7). The typical zero age model will have 300 zones or more, dependent on its mass. It is worth noting that the majority of the zones within any model generated by the code represent only very small changes in radius or mass, and are clustered either near the photosphere or at the core. The majority of the mass of any model is in a comparatively small number of zones nearer the centre of the range in radius. This is due to large changes in the variables describing the stellar material across the outer and core zones. Changes in variables across the zones between these parts of the model are smaller, and hence there are less zones in these areas. The largest changes in variables, especially temperature and pressure, in the outer zones lie across the hydrogen ionisation region.

The two graphs below give the distribution of zones with mass and radius in a $2M_{\text{sun}}$ zero-age model, and clearly show that the majority of zones in this model are either in the outermost layers of the model, or close to the centre. These graphs are representative of most models in this respect, even highly evolved ones.

Mass vs Zone Number



Zone Number vs Radius



4.9 The Calculation Of Mass Loss Rates

4.9.1 Mass Loss From Stellar Winds

The author's single largest modification to the code is the algorithm which computes mass loss rates and the resultant chemical changes in the outer layers of a model. The author uses a formula for mass loss rates given by Nieuwenhuijzen and de Jager (1990). The amount of mass lost in solar units, ΔM , in a given time step in years, Δt , is given by

$$\Delta M = 9.631^{-15} \left(\frac{L}{L_{\text{sun}}} \right)^{1.42} \left(\frac{M}{M_{\text{sun}}} \right)^{0.16} \left(\frac{R}{R_{\text{sun}}} \right)^{0.81} \Delta t, \quad (4.9.1-1)$$

where the other quantities have their normal meanings. This loss in mass is assumed to come about due to a radiatively driven stellar wind. However, this is an empirically derived equation, and an understanding of the mechanism causing this stellar wind, and thus the mass loss, is not immediately necessary. For an extensive treatment of the physics of radiation-driven stellar winds, the reader might refer to the papers by Kudritzki, Pauldrach and Puls (1986, 1989), which build upon the older work by Castor, Abbott and Klein (1975).

Equation (4.9.1-1) was derived by Nieuwenhuijzen and de Jager from an examination of a selection of currently observable stars for which the mass loss rate (from the stellar wind velocity and other data), luminosity, mass and radius could be determined. It was then tested on a second selection of similar data, and achieved a good performance against the real mass loss rates of these stars. Obviously, equation (4.9.1-1) cannot be checked against observable stars with metallicities comparable to or as low as population III metallicities, and was in fact derived from observations of stars with far higher metallicities. However, the mass loss rate given by equation (4.9.1-1) depends upon the fundamental parameters M , L and R . For a given mass M , the parameters L and R depend entirely upon the composition of the star considered. We thus feel it is at least slightly reasonable to extend the use of equation (4.9.1-1) in this case to stars of a composition very different from those used to derive it. In fact, if we look at the literature, we find very few suitable formula giving mass loss rates for use in this type of modelling. The most commonly used are derived empirically from observations of a small group of those stars with high velocity stellar winds - O stars, for example, by Cassinelli and Lamers (1987), or Lamers (1982). Most are also very specific in use - red giants, or early-type stars, only. The equation given above, (4.9.1-1), is the most general available, but is still strictly only valid for stars of population I.

4.9.2 Mass Loss From Thermonuclear Reactions and Radiation

In addition to the mass loss rate given by equation (4.9.1-1) we should also consider the 'loss' in mass due to thermonuclear reactions. The difference in mass between reactants and products in any thermonuclear reaction is known as the 'mass defect'. In the case of fusion reactions, we may give the mass defect ΔM as

$$\Delta M = \left(\sum_i M_i \right) - M_f \quad , \quad (4.9.2-1)$$

where the initial nuclei i have masses M_i and the final nucleus has a mass M_f . Rather than considering the mass defect of all thermonuclear reactions in a model, however, we can calculate the mass lost from radiation leaving the star, or luminosity. So we have that the mass loss rate due to radiation released by thermonuclear reactions for any given model is

$$\Delta M = \left(\frac{L_{\text{sun}}}{M_{\text{sun}}} \right) \frac{L}{c^2} \Delta t \quad , \quad (4.9.2-2)$$

where ΔM and L will be in solar units.

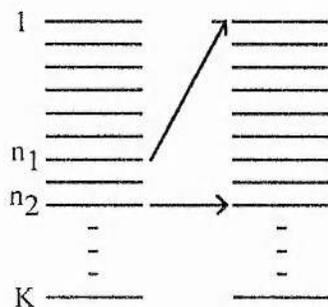
4.10 Composition Changes Due to Mass Loss

A change in mass taken from equations (4.9.1-1) and (4.9.2-2) during a time step is applied to the model after it has converged. To simulate the outward flow of mass in the model, while preserving the temperature, pressure and density structure in the outer zones, the compositions of the outer zones are altered. We should note that only the stellar wind has the effect of causing an outward flow of material particles. Mass loss due to radiation released in thermonuclear reactions, determined from equation (4.9.2-2), can be assumed not to contribute to this flow of material. As noted in the previous chapter, it is important to keep the structure of these zones similar between models - simulating mass loss by simply removing zones will present the code with a model that gives corrections which will not converge. We can note that the outermost zones are in many ways the most sensitive to perturbation.

The alteration of the chemical compositions of the outer zones is a mapping, carried out as follows. If the total mass lost by the star due to its stellar wind, from equation (4.9.1-1) only, over all time steps is given by ΔM_t , then we can select two zones $n=n_1$ and $n=n_2$ from the following equations :

$$M_{n_1} = M - \Delta M_t, \quad M_{n_2} = M - 2\Delta M_t, \quad (4.10-1)$$

where M is the total mass of the model considered. The values of the mass fractions of hydrogen, helium, and the various metals, given by the X_i from equations (2.1-5), at zone n_1 can then be mapped to the outermost zone $n=1$, as illustrated below :



The values of the mass fractions X_i in zones inside the range $n=[1, n_2]$ are then determined by interpolation by zone number between the values now at the ends of the range. As the outermost zones will all have a very similar mass, this is as valid as the normally more accurate interpolation by mass. In this way, an outward flow of material due to the stellar wind is simulated by altering the chemical structure of the outer zones, rather than the physical structure. In this mapping of values, the author is ignoring the effects of any atmosphere the star may have, and is in fact considering it to be an extension of the outermost zone, if the input parameters allow the code to consider it at all.

4.11 The Timestep for Evolution

The timestep, Δt , selected for a given iterative procedure given in the previous chapter can be altered during this procedure, according to criteria given within the code. For example, if the previous model converged within ten iterations, then the current value for the timestep Δt is doubled to become $2\Delta t$ at the start of the next iterative procedure. During the iterative procedure, the relative value of the corrections, given by $\delta B_i/B_i$, $\delta A_i/A_i$ and $\delta C_i/C_i$, may not decrease smoothly from iteration to iteration. Reasons for this have been discussed in the previous chapter. If these corrections increase from one iteration to the next :

$$\left(\frac{\delta A_i}{A_i} \right)_{k+1} > \left(\frac{\delta A_i}{A_i} \right)_k, \quad (4.11-1)$$

where subscripts k and $(k+1)$ refer to consecutive iterations, then the code will alter the timestep to become $0.75\Delta t$.

It may otherwise be the case that the relative value of the corrections is dropping too slowly. If we have that

$$\left| \left(\frac{\delta A_i}{A_i} \right)_k - \left(\frac{\delta A_i}{A_i} \right)_{k+1} \right| < \varepsilon \quad , \quad (4.11-2)$$

where k and $(k+1)$ refer to consecutive iterations as above, and ε is taken as being the convergence value, then the code will again alter the timestep to become $0.75\Delta t$. Lastly, the time step is restored to its initial value for that iterative procedure if the value of the relative corrections has decreased by a factor of ten or more since the last change in timestep during that evolutionary calculation. It is worth noting that the code contains many other criteria for the alteration of the timestep or the convergence value of the corrections, most of which are not actually used. This section of the code is very dense, and is altered at the reader's risk!

4.12 Limits on the Timestep

There are effectively two upper limits applied to the value of the timestep by the code, and it will not exceed the lower of these limits in any alteration to the timestep that has been made. If an alteration to Δt exceeds the lower of the limits, then Δt will be made equal to the lower limit. The first limit is that imposed within the input parameters, while the second comes from within the code. This second upper limit requires that in any given timestep, in any given zone of a model

$$\Delta X_j \leq 0.05 \quad . \quad (4.12-1)$$

Here, ΔX_j is the change in mass fractions X_j during that time step. If any value of ΔX_j exceeds 0.05, then the timestep is changed to become $0.5\Delta t$.

The second upper limit is imposed to minimise the distortions caused by an approximation, or shortcut, used in the code. At any zone, if the mass fraction of any element is reduced below zero by the changes in the X_j taken from equation (2.1-5), then the mass fraction of that element is taken by the code to be zero. However, the energy generation and reaction rates in that zone, for that time step, are not changed to allow for the mass fraction X_j being reduced to zero rather than to below zero. To allow for this would require several large changes in the structure of the code, and would slow down its operation. A similar shortcut is made should any mass fraction in any zone increase to be greater than one, which in theory could perhaps happen in an evolving star when hydrogen is exhausted in a zone. If the mass fraction of metals is low, then the final timestep leading to the exhaustion of hydrogen could lead to a mass fraction of helium that was greater than one. Again the energy

generation rates and reaction rates are not changed - the sum of all the mass fractions is reset to one.

We can see that the limit to the value of ΔX_i at any zone is imposed to prevent these approximations, or shortcuts, causing the code to produce unphysical results. We should bear in mind that the above shortcuts should only be being used in a few zones, of several hundred in a model, in any timestep. Thus any effect that they have will be small in any case.

4.13 Input Requirements of the Code

The following discussion assumes the reader to be familiar with the use of FORTRAN in a UNIX-based system. The code requires an input file and one or two data files. The first, essential, data file must contain the data from the OPAL opacity tables, while the second, optional, data file contains the numerical solutions for all the variables of a model star in binary format - i.e. an output file from the code. The reader would have to examine the code to determine the format required for the OPAL opacities. The input file contains the control parameters used in the Henyey routine - its layout and an explanation of the various control parameters follows (it would be necessary for the reader to examine FORMAT statements in the opening lines of the code to determine the precise layout required for the input file) ;

```
KIN KCM KRZ KDE KSC KPR NDS1 NDS2 NDS3 KEV
STARMS FSCM FSCL FDVM EPSD
DELTY XIN YIN ZIN RMIX
TMAX MLOSS
ENDOFDATA
```

KIN ; this parameter determines whether or not the code accepts a second data file containing the numerical solution of a stellar model. If KIN=1 then the TRC code will generate its own initial model. Should KIN=2 then the code searches for a file containing the data from a model, and NDS1 becomes a null parameter.

KCM ; the number given here alters the normal zoning procedure in the code to add that number of zones to the central section of the model. It has been found that this is the area of the models that causes the most difficulty when using the code, and it is therefore useful to have some external control over the zoning procedure in that part of the model.

KRZ ; this parameter determines whether or not the code will alter the placing and number of zones in the model. If KRZ=0 then the code will not alter the zone structure, and if KRZ=1 then each model will be rezoned before any calculations are performed to form a new model.

KDE ; this parameter determines which mean the code uses during the conversion of the differential equations of stellar evolution to

difference equations. Various settings of KDE make the code use either the arithmetic or geometric mean for different equations, as seen in equations (3.2-2, 3). The default setting of KDE=1 means the code uses the arithmetic mean when forming all the difference equations.

KSC ; this parameter determines whether or not an atmosphere is considered when a model is generated. If KSC=1 then no atmosphere is calculated for the model. If KSC=2, then the atmosphere is calculated, and the output values of pressure, density and radius are given at the 'surface' of the atmosphere, rather than the photosphere.

KPR ; this parameter specifies the amount of data that goes to the standard output file generated by the code. If KPR=0 then none of the energy generation rates in zones at the centre and hydrogen burning shell (if any) will be printed. If KPR=1 then this data will be included in the output.

NDS1 ; providing that KIN=1 then this determines what sort of model the TRC code uses to provide the necessary initial values. If NDS1<10 then a Lane-Emden model is generated, and if NDS1=10, a far simpler linear model is generated instead. If NDS1>10, then the code generates the initial values by integrating step by step from trial values of L and R at the photosphere, and adjusting the results near the centre so that they fit the central boundary conditions. The initial values in this last case are stored in a file which is named 'fort.NDS1'.

NDS2 ; this number is used to specify the file number of the file the code will look for, if KIN=2. The file is considered to be called 'fort.NDS2'.

NDS3 ; this number is used to specify the file number of the file the code will write the numerical solutions for the generated model to. As above, this is considered to be 'fort.NDS3'. During an evolutionary calculation, involving many changes of model, the code will alternate between writing to 'fort.NDS3' and 'fort.(NDS3+1)'.

KEV ; this parameter determines whether or not the code will perform an evolutionary calculation. If KEV=0, then it will not, and will attempt to form a model with parameters determined by STARMS, XIN, YIN, and ZIN. In this case, DELTY, TMAX and MLOSS are null parameters (but must still be included in the input file, of course). If KEV= 1 or 2 then the code will enter an evolutionary calculation, in which case STARMS, XIN, YIN and ZIN become null parameters - these will be determined for the new model within the code. If KEV=1, then the equation of energy conservation is given by :

$$\frac{\partial l}{\partial m} = \epsilon_n - \epsilon_v - T \frac{\partial S}{\partial t} \quad , \quad (4.13-1)$$

and if KEV=2, then the equation becomes :

$$\frac{\partial L}{\partial m} = \varepsilon_n - \varepsilon_v - \frac{\partial U}{\partial t} - P \frac{\partial V}{\partial t}. \quad (4.13-2)$$

Finally, if KEV=10, 11, or 12, then it will have the same effect as KEV=0, 1 or 2 respectively, but an inertial term will be included in the equation of hydrostatic equilibrium, (2.7-4), as described earlier in this chapter.

STARMS ; this is simply the mass of the model the code is to generate, in solar mass units. When KEV=0, this is a null parameter.

FSCM ; this is a scaling constant that is used to vary the range of the parameter used to represent mass in the TRC code. This parameter, ξ_n at a zone n , is given by ;

$$\xi_n = \ln \left[1 - \frac{m_n}{M^* (1 + \eta)} \right], \quad M^* = \text{FSCM} \times M_{\text{star}}, \quad (4.13-3)$$

where $\eta \ll 1$. The code uses a value $\eta = 10^{-3}$, and in most cases FSCM need only be set to one.

FSCL ; this is another scaling constant, similar in use to the above, used to vary the range of the parameter used to represent luminosity in the code. This parameter, Λ_n at a zone n , is given by ;

$$\Lambda_n = \ln \left[1 + \frac{l_n}{L^*} \right], \quad L^* = \text{FSCL} \times L_{\text{star}}, \quad (4.13-4)$$

and again in most cases, FSCL need only be set to one. If $l_n \leq -L^*$, in late stages of evolution, then Λ_n is set to a large negative number.

FDVM ; this scaling constant applies to the conditions under which the code will rezone a model, and so is obviously a null parameter if KRZ=0. The code requires the differences between variables v and v_1 , given by $dv = |v - v_1|$, in two adjacent zones, to be within a range given by :

$$0.2 \times \text{FDVM} < \frac{dv}{v} < \text{FDVM}, \quad (4.13-5)$$

If the model contains values of dv/v which fall outside the above range, then extra zones will be inserted in the problem areas until the above condition is satisfied throughout the model. The above applies to all the variables except for the temperature T and the pressure P . Because of the

sensitivity of the energy generation rates to small changes in temperature, these two variables have the conditions :

$$0.2 \times FDVM < \frac{dT}{T} < 0.25 \times FDVM, \quad (4.13-6)$$

$$0.2 \times FDVM < \frac{dP}{P} < 1.5 \times FDVM, \quad (4.13-7)$$

The range for the pressure has been relaxed in order to prevent too many new zones being added in due to the temperature restriction.

EPSD ; this is the limiting value for corrections applied during any iterative procedure performed by the code, relative to the variables these corrections are being applied to. These relative corrections are given by $\delta B_i/B_i$, $\delta A_i/A_i$ or $\delta C_i/C_i$, the functions and corrections from equations (3.2-19, 21). If the corrections drop below this relative value, then the iteration is considered to be converged. Typically, this is set at somewhere between 10^{-4} and 10^{-6} . This test normally applied to the root mean square value of the corrections, but it can be applied to the largest correction. If the latter option is chosen, then a larger value for EPSD should be selected.

DELTY ; this is the initial time step for an evolutionary calculation to begin at, measured in years.

XIN ; this gives the initial mass fraction of hydrogen present throughout the model that the code is to generate. During evolutionary calculations, this is a null parameter. Note that the distribution of elements is assumed to be uniform throughout the model if this parameter, and the following two, are used.

YIN ; this gives the initial mass fraction of helium present in the model to be generated, and is a null parameter during evolutionary calculations.

ZIN ; this gives the initial mass fraction of metals present in the model to be generated. In the current version of the code, this is assumed to be composed of 60% carbon and 40% oxygen by mass. This is a null parameter during evolutionary calculations.

RMIX ; if RMIX > 0, then RMIX becomes the mixing length parameter for convection, α , and the mixing length is defined by $l_m = \alpha H_p$, where H_p is the pressure scale height. The value of the temperature gradient, ∇ , in the equations of stellar evolution is arrived at by solution of equations (2.2-3, 6). If RMIX < 0, then the value used for ∇ is ∇_{ad} , the adiabatic value, and the mixing length parameter α becomes $|RMIX|$. If

RMIX=0, then the transport of energy within the star is assumed to be due entirely to radiation and conduction, and ∇ is replaced by ∇_{rad} , the radiative value.

TMAX ; this determines the maximum time step the code can apply during an evolutionary calculation, measured in years.

MLOSS ; this determines whether or not changes in model mass due to stellar wind and nucleosynthesis are considered during an evolutionary calculation. If MLOSS=0 then the model is assumed to suffer no changes in mass during evolution, and if MLOSS=1 then mass changes are considered. This is a null parameter if KEV=0.

4.14 Output From the Code

The output from the code consists of the standard output, which may be directed to a file, and up to four data files, one of which contains the numerical solutions to the model generated in a binary format, named as described above. The other three files reproduce data from the standard output in a format compatible with most graphical and spreadsheet packages, and are named 'fort.7', 'fort.8' and 'fort.10'. The actual data contained in each column of these files is given below :

fort.7 ; Absolute B-magnitude, time in years that the star has been evolved, the logarithm of T at the photosphere, the logarithm of the luminosity at the photosphere in solar units, the logarithm of the acceleration due to gravity at the photosphere.

fort.8 ; time in years that the star has been evolved, mass lost during that step, the logarithm of the luminosity at the photosphere in solar units, T_{eff} at the photosphere, the mass fraction of helium at the photosphere, the mass fraction of metals at the photosphere, total mass lost during evolution.

fort.10 ; time in years that the star has been evolved, the mass of the convective core, the radius of the convective core, the mass of the hydrogen-depleted core, the radius of the hydrogen-depleted core, the mass of the helium-depleted core, the radius of the helium-depleted core.

These three files are not used if evolution is not being performed by the code. During evolution, the code will write the above data to these files at each step of evolution, so that a record of changes in the parameters of interest is recorded.

The standard output is of the following format ; before iteration begins, the code will output the following :

```
HENYEY : REZONE MODEL WITH FACTOR  A
      B   C   D   E   F   G
```

Here A is factor expressing (B/F) , B is the old number of zones in the model, and F is the new number of zones. C is the number of zones removed, D is the number of zones added, and E is given by $(D - C)$, or the total number of zones added. If the code has reached the limit number of 512 zones, then G is the number of zones that would have been added by the rezoning procedure.

During the iterative procedure, at the conclusion of each iteration, the code will output the following :

$A \ B \ C \ D \ E \ F \ G \ H \ I$

In this case, A and B are control parameters within the code that measure how far and how well the iterative procedure is progressing. C measures how much the time step has been cut by since the previous step, and D is the current time step, in years. E and F are the most important of these outputs - E is the relative size of the correction being considered in that step, given by $\delta B_i/B_i$, $\delta A_j/A_j$ or $\delta C_i/C_i$ from the previous chapter, and F is the fraction of the correction that is being applied. G , H , and I are the radius in solar units, luminosity at the photosphere, also in solar units, and the temperature at the photosphere, respectively. Should the iterative procedure be progressing badly, it is possible that the new pressure, P , will fall outside a range in which the model would be stable, in one or more zones. If this is the case, the code will change the value of P and the temperature T in these zones, and proceed with a new step. The following will be printed for each zone thus affected :

PRESSURE WARNING : ZONE,T,P,C $A \ B \ C \ D$

Here, numbers $A - D$ are the number of the zone affected, the effective temperature at that zone, and the pressure and density at that zone. Both the pressure and density will be in c.g.s. units.

Once the iterative procedure has converged, the code will output the following, dependent on the control parameters. If mass loss is being considered, then :

MASS LOSS BY NIEUWENHUIJZEN AND DE JAGER
 STEP MASS LOSS = A
 NUMBER OF ZONES "REMOVED" = B
 TOTAL MASS LOST = C

Here, A gives the mass lost in that time step, in solar mass units. B gives the number of zones in the model, from the photosphere inwards, that have had their chemical compositions changed by the mass loss procedure, and C gives the total mass lost over all steps to date, again in solar mass units.

If one or more convective regions exist in the model, then data for each region will be outputted as the following :

```
CONVECTION MRTD ZONE  A B C D E
                    ZONE  F G H I J
```

Below all of the regions listed, the following data will be listed for each convective region in turn :

```
A F K L M
```

Here, *A* and *F* are the zone numbers of the outer and inner limits of the convective region, respectively. *B* and *G* are the fraction of the model's mass contained within these zones, *C* and *H* are the fractional radius of the model at these zones, *D* and *I* are the effective temperatures at these zones, and *E* and *J* are the densities at these zones. *K*, *L* and *M* are the new mass fractions of hydrogen, helium and metals throughout the region, respectively.

Should the model have exhausted hydrogen, or hydrogen and helium, in its core, then one or both of the following messages will be outputted :

```
H DEPLETED TO  A  AT  M,R,T,D  B  C  D  E
```

```
HE DEPLETED TO  F  AT  M,R,T,D  G  H  I  J
```

Here, *A* and *F* are the outermost zones in which hydrogen and helium respectively are depleted to. Numbers *B* - *E* are the fractional mass contained within zone *A*, the fractional radius at zone *A*, and the effective temperature and density in c.g.s. units at zone *A*, respectively. Numbers *G* - *H* give the same data for zone *F*. The code will then output the logarithmic energy generation rates for the first, innermost, thirty-two zones. If there is hydrogen exhaustion at the centre of the model, then it will also output the logarithmic energy generation rates for the thirty-two zones immediately outside the hydrogen-deleted core - which should provide some information on the shell burning in the model. The original rates used in the model are given in c.g.s. units.

Then the code will output the bulk of the model data as follows :

	STEP	ITNS	TIME	STEP	TOTL	TIME	IMASS	CMASS	RADIUS	LUMINOSITY
	TP	PP	DP	XP	YP	ZP				
	TC	PC	DC	XC	YC	ZC				
	ZC12P	ZO16P	ZNE20P	ZMG24P	ZSI28P	ZS32P				
	ZC12C	ZO16C	ZNE20C	ZMG24C	ZSI28C	ZS32C				
A	B	C	D	E	F	G	H			
	I	J	K	L	M	N				
	O	P	Q	R	S	T				
	C12 _p	O16 _p	Ne20 _p	Mg24 _p	Si28 _p	S32 _p				
	C12 _c	O16 _c	Ne20 _c	Mg24 _c	Si28 _c	S32 _c				

Here, *A* denotes the step number, and *B* gives the number of zones. *C* and *D* are the current time step and the total evolutionary time respectively, both in years. *E* and *F* are the initial (zero-age) mass of the model, and the current mass of the model, both in solar mass units, while *G* is the radius of the model in solar units, and *H* is the luminosity at the photosphere, also in solar units. Numbers *I* - *N* give, respectively, the effective temperature at the photosphere, pressure at the photosphere, density at the photosphere, and the mass fractions of hydrogen, helium and metals at the photosphere. Numbers *O* - *T* give the same properties and proportions at the central zone of the model. Both pressure and density are given in c.g.s. units. The last two rows of numbers give the mass fractions of metals in the photosphere (subscript '*p*') and the central zone of the model (subscript '*c*'). The metals considered are, left to right in each row, Carbon-12, Oxygen-16, Neon-20, Magnesium-24, Silicon-28, and Sulphur-32, the number denoting atomic weight, as is usual. We should note that the Nitrogen-14 content is not explicitly accounted for, as explained earlier in this chapter. It is assumed to be included in the *C12* mass fraction, or if this is zero, the *O16* mass fraction.

Lastly, the code outputs the following :

OUTPUT MODEL TO DATA SET NUMBER						A
H-R DATA:	AGE6	LOGT	LOGL	BMAG	LOGG	
	B	C	D	E	F	G
APSIDAL MOTION:	K	LOGK				
	B	I	J	K		

Here, *A* denotes the current file being used to store the model data in - the file being named 'fort.A'. *B* is the number of the current time step, and *C* is the number of zones in the model. Numbers *D* - *F* give, respectively, the age in units of 10^6 years, the logarithm of T_{eff} at the photosphere, the logarithm of the luminosity in solar units at the photosphere, the absolute B-magnitude of the model, and the logarithm of the surface gravity in c.g.s. units at the photosphere. *I* gives a number related to the number of zones, *C*, by $I = (99C - 100)/(C + 100)$. *J* and *K* give the apsidal motion and the logarithm of the apsidal motion respectively.

4.15 Sample Output for a Single Timestep

The above format is repeated for each step of evolution, and separated by a row of stars. An example of the output for a single step of an evolution calculation is given below. A $2M_{\text{sun}}$ star in its hydrogen shell burning stage is being modelled.

```

HENYEY : REZONE MODEL WITH FACTOR  1.000000000000
  439  1  0  1  438  0
  439  0  0  0  438  0
  1  1 1.000E+00 5.000E+05 3.350E-02 1.000E+00 1.664E+00 2.078E+02 1.706E+04
  2  1 1.000E+00 5.000E+05 5.443E-03 1.000E+00 1.661E+00 2.082E+02 1.709E+04
  3  1 1.000E+00 5.000E+05 1.367E-04 1.000E+00 1.661E+00 2.082E+02 1.709E+04
  4  1 1.000E+00 5.000E+05 1.389E-05 1.000E+00 1.661E+00 2.082E+02 1.709E+04
MASS LOSS BY NIEUWENHUIJZEN AND DE JAGER
STEP MASS LOSS = 1.020E-05
NUMBER OF ZONES "REMOVED" = 148
TOTAL MASS LOST = 4.540E-03
  CONVECTION MRTD ZONE  48 1.000E+00 9.951E-01 3.636E+04 1.516E-08
                        ZONE  63 1.000E+00 9.910E-01 5.268E+04 4.409E-08
  48  63 7.500E-01 2.500E-01 1.000E-12
  H DEPLETED TO 380 AT M,R,T,D 3.612E-02 1.265E-02 7.423E+07 8.179E+03
-11.020 -11.020 -11.021 -11.021 -11.021 -11.021 -11.021 -11.021
-11.021 -11.021 -11.022 -11.022 -11.023 -11.025 -11.027 -11.031
-11.034 -11.037 -11.039 -11.042 -11.044 -11.046 -11.050 -11.054
-11.061 -11.067 -11.073 -11.079 -11.084 -11.090 -11.103 -11.115
  0.773  2.916  4.764  5.816  6.470  5.733  5.586  5.542
  5.569  5.616  5.687  5.756  5.838  5.913  5.996  6.074
  6.156  6.234  6.313  6.385  6.455  6.518  6.575  6.623
  6.661  6.689  6.705  6.710  6.701  6.682  6.648  6.606
STEP ITNS TIME STEP TOTL TIME  IMASS  CMASS  RADIUS LUMINOSITY
  TP  PP  DP  XP  YP  ZP
  TC  PC  DC  XC  YC  ZC
  ZC12P  ZO16P  ZNE20P  ZMG24P  ZSI28P  ZS32P
  ZC12C  ZO16C  ZNE20C  ZMG24C  ZSI28C  ZS32C
  41 401 5.000E+05 8.422E+08 2.000E+00  1.995E+00  1.649E+00  2.101E+02
      1.704E+04 3.391E+03 1.402E-09  7.500E-01  2.500E-01  1.000E-12
      7.574E+07 8.661E+19 1.566E+04  0.000E+00  1.000E+00  8.324E-09
      4.000E-13 6.000E-13 2.783E-113 2.859E-234 2.868E-315 0.000E+00
      8.323E-09 7.019E-13 3.551E-25  5.465E-43  5.046E-62 1.198E-106
OUTPUT MODEL TO DATA SET NUMBER 103
H-R DATA: AGE6 LOGT LOGL BMAG LOGG
  41 438 8.422E+02 4.231E+00 2.322E+00-5.806E+00 4.304E+00
APSIDAL MOTION: K LOGK
  41 87 2.192E-03-2.659E+00
*****

```

4.16 The Structure of the Code

The code is somewhat densely written, and can be difficult to interpret at many points, even by those with experience in FORTRAN and the particular method of stellar evolution used. Accordingly, the author presents a brief guide to the operations performed by the routine and

subroutines making up the code. Only those subroutines that are pertinent in the evolutionary use of the code will be so described.

The code is stored in three different files, 'mlevol.f', 'sutilsz.f' and 'trceost.f'. The first of these contains the main structure of the Henyey method, the second contains mathematical methods and algorithms, while the third contains subroutines dealing with the underlying physics of the stellar matter. Where possible, the following descriptions have been ordered as the subroutines appear in the listings. Other subroutines are present in the listings, but perform functions that were not needed for the purposes of this work. For example, there are the subroutines needed to calculate aspects of stellar pulsation, or to form initial models if none are available. The full listing of the code may be found in appendix D.

4.16.1 Henyey Routine

This is the main routine of the code, and performs the iterative procedure required to evolve one stellar model from another. The precise computation methods used are described in detail earlier in this work. In essence, the heney routine first determines the new element abundances throughout the model at some later time, using parameters from the input model. It then calls upon the pander and enuczr subroutines for the necessary parameters and derivatives at each zone, proceeds to iterate to form a new model using the process described in chapter three. After the new model has been formed, the effects of mass loss and convection for that timestep are calculated. The only subroutines to be called on after a model has been converged are conv and capsis. The only alterations made to the model once it has been converged, but before it is written to an output file, come from conv and the mass loss algorithm contained in the heney routine. The majority of the modifications made by the author to the code occur within the heney routine, including the section of code to calculate and implement mass loss due to stellar wind.

4.16.2 Kdelta Subroutine

This subroutine defines the Kroneckers delta, $\delta(i, j)$ for integer values of i and j .

4.16.3 Deltak Subroutine

This defines the same function as the kdelta subroutine, but for real values of i and j .

4.16.4 Atmosp Subroutine

This subroutine calculates the values of the variables in the stellar atmosphere, if it is called upon, by an inwards integration towards the photospheric conditions. As mass changes little over the atmosphere, the

optical depth τ is used as the independent variable instead, $\tau = 2/3$ representing the photosphere in the Eddington approximation.

4.16.5 Conv Subroutine

The conv subroutine is used to calculate if and where in the star convection zones exist, and then redistribute chemical abundances in any convection zones. It is assumed that elements will be evenly mixed throughout any zone. In addition, this subroutine checks on whether hydrogen or helium have been exhausted at any points throughout the star. Convection is the last alteration made to any of a converged model's parameters before it is stored for the next cycle of iterations.

4.16.6 Capsis Subroutine

This is used to find the apsidal motion values K and $\log K$, the last values to be printed in an output.

4.16.7 Pander Subroutine

This subroutine organises the calculation of all the derivatives and physical quantities required for the matrix elements of the Henyey and associated matrices described in chapter three. It calls upon the coneqs subroutine.

4.16.8 Coneqs Subroutine

This subroutine also organises the calculation of the derivatives and physical quantities through calling upon the estate, opacity and energy subroutines.

4.16.9 Estate Subroutine

This subroutine calculates the equation of state and associated differentials. It calls upon estatenn, partfn and free.

4.16.10 Free Subroutine

The free subroutine is used to calculate the fraction of particles A and B that are free, from any molecular reaction of the form



These are of course established from the solutions of a quadratic equation for each fraction.

4.16.11 Estatenn Subroutine

This subroutine solves the Saha equations needed to form the equation of state. It calls upon `partfnn` and `fdis`.

4.16.12 Partfn Subroutine

`Partfn` organises the calculation of the partition functions needed before the Saha equations can be solved. It calls upon `partfnn`.

4.16.13 Partfnn Subroutine

`Partfnn` calculates the partition functions needed before the Saha equations can be solved, using the nearest neighbour approximation. It calls upon `erfcl`.

4.16.14 Erfcl Subroutine

This is an error function and complement subroutine, which calculates the incomplete error function for the `partfnn` subroutine.

4.16.15 Opacty Subroutine

This subroutine coordinates the calculation or reading of the opacity at each zone. It also calculated the associated derivatives of the opacity. Although it can fit an opacity, from Christy (1966), or read data from several older sets of opacity tables, this version of the TRC code restricts us to using the OPAL and Carson-Sharp opacity tables. It calls upon the `opal` subroutine for the former, and the `rmkmpz` subroutine for the latter. The earlier opacity tables were calculated by Iglesias and Rogers (1991), and are contained in the subroutines `rkirp1` and `rkirp2`. These are called upon by `rkirpz`, which performs any interpolations needed.

4.16.16 Rmkmpz Subroutine

This subroutine calls upon the Carson-Sharp molecular opacity tables, which are contained in subroutines `rmkmp1`, `rmkmp2`, `rmkmp3` and `rmkmp4`. It then interpolates to find the opacity at the values given.

4.16.17 Rmkmp1 Subroutine

This contains the Carson-Sharp tables for material of a population I composition.

4.16.18 Rmkmp2 Subroutine

This contains the Carson-Sharp tables for material of a population II composition.

4.16.19 Rmkmp3 Subroutine

This contains the Carson-Sharp tables for material of a population III, zero-metal, composition.

4.16.20 Rmkmp4 Subroutine

This contains the Carson-Sharp tables for material consisting entirely of hydrogen.

4.16.21 Opals Subroutine

This reads values for the opacity from the file 'opalt.d', which should contain the OPAL opacity values. The reader is advised to check this section of the code to find the necessary format for 'opalt.d'. This subroutine calls upon the flag subroutine to perform the necessary interpolation in two dimensions (temperature and composition) that the tables require.

4.16.22 Flag Subroutine

As noted above, this routine performs two-dimensional Lagrangian interpolation on values read from the OPAL tables.

4.16.23 Enuc, Enucz, Enucz Subroutines

These subroutines determine the rates of change of element abundances due to various nuclear reactions at points throughout the star, and the energy produced by these reactions. By no means all the reactions we would expect to find in a stellar interior have been included in these routines, but the most important ones have been. In the current version of the code, enucz is the only one of the three subroutines to be called upon. The other two are still present for reference purposes, as they contain less of the nuclear reactions that become important in the later stages of stellar evolution.

4.16.24 Dfis Subroutine

This subroutine is called upon by the fdis subroutine to calculate the inverse of some integrals.

4.16.25 Fdis Subroutine

The fdis subroutine calculates the Fermi-Dirac integrals used in the degenerate electron pressure equations. The middle ranges of these functions, which can otherwise be time-consuming to calculate, are fitted by a method put forward by Cody and Thatcher (1967).

4.17 The Amended Version of the Code

In the initial version of the code, detailed earlier in this chapter, we have assumed that we need not keep track of the mass fraction of ^{14}N . We have also assumed that the mass fraction of metals, Z , can be represented in the equation of state by the simple proportions 0.3, 0.1 and 0.6 of ^{12}C , ^{14}N and ^{16}O respectively. In the later stages of evolution, as the mass fraction of metals increases in a star, we can see that this second approximation will become less and less useful as helium and carbon burning set in. The α -reactions, $^{12}\text{C} - ^{12}\text{C}$ fusion, and at the higher temperatures $^{16}\text{O} - ^{16}\text{O}$ fusion, will progressively change the initial mass fractions of the metals. In addition, failing to take account of the mass fraction of ^{14}N explicitly can lead to problems as the mass fraction of ^{12}C in any part of the star decreases to zero - we can find ourselves burning more carbon than we have!

The amended version of the code changes the equation of state to allow it to take the actual mass fractions, at any point, of ^{12}C , ^{14}N and ^{16}O . As the equation of state considers only these three metals, the mass fraction of oxygen is taken as being equal to the mass fraction of all metals heavier than nitrogen. In addition, the amended version takes account of the mass fraction of ^{14}N explicitly. Hence, it is no longer assumed to be included in the values given at output for the mass fractions of ^{12}C or ^{16}O . These mass fractions now represent only these metals, while the actual mass fraction of ^{14}N is now always given by

$$z_{\text{N}^{14}} = 0.1z \quad . \quad (4.17-1)$$

The use of the two versions of the equation of state is in fact a useful check on its importance in determining the parameters of models generated using the code.

Chapter Five

Checking the TRC Code

5.1 The Consistency of the Code

There are several different assumptions that we have made in order to facilitate the use of the code. For example, that there is no appreciable difference in identical models evolved using different values for the timesteps. In this chapter, we examine the more important of these assumptions, with a view to justifying them in the context of the use of the code.

5.2 Errors Introduced Due to the Choice of Timestep

The author would like to be sure that the code, as a whole, produces self-consistent results. Of especial concern in this respect are the effects of the changes in chemical composition applied after convergence by the sections of the code calculating the effects of mass loss and convection. Fortunately a method of checking on this is immediately obvious - we can evolve two initially identical models to the same point, say to some time Δt , one in i timesteps and the other in $2i$ timesteps. We can then compare the results to check on the self-consistency of the code.

We have used the above method for a pair of zero-age, zero-metal, $15M_{\text{SUN}}$ models. The results are detailed in the following table, where the subscripts 'p' and 'c' denote photospheric and central values, respectively :

Step No.	T (yrs)	L/L _{sun}	R/R _{sun}	T _p	P _p	T _c	P _c
1	5×10^4	2.096×10^4	1.372	5.820×10^4	1.111×10^5	9.917×10^7	2.517×10^{18}
40	2×10^6	2.277×10^4	1.451	5.771×10^4	1.022×10^5	1.002×10^8	2.430×10^{18}
100	5×10^6	2.560×10^4	1.649	5.560×10^4	8.073×10^4	9.980×10^7	2.058×10^{18}
160	8×10^6	2.927×10^4	2.013	5.186×10^4	5.453×10^4	1.001×10^8	1.690×10^{18}
2	5×10^4	2.152×10^4	1.344	5.948×10^4	1.213×10^5	1.030×10^8	2.952×10^{18}
80	2×10^6	2.281×10^4	1.467	5.770×10^4	1.020×10^5	1.002×10^8	2.428×10^{18}
200	5×10^6	2.553×10^4	1.683	5.531×10^4	7.891×10^4	9.970×10^7	2.023×10^{18}
320	8×10^6	2.953×10^4	2.069	5.158×10^4	5.301×10^4	1.002×10^8	1.672×10^{18}

We can see that there is some variance in the parameters listed, but hardly enough to prejudice any results obtained by using the code. The discrepancies are larger initially, and 'smooth out' as the evolution continues.

The most important parameter from this point of view is the temperature, as many of the reaction rates used in the code have a very large temperature-dependence. We can see from the table above that the photospheric and central temperatures vary the least of all the parameters between the two models - a few percent at most. This degree of uncertainty in the parameters of the models evolved using the code should be taken as the minimum uncertainty expected. In any case, it should be small compared to differences due to composition, other changes in the initial conditions of the models, or changes to the parameters used in the code.

5.3 The Choice of Temperature Gradient, ∇ , ∇_{ad} or ∇_{rad}

All of the models to be presented in this work were produced and evolved taking the temperature gradient to be adiabatic in nature, using the value for ∇_{ad} given by equation (2.2-1). This was done in order to slightly speed calculation times by removing the need to calculate ∇ through the cubic equations (2.2-3, 6). In addition, the code has less trouble in converging models in the later stages of evolution if the adiabatic gradient is used.

The author has allowed three identical models to evolve, through approximately 400 timesteps each to a time Δt , using the three different temperature gradients discussed in chapter two. This time Δt was picked as an evolved time at which all the models coincided, otherwise other differences would be introduced. This can be used as a means of showing how valid the assumption of an adiabatic temperature gradient is. The final sets of values from the output files 'fort.7', 'fort.8' and 'fort.10' for each model are tabled below for comparison.

'fort.7'

	B-magnitude	Δt (yrs)	$\log T_{phot}$	$\log (L/L_{sun})$	$\log g$
∇_{rad}	-1.140×10^1	1.227×10^7	4.670×10^1	4.540×10^1	4.740×10^1
∇_{ad}	-1.117×10^1	1.227×10^7	4.670×10^1	4.700×10^1	4.540×10^1
∇	-1.117×10^1	1.227×10^7	4.680×10^1	4.680×10^1	4.600×10^1

'fort.8'

	∇_{rad}	∇_{ad}	∇
Δt (yrs)	1.227×10^7	1.227×10^7	1.227×10^7
$\Delta M/M_{sun}$	8.430×10^{-4}	6.250×10^{-5}	6.180×10^{-5}
L/L_{sun}	3.480×10^4	4.990×10^4	4.830×10^4
T_{phot}	4.700×10^4	4.630×10^4	4.760×10^4
y_{phot}	2.500×10^{-1}	2.500×10^{-1}	2.500×10^{-1}
z_{phot}	1.420×10^{-117}	2.280×10^{-102}	2.280×10^{-103}
$\Sigma (\Delta M/M_{sun})$	1.210×10^{-1}	1.730×10^{-1}	1.650×10^{-1}

'fort.10'

	∇_{rad}	∇_{ad}	∇
Δt (yrs)	1.227×10^7	1.227×10^7	1.227×10^7
m_{cc}/M	0.000	0.133	0.108
r_{cc}/R	0.000	0.442	0.469
m_{hdc}/M	0.000	0.203	0.184
r_{hdc}/R	0.000	0.590	0.630
m_{hedc}/M	0.000	0.000	0.000
r_{hedc}/R	0.000	0.000	0.000

In the above tables, the subscripts '*phot*' denote values taken at the photosphere, '*cc*' denote the convective core, '*hdc*' denote the hydrogen-depleted core and '*hedc*' denote the helium-depleted core. *Y* and *Z* are the mass fractions of helium and heavier elements, respectively.

We can see that there are differences of a few percent between most of the values given for ∇ and ∇_{ad} , of the same order as the differences due to changes in time step shown previously. It should be noted that these models each took a different number of timesteps to reach the same point in time Δt . This is unavoidable given the way in which the code changes the time step automatically during a run. However, this does mean that differences between values may have been introduced in this way. The largest difference is in the photospheric value for *Z* - an order of magnitude. These values for *Z* are far too small, however, for this difference to be regarded as a problem. In fact mass fractions this small are purely the product of arithmetic - if we consider them compared to the $\sim 10^{57}$ particles in the star, then we can see that they should be taken as zero. In addition, we have that the sizes of the various cores differ by 10-20%. In the case of the convective core this can be regarded as a direct consequence of changing the temperature gradient, convection of course depending directly upon the value of ∇ throughout the model. Of course, there will be no convection in the completely radiative model. For the hydrogen-depleted core, these differences are more of a hindrance, as they will reduce the importance of any changes we may observe in internal structure between models of different metallicities.

These differences described in this section should be born in mind when examining the rest of the model data presented in this work. We can see that the values for the convective and depleted cores are perhaps not as reliable as the other parameters given for a model. However, the above results would seem to indicate that the use of the adiabatic temperature gradient is a good approximation to make, in terms of a compromise between accuracy and speed of computation.

5.4 The Behaviour of the Rate of Mass Loss With Metallicity

It is interesting to note the behaviour of the mass loss formula given by Nieuwenhuijzen and de Jager (1990), equation (4.9.1-1), with decreasing metallicity. If we assume mass loss only due to a radiation-driven stellar wind then we should be able to say that as metallicity decreases, so the opacity due to H^- ions will also decrease. Hence, there will be less interaction between the outflowing radiation and the stellar material, so a lower velocity for the stellar wind, and thus a lower rate of mass loss. It has been noted by Kudritzki, Pauldrach and Puls (1987), using a similar argument, that the average mass loss rate of bright stars appears to be lower in the Magellanic clouds than in our galaxy. The Magellanic clouds of course have a lower metallicity than that of our galaxy.

Using the code, the author formed a series of $1M_{\text{sun}}$ zero-age models, with metallicities varying from that expected in population I stars to zero-metal objects, and calculated a mass loss rate from equation (4.9.1-1) for each. The mass of the models was chosen to simplify the equation, as the mass term in (4.9.1-1) now simply becomes unity. In the following table the rate of mass loss $\Delta M/\Delta t$ is given in solar mass units per year.

Z	R/R _{sun}	L/L _{sun}	$\Delta M/\Delta t$
2×10^{-2}	9.829×10^{-1}	5.372×10^{-1}	7.172×10^{-16}
1×10^{-2}	1.040×10^0	7.977×10^{-1}	1.316×10^{-15}
1×10^{-3}	9.252×10^{-1}	1.459×10^0	2.822×10^{-15}
1×10^{-4}	8.467×10^{-1}	1.619×10^0	3.045×10^{-15}
1×10^{-5}	8.328×10^{-1}	1.638×10^0	3.054×10^{-15}
1×10^{-6}	8.297×10^{-1}	1.646×10^0	3.066×10^{-15}
1×10^{-7}	8.311×10^{-1}	1.641×10^0	3.057×10^{-15}
1×10^{-8}	8.313×10^{-1}	1.641×10^0	3.058×10^{-15}
1×10^{-9}	8.313×10^{-1}	1.641×10^0	3.058×10^{-15}
1×10^{-10}	8.313×10^{-1}	1.641×10^0	3.058×10^{-15}
1×10^{-12}	8.313×10^{-1}	1.641×10^0	3.058×10^{-15}
0	8.313×10^{-1}	1.641×10^0	3.058×10^{-15}

We can see that for metallicities $Z \leq 10^{-8}$, changes in the value of Z cease to affect the luminosity and radius of the model, and thus also the rate of mass loss by equation (4.9.1-1). This desensitising of stellar parameters to changes in Z at very low values of Z is well known. See, for example, the work by Wagner (1974) or Chieffi and Tornambé (1986). We should note that by equation (4.9.1-1), we have that the rate of mass loss $\Delta M/\Delta t$ increases with decreasing metallicity Z. This is contrary to the predictions of Kudritzki, Pauldrach and Puls. Given the differences between our methods, it is probable that their prediction is the right one. However, they were not considering stars with metallicities as low as the ones we are ultimately interested in. In any case, the actual rates of mass loss given by equation (4.9.1-1) are very low, and thus should not affect the the general results of the work, even if overestimated.

Chapter Six

Results

6.1 The Aims of the Work

At this point, as we prepare to present our results, it would be useful to restate our aims in this work. From the results presented in this chapter, we hope to show that small differences in levels of metallicity, near $Z=0$, will result in changes in stellar structure and evolution. It has been suggested that the stellar parameters, R , T and L become insensitive to changes in metallicity below $Z \sim 10^{-6}$. See, for example, the work by Wagner (1974). However, we hope to show that the internal structure of a star is not insensitive to these changes. We should also be able to use the results to come to some conclusions as to the composition of the material returned to the interstellar medium by stars of population III due to mass lost during their evolution. We hope to present a case for a population III that produces metals at a high enough rate to explain the lowest observed metallicities in the oldest examples of population II stars. We should also, from our results of the yields of population III stars at different masses and metallicities, be able to present a range of possible distribution functions for a population III. Further conclusions and speculation can follow these results.

6.2 The Choice of Initial Mass and Metallicity

In this work the author will give the results from modelling the evolution of a series of population III stars. The author has selected models with masses of $2M_{\text{sun}}$, $5M_{\text{sun}}$, $10M_{\text{sun}}$ and $15M_{\text{sun}}$. These masses are chosen because we are concerned with the early enrichment of the interstellar medium. Larger stars, in general, burn their nuclear fuel more quickly and thus have shorter lifetimes. See, for example, Arnett (1978) for a discussion of the yields of heavy elements at varying stellar masses. The author has chosen to evolve models, at these masses, with three different chemical compositions. Each set of models has a uniform initial chemical distribution, and can be classed as follows :

Population IIIa ; $X=0.25$ $Y=0.75$ $Z=10^{-10}$

Population IIIb ; $X=0.25$ $Y=0.75$ $Z=10^{-12}$

Population IIIc ; $X=0.25$ $Y=0.25$ $Z=0.00$

where X , Y and Z are the mass fractions of hydrogen, helium and metals respectively. The initial split of Z is assumed to be 0.3, 0.1 and 0.6 for carbon, nitrogen and oxygen respectively. We should note that $Z \sim 10^{-10} - 10^{-12}$ is the expected range of metals that would have been produced by cosmological mechanisms in the early universe. See, for example, the work by Applegate, Hogan and Scherrer (1988) or Steigman (1985).

These twelve models, four different masses for each of populations IIIa, IIIb and IIIc, were each evolved using the initial code and the amended code described in section 4.17. Thus the evolution of twenty-four models will be presented in total.

6.3 Other Parameters

The stellar atmospheres exterior to the photosphere possessed by the models were not considered in this work. All of the models were evolved with a consideration of mass loss and the associated chemical changes in their outer layers. All of the models were evolved until the code consistently failed to converge, or the timestep necessary for convergence fell below 10^{-4} of the total time evolved at that point. At the points at which the models failed to converge for the first time, the 'patch-on' acceleration term described in chapter four was introduced into the equation of hydrostatic equilibrium. The convergence criterion itself was set at 10^{-4} of the root mean square value of all the corrections made throughout the model. As was noted in the previous chapter, all the models were evolved using the approximation of adiabatic convection, ~~was considered~~. All models were generated using the equation of energy conservation given in (4.13-2).

6.4 Tables of Properties for Zero-Age and Evolved Models

The following tables give various properties for the models evolved in this work, at zero age and at the point of core helium exhaustion (for the $15M_{\text{SUN}}$ and $10M_{\text{SUN}}$ models) or the point at which core helium ignition is about to occur (for the lower mass models). In these tables, the subscripts '*phot*' indicate photospheric values and '*hdc*' indicates a value associated with the hydrogen-depleted core. The subscript '*c*' indicates a value at the centre of a model. Where appropriate, the figures given in these tables are either in solar units or c.g.s. units.

An important note; in this work, 'depleted' is given its meaning of 'depleted to zero' or 'exhausted' when used in connection with the cores of models. For example, a hydrogen-depleted core, in the context of this work, is composed entirely of helium and heavier elements.

15M_{sun}

Initial Code

Amended Code

	Pop IIIa	Pop IIIb	Pop IIIc	Pop IIIa	Pop IIIb	Pop IIIc
Zero-Age						
Δt (yrs)	-	-	-	-	-	-
log (L/L _{sun})	4.328	4.342	4.342	4.328	4.342	4.342
R/R _{sun}	1.462	1.355	1.351	1.462	1.355	1.351
log T _{phot}	4.763	4.783	4.783	4.763	4.783	4.783
T _c (K)	9.783×10 ⁷	1.064×10 ⁸	1.066×10 ⁸	9.783×10 ⁷	1.064×10 ⁸	1.066×10 ⁸
P _{phot}	1.078×10 ⁵	1.322×10 ⁵	1.322×10 ⁵	1.078×10 ⁵	1.322×10 ⁵	1.322×10 ⁵
P _c	3.431×10 ¹⁸	2.401×10 ¹⁸	3.399×10 ¹⁸	3.431×10 ¹⁸	2.401×10 ¹⁸	3.399×10 ¹⁸
ρ_c	1.578×10 ²	2.057×10 ²	2.072×10 ²	1.578×10 ²	2.057×10 ²	2.072×10 ²
Evolved						
Δt (yrs)	1.205×10 ⁷	1.270×10 ⁷	1.238×10 ⁷	1.251×10 ⁷	1.262×10 ⁷	1.255×10 ⁷
log (L/L _{sun})	4.722	4.726	4.710	4.735	4.720	4.712
R/R _{sun}	3.861	4.024	3.653	4.903	3.845	3.724
log T _{phot}	4.646	4.638	4.655	4.636	4.647	4.653
T _c (K)	2.304×10 ⁸	2.323×10 ⁸	2.238×10 ⁸	2.946×10 ⁸	2.325×10 ⁸	2.285×10 ⁸
P _{phot}	2.071×10 ⁴	1.922×10 ⁴	2.330×10 ⁴	1.869×10 ⁴	2.091×10 ⁴	2.285×10 ⁴
P _c	2.979×10 ¹⁹	3.029×10 ¹⁹	2.658×10 ¹⁹	8.971×10 ¹⁹	3.157×10 ¹⁹	2.941×10 ¹⁹
ρ_c	2.078×10 ³	2.084×10 ³	1.897×10 ³	5.060×10 ³	2.178×10 ³	2.055×10 ³
ΔM (M _{sun})	1.768×10 ⁻¹	1.893×10 ⁻¹	1.775×10 ⁻¹	1.905×10 ⁻¹	1.825×10 ⁻¹	1.785×10 ⁻¹
mhdc/M	2.167×10 ⁻¹	2.167×10 ⁻¹	2.119×10 ⁻¹	2.216×10 ⁻¹	2.167×10 ⁻¹	2.167×10 ⁻¹
mhedc/M	-	-	-	-	-	-

10M_{sun}

Initial Code

Amended Code

	Pop IIIa	Pop IIIb	Pop IIIc	Pop IIIa	Pop IIIb	Pop IIIc
ZAMS						
Δt (yrs)	-	-	-	-	-	-
log (L/L _{sun})	3.830	3.853	3.835	3.830	3.853	3.835
R/R _{sun}	1.277	1.259	1.258	1.277	1.259	1.258
log T _{phot}	4.668	4.672	4.672	4.668	4.672	4.672
T _c (K)	8.020×10 ⁷	8.152×10 ⁷	8.154×10 ⁷	8.020×10 ⁷	8.152×10 ⁷	8.154×10 ⁷
P _{phot}	5.647×10 ⁴	5.845×10 ⁴	5.848×10 ⁴	5.647×10 ⁴	5.845×10 ⁴	5.848×10 ⁴
P _c	2.004×10 ¹⁸	2.155×10 ¹⁸	2.157×10 ¹⁸	2.004×10 ¹⁸	2.155×10 ¹⁸	2.157×10 ¹⁸
ρ_c	1.684×10 ²	1.783×10 ²	1.784×10 ²	1.684×10 ²	1.783×10 ²	1.784×10 ²
Evolved						
Δt (yrs)	2.136×10 ⁷	2.280×10 ⁷	2.043×10 ⁷	2.106×10 ⁷	2.193×10 ⁷	2.205×10 ⁷
log (L/L _{sun})	4.726	4.243	4.286	4.234	4.233	4.248
R/R _{sun}	4.024	2.467	2.482	2.533	2.343	2.496
log T _{phot}	4.638	4.625	4.624	4.617	4.634	4.623
T _c (K)	2.323×10 ⁸	2.059×10 ⁸	2.086×10 ⁸	2.094×10 ⁸	1.980×10 ⁸	2.039×10 ⁸
P _{phot}	2.554×10 ⁴	2.358×10 ⁴	2.343×10 ⁴	2.232×10 ⁴	2.579×10 ⁴	2.320×10 ⁴
P _c	3.250×10 ¹⁹	3.324×10 ¹⁹	3.524×10 ¹⁹	4.006×10 ¹⁹	2.837×10 ¹⁹	3.240×10 ¹⁹
ρ_c	2.795×10 ⁴	2.920×10 ³	3.065×10 ³	3.501×10 ³	2.510×10 ³	2.828×10 ³
ΔM (M _{sun})	5.504×10 ⁻²	5.795×10 ⁻²	5.187×10 ⁻²	5.337×10 ⁻²	5.579×10 ⁻²	5.707×10 ⁻²
m _{hdc} /M	1.886×10 ⁻¹	1.841×10 ⁻¹	6.311×10 ⁻²	1.864×10 ⁻¹	1.841×10 ⁻¹	1.886×10 ⁻¹
m _{hedc} /M	-	-	-	-	-	-

5M_{sun}

Initial Code

Amended Code

	Pop IIIa	Pop IIIb	Pop IIIc	Pop IIIa	Pop IIIb	Pop IIIc
ZAMS						
Δt (yrs)	-	-	-	-	-	-
log (L/L _{sun})	2.889	2.889	2.889	2.889	2.889	2.889
R/R _{sun}	1.104	1.104	1.103	1.104	1.104	1.103
log T _{phot}	4.464	4.464	4.464	4.464	4.464	4.464
T _c (K)	4.978×10 ⁷	4.984×10 ⁷	4.978×10 ⁷	4.978×10 ⁷	4.984×10 ⁷	4.978×10 ⁷
P _{phot}	2.608×10 ⁴	2.608×10 ⁴	2.609×10 ⁴	2.608×10 ⁴	2.608×10 ⁴	2.609×10 ⁴
P _c	1.028×10 ¹⁸	1.032×10 ¹⁸	1.029×10 ¹⁸	1.028×10 ¹⁸	1.032×10 ¹⁸	1.029×10 ¹⁸
ρ_c	1.444×10 ²	1.447×10 ²	1.445×10 ²	1.444×10 ²	1.447×10 ²	1.445×10 ²
Evolved						
Δt (yrs)	7.590×10 ⁷	7.713×10 ⁷	8.062×10 ⁷	7.660×10 ⁷	7.710×10 ⁷	7.600×10 ⁷
log (L/L _{sun})	3.324	3.334	3.372	3.332	3.334	3.263
R/R _{sun}	1.403	1.400	1.535	1.429	1.398	1.398
log T _{phot}	4.516	4.520	4.513	4.513	4.519	4.501
T _c (K)	8.495×10 ⁷	8.847×10 ⁷	9.375×10 ⁷	8.545×10 ⁷	8.869×10 ⁷	8.869×10 ⁷
P _{phot}	2.319×10 ⁴	2.358×10 ⁴	2.139×10 ⁴	2.252×10 ⁴	2.347×10 ⁴	2.231×10 ⁴
P _c	2.210×10 ¹⁹	2.200×10 ¹⁹	3.879×10 ¹⁹	2.436×10 ¹⁹	2.207×10 ¹⁹	2.207×10 ¹⁹
ρ_c	3.999×10 ³	3.832×10 ³	6.285×10 ³	4.370×10 ³	3.833×10 ³	3.833×10 ³
ΔM (M _{sun})	1.084×10 ⁻²	1.119×10 ⁻²	1.227×10 ⁻²	1.106×10 ⁻²	1.119×10 ⁻²	1.086×10 ⁻²
m _{hdc} /M	5.941×10 ⁻²	6.485×10 ⁻²	8.898×10 ⁻²	7.041×10 ⁻²	6.471×10 ⁻²	9.297×10 ⁻⁴
m _{hcdc} /M	-	-	-	-	-	-

2M_{sun}

Initial Code

Amended Code

	Pop IIIa	Pop IIIb	Pop IIIc	Pop IIIa	Pop IIIb	Pop IIIc
ZAMS	-	-	-	-	-	-
Δt (yrs)	-	-	-	-	-	-
log (L/L _{sun})	1.471	1.471	1.471	1.471	1.471	1.471
R/R _{sun}	9.362×10 ⁻¹	9.363×10 ⁻¹	9.363×10 ⁻¹	9.362×10 ⁻¹	9.363×10 ⁻¹	9.363×10 ⁻¹
log T _{phot}	4.145	4.145	4.145	4.145	4.145	4.145
T _c (K)	2.499×10 ⁷	2.499×10 ⁷	2.500×10 ⁷	2.499×10 ⁷	2.499×10 ⁷	2.500×10 ⁷
P _{phot}	3.224×10 ³					
P _c	4.184×10 ¹⁷	4.186×10 ¹⁷	4.188×10 ¹⁷	4.184×10 ¹⁷	4.186×10 ¹⁷	4.188×10 ¹⁷
ρ _c	1.180×10 ²					
Evolved						
Δt (yrs)	8.057×10 ⁸	9.228×10 ⁸	8.191×10 ⁸	8.385×10 ⁸	8.205×10 ⁸	8.742×10 ⁸
log (L/L _{sun})	2.253	2.581	2.332	2.454	2.338	2.417
R/R _{sun}	1.448	1.969	1.637	2.930	1.676	1.676
log T _{phot}	4.254	4.130	4.233	4.137	4.231	4.225
T _c (K)	7.093×10 ⁷	8.874×10 ⁷	7.624×10 ⁷	9.448×10 ⁷	7.551×10 ⁷	7.551×10 ⁷
P _{phot}	3.401×10 ³	4.284×10 ³	7.663×10 ²	9.572×10 ²	3.316×10 ³	2.825×10 ³
P _c	2.934×10 ¹⁹	1.720×10 ²⁰	8.617×10 ¹⁹	4.264×10 ²⁰	9.403×10 ¹⁹	9.403×10 ¹⁹
ρ _c	2.552×10 ⁴	6.125×10 ³	1.552×10 ⁴	5.199×10 ⁴	1.686×10 ⁴	1.686×10 ⁴
ΔM(M _{sun})	3.951×10 ⁻³	7.236×10 ⁻³	4.208×10 ⁻³	4.678×10 ⁻³	4.232×10 ⁻³	5.243×10 ⁻³
m _{hdc} /M	-	2.353×10 ⁻¹	4.916×10 ⁻²	1.663×10 ⁻¹	4.906×10 ⁻²	1.096×10 ⁻¹
m _{hedc} /M	-	-	-	-	-	-

We can compare selected parameters from the zero-age models of population IIIa with the parameters given for models of the same initial mass and metallicity by Cassisi and Castellani (1993). We find that the models presented in this work are more luminous and have higher photospheric temperatures at a given mass. In addition, the core temperatures and densities are higher for models presented in this work. Selected figures from the work by Cassisi and Castellani are given in the table below for comparison.

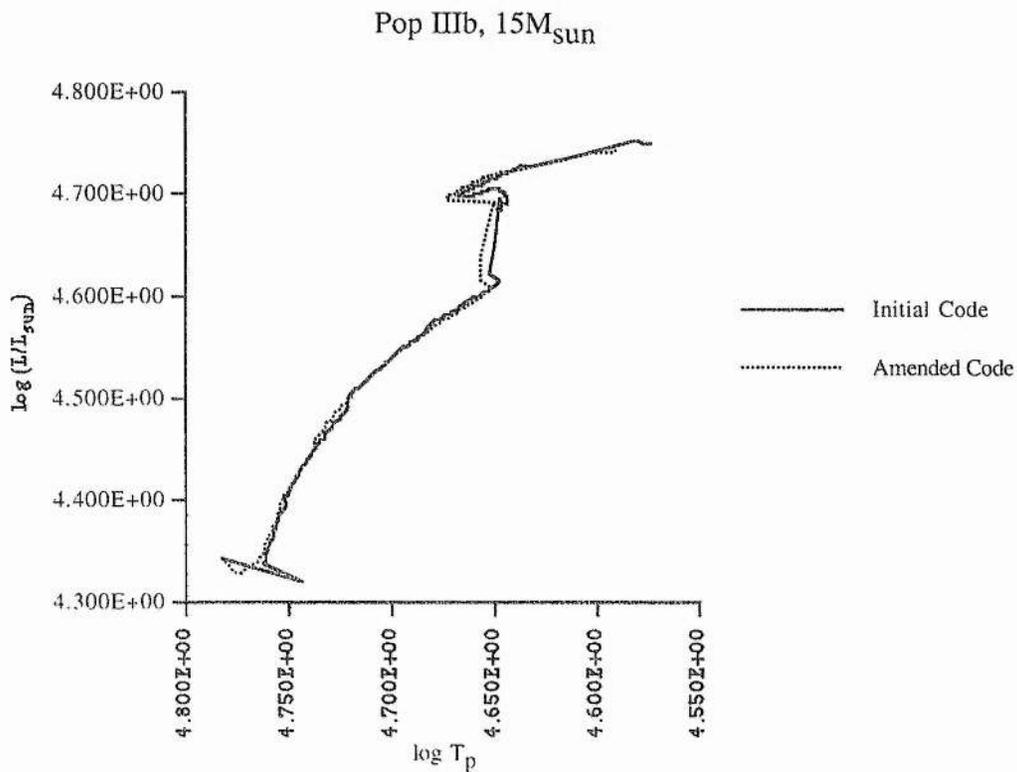
M (M _{sun})	log (L/L _{sun})	log T _{phot}	T _c	ρ _c
15	4.308	4.747	9.376×10 ⁷	1.442×10 ²
10	3.798	4.643	7.430×10 ⁷	1.387×10 ²
5	2.838	4.364	4.624×10 ⁷	1.175×10 ²

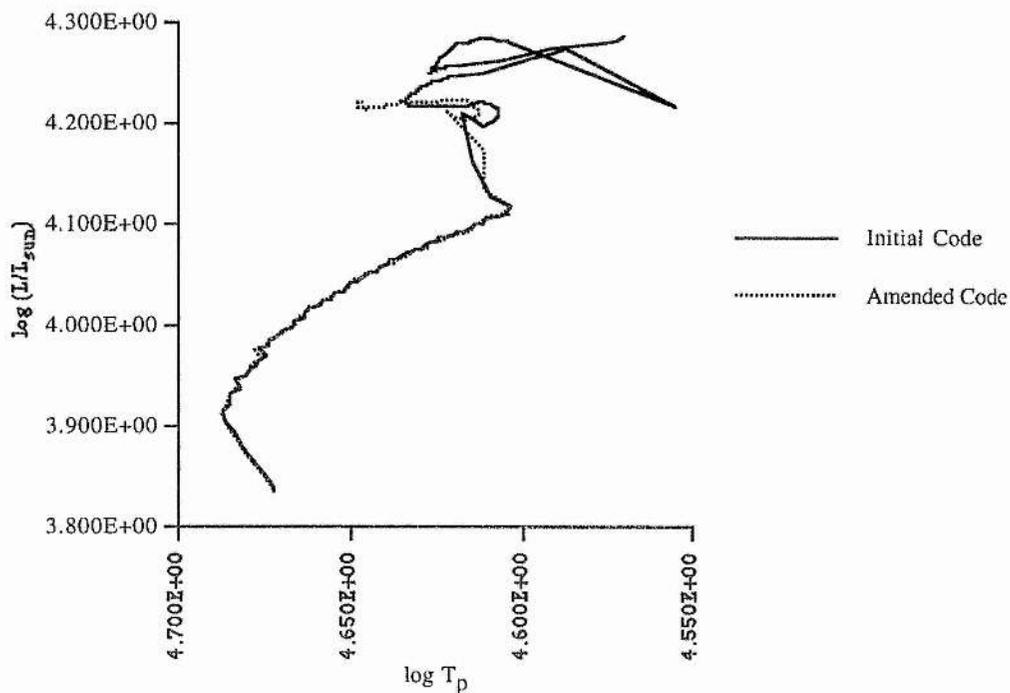
We should note from the preceding tabled results that the total mass lost to these stars is very little in all cases, and as commented upon in the previous chapter, probably overestimated as well.

6.5 Differences in Models Due to the Amendment of the Code

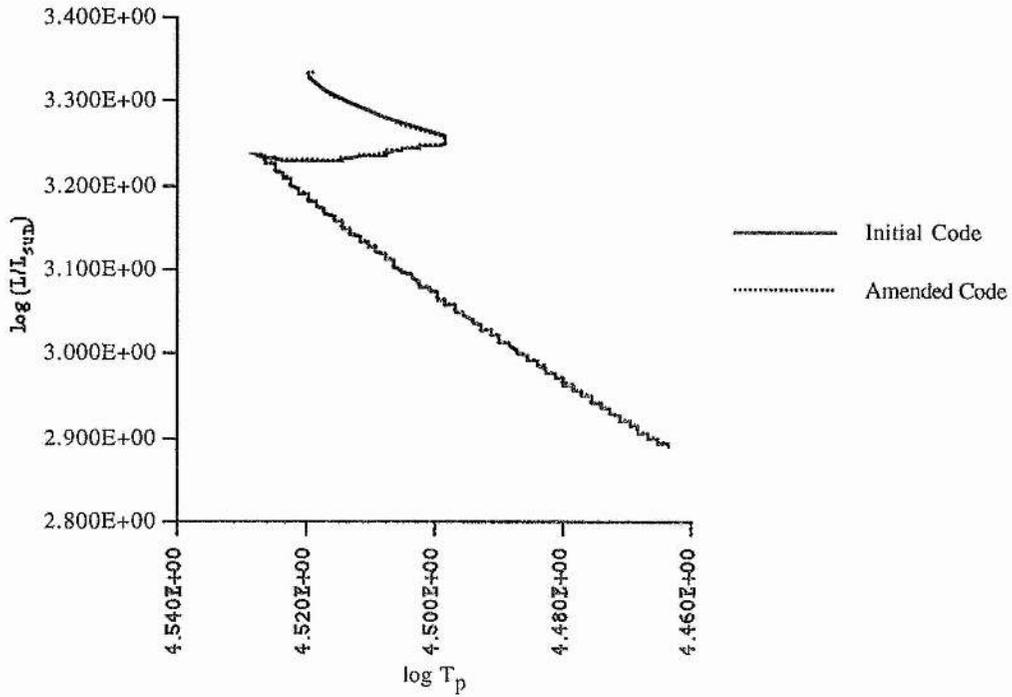
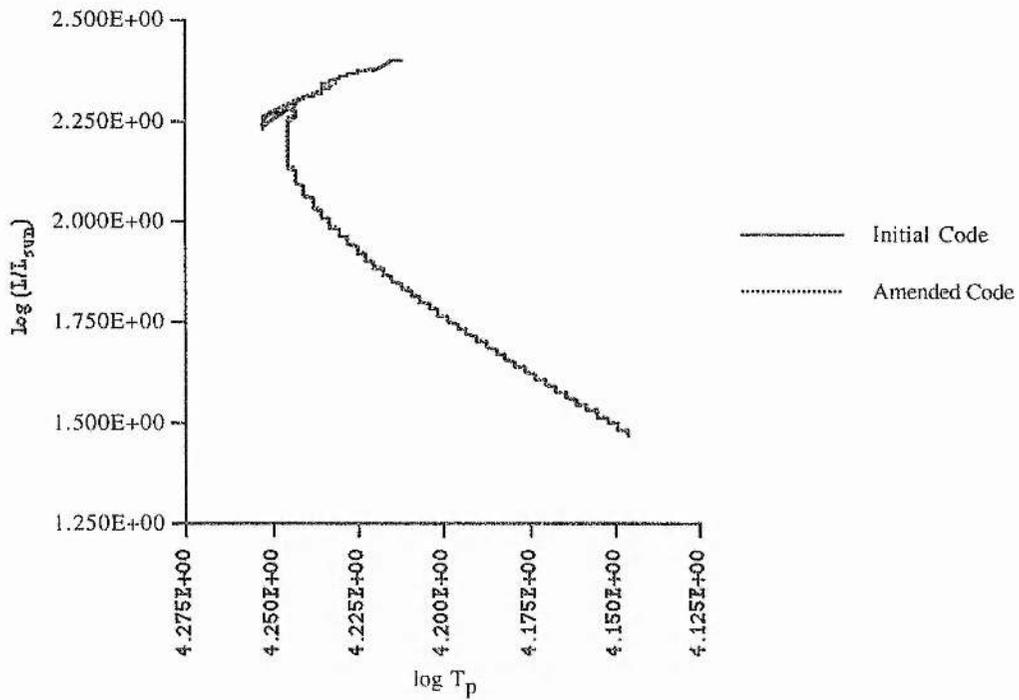
6.5.1 H-R Diagrams

We can show differences in the evolution of the models by the appropriate H-R diagrams. The author has selected the population IIIb models as a median composition to illustrate the differences made by the assumptions in the two versions of the code. Some of the aspects of the diagrams presented in this section will be described more fully later in this chapter. For the moment, the author will concentrate on differences arising from the amendment to the code.



Pop IIIb, $10M_{\text{sun}}$ 

Although there is no helium flash as such in these more massive stars, we can see that the greatest differences occur in T_p at a given luminosity at the point at which hydrogen in the core nears or reaches exhaustion ($4.1 \leq \log L/L_{\text{sun}} \leq 4.2$). In addition, there are the differences in T_p immediately after this point, the start of shell hydrogen burning. This difference in T_p is more pronounced in the $15M_{\text{sun}}$ model than in the $10M_{\text{sun}}$ model.

Pop IIIb, $5M_{\text{sun}}$ Pop IIIb, $2M_{\text{sun}}$ 

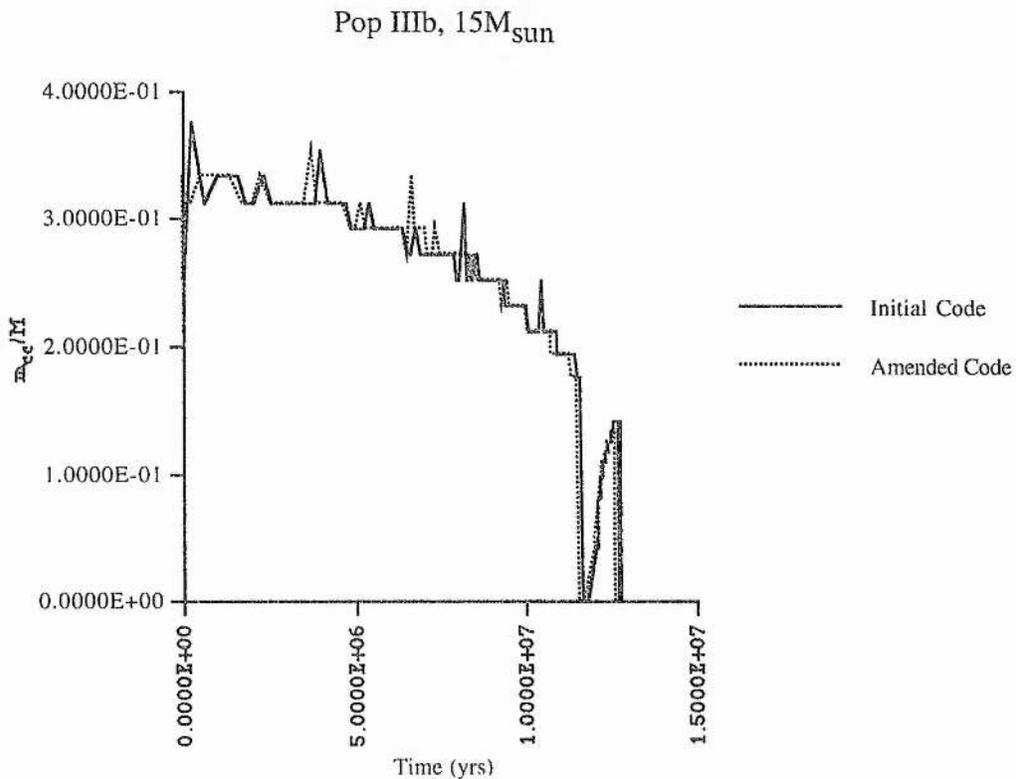
We can see that in the lower mass models, of $5M_{\text{sun}}$ and $2M_{\text{sun}}$, differences in the H-R diagrams are too small to become noticeable, even towards the later stages. At these later stages in a model's evolution, a

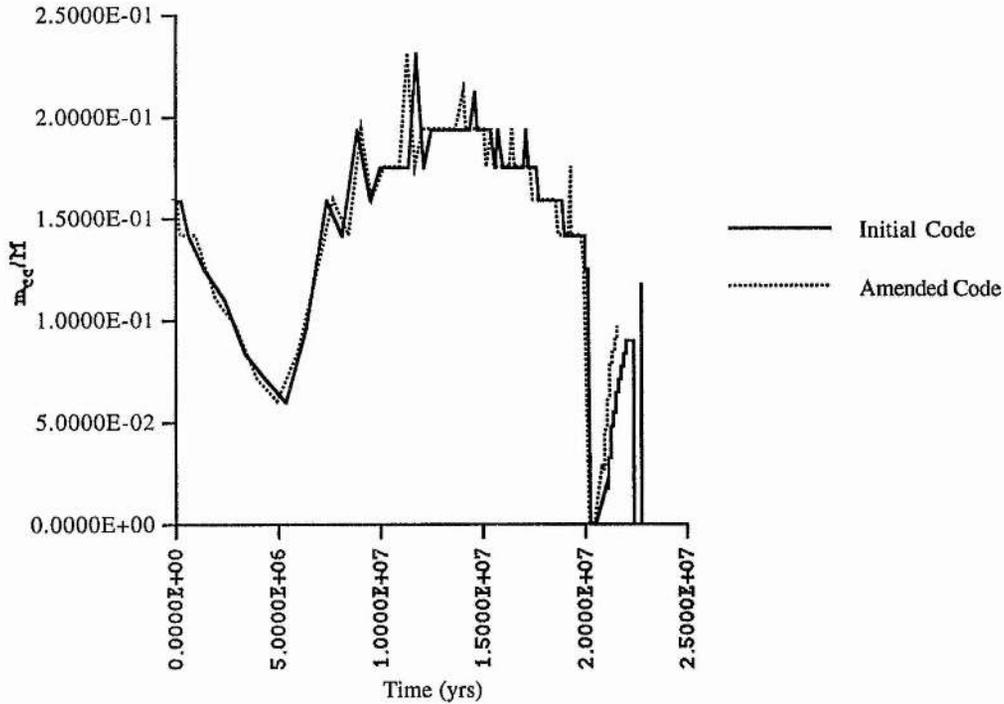
hydrogen-depleted core is forming, and the model is nearing a helium-flash event.

Aside from the regions in the H-R diagrams of the more massive stars where core helium-burning has set in fully, we can view the differences in the H-R diagrams above as differences in luminosity L at a given temperature T . These come about due to radius changes, which we may ascribe to differences in the behaviour of the convection zones in the models. For the $15M_{\text{sun}}$ and $10M_{\text{sun}}$ models, the evolutionary tracks of those evolved with different equations of state diverge as the core hydrogen is nearing exhaustion. The tracks later converge again once the hydrogen at the core is finally exhausted.

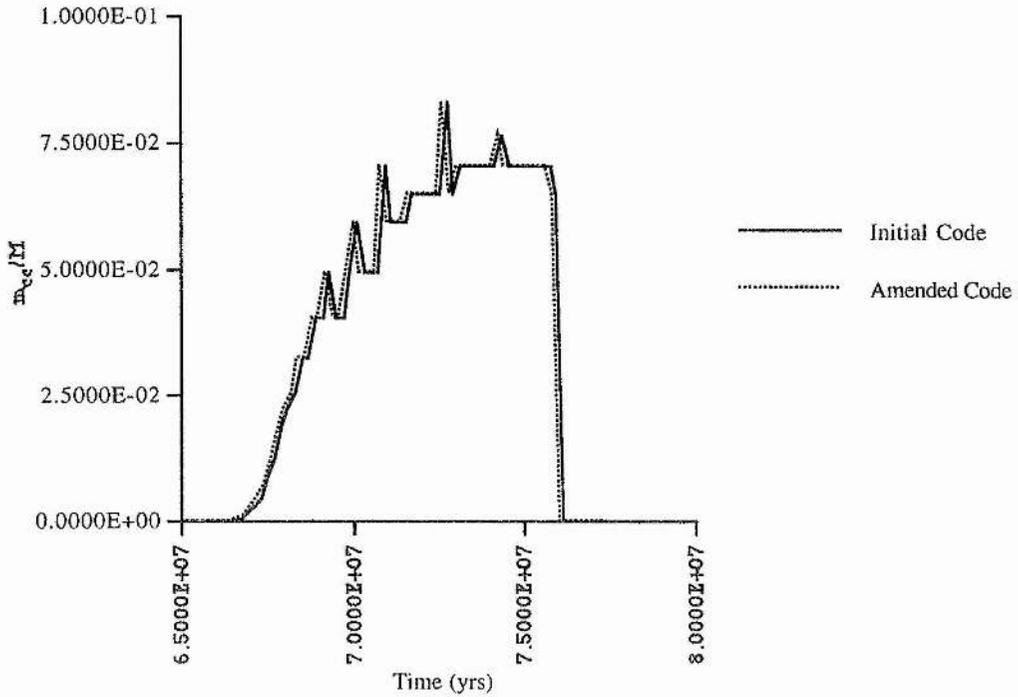
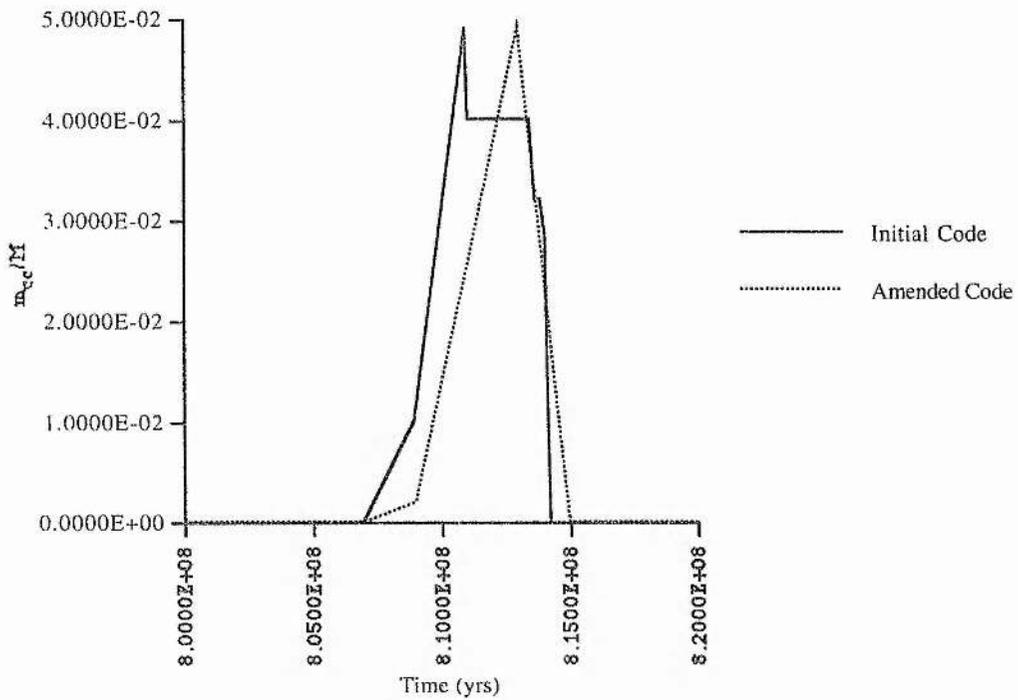
6.5.2 Convective Cores

We are interested in the behaviour of the cores of our evolving models, with changing initial compositions. Hence, it is useful to note how the behaviour of the convective core differs between the initial and amended code.



Pop IIIb, $10M_{\text{sun}}$ 

In the case of the $15M_{\text{sun}}$ and $10M_{\text{sun}}$ models, we find that the radius of the convective core changes very little between the initial and amended code. Hence the minor differences in the mass of the convective core are indicative of small density differences between the cores of the models, or, possibly, of numerical noise introduced by the omission of semiconvection. The way that the plots seem to be displaced from one another would imply that the cores of the stars are going through similar events in their lifetimes, but that the cores calculated using the amended code have evolved slightly faster.

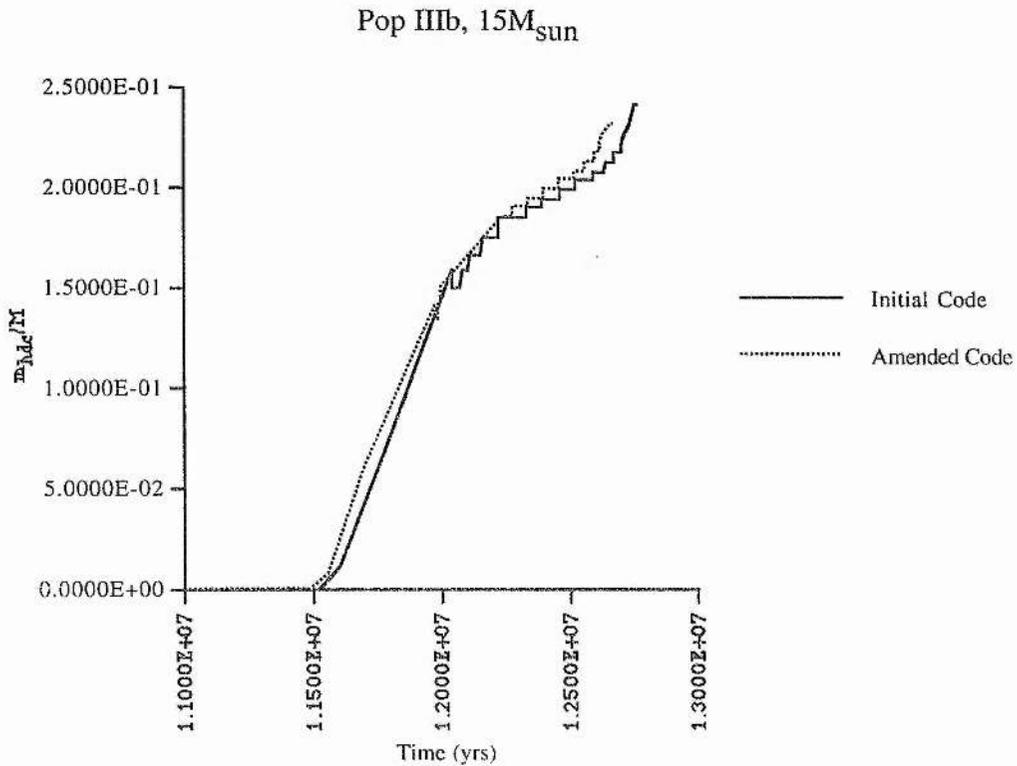
Pop IIIb, $5M_{\text{sun}}$ Pop IIIb, $2M_{\text{sun}}$ 

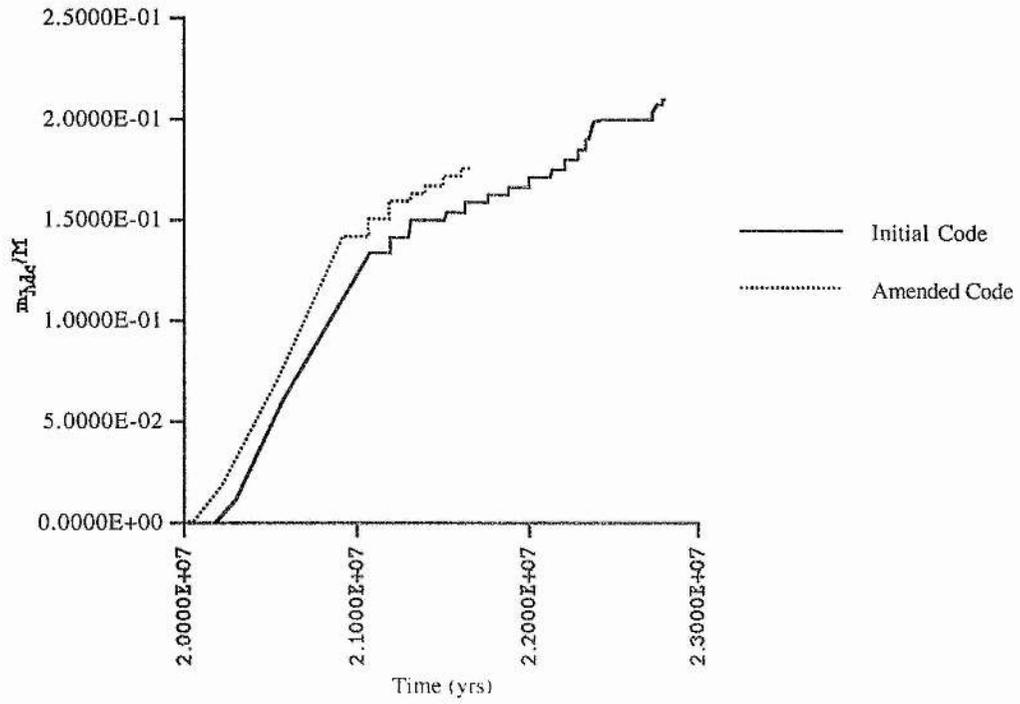
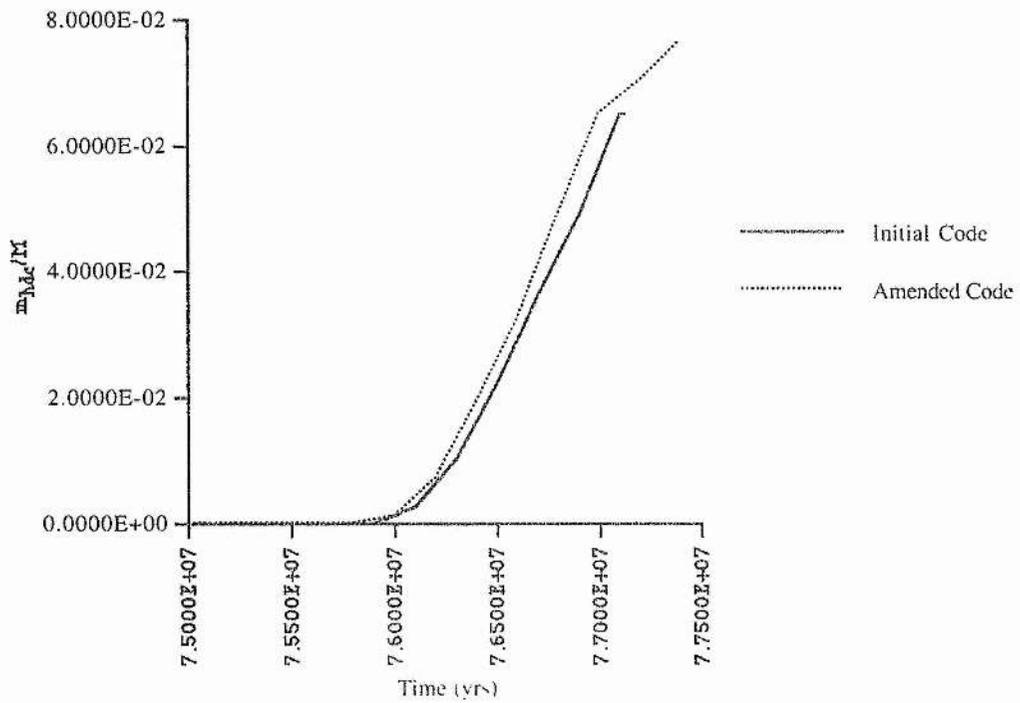
In these lower mass stars, we have concentrated on the second episode of core convection, as the models near hydrogen exhaustion at the core. We can see that in the $5M_{\text{sun}}$ models, the core is again evolving

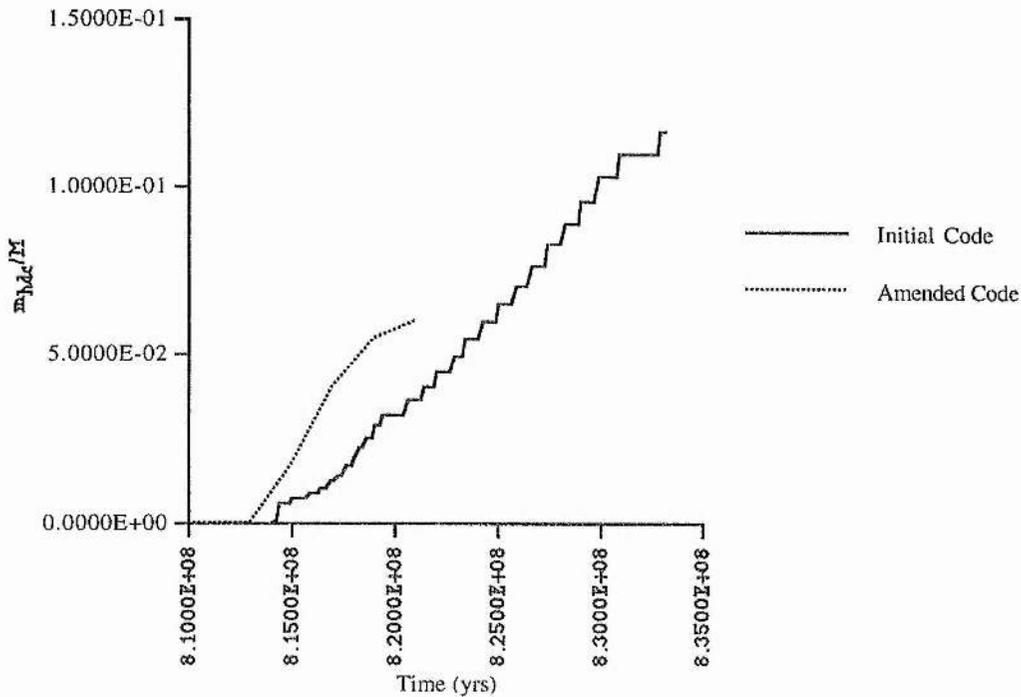
slightly faster under the amended code, while still undergoing the same events as the core evolving under the initial code. However, for the $2M_{\text{sun}}$ models, we can see that the case is the reverse, and the two convective cores are quite different in their behaviour. The structure of the convection zones in the context of the evolution of these stars will be discussed later in this chapter.

6.5.3 Hydrogen-Depleted Cores

The rate at which the hydrogen-depleted core forms in identical models evolved under the initial and amended code are similar, as can be seen from the following diagrams.



Pop IIIb, $10M_{\text{sun}}$ Pop IIIb, $5M_{\text{sun}}$ 

Pop IIIb, $2M_{\text{sun}}$ 

We can see, however, that the hydrogen-depleted cores evolve earlier when the amended code is used to calculate the models. Unlike differences in the H-R diagrams, this difference in hydrogen-depleted cores is equally apparent in the models of a lower mass.

6.5.4 Discussion

In general, we can note from the diagrams in this section that models calculated using the initial code could be evolved for a greater length of time, without departing from hydrostatic equilibrium, than those calculated using the amended code. Models generated by the amended code have evolved convective and hydrogen-depleted cores earlier than models generated by the initial code. However, time differences aside, the structures of these cores are very similar in all but the $2M_{\text{sun}}$ model.

The H-R diagrams show very similar evolutionary tracks of models generated by the initial and amended code until the later stages of evolution. Here, shell hydrogen burning or core helium burning would be the dominant energy source for the more massive models. We can broadly conclude that it is differences in the convective behaviour of the models associated with these strong and concentrated energy sources that are causing the differences we see between these models. This sort of

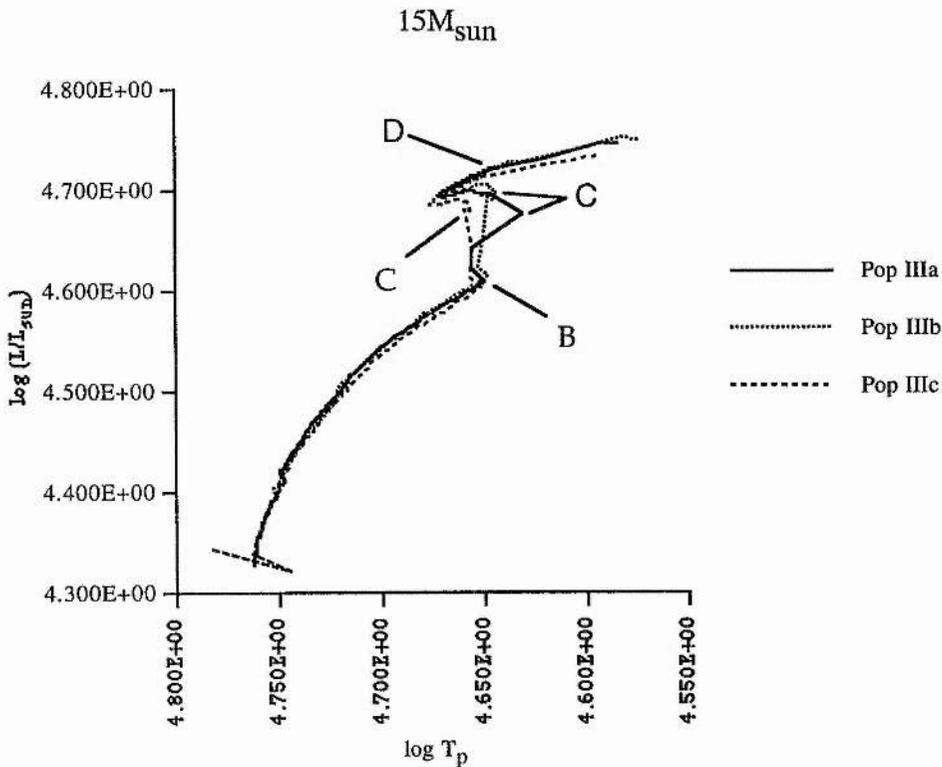
6.6 Differences In Models With Initial Metallicity

In this section, we will present the results for the models generated using the initial code only, for reasons of clarity. Most of the conclusions that are reached in this chapter may be gained from a consideration of either set of models, and so including both here would not help the reader to follow our arguments. The diagrams presented in this chapter, but for models generated using the amended code, can be found in appendix B.

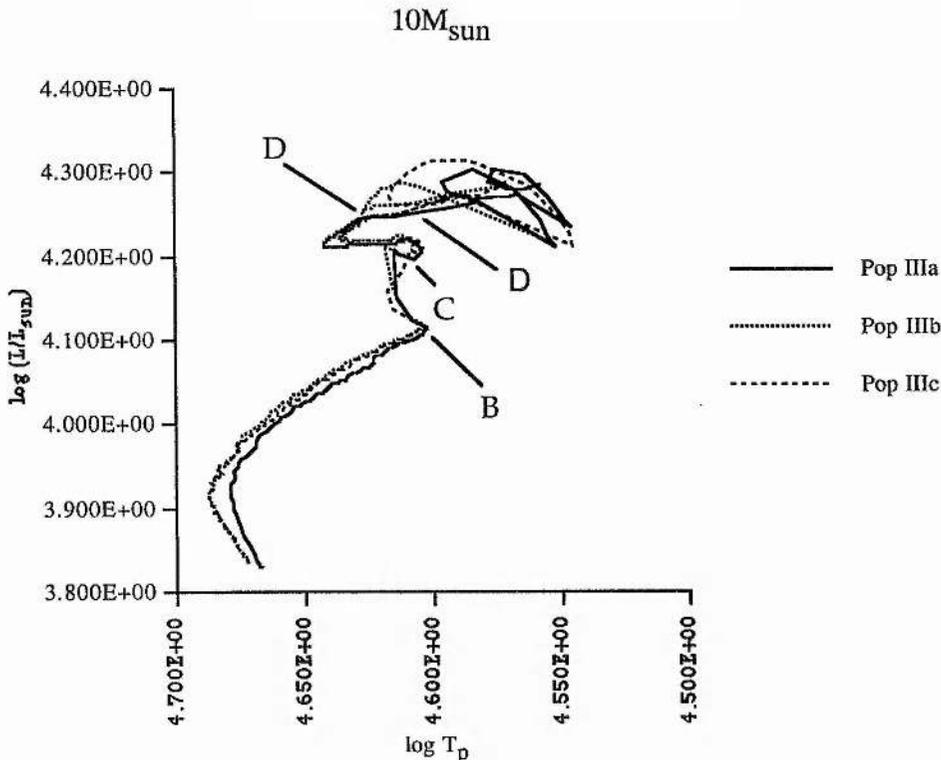
6.6.1 H-R Diagrams

Each of the following H-R diagrams shows evolutionary tracks for one mass with the three different initial compositions (populations IIIa, IIIb and IIIc, corresponding to zero-age values for Z of 10^{-10} , 10^{-12} and zero respectively). We can assign letters to the points of interest in the evolution of our models, and the diagrams are appropriately labelled;

- A - the start of the second episode of core convection in $5M_{\text{sun}}$ and $2M_{\text{sun}}$ models.
- B - the point at which helium burning becomes an important energy source in $15M_{\text{sun}}$ and $10M_{\text{sun}}$ models.
- C - the formation of a hydrogen-depleted core.
- D - the formation of a helium-depleted core.

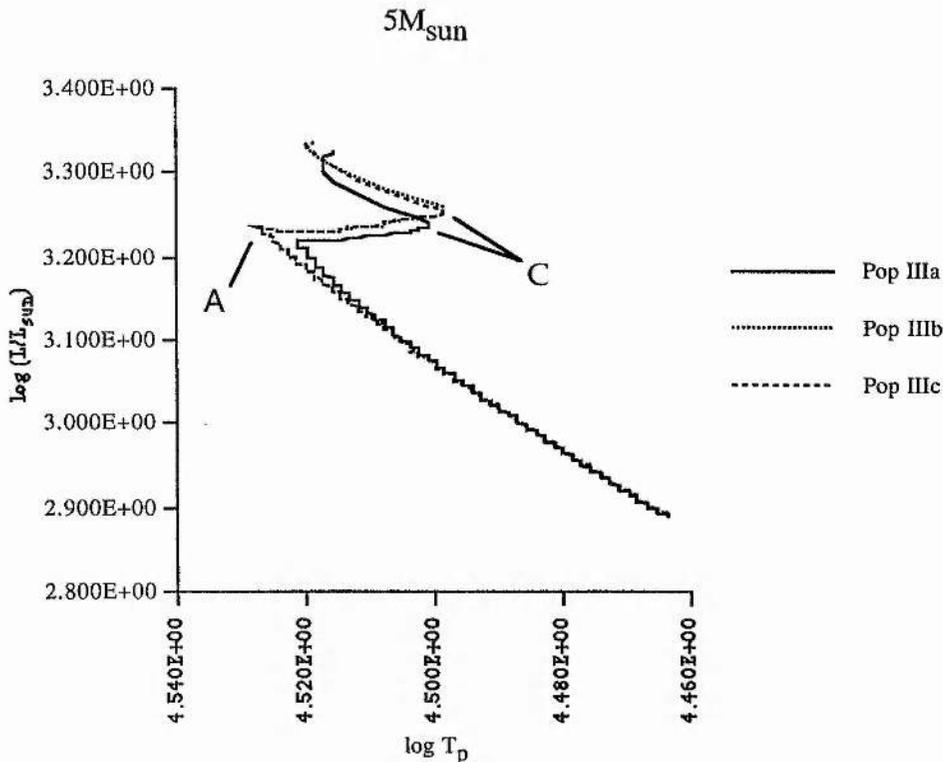


In the H-R diagram for the mass $15M_{\text{sun}}$ models, *B* is the point at which the core conditions of the models allow helium-burning to become an important energy source. This occurs at times of 1.065×10^7 yrs, 1.117×10^7 yrs and 1.104×10^7 yrs for populations IIIa, IIIb and IIIc respectively. The following rapid increase in photospheric luminosity is due to the energy produced by this core helium burning, this increase only slowing at the points labelled *C*. These are the points at which the hydrogen-depleted core begins to form, reducing the energy output at the centre of the models. These are also the points at which the convective core vanishes in these models, to return a little while later at the high-temperature point of the evolutionary track. Both of these events are shown in diagrams later in this chapter. Points *C* represent times of 1.142×10^7 yrs, 1.206×10^7 yrs and 1.174×10^7 yrs for populations IIIa, IIIb and IIIc respectively. *D* is the point at which the model of population IIIc composition begins to form its helium-depleted core, at a time $t = 1.262 \times 10^7$ yrs, reaching a central temperature $T_c = 2.517 \times 10^8$ K. At this point, the convective core vanishes for the second time. Neither of the other two models, of populations IIIa and IIIb, evolved helium-depleted cores before failing to converge.



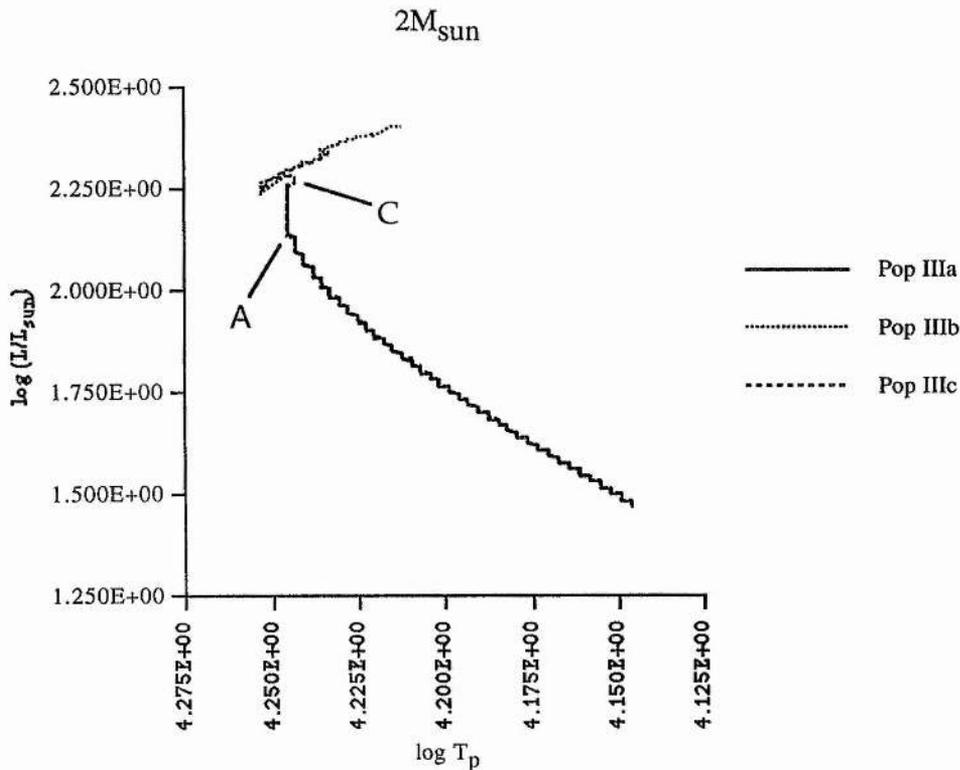
We may now discuss the evolutionary tracks for the $10M_{\text{sun}}$ models, shown above. As for the H-R diagram for the $15M_{\text{sun}}$ models, here *B* is the point at which the core conditions allow helium burning to become an important source of energy. This point corresponds to times of

1.886×10^7 yrs, 2.017×10^7 yrs and 1.784×10^7 yrs for models of population IIIa, IIIb and IIIc respectively. C is the point at which the hydrogen-depleted core begins to form, and the convective core first vanishes. This point occurs at times of 1.875×10^7 yrs, 2.115×10^7 yrs and 1.784×10^7 yrs for models of population IIIa, IIIb and IIIc respectively. As for the $15M_{\text{sun}}$ models, the convective cores in the $10M_{\text{sun}}$ models reappear at the high-temperature point of the evolutionary track after this. We should note, however, that for the $10M_{\text{sun}}$ models, the hydrogen-depleted core evolves earlier in the population IIIc model than in the population IIIa model. The higher temperature point labelled *D* indicates the point at which both the population IIIa and IIIb models form helium-depleted cores. At this point, the convective core vanishes for the second time. These points occur at times $t=2.139 \times 10^7$ yrs and $t=2.236 \times 10^7$ yrs, at core temperatures $T_c=2.209 \times 10^8$ K and $T_c=2.185 \times 10^8$ K for models of population IIIa and IIIb composition respectively. The second label *D* indicates the point at which the population IIIc model forms a helium-depleted core, and at which the convective core vanishes again. This occurs at a time $t=2.055 \times 10^7$ yrs and a core temperature $T_c=2.534 \times 10^7$ K.



In the H-R diagram for models of $5M_{\text{sun}}$, we can note that the differences between the population IIIa model and the models of populations IIIb and IIIc appear more clearly than in the H-R diagrams for the more massive models. We can see that there is little difference between the evolutionary tracks of the models of population IIIb and IIIc at

this mass. As before, the labelled points correspond to important changes in the internal structure of the models. A corresponds to the point at which the second episode of central convection is beginning in the models. This occurs at times of 6.70×10^7 yrs for models of population IIIb and IIIc compositions, and at 6.59×10^7 yrs for the model of population IIIa composition. C indicates the points at which the models begin to form hydrogen-depleted cores, and at which the second episode of central convection ceases. This occurs at times of 7.59×10^7 yrs, 7.71×10^7 yrs and 7.60×10^7 yrs for the models of populations IIIa, IIIb and IIIc respectively.



It can be seen that the $2M_{\text{sun}}$ models show no noticeable differences in evolutionary tracks until after point C. As for the H-R diagram for the $5M_{\text{sun}}$ models, A indicates the point at which the second episode of core convection begins, as hydrogen in the core nears exhaustion. This occurs at times of 8.124×10^8 yrs and 8.110×10^8 yrs for models of population IIIb and IIIc respectively. The author found that the $2M_{\text{sun}}$ model of population IIIa did not experience a second episode of core convection. C indicates the point at which the models began to form hydrogen-depleted cores. At this point, in addition, the second episode of core convection ceased. This occurred at times of 8.144×10^8 yrs and 8.130×10^8 yrs for the models of population IIIb and IIIc respectively. The model of population IIIa composition did not form a hydrogen-depleted core before failing to converge.

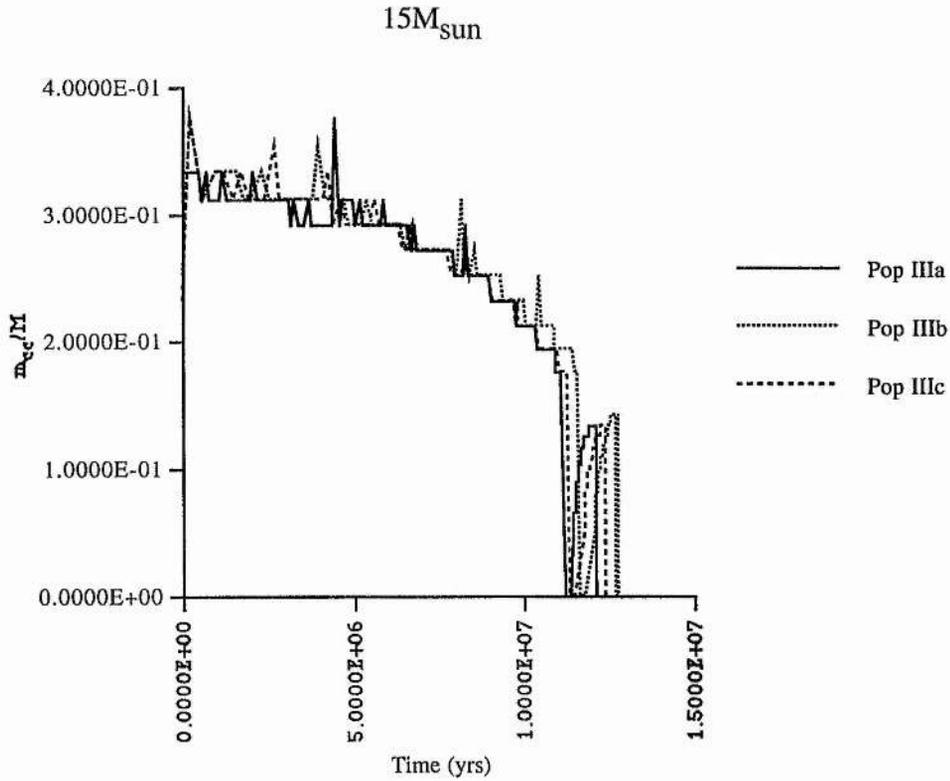
6.6.2 Discussion of the H-R Diagrams

Broadly, we can note that for the more massive models of $15M_{\text{sun}}$ and $10M_{\text{sun}}$ detailed in the previous section, the largest differences occur when the core hydrogen is nearing exhaustion. This stage occurs after core helium burning has become an important source of energy. There are also noticeable differences during the shell hydrogen burning stage after the hydrogen-depleted core has been formed. At least some of these later differences would appear to spring from the different times at which the models exhaust their central helium. The less massive models of $5M_{\text{sun}}$ and $2M_{\text{sun}}$ show fewer differences in their evolutionary tracks, but these also begin when their core hydrogen is nearing exhaustion. None of these less massive models evolved to the point of helium ignition before failing to converge, and their final core temperatures remained lower than the 10^8K required for this event.

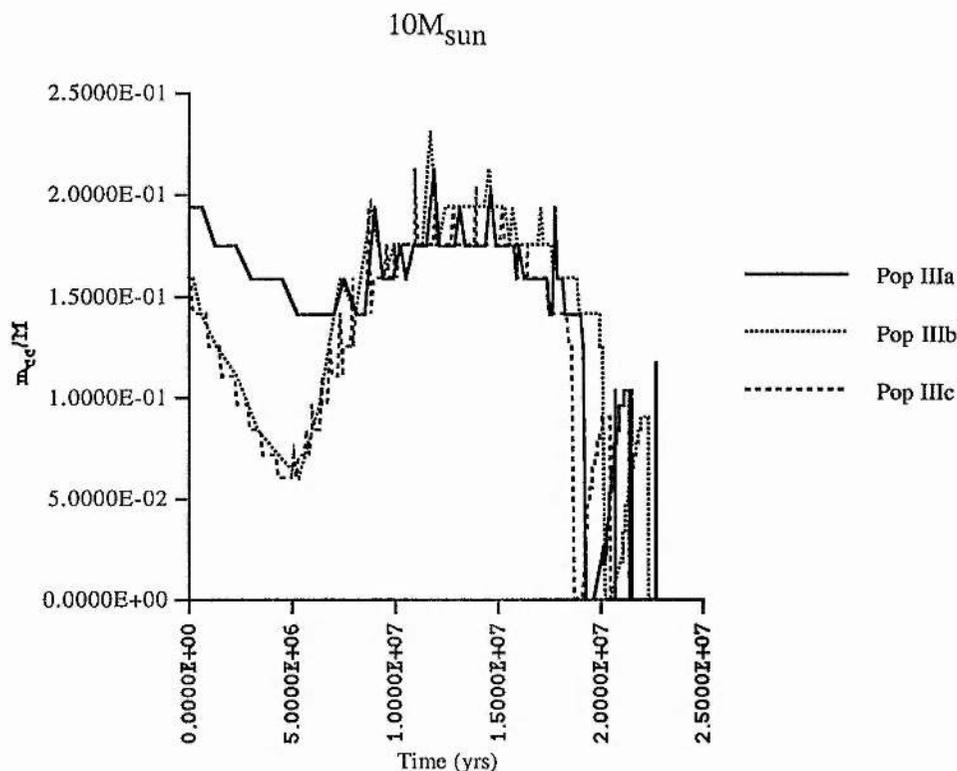
The most important differences between models of population IIIa, IIIb and IIIc are in the evolutionary times at which the events described in the previous section occur. The evolutionary tracks of the models of the same initial mass are very broadly similar, but the same points noted on them occurred at different times. In other words, nearly all of these models evolve through the same events in their lifetimes, but these events occur at different times for models of the same mass but differing initial composition. We can examine the times given for the appearance and disappearance of the convective cores, and for the exhaustion of core hydrogen in all the models. We find that in the $15M_{\text{sun}}$, $5M_{\text{sun}}$ and $2M_{\text{sun}}$ models where these events have occurred, the population IIIa models evolved to these events most quickly, followed by the models of population IIIc and then those of population IIIb. For the $10M_{\text{sun}}$ models, the order was population IIIc, IIIa and then IIIb. Why should they evolve to a given event in this order? This question is discussed at the end of this chapter.

6.6.3 Convective Cores

In this section, we present diagrams depicting the fractional mass comprising the convective cores of the models evolved using the initial code. Diagrams for the models evolved using the amended code may be found in appendix B.

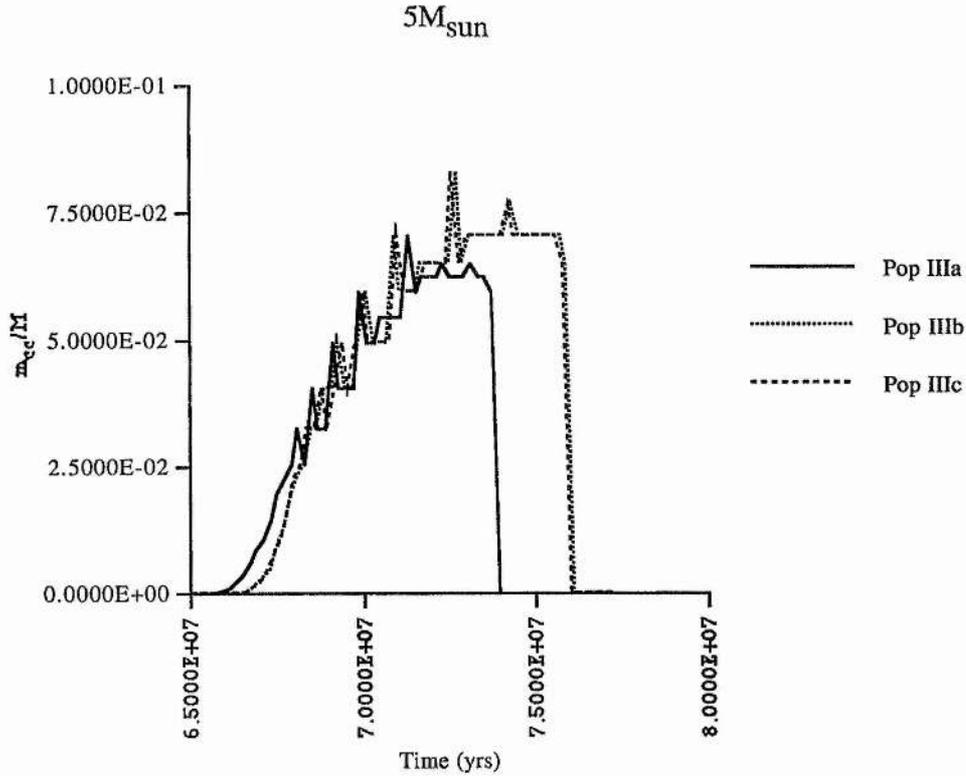


The diagram for the $15M_{\text{sun}}$ models shows that the models of population IIIa, IIIb and IIIc have similar convective core structures. The first time at which the convective core disappears corresponds to points C in the H-R diagram for the $15M_{\text{sun}}$ models, and the time at which the hydrogen-depleted core is beginning to form. The second time at which it disappears corresponds to point D in the H-R diagram, and the central hydrogen becoming exhausted for the population IIIc model, or nearing exhaustion for the population IIIa and IIIb models.

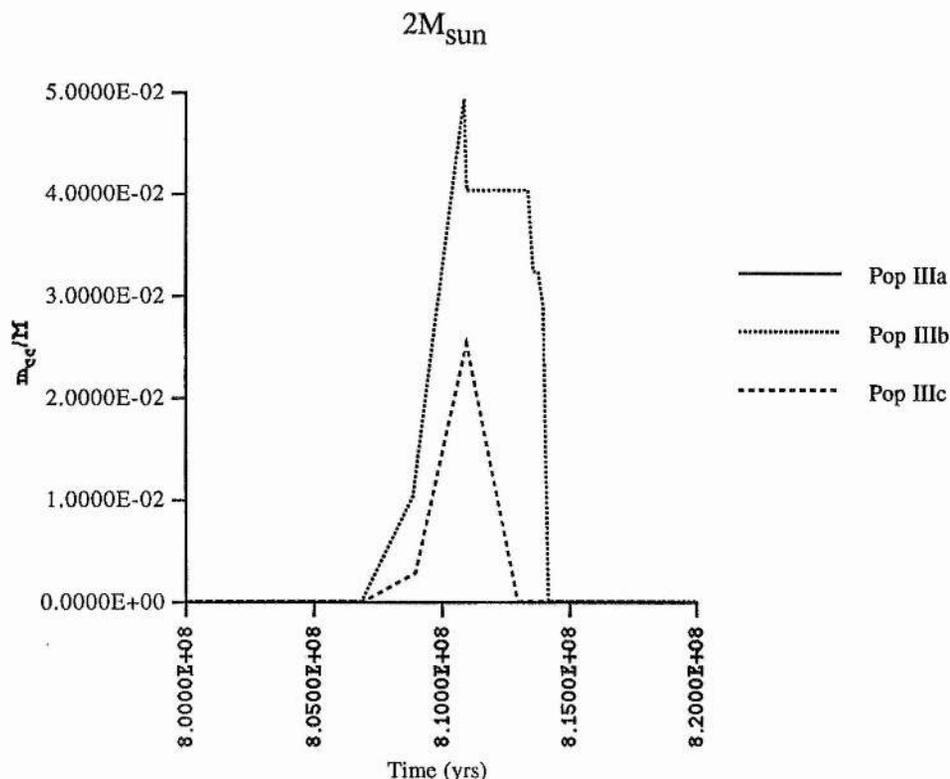


The largest difference shown in the diagram for the $10M_{\text{sun}}$ models is the mass of the convective core early in the evolution of the models. The convective core of the population IIIa model is far more massive than those of the other models - almost twice as massive at a time $t=5 \times 10^6$ yrs. The stage at which the convective core first disappears corresponds to point C in the H-R diagram for $10M_{\text{sun}}$ models, and the stage at which the hydrogen-depleted core starts to form. We should note that here, the population IIIc model reaches this stage in its evolution earlier than the population IIIa model - the reverse of the case for the $15M_{\text{sun}}$ models. The second stage at which the convective core disappears corresponds to points D in the H-R diagram, and the stages at which the helium-depleted cores form in the $10M_{\text{sun}}$ models of different initial composition.

We may now look at the lower mass models of $5M_{\text{sun}}$ and $2M_{\text{sun}}$. These models experience two widely separated periods of convection at their cores. The first is as they are leaving the main sequence, and there are no significant differences in the convective core masses or radii between the different compositions during this period. Accordingly, the diagrams below are of the second convection period, as the models approach hydrogen exhaustion at the core. The full diagrams may be found in appendix C. The end of these second periods of core convection corresponds to the end of the first period of core convection in the more massive models, in that both these points are associated with the exhaustion of hydrogen in the core of a model.



We can see from the diagram for the $5M_{\text{sun}}$ models that there is little difference in the structure of the convective core between the models of population IIIb and IIIc. The convective core of the population IIIa model appears earlier, does not become as massive and disappears earlier than the convective cores of the other two models. The times at which these convective cores appear correspond to point A in the H-R diagram for the $5M_{\text{sun}}$ models. The times at which the convective cores disappear correspond to the points labelled C in the H-R diagram, and the times at which the hydrogen-depleted core begins to form.



We can see that the $2M_{\text{sun}}$ model of population IIIa did not experience a second period of core convection. This would eliminate the convective process as being responsible for this model not evolving a hydrogen-depleted core, and we must look elsewhere for an explanation. The convective core of the population IIIc model is much less massive than that of the population IIIb model, although there is little difference in the duration of this second period of convection between the two models. The appearance of these convective cores corresponds to point A in the H-R diagram for $2M_{\text{sun}}$ models, as the core hydrogen is nearing exhaustion. The disappearance of the convective cores corresponds to point C in this H-R diagram, and the time at which the hydrogen-depleted core begins to form.

6.6.4 Discussion of the Convective Core Diagrams

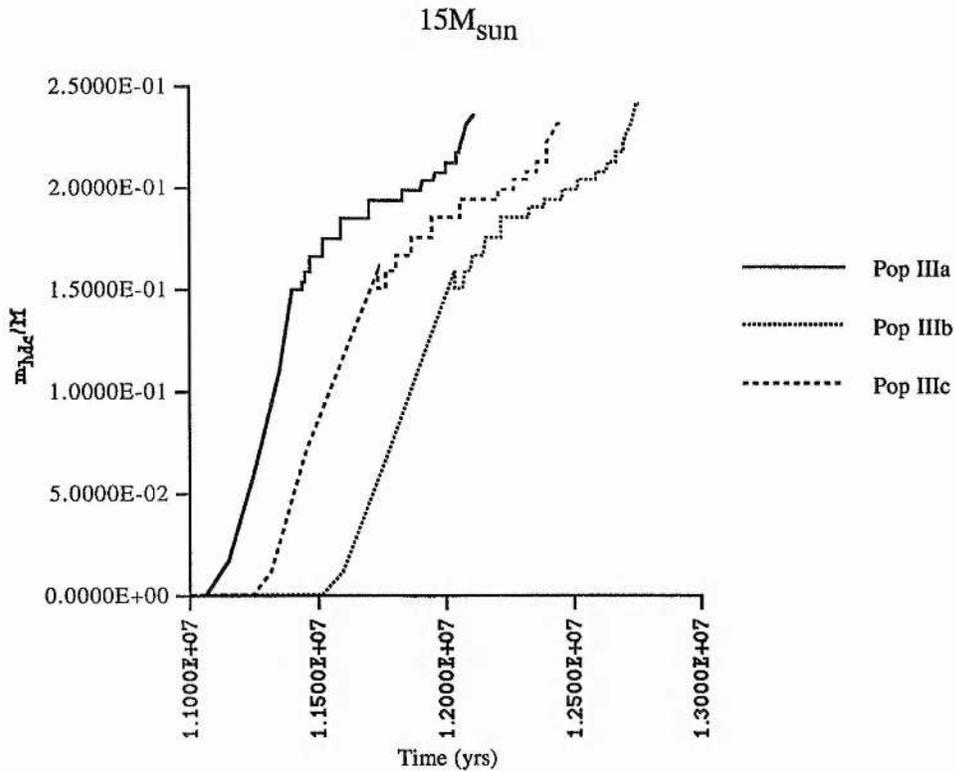
The differences in the evolution of the convective cores between the three initial compositions are varying according to the mass of the models. However, there are a few broad and sweeping statements that can be made about the changes in core convection with initial composition. The times at which the convective core of a model vanishes are connected with the exhaustion of hydrogen or helium at the cores of the models. This can be noted from the section describing the H-R diagrams associated with the models presented in this work. If differences in convective cores are associated with chemical changes, then this would be in accord with the lack of differences between the initial convective episodes in the lower

mass models. At these models zero-age, any chemical differences are still minor. Thus there will be no noticeable differences in the convective core masses. By way of comparison, there are noticeable differences with initial compositions in the early evolution of the convective cores of the more massive models. The reasons for this will be discussed at the end of the chapter.

As a final note, it is possible that the jagged nature of the convective core diagrams is due to the neglect of semiconvection in the code.

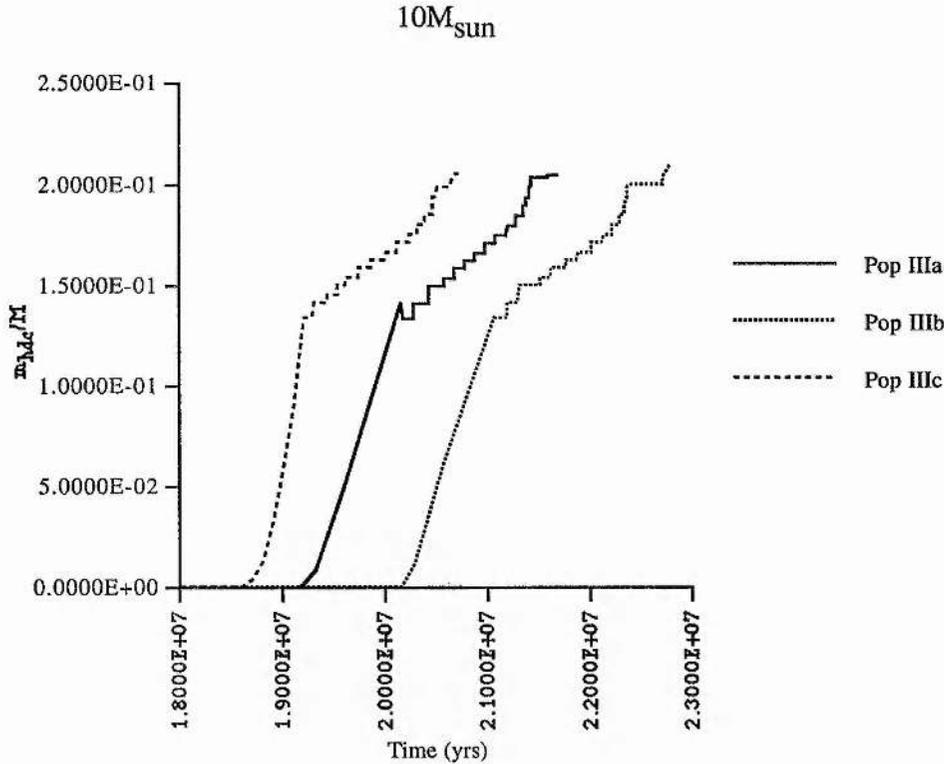
6.6.5 Hydrogen-Depleted Cores

Lastly, we can compare the rates at which the hydrogen depleted cores formed in models of different populations, and the times at which they begin to form. The following first set of diagrams show the core masses with time for models evolved using the initial code. The diagrams for the models evolved using the amended code may be found in appendix B.

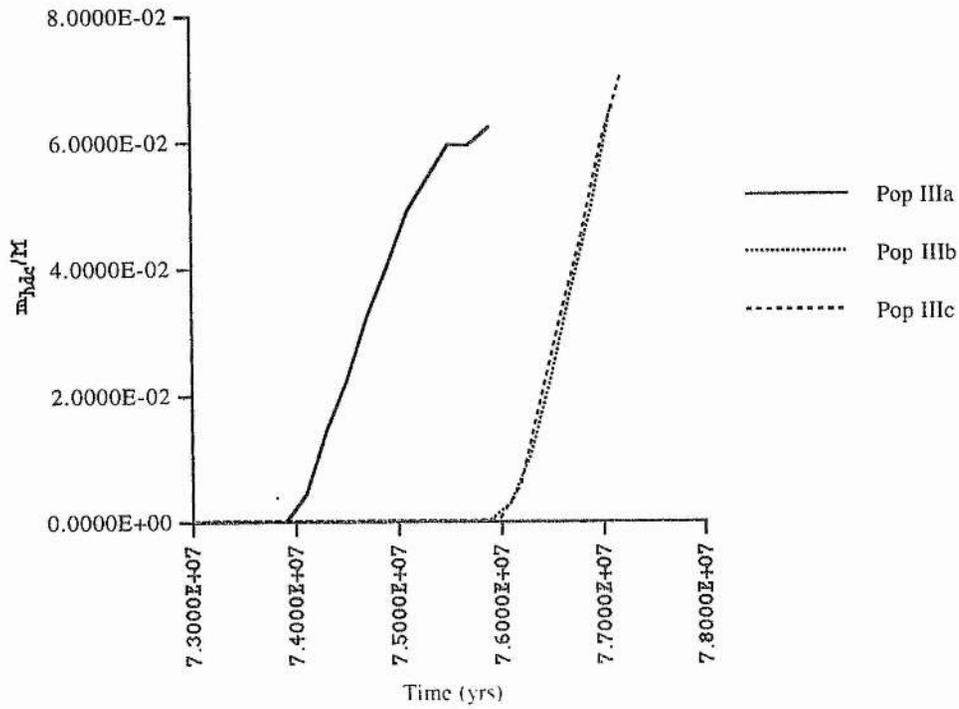


We can see from the diagram for the $15M_{\text{sun}}$ models that there are three distinct periods of core growth in all three models. The rates of growth are similar for the first and third period, and slower for the second period in between. The time at which the second period begins indicates that it begins when the convective core has reappeared. The second period

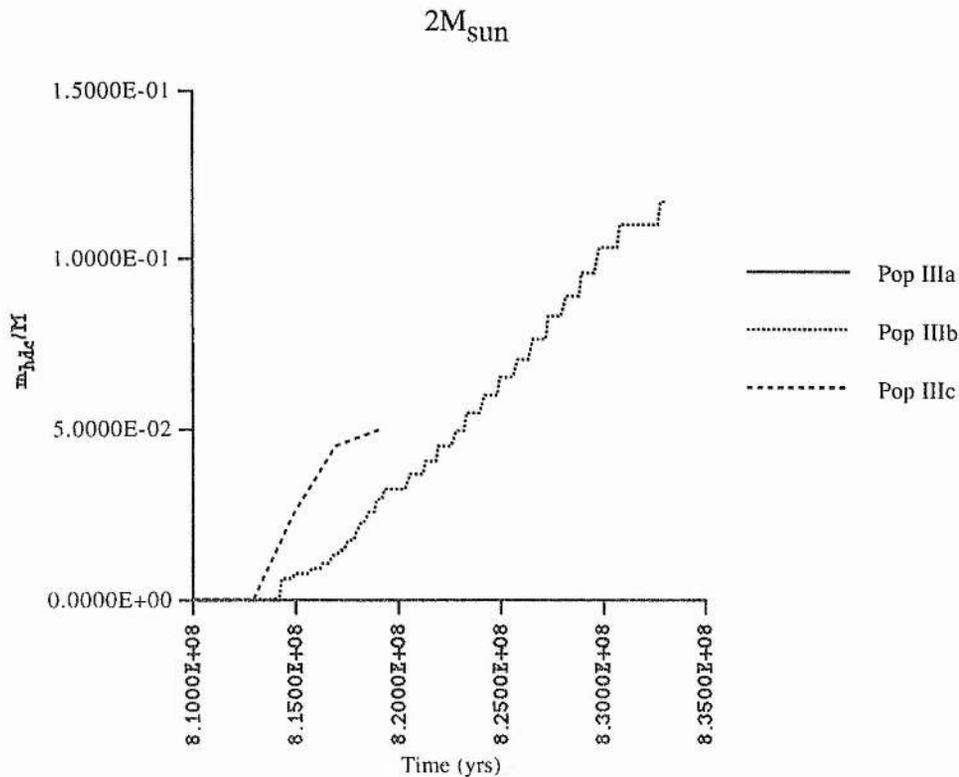
ends when the convective core disappears, as the central helium nears exhaustion. At the start of the second period, there is a brief and small decrease in the mass of the hydrogen-depleted core, as the convective core brings in hydrogen from the hydrogen-burning shell. The start of the second period of growth marks the high temperature point on the H-R diagram for the $15M_{\text{SUN}}$ models.



The diagram for the $10M_{\text{SUN}}$ models illustrates the same events as does the diagram for the $15M_{\text{SUN}}$ models. Again, we can see the three distinct periods of core growth, and the different growth rates. Again, the second period begins when the convective core in these models reappears and ends when the convective core disappears. We can also see, although less clearly than in the $15M_{\text{SUN}}$ models, the small and brief decrease in the mass of the hydrogen-depleted core as the convective core reappears. We can again note that the order of the population IIIc and population IIIa models is the reverse of that in the models at other masses.

$5M_{\text{sun}}$ 

In the diagram for the $5M_{\text{sun}}$ models, we can see far less difference between the population IIIb model and the population IIIc than was noticeable for the more massive models. There was no core convection while these hydrogen-depleted cores were forming.



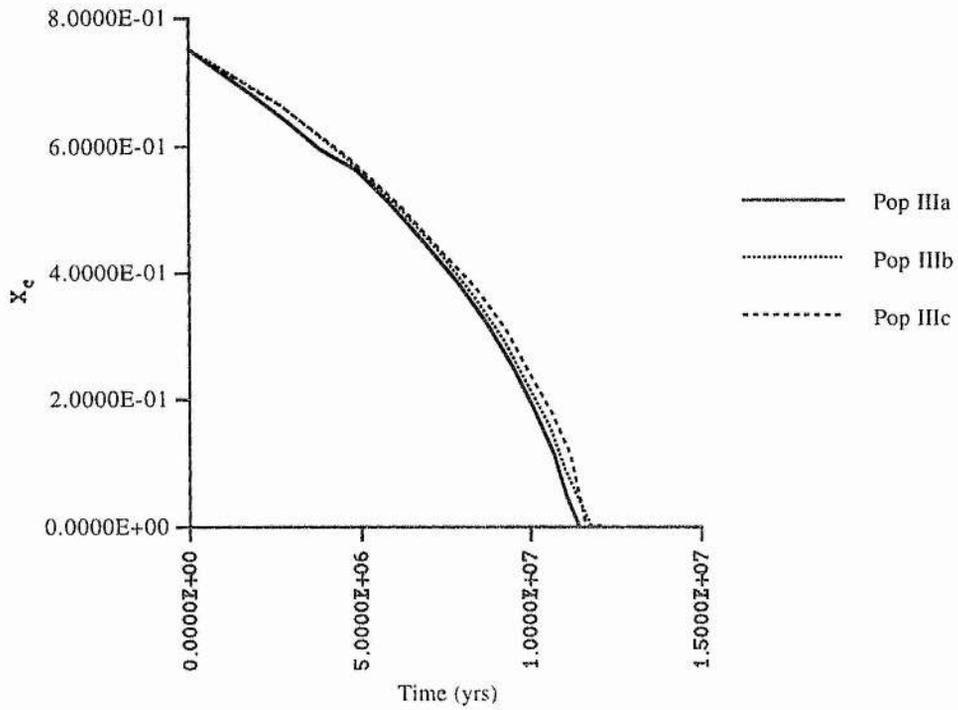
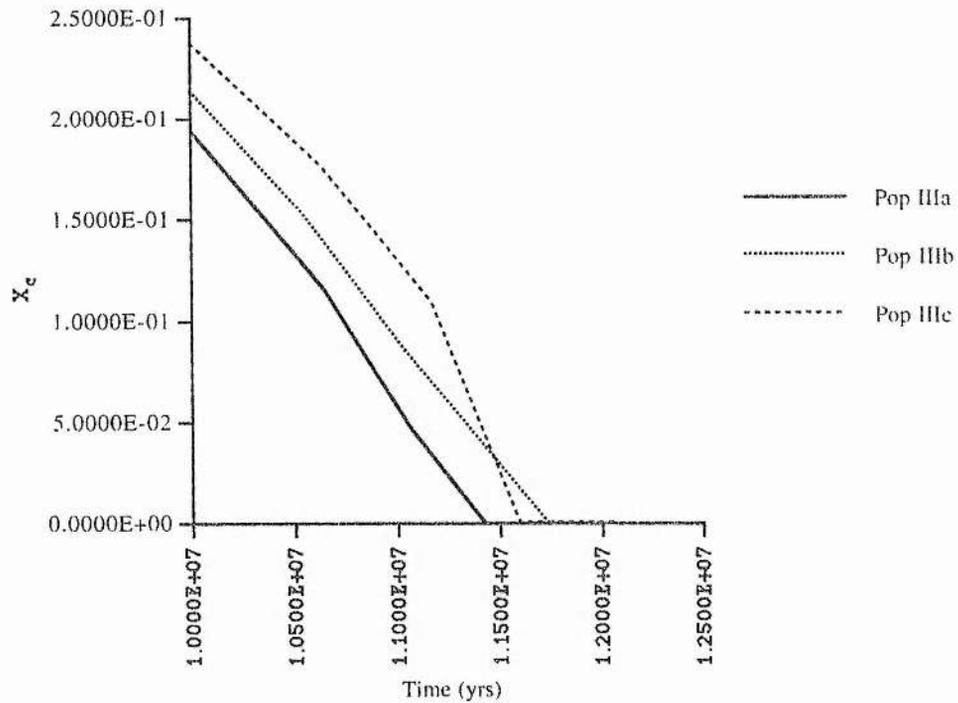
In this last diagram for the $2M_{\text{sun}}$ models, we can see that the population IIIa model did not evolve a hydrogen-depleted core before failing to converge.

6.6.6 Discussion of the Hydrogen-Depleted Core Diagrams

The diagrams of the evolution of the hydrogen depleted cores in the models presented in this work confirm what has already been stated. We can see that those models with population IIIa metallicities evolve hydrogen-depleted cores more quickly than the other models at the same mass, followed by the population IIIc models and then the population IIIb models. In the lower mass models, the difference between populations IIIb and IIIc becomes much less pronounced. However, we can still note that at these lower masses that the population IIIc models form their hydrogen-depleted cores prior to models of population IIIb of the same mass.

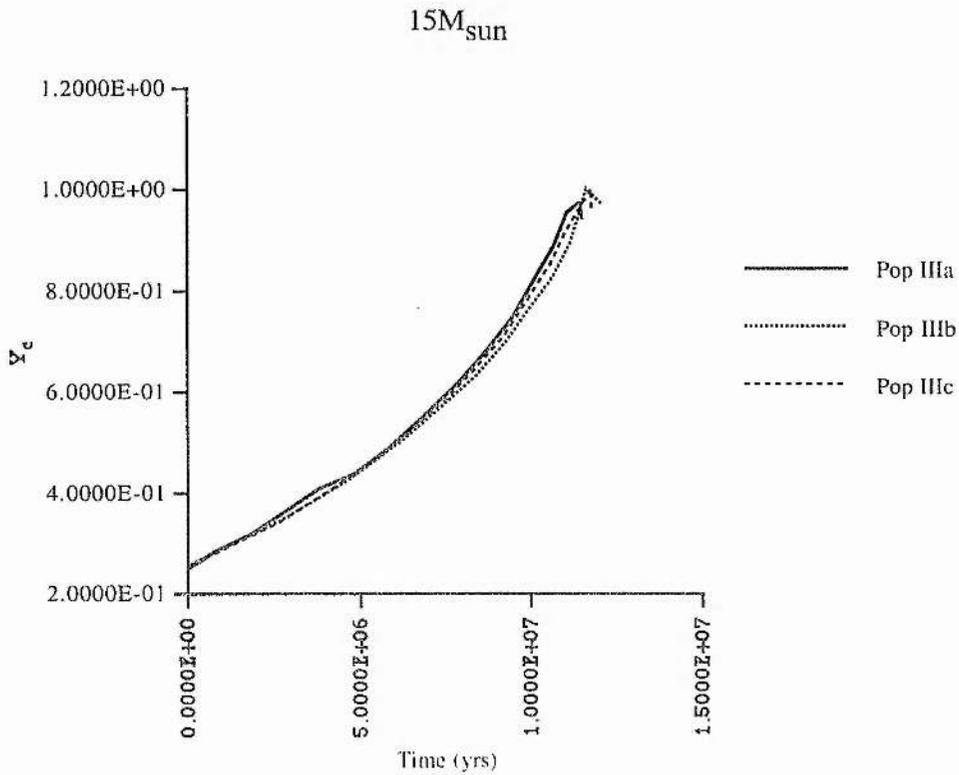
6.6.7 Central Chemical Evolution

In this section we present diagrams showing the chemical evolution at the centres of the models. In the same manner as the previous section, each diagram will give data for a given mass and the three different initial compositions. Firstly we have the diagrams for the $15M_{\text{sun}}$ models, the second diagram giving better resolution of the order in which the values of X_{C} reach zero.

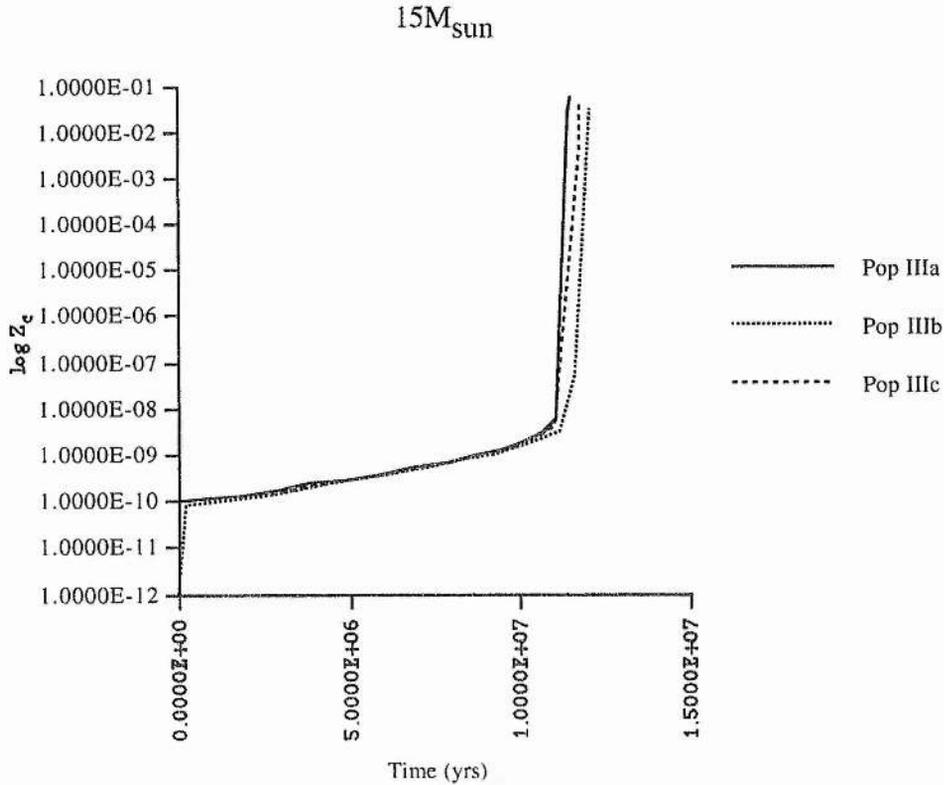
$15M_{\text{sun}}$  $15M_{\text{sun}}$ 

The $15M_{\text{sun}}$ diagrams for the mass fraction X_c shows that the decline of hydrogen at the centre of these models is occurring in much the

same fashion. It is interesting to note that the population IIIc model, which exhausted its core hydrogen prior to the population IIIb model, has a larger mass fraction X_C than the population IIIb model at any given time until just before the exhaustion of hydrogen, when it drops below the population IIIb value.

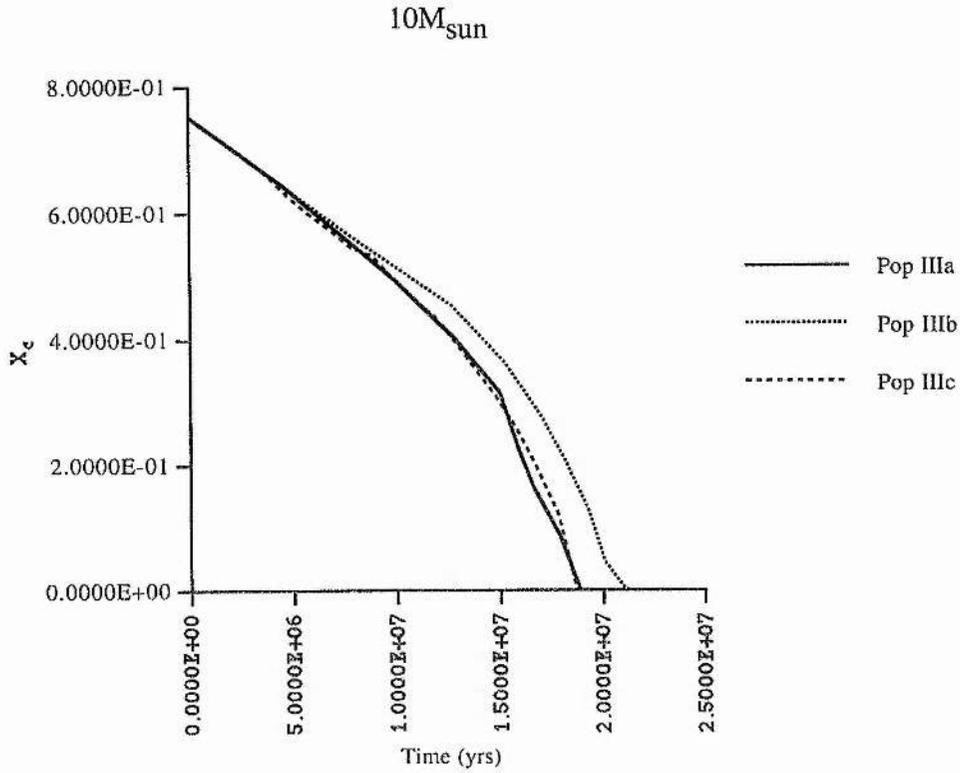


Again, from the graph for the core value of Y_C in the $15M_{\text{sun}}$ models, we can see that the changes in the mass fraction Y_C are broadly similar for the three models.

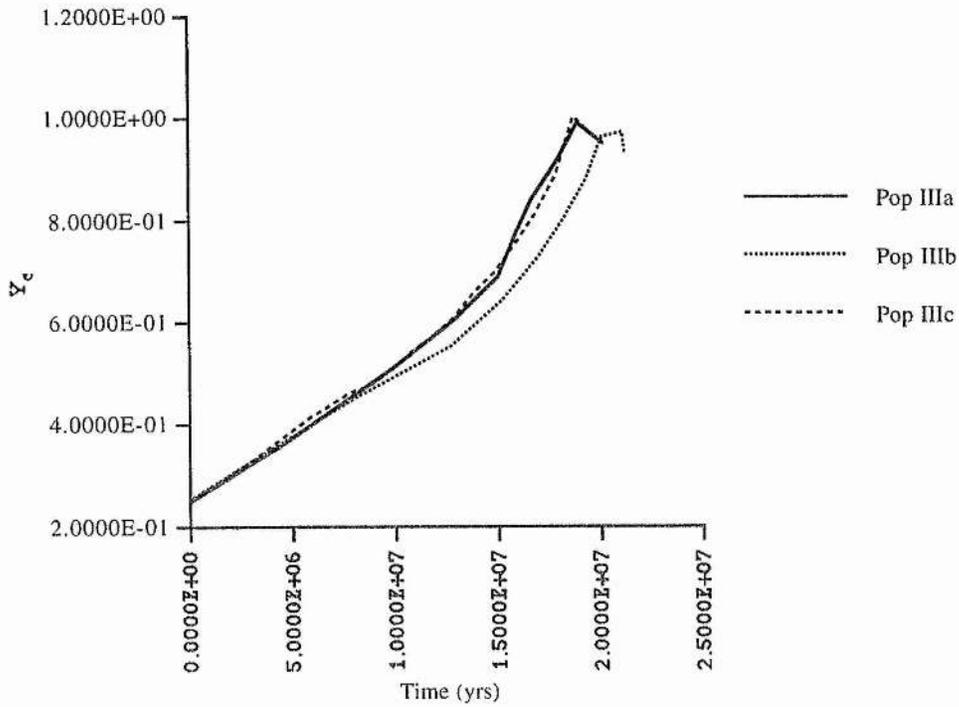


We can see from the diagram of Z_c for the $15M_{\text{sun}}$ models that once the models of population IIIb and IIIc have 'caught up' to the core metallicity of the population IIIa model, the evolution of Z_c is very similar up to the point at which helium burning becomes important. The sharp increase in the rate of increase of Z_c corresponds to point A in the H-R diagram for the $15M_{\text{sun}}$ models given in section 6.6.1. After this point, the rate of helium burning becomes great enough that small differences in core conditions are reflected noticeably in the value of Z_c with time.

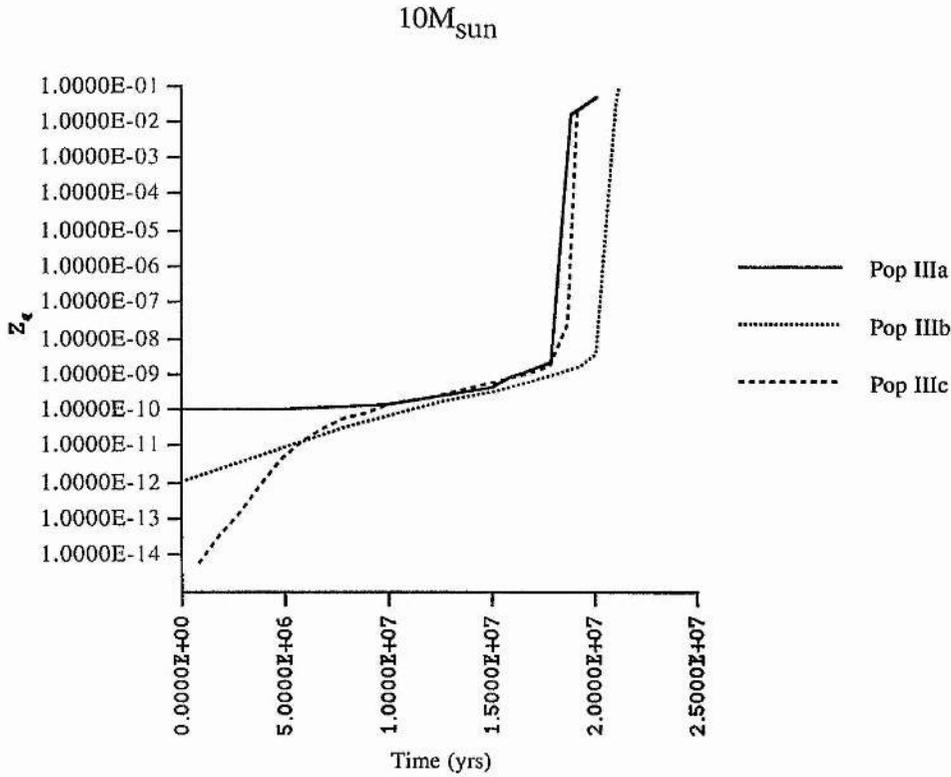
The next diagrams show the central chemical evolution diagrams for the $10M_{\text{sun}}$ models.



We can see from the $10M_{\text{sun}}$ diagram for the mass fraction X_C that the decrease in X_C for the population IIIb model is clearly less rapid from an early stage.

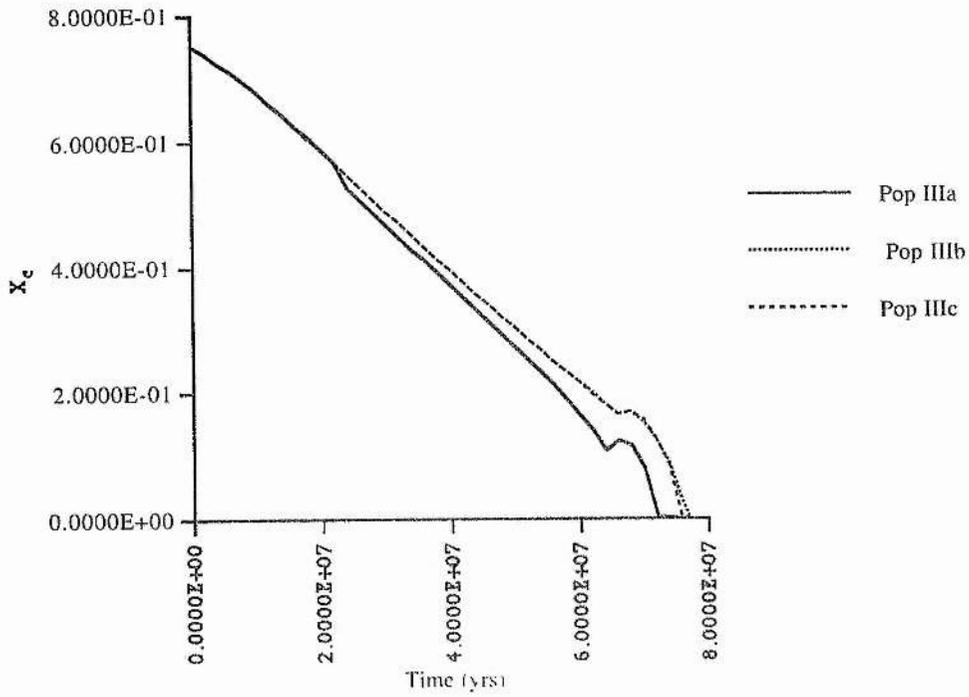
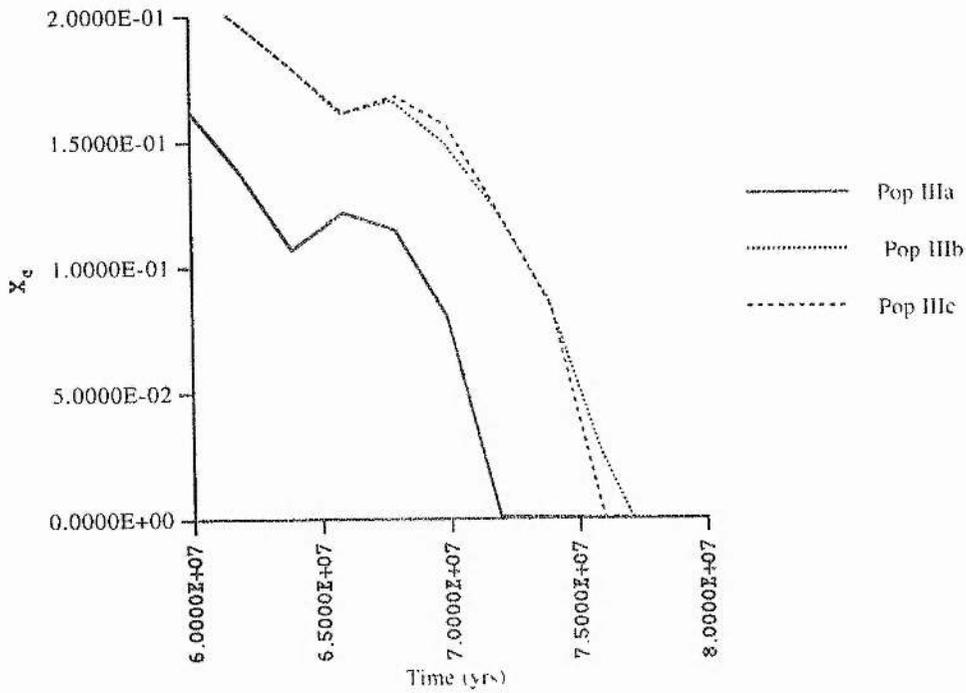
$10M_{\text{sun}}$ 

The early difference between the $10M_{\text{sun}}$ model of population IIIb and the other two models at this mass shows up clearly in the diagram for the mass fraction Y_C as well. The lower rate of decrease for X_C leads to a smaller rate of increase of Y_C .



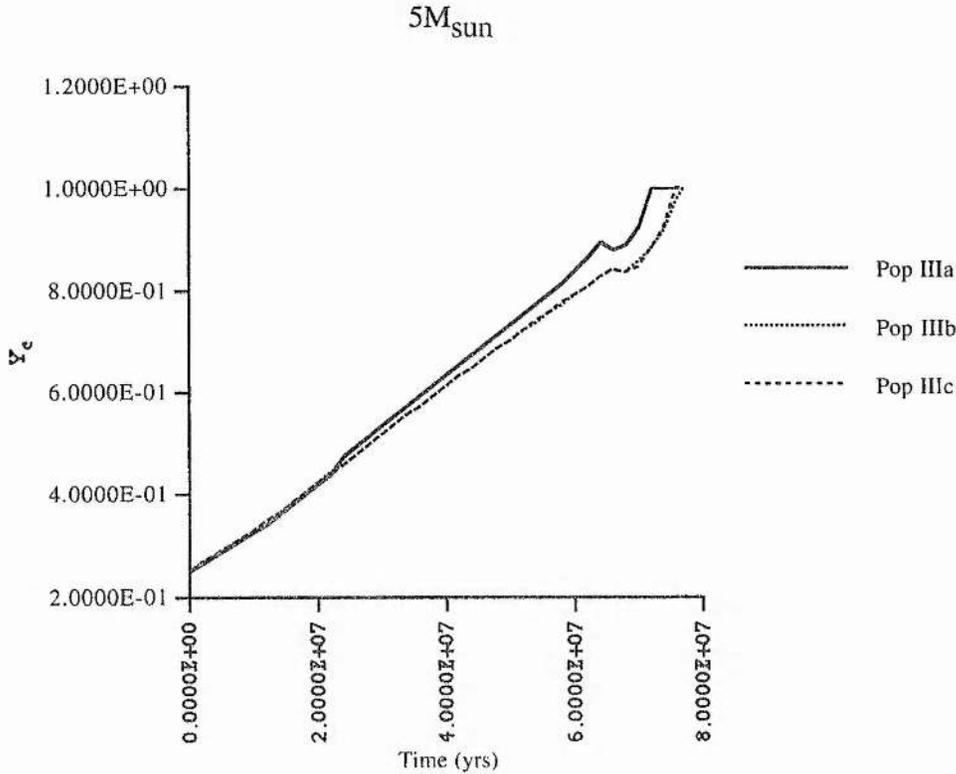
The diagrams for X_C and Y_C for the $10M_{\text{sun}}$ models implied that there is a lower nucleosynthesis rate in the core of the population IIIb model after an early age. The diagram giving the behaviour of the mass fraction Z_C at the same mass would seem to bear this out. The population IIIb model never 'catches up' to the metallicity of the population IIIa model as the population IIIb model of $15M_{\text{sun}}$ did. However, we can see the same general behaviour of Z_C as was illustrated by the $15M_{\text{sun}}$ models - an initial 'catching up' period, followed by a slow increase in Z_C and then a rapid rate of increase when helium-burning becomes an important energy source at the core. The point at which the rates change corresponds to point A in the H-R diagram for the $10M_{\text{sun}}$ models.

Next we have the diagrams for the core chemical evolution of the models of $5M_{\text{sun}}$. The second diagram concentrates on the latter chemical evolution of the core, as X_C approaches zero.

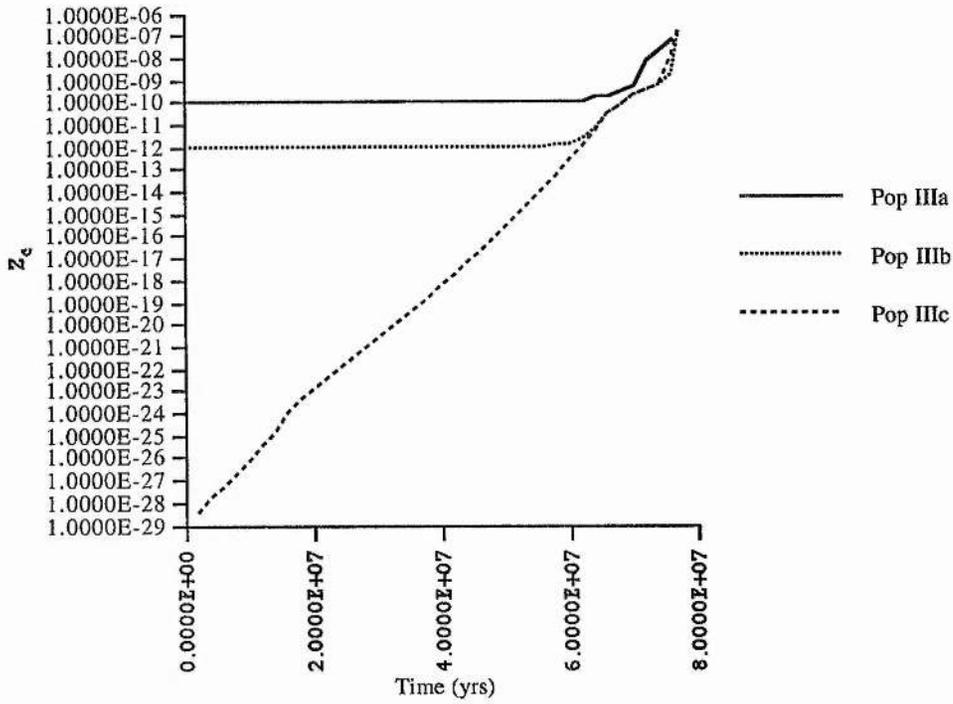
$5M_{\text{sun}}$  $5M_{\text{sun}}$ 

The diagrams for X_C in the $5M_{\text{sun}}$ models show an interesting feature - an increase in X_C in the models at a time of $6.7-7.0 \times 10^7$ years. This is due to

the behaviour of the convective core of these stars at this time, as it convects more hydrogen into the core from regions where the rate of burning has not been so high. We can see that the population IIIa model exhibits a faster decline in the value of X_C than the other two models, indicating a higher burning rate.

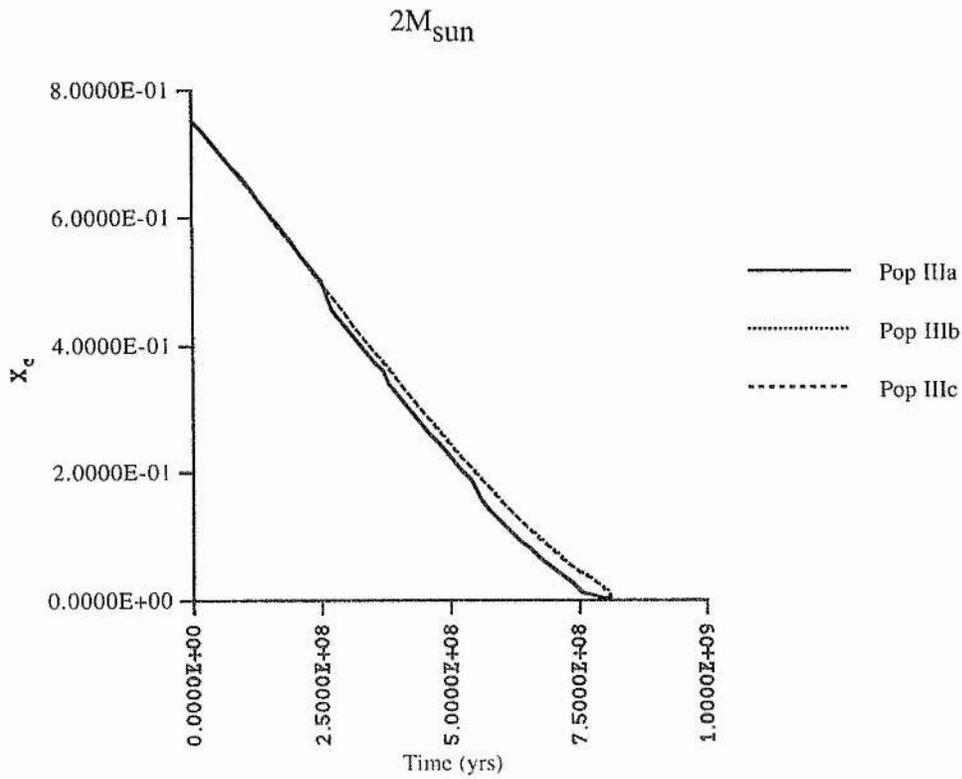


The diagram of Y_C in the $5M_{\text{sun}}$ models also indicates that the population IIIa model has a higher burning rate, as Y_C increases more rapidly in this model. The final constant value of Y_C for the population IIIa model occurs as the exhaustion of core hydrogen draws near, as the rates of hydrogen and helium burning balance each other. The brief decreases in Y_C in the models occur as the helium burning begins to become significant.

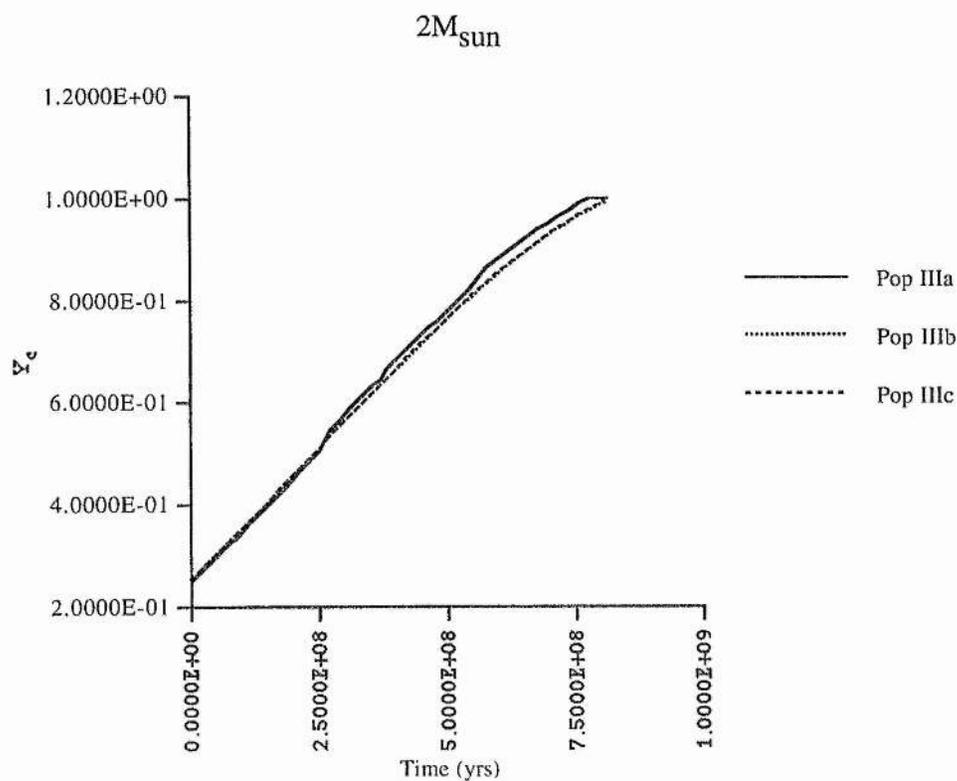
$5M_{\text{sun}}$ 

We can observe the same sort of behaviour of Z_c in the $5M_{\text{sun}}$ models as we could in the early evolution of the more massive models. The metallicities of the population IIIb and IIIc models 'catch up' with that of the population IIIa model before helium burning becomes significant enough to noticeably alter the value of Z_c in the population IIIa model.

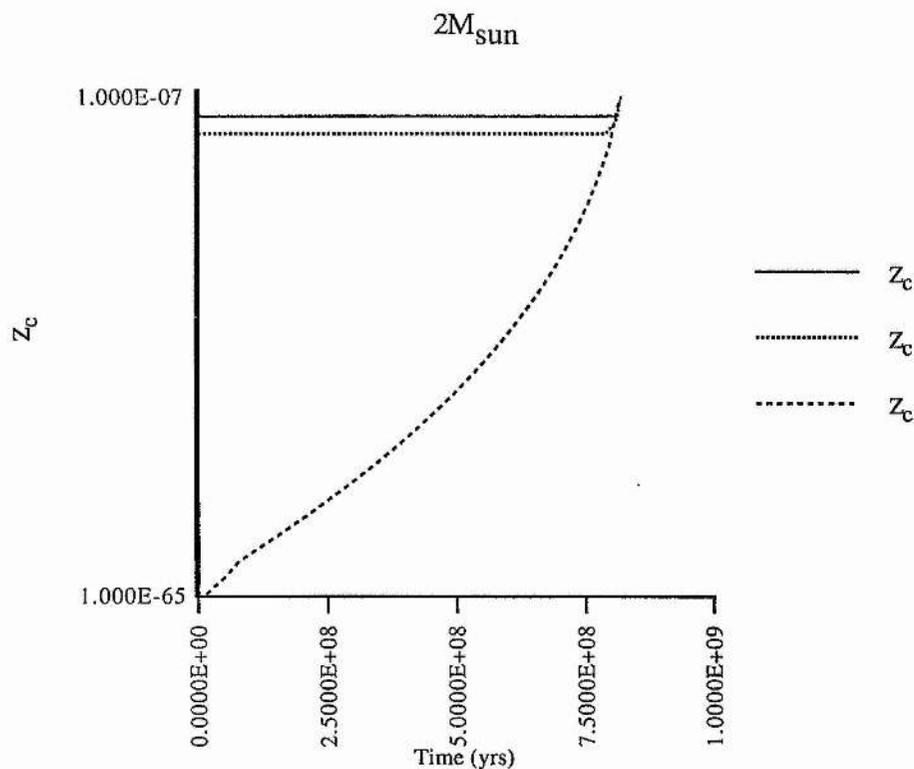
Lastly, we have the diagrams for the $2M_{\text{sun}}$ models.



The diagram showing the value of X_C for the $2M_{\text{sun}}$ models illustrates the same trend as for the $5M_{\text{sun}}$ models. The population IIIa model has a higher central rate of hydrogen burning that the other two models, as evidenced by the more rapid decrease in X_C .



That the population IIIa model of $2M_{\text{sun}}$ has a higher core rate of hydrogen burning is also illustrated by the values of Y_C for the $2M_{\text{sun}}$ models. We can note that unlike the $5M_{\text{sun}}$ models, there is no decrease in the values of Y_C at any point in the evolution of the $2M_{\text{sun}}$ models.



The absence of any decrease in Y_c in the $2M_{\text{sun}}$ models can be attributed to a far lower rate of helium burning. This is evidenced by the diagram of Z_c for the $2M_{\text{sun}}$ models. Unlike the diagram of Z_c for the $5M_{\text{sun}}$ models, there is no rapid increase in Z_c after the population IIIb and IIIc models have 'caught up' with the population IIIa model.

6.6.8 Discussion of the Central Chemical Evolution Diagrams

From the diagrams for the core chemical evolution, we can identify several of the important events in the lifetimes of the models we have studied. For example, core convection leading to a brief increase in X_c in the $5M_{\text{sun}}$ models, and the point at which helium burning becomes an important source of energy in the $15M_{\text{sun}}$ and $10M_{\text{sun}}$ models. We can see from the diagrams of X_c for the various masses that differences in the central mass fraction of hydrogen between the models of population IIIa, IIIb and IIIc only become very apparent in later evolution. These changes are largely due to the differences in Z_c and the effects of the CNO cycle, as discussed later in the chapter. We can also note that, at a given mass, the models of population IIIb and IIIc exhibit higher core rates of helium burning than the model of population IIIa, and thus a faster increase of Z_c , until they have reached the same metallicity as the population IIIa model. The reasons for this initial rapid 'catching up' in Z_c will also be discussed later in the chapter.

6.7 Other Results

We found that the use of the 'patch-on' acceleration term in the hydrostatic equation, described in section 4.7, made no appreciable difference to the convergence or lack of convergence of any of the models. This could either be due to the smallness of the acceleration term, or to the treatment of the problem, described in chapter three, being quasi-hydrostatic in nature. It is possible that a full consideration of an inertial term would include changing the method of solution at the most basic level.

None of the models evolved by either the initial or amended code experienced significant photospheric enrichment of either helium or metals. Neither the outer convection zones or the mechanism of mass loss reached deeply enough into the models to bring up enriched material to the upper layers. Due to the lack of enriched material at the photosphere, we could not consider there to be any significant enrichment of the interstellar medium due to the mass lost by these models as they evolved under the code.

6.8 Discussion and Conclusions

The minor differences between the zero-age models of population IIIa presented in this work and the same models presented by Cassisi and Castellani (1993), detailed in section 6.4, can be ascribed to the differences in the parameters and tables used by Cassisi and Castellani and the author of this work. The initial mass fraction of helium selected by Cassisi and Castellani is $Y=0.23$. While both works make use of the same reaction rates, Cassisi and Castellani use the LAOL and Cox-Tabor tables for the opacities. In addition, rather than calculating the thermodynamic properties, for $T > 10^6$ these authors use tables of values. Finally, the mixing length parameter in their work has been set to $\alpha=1.6$, and they did not consider any form of mass loss.

From the results presented in this chapter, we have seen that models with different initial metallicities evolve through the same events at different times. The timing of these events, the core exhaustion of hydrogen and the associated behaviour of the convective core, are dependent on the rate at which a model processes its nuclear fuels. Some core properties of the models at zero-age are given in the following table. Values are given in solar or c.g.s units as appropriate. ϵ_{ppc} , ϵ_{CNO} and $\epsilon_{3\alpha}$ are the energy generation rates at the centre of a model for the pp-chain, the CNO cycle and the triple- α cycle respectively.

	T_c (K)	ρ_c	ϵ_{ppc}	ϵ_{CNO}	$\epsilon_{3\alpha}$
Pop IIIa					
15M _{sun}	9.783×10^7	1.578×10^2	9.428×10^3	1.808×10^4	6.038×10^{-6}
10M _{sun}	8.020×10^7	1.684×10^2	6.706×10^3	2.373×10^3	6.328×10^{-10}
5M _{sun}	4.978×10^7	1.444×10^2	1.915×10^3	6.651×10^0	5.802×10^{-24}
2M _{sun}	2.499×10^7	1.180×10^2	2.139×10^2	2.010×10^{-4}	2.236×10^{-61}
Pop IIIb					
15M _{sun}	1.064×10^8	2.057×10^2	1.446×10^4	5.558×10^2	2.991×10^{-4}
10M _{sun}	8.152×10^7	1.783×10^2	7.344×10^3	2.955×10^1	1.634×10^{-9}
5M _{sun}	4.984×10^7	1.447×10^2	1.917×10^3	6.665×10^{-2}	5.861×10^{-24}
2M _{sun}	2.499×10^7	1.180×10^2	2.139×10^2	2.010×10^{-6}	2.236×10^{-61}
Pop IIIc					
15M _{sun}	1.066×10^8	2.072×10^2	1.463×10^4	0.0	3.302×10^{-4}
10M _{sun}	8.154×10^7	1.784×10^2	7.352×10^3	0.0	1.654×10^{-9}
5M _{sun}	4.978×10^7	1.445×10^2	1.917×10^3	0.0	5.869×10^{-24}
2M _{sun}	2.500×10^7	1.180×10^2	2.141×10^2	0.0	2.330×10^{-61}

We have noticed that the models with lower initial metallicities have higher initial rates of nucleosynthesis of metals, which can be explained from the rates given in the table above. At zero-age, a population IIIc model with $Z=0$ cannot generate energy using the CNO cycle. This leads to a lower radiation pressure to support the model against its collapse, and thus a core that is more dense. We can see that the population IIIc models initially have a higher rate of energy generation for the triple- α process, due to the increased density and temperature at the core, and hence the nucleosynthesis of metals in these models will initially occur at a higher rate. This higher rate should continue until the model has formed enough metals for the CNO cycle to become active. As all models of population III burn a small amount of helium in the triple- α process from their zero-age onwards, the nucleosynthesis of the required mass fraction of metals does not take very long. Once the models of population IIIb and IIIc have 'caught up' to the metallicity of the population IIIa model, at a given mass, their core energy generation rates, and thus nucleosynthesis rates, after this point will be similar. This will last until helium burning becomes an appreciable energy source, and small differences in the parameters at the core lead to differing rates in different models.

The rates at which the models of population IIIa, IIIb and IIIc at a given mass burn hydrogen differ largely due to the effects of the CNO cycle. The burning rate associated with the CNO cycle has a higher temperature dependence than that associated with the pp-chain, and provides a greater contribution to the total burning rate for $T > 1.5 \times 10^7$. All of the models evolved in this work initially have core temperatures $T_c > 1.5 \times 10^7$. The efficiency of the CNO cycle at a given temperature is dependant on the level of metallicity in the model, so models with a higher initial metallicity will initially burn hydrogen at a faster rate. By this consideration, at a given mass the population IIIa model should

exhaust their central hydrogen first, followed by the population IIIb model and then the population IIIc model. However, we have found that at a given mass the models of population IIIb formed their hydrogen-depleted cores later than those of population IIIc, and for the $10M_{\text{SUN}}$ models the population IIIc model exhausted its central hydrogen before either of the other two models. We can see from the diagrams of the evolution of the core chemical composition presented in section 6.6.7 that in most cases the order in which the models of a given mass will deplete their core hydrogen is not apparent from an examination of the early parts of the diagrams. The rate at which hydrogen is burned is similar between models early in their lifetimes, and the rate in different models only diverges later. This early similarity in rates is due to the rapid increase of the metallicity in the models of population IIIb and IIIc, which rises high enough to make the rates for the CNO cycle comparable at an early stage. In those models that do not nucleosynthesize metals to the same level as other models at the same mass, the CNO rate is lower, and hence the conversion of hydrogen to helium occurs at a slower rate. For example, the $10M_{\text{SUN}}$ model of population IIIb does not produce metals at the core as rapidly as the other models at the same mass. The difference in the evolution of the mass fraction of hydrogen at the core between this and the other two models of the same mass is very apparent. We can extend this observation to argue that all the differences in these models, at least up until the exhaustion of the core hydrogen, can be ascribed as having the varying rates of the CNO cycle as their root cause.

We can explain the initially large convective core in the $10M_{\text{SUN}}$ model of population IIIa from the table of energy generation rates at zero-age given earlier in this section. The sum of the core rates ϵ_{ppc} and ϵ_{CNO} for this model is nearly double that of these rates summed for the population IIIb and IIIc models at this mass. This increased energy production will lead to a steeper temperature gradient in the core of the model, and hence a larger convective core. We can note that this scale of difference between the total core rates of the models of population IIIa, IIIb and IIIc only occurs for the $10M_{\text{SUN}}$ models, explaining why we only see this large initial difference in convective core size in the models of this mass.

We can note that our $5M_{\text{SUN}}$ models failed to evolve to a stage of core helium-burning, a possible indication that they would have undergone a helium flash event. However, helium burning set in more gradually in the more massive models. This could place the upper mass limit for a helium flash to occur in population III somewhere between $5M_{\text{SUN}}$ and $10M_{\text{SUN}}$, most likely in the lower end of this range. This can be compared with the upper limit of $4M_{\text{SUN}}$ for the helium flash to occur in zero-metal stars, as stated by Fujimoto, Iben, Chieffi and Tornambé (1984). Their paper was a much more directed study of the phenomenon of helium shell flashes in zero-metal stars. These authors approached the modelling of these stars in a different manner to that adopted in this work, using a semi-analytical consideration of the shell burning regions.

The lack of enrichment of either helium or metals at the photosphere of the models presented in this work would be expected for the $5M_{\text{sun}}$ and $2M_{\text{sun}}$ models. These lower mass models have not undergone helium ignition at the core. However, the lack of a 'dredge-up' of enriched material in the more massive models was surprising. Chieffi and Tornambé (1984) found that the first dredge-up of enriched material, which would be expected during core helium-burning in stars of population I or II, did not occur in stars of population III. However, they found that the second dredge-up, after the formation of a helium depleted core, did occur. This is not the case in any of the $15M_{\text{sun}}$ or $10M_{\text{sun}}$ models evolved in this work. This second dredge-up would have to be connected with a convective episode during shell helium burning, but none of the $15M_{\text{sun}}$ or $10M_{\text{sun}}$ models generated by the code exhibited this. It is possible that the onset of a convection zone or zones extensive enough to cause this dredge-up would also cause a model to depart from hydrostatic equilibrium, and thus fail to converge under the code. It could be this failure of convergence prior to the second dredge-up which prevents us from observing it in our evolved models. The lack of enriched material at the photosphere means that no enriched material was returned to the interstellar medium during the evolution of these models. Thus we must discard our original assumption that the stellar wind of these models would have returned enriched material to the interstellar medium. The implication of this is that any return of enriched material would have to occur during a supernova event at the end of the life of a star. The work of Jura (1986) suggests that even intermediate mass ($M \geq 1.5M_{\text{sun}}$) stars of low metallicity may end their lives as supernovae, even if not in interacting binary systems.

It is interesting to note, in the light of one of our $2M_{\text{sun}}$ models failing to exhaust its central hydrogen, that the same behaviour was observed in a $0.9M_{\text{sun}}$ zero-metal model by Guenther and Demarque (1983). If, however, we compare the evolved time of this $2M_{\text{sun}}$ model with the other models of the same mass, we can see that it is the youngest of them all, at 8.057×10^8 years. From the diagrams for the other $2M_{\text{sun}}$ stars, we can see that the hydrogen-depleted and convective cores did not start forming until after 8.057×10^8 years. So we should state that this star is unusual for an early failure to converge, before the formation of this core, rather than for the lack of it. This failure to converge could be attributed to a departure from hydrostatic equilibrium due to the onset of the second convection episode. As we can see from the diagrams for the other $2M_{\text{sun}}$ models, the central convection zone grows very quickly, which could account for a sufficient departure from hydrostatic equilibrium to prevent the code from converging a model.

It has been noted by several authors that a lower limit to the effects of changing metallicity upon stellar parameters appears to exist. For $Z < Z_{\text{u}}$, where Z_{u} is some limiting value of Z , there are no apparent differences

between stars of the same mass but with different values of Z . Wagner (1974) set this limit $Z_{\text{U}} \sim 10^{-6}$, but Chieffi and Tornambé (1986) found the dependence of selected parameters upon Z to continue as low as $Z=10^{-8}$. Finally, Cassisi and Castellani (1993) found that differences in the structure of stars due to differing Z exist for values of Z as low as $Z=10^{-10}$. In this work we have found that differences in the evolutionary behaviour of our models exist between the initial metallicities of $Z=10^{-10}$, 10^{-12} and zero, indicating that the sensitivity of many stellar parameters to the metal content of the star extends all the way down to $Z=0$. However, we can note that there are no noticeable differences in the basic stellar parameters R , T and L between the lower mass zero age models of population IIIa, IIIb and IIIc. As the zero-age models at higher masses do exhibit differences due to different initial metallicities, we can say that differences due to different initial metallicities appear to decrease with mass. We can see from the H-R diagrams presented in section 6.6.1 that the lower mass models exhibit the same evolutionary tracks for models of population IIIb and IIIc. This is not the case for the more massive models, indicating again that the sensitivity to the initial value of Z appears to decrease with mass. We agree with Cassisi and Castellani in attributing these differences at zero age to the energy produced by the CNO cycle in these models. In the lower mass models, the interior temperatures are correspondingly lower, and thus the CNO cycle produces less energy. Any differences in this lower rate of energy generation due to differing values of Z will thus be less noticeable or entirely negligible.

Chapter Seven

Chemical Evolution by a Population III

7.1 The Processing of Material by a Population III

The 'bulk yield' of a star refers to the amount by which it changes the composition of the material of the interstellar medium over its lifetime. In general, for stars not in binary or multiple systems, the largest loss of material occurs towards the end of their life. This can either be in the form of a large outflow of mass due to a stellar wind, or be due to a supernova event. Some of this material will have been processed in the stellar interior, and will have a significantly different composition to the interstellar medium. In order to calculate or estimate the bulk yields of the models presented in this work, several problems must be overcome. Firstly, none of the models presented in this work are fully evolved, and so their current distribution of elements will almost certainly not be the same as their final distribution of elements. In addition, there is the problem of stellar remnants. Not all of the mass of a star will be returned to the interstellar medium by the mechanisms that we are considering. Some remnant of the core will remain as a class of white dwarf, or possibly a neutron star or even a black hole. Lastly, there is the question of which models will undergo supernova events, and what effect this will have on the composition of the material ejected by such an event.

7.2 The Role of Helium Star Studies

In order to gain information on the evolution of the core of a star past the point at which a quasi-hydrostatic treatment of the problem, such as is described in this work, becomes difficult or impossible to use, we can turn to the studies of helium stars. We can identify the hydrogen-depleted cores of the models presented in this work with helium stars of the same mass. These helium stars would then have been evolved to the point of their presupernova collapse, with particular attention to the nucleosynthesis of metals up to this point. The study of helium stars considered in this work was carried out by Arnett (1978). We can make this correspondence of helium star with hydrogen-depleted core directly for the models of $15M_{\text{sun}}$ and $10M_{\text{sun}}$, taking the mass of the core at its final and largest value. However, the $5M_{\text{sun}}$ and $2M_{\text{sun}}$ models present a problem. These models have not evolved to the same point in their lifetimes as the more massive models have, to the point of the formation of a hydrogen-depleted core. We therefore need to find a way of predicting the size of the hydrogen-depleted cores which these less massive models would have evolved by this later point in their lifetimes.

7.3 Obtaining the 'Final' Core Masses for the Lower Mass Models

In order to predict the mass of the hydrogen-depleted cores eventually evolved by the less massive models, the author referred to data from the work by Limongi and Tornambé (1991). These authors presented evolutionary data, including lifetimes, for helium star models in the mass range $0.4M_{\text{sun}}$ to $1.0M_{\text{sun}}$. The models were evolved from their zero age

up to their departure from quasi-hydrostatic equilibrium after forming helium-depleted carbon-oxygen cores. This stage in their lifetime corresponds to the final point reached by the cores of the $15M_{\text{SUN}}$ and $10M_{\text{SUN}}$ population III models presented in this work. Similar data to that from Limongi and Tornambé (1991) is presented by Paczynski (1971), but the author chose to use the later work. The reader may like to refer to the work by Paczynski by way of comparison with the later results of Limongi and Tornambé.

From the evolutionary data collected for the $5M_{\text{SUN}}$ and $2M_{\text{SUN}}$ models evolved by the code in this work, we can establish an average 'rate of growth' for the hydrogen-depleted cores. We may then write the mass of the hydrogen-depleted core, m_{hdc} in solar mass units, in terms of this rate of growth and time, t in years. This becomes

$$m_{\text{hdc}} = \frac{dm_{\text{hdc}}}{dt} (t - t_{\text{hdc}}) , \quad (7.3-1)$$

where t_{hdc} is the time in years at which the hydrogen-depleted core began to form. From the work by Limongi and Tornambé, we can form a function relating the final age t of their helium star model to its mass m . For models in the mass range $0.8M_{\text{SUN}}$ to $0.9M_{\text{SUN}}$ this function is linear, and takes the form

$$m = 0.9 - 4.032 \times 10^{-11} (t - 2.585 \times 10^7) . \quad (7.3-2)$$

If we now identify m and t from equation (7.3-2) with m_{hdc} and t from equation (7.3-1), we have a pair of simultaneous equations which can be solved to find m_{hdc} for any of the low mass models in this work. In addition to the mass of the core, Limongi and Tornambé give the mass fraction of metals in the core at the final point of evolution. Once we know the masses of our cores, we can interpolate to find mean values for the metallicity in these cores, Z_{hdc} . These values will of course be for the stage of evolution comparable to that reached by the high mass models. Needless to say, this is not a very safe procedure. We are identifying a core of increasing mass with helium stars of static mass, and any answer we achieve should be used with circumspection.

7.4 Core Data for the Lower Mass Models

Here, the author presents the results of the calculations resulting from the discussion in the previous section.

	t_{hdc} (yrs)	dm_{hdc}/dt ($M_{\text{sun}} \text{ yr}^{-1}$)	m_{hdc} (M_{sun})	Z_{hdc}
Population IIIa				
$5M_{\text{sun}}$	7.390×10^7	3.105×10^{-8}	0.8978	0.7118
$2M_{\text{sun}}$	8.150×10^8	7.064×10^{-9}	0.8657	0.7374
Population IIIb				
$5M_{\text{sun}}$	7.590×10^7	5.276×10^{-8}	0.8978	0.7118
$2M_{\text{sun}}$	8.130×10^8	8.066×10^{-9}	0.8661	0.7371
Population IIIc				
$5M_{\text{sun}}$	7.580×10^7	5.029×10^{-8}	0.8790	0.7268
$2M_{\text{sun}}$	8.130×10^8	8.066×10^{-9}	0.8662	0.7370

These figures for the hydrogen-depleted cores of these models can now be identified with helium stars as we could with the cores of the more massive models. Fortunately, all of these core masses fall within the range for equation (7.3-2). Further equations did not have to be formed.

7.5 Stellar Remnants

It is important for us to know how much of the stellar core will be left as a remnant if we are to calculate the degree to which the models presented in this work would augment the interstellar medium. Unfortunately the fraction of the core left as a remnant can only be estimated. Various authors have suggested using the Chandrasekhar mass of $\sim 1.4M_{\text{sun}}$ for the remnant mass, but this would be far too large for the purposes of this work. It is in fact larger than the core masses given in the previous section. This figure is more useful when more massive stars are being considered. Arnett (1978) suggests that a good approximation is to assume that the iron core is left as a remnant after a supernova event. In his paper, he gives a method of calculating this remnant from abundances in his 'pre-supernova' models of helium stars. In essence, the mass of the remnant, m_{rem} is given by

$$m_{\text{rem}} = m_{\text{hdc}} \langle X_{\text{Ni}} \rangle \quad (6.5-1)$$

where m_{hdc} is the mass of the hydrogen-depleted core, or helium star, and X_{Ni} is the mass fraction of all metals heavier than silicon. It is this second definition that will be used here to establish the mass of remnants left after the supposed 'death' of the models presented. In this case, the remnant can be considered to consist entirely of metals - any and all helium left in the core will be ejected at the 'death' of a model. If small enough, this remnant will then be some form of white dwarf.

Arnett identifies helium stars with the cores of hydrogen stars less massive than models presented in this work with equivalent hydrogen-depleted cores. His work is primarily concerned, however, with identifying core masses for stars of population I. For example, he suggests that we identify a helium core of $3M_{\text{sun}}$ as having formed in a star

initially of $10M_{\text{SUN}}$. From the results of the previous chapter, we can see that the hydrogen-depleted cores formed by the $15M_{\text{SUN}}$ population III models are only slightly more massive than $3M_{\text{SUN}}$. This tendency for models of population III composition to form less massive hydrogen-depleted cores has been remarked upon by Cassisi and Castellani (1993), amongst other authors. Values for X_{Ni} were determined for the models presented in this work by interpolating between values given by Arnett at various masses and hence metallicities. The author has chosen to identify which models to interpolate between by appropriate metallicity rather than mass. Those evolved helium stars presented by Arnett which were of similar mass to the hydrogen-depleted cores evolved by the $15M_{\text{SUN}}$ and $10M_{\text{SUN}}$ models in this work had lower metallicities than these cores. This result was obviously not desirable. Hence the author selected helium cores of a greater mass, but which had the same mean metallicities as the hydrogen-depleted cores shown in this work. The cores formed by the models of mass $5M_{\text{SUN}}$ and $2M_{\text{SUN}}$ were too small for Arnett's work to be used to determine a remnant mass from.

7.6 Supernova Events and Endstates

There is a vast amount of literature covering all aspects of supernovae, very little of which, unfortunately, is applicable in respect of this work. The interested reader might like to refer to the review by Kirshner (1981), or similar more recent articles. What we wish to know is whether or not the models presented in this work will eventually become supernovae, and what effect this will have on the composition of the material returned to the interstellar medium by this mechanism. Jura (1986) demonstrates that even intermediate mass ($>1.5M_{\text{SUN}}$) stars of population II composition may become supernovae. It is plausible to extend this trend of supernovae at lower masses to models of population III composition. So we can consider all the models presented in this work to end their lives as supernovae, returning most of their masses to the interstellar medium, except for that remaining in a remnant. However, from work by Woosley and Weaver (1982) we can see that stars of $<10M_{\text{SUN}}$ contribute no significant enrichment to the interstellar medium over their lifetimes. This conclusion is also implied in the results given by Arnett (1978), in that the minimum mass of the main sequence population I model that his helium stars, or cores, can be properly identified with is $10M_{\text{SUN}}$. For the purposes of this work, we could assume that those models of $5M_{\text{SUN}}$ and $2M_{\text{SUN}}$ produce no significant changes in metallicity in the material that they return to the interstellar medium over their lifetimes. However it would seem a safer procedure to extrapolate to find the metallicity of the ejected material at lower masses from a function fitted to the results at higher masses.

During a supernova event, a significant amount of nucleosynthesis may occur during the explosion itself. A treatment of this explosive nucleosynthesis is unfortunately beyond the scope of this work. However, we can use the assumption that such explosive nucleosynthesis only

changes the relative proportions of heavy elements within Z , and does not increase the overall value of Z . The reader may like to refer to comments by Arnett (1977) on this topic.

7.7 The Composition of the Ejecta by Mass and Initial Metallicity

From the above discussion, we can take the average metallicity of the ejected material from a model initially of mass M to be given by

$$Z_{ej} = \frac{M_{hdc}(M) \times Z_{hdc}(M) - M_{rem}(M)}{M - M_{rem}(M)}, \quad (7.7-1)$$

where $M_{hdc}(M)$ and $M_{rem}(M)$ are the masses of the hydrogen-depleted core and the remnant at initial mass M , all in solar mass units. $Z_{hdc}(M)$ is the average metallicity of the core of the model. Equation (7.7-1) implies that, as noted previously, the remnant will consist entirely of metals. The equation also assumes that the mass of metals outside the core is small enough in comparison to be neglected. In addition, we should note that the small amount of mass lost during the evolutionary calculations is included in equation (7.7-1) as a result of taking the initial mass rather than the final mass of the evolved models.

	$M_{hdc}(M_{sun})$	$M_{rem}(M_{sun})$	Z_{hdc}	Z_{ej}
Population IIIa				
15 M_{sun}	3.385	7.383×10^{-1}	5.801×10^{-1}	8.592×10^{-2}
10 M_{sun}	2.014	4.429×10^{-1}	5.790×10^{-1}	7.567×10^{-2}
Population IIIb				
15 M_{sun}	3.452	6.328×10^{-1}	6.009×10^{-1}	1.003×10^{-1}
10 M_{sun}	2.047	4.898×10^{-1}	5.668×10^{-1}	7.049×10^{-2}
Population IIIc				
15 M_{sun}	3.314	6.594×10^{-1}	5.917×10^{-1}	9.075×10^{-2}
10 M_{sun}	2.004	6.182×10^{-1}	5.159×10^{-1}	4.430×10^{-2}

In the above table, M_{hdc} is the mass of the hydrogen-depleted core, M_{rem} the mass of the remnant, Z_{hdc} is the mean metallicity of the hydrogen-depleted core as a whole, and Z_{ej} is the mean metallicity of the material returned to the interstellar medium by a supernova event and the prior mechanism of a stellar wind. This material of course includes the unprocessed mass lying outside the hydrogen-depleted core. The masses of these remnants indicates that they would be some form of white dwarf.

We can consider the amount of helium ejected by the same processes in much the same way. The mass fraction of helium in the material returned to the interstellar medium by a model can be given by

$$Y_{ej} = \frac{M_{hdc}(M) \times (1 - Z_{hdc}(M)) + 0.25 \times (M - M_{hdc})}{M - M_{rem}}. \quad (7.7-2)$$

Here, we are including the helium exterior to the hydrogen-depleted core in our calculation of Y_{ej} , taking the mass fraction exterior to the core to be at its initial value of $Y=0.25$. We are assuming that any hydrogen-burning shell is thin enough that any additional helium over that given by equation (7.7-2) contained within it can be neglected. The mass fractions of helium for the more massive stars are given in the table below;

	$1 - Z_{hdc}$	Y_{ej}
Population IIIa		
$15M_{sun}$	4.199×10^{-1}	3.033×10^{-1}
$10M_{sun}$	4.210×10^{-1}	2.976×10^{-1}
Population IIIb		
$15M_{sun}$	3.991×10^{-1}	2.968×10^{-1}
$10M_{sun}$	4.332×10^{-1}	3.023×10^{-1}
Population IIIc		
$15M_{sun}$	4.083×10^{-1}	2.981×10^{-1}
$10M_{sun}$	4.841×10^{-1}	3.165×10^{-1}

7.8 Functions for Z_{ej} and Y_{ej} With Initial Mass

From the values tabled in the previous section, we can determine a function for the value of the mass fraction Z_{ej} at some initial mass. If, for example, we were to use the assumption that the metallicity is negligible in the ejected matter for masses $M \leq 5M_{sun}$, then we can find a good fit for the metallicity data from the function

$$Z_{ej}(M) = \begin{cases} 0 & M < 5M_{sun} \\ a(M-5)^b & M \geq 5M_{sun} \end{cases} \quad (7.8-1)$$

Here, M is in solar mass units, while a and b are parameters that can be determined for each initial metallicity by solving the simultaneous equations for masses $15M_{sun}$ and $10M_{sun}$ given by

$$\begin{aligned} Z_{ej}(15) &= a \times 10^b \\ Z_{ej}(10) &= a \times 5^b \end{aligned} \quad (7.8-2)$$

$Z_{ej}(15)$ and $Z_{ej}(10)$ are the values of $Z_{ej}(M)$ for initial masses of $15M_{sun}$ and $10M_{sun}$ respectively. For the three different initial values of Z , the values of the parameters a and b are given in the following table;

	Initial Z	a	b
Population IIIa	10^{-10}	7.085×10^{-2}	1.872×10^{-1}
Population IIIb	10^{-12}	5.274×10^{-2}	3.552×10^{-1}
Population IIIc	0.0	3.249×10^{-2}	5.333×10^{-1}

We can see that the function given in equation (7.8-1) has several disadvantages. The most obvious of these is that the value for Z_{ej} does not tend to a finite value as M increases. We would have to impose a cut-off point to the function to avoid any form of integration or summation going to infinity. This cut-off point could be given by mass or metallicity, but would have to be selected in a fairly arbitrary manner given our lack of knowledge of the behaviour of these stars. In addition, it would be better if we determined the values of Z_{ej} at lower masses from a function fitted to the known values rather than making prior assumptions. It would therefore be preferable for us to fit a function for Z_{ej} that avoided these problems. For example, we can take a function for Z_{ej} of the form

$$Z_{ej}(M) = 1 - e^{-aM} \quad (7.8-3)$$

where as before, M is in solar units and a is an adjustable parameter. We can then find the best fit to the values for Z_{ej} that we have. The values of a for the three different initial metallicities are given in the following table;

	Initial Z	a
Population IIIa	10^{-10}	5.437×10^{-3}
Population IIIb	10^{-12}	6.929×10^{-3}
Population IIIc	0.0	7.179×10^{-3}

The function given in equation (7.8-3) gives the values $Z=0$ at the lower boundary of $M=0$ and $Z=1$ at the upper boundary of $M=\infty$. These values preclude us from having to truncate this function at all should we need to sum or integrate it.

We can use equation (7.8-3) to determine the values of Z_{ej} for the $5M_{\text{sun}}$ and $2M_{\text{sun}}$ models with the three initial metallicities. Once we have calculated these values, we can invert equation (7.7-1) to determine the remnant mass M_{rem} for these models as well. These values are given in the following table, with M_{rem} in solar units and as a fraction of the initial mass.

	Z_{ej}	$M_{\text{rem}} (M_{\text{sun}})$	M_{rem}/M
Population IIIa			
$5M_{\text{sun}}$	2.682×10^{-2}	0.5188	1.038×10^{-1}
$2M_{\text{sun}}$	1.082×10^{-2}	0.6235	3.118×10^{-1}
Population IIIb			
$5M_{\text{sun}}$	3.405×10^{-2}	0.4853	9.706×10^{-2}
$2M_{\text{sun}}$	1.251×10^{-2}	0.6212	3.106×10^{-1}
Population IIIc			
$5M_{\text{sun}}$	3.526×10^{-2}	0.4795	9.590×10^{-2}
$2M_{\text{sun}}$	1.426×10^{-2}	0.6187	3.094×10^{-1}

These masses indicate that these remnants would be some form of white dwarf.

We can determine a function to fit the values we have obtained for Y_{ej} in much the same manner as for Z_{ej} . We need a function that takes the value $Y_{ej}=0.25$, the initial mass fraction of helium, at $M=0$. In addition, the function would have to take the value $Y_{ej}=0$ at $M=\infty$, as we have already defined Z_{ej} as being unity at this mass. While it would be possible to define a function consisting of an increasing exponent multiplied by a decreasing exponent, it seems simplest to divide the range at the highest value for Y_{ej} and fit a curve to either side of this dividing point. For example, for models of population IIIc we can take the dividing point at $M=10M_{sun}$ as the highest value of Y_{ej} and fit curves of the form

$$Y_{ej}(M) = \begin{cases} 0.25e^{aM} & M < 10M_{sun} \\ Y_{ej}(10)e^{-b(M-10)} & M \geq 10M_{sun} \end{cases} \quad (7.8-4)$$

$Y_{ej}(10)$ is the mass fraction of helium in the ejected matter for a $10M_{sun}$ model. The parameters a and b can be determined from the values for $Y_{ej}(10)$ and $Y_{ej}(15)$, and are given by $a=2.359 \times 10^{-2}$, $b=1.199 \times 10^{-2}$. For models of population IIIa, the highest value of Y_{ej} is at $M=15M_{sun}$, so the fit will be different. In this case, the curves for $Y_{ej}(M)$ can be given by

$$Y_{ej}(M) = \begin{cases} 0.25e^{aM} & M < 15M_{sun} \\ Y_{ej}(15)e^{-b(M-15)} & M \geq 15M_{sun} \end{cases} \quad (7.8-5)$$

Here, the best for masses $M < 15M_{sun}$ is obtained by taking $a=1.288 \times 10^{-2}$. This is not an exact fit, however, for the value of $Y_{ej}(M)$ at $10M_{sun}$. The value for b becomes more or less arbitrary, as we do not have any more points to fit the curve to. We may best take b as the same value as for the models of population IIIb as the closest population in initial composition. The population IIIb models have their highest value of Y_{ej} at a mass of $10M_{sun}$, the same as the models of population IIIc. Therefore we may use the same procedure, and fit the functions given in equation (7.8-4) in the same way. In this case, we find that $a=1.900 \times 10^{-2}$ and $b=3.659 \times 10^{-3}$. From the equations (7.8-4, 5) we can determine the values of Y_{ej} for the models of $5M_{sun}$ and $2M_{sun}$, and these values are given for the three initial metallicities in the following table.

	Population IIIa Y_{ej}	Population IIIb Y_{ej}	Population IIIc Y_{ej}
$5M_{sun}$	0.2666	0.2749	0.2813
$2M_{sun}$	0.2565	0.2579	0.2621

7.9 The Rates of Increase of Helium and Metals From Nuclear Processing

As we wish to use our results to determine an initial mass function for our population III, it is necessary to express the rates at which the models of different masses increase the mass fractions Y and Z in the interstellar medium. These rates can be determined, for a model of a given mass M , from the following simple equations ;

$$\begin{aligned} \frac{dZ}{dt} &= \dot{Z}_M = \frac{Z_{ej}(M) - Z_i}{t_L(M)} \\ \frac{dY}{dt} &= \dot{Y}_M = \frac{Y_{ej}(M) - Y_i}{t_L(M)} \end{aligned} \quad (7.9-1)$$

Here, $t_L(M)$ is the total lifetime of a star of initial mass M , including the time taken for the star to form. Z_i and Y_i are the initial mass fractions of metals and helium respectively. We can estimate $t_L(M)$ to be given by

$$t_L(M) = 2 \times t_M \quad (7.9-2)$$

where t_M is the time taken for a model of mass M to evolve from zero-age up until the formation of a helium-depleted core, or in other words the evolved times given for the more massive models in chapter six. For the lower mass models, we assume t_M to be approximately the time taken to reach core helium ignition. Of course, by proposing that dY/dt and dZ/dt are given by (7.9-1) we are assuming that the changes in Y and Z are occurring at a steady rate. This is not actually the case, of course. As we have already stated, the models presented in this work provide no significant increase in Y or Z for the larger part of their lifetimes. We assume all the enriched material to be returned at one time, during a supernova event, in a burst of enrichment. However, not all stars in a population will form at the same time, and stars of different masses will 'die' at differing times in any case. The fact that there is a spectrum of stars at different masses will help to smear out the rates of enrichment, so that they can be approximated by simple functions such as equation (7.9-1), rather than by some form of step function.

We could determine functions for dY/dt and dZ/dt by finding a function expressing t_L with mass and dividing the functions obtained for Y_{ej} and Z_{ej} in the previous section by this new function. However, it is easier to fit a function to the values of dY/dt and dZ/dt determined directly from the data. For the more massive stars, we can determine dY/dt and dZ/dt from the values of Y_{ej} , Z_{ej} and the evolved times Δt .

	Δt (yrs)	dY/dt (yrs ⁻¹)	dZ/dt (yrs ⁻¹)
Population IIIa			
15M _{SUN}	1.211×10 ⁷	2.199×10 ⁻⁹	3.547×10 ⁻⁹
10M _{SUN}	2.168×10 ⁷	1.098×10 ⁻⁹	1.745×10 ⁻⁹
Population IIIb			
15M _{SUN}	1.276×10 ⁷	1.835×10 ⁻⁹	3.932×10 ⁻⁹
10M _{SUN}	2.280×10 ⁷	1.147×10 ⁻⁹	1.546×10 ⁻⁹
Population IIIc			
15M _{SUN}	1.246×10 ⁷	1.929×10 ⁻⁹	3.642×10 ⁻⁹
10M _{SUN}	2.046×10 ⁷	1.625×10 ⁻⁹	1.083×10 ⁻⁹

We can now fit both dY/dt and dZ/dt to functions in mass. We would require that these gave $dY/dt=dZ/dt=0$ at mass $M=0$ as no nuclear processing would be occurring, and that the functions will not become large too quickly above a mass of $15M_{SUN}$. There are other constraints on the choice of the functions, and the reader should refer to appendix A for a general treatment of the problem. For the value of dZ/dt at a mass M we will use a function of the form

$$\frac{dZ}{dt} = \dot{Z}_M = aM^b, \quad (7.9-3)$$

where a is a parameter to be determined. For the value of dY/dt we can use a function of the same form, so that

$$\frac{dY}{dt} = \dot{Y}_M = cM^b, \quad (7.9-4)$$

where c is another parameter which is to be determined. We have adopted a value of $b=4$ as the most appropriate in this case, as the rates of nucleosynthesis should vary with mass in much the same way as the luminosity. Mass-luminosity functions can be given by the same form of power law shown in equations (7.9-3, 4) with a value of b in the range $3 < b < 5$. With the use of a single parameter in each equation, we can find a best fit to the values of dY/dt and dZ/dt at $15M_{SUN}$ and $10M_{SUN}$. The functions (7.9-3, 4) will tend to increase rapidly with M due to the high value of b . However, this is tolerable for reasons discussed in appendix A. The values of the parameters a and c in functions (7.9-3, 4) for models of population IIIa, IIIb and IIIc are given in the following table.

	Initial Z	a	c
Population IIIa	10 ⁻¹⁰	1.161×10 ⁻¹³	7.662×10 ⁻¹⁴
Population IIIb	10 ⁻¹²	1.223×10 ⁻¹³	7.547×10 ⁻¹⁴
Population IIIc	0.0	8.835×10 ⁻¹⁴	1.003×10 ⁻¹³

We can now use the functions (7.9-3,4) to find the values for dY/dt and dZ/dt at masses of $5M_{\text{sun}}$ and $2M_{\text{sun}}$. Accordingly, these values are tabled below.

	dY/dt (yrs ⁻¹)	dZ/dt (yrs ⁻¹)
Population IIIa		
$5M_{\text{sun}}$	4.789×10^{-11}	7.256×10^{-11}
$2M_{\text{sun}}$	1.226×10^{-12}	1.858×10^{-12}
Population IIIb		
$5M_{\text{sun}}$	4.716×10^{-11}	7.644×10^{-11}
$2M_{\text{sun}}$	1.208×10^{-12}	1.957×10^{-12}
Population IIIc		
$5M_{\text{sun}}$	6.269×10^{-11}	5.521×10^{-11}
$2M_{\text{sun}}$	1.605×10^{-12}	1.414×10^{-12}

7.10 The Rate at Which Material Is Deposited Into Remnants

From our results, we can determine the rate at which material is deposited in stellar remnants. Consider the equation

$$\frac{d(M_{\text{rem}} / M)_M}{dt} = \frac{(M_{\text{rem}} / M)_M}{t_L} \quad (7.10-1)$$

where $t_L(M)$ is the lifetime of a model of initial mass M , given in equation (7.9-2), and $(M_{\text{rem}}/M)_M$ is the fractional mass of the remnant of a model of initial mass M . This equation (7.10-1) gives the rate at which mass is deposited into stellar remnants by models of mass M . However, we can only consider this rate $d(M_{\text{rem}}/M)/dt$ to be an initial rate with respect to the total mass that was initially available to form stars. As stars 'die' and form remnants, the mass available to form new stars is reduced by the mass now contained in these remnants. So if the next generation of stars forms remnants of the same fractional mass, the fraction of the total mass (including that already in stellar remnants) that is deposited as stellar remnants by this generation will be lower than by the previous generation. So we should write the rate $d(M_{\text{rem}}/M)/dt$ as a function of time t , in which we take account of the fraction of the total initial mass that is in contained in stellar remnants at time t . We can write

$$\left(\frac{d(M_{\text{rem}} / M)_M}{dt} \right)_t = \left(\frac{d(M_{\text{rem}} / M)_M}{dt} \right)_{t=0} (1 - M_t) \quad (7.10-2)$$

where M_t is the fraction of the total initial mass contained in stellar remnants at some time t . In order to calculate or even approximate this fraction, we must have a function for $d(M_{\text{rem}}/M)/dt$ as a function of

mass, and a function describing the distribution of mass in the stars of a population III. This will be discussed later in this chapter.

From the values for M_{rem} given in preceding sections, we may fit a function for $d(M_{\text{rem}}/M)/dt$ with mass. The following table gives the values of M_{rem}/M and $d(M_{\text{rem}}/M)/dt$ in years⁻¹ calculated from equation (7.10-1). The rate given here should be considered the initial rate.

	M_{rem}/M	$d(M_{\text{rem}}/M)/dt$
Population IIIa		
15 M_{sun}	4.922×10^{-2}	2.032×10^{-9}
10 M_{sun}	4.429×10^{-2}	1.021×10^{-9}
5 M_{sun}	1.038×10^{-1}	6.838×10^{-10}
2 M_{sun}	3.118×10^{-1}	1.934×10^{-10}
Population IIIb		
15 M_{sun}	4.922×10^{-2}	1.653×10^{-9}
10 M_{sun}	4.899×10^{-2}	1.074×10^{-9}
5 M_{sun}	9.706×10^{-2}	6.292×10^{-10}
2 M_{sun}	3.106×10^{-1}	1.683×10^{-10}
Population IIIc		
15 M_{sun}	4.397×10^{-2}	1.770×10^{-9}
10 M_{sun}	6.182×10^{-2}	1.491×10^{-9}
5 M_{sun}	9.590×10^{-2}	5.948×10^{-10}
2 M_{sun}	3.094×10^{-1}	1.889×10^{-10}

We can fit the values for $d(M_{\text{rem}}/M)/dt$ to a simple linear function in mass, given by

$$\frac{d(M_{\text{rem}}/M)_M}{dt} = dM \quad , \quad (7.10-3)$$

where d is an adjustable parameter. The best fits for the three initial metallicities are given below ;

$$\begin{aligned} \text{Population IIIa,} & \quad d=1.178 \times 10^{-10}, \\ \text{Population IIIb,} & \quad d=1.069 \times 10^{-10}, \\ \text{Population IIIc,} & \quad d=1.201 \times 10^{-10}. \end{aligned}$$

7.11 The Distribution and Initial Mass Functions for a Population III

The discussion of functions describing the values of dY/dt and dZ/dt with mass can lead to the determination of a set of number probability distributions with mass and initial mass functions from the results presented in this work. The number probability distribution describes the fraction of the total number of stars which lie within a given range of mass. In a general form, we have that

$$\Phi_M dM = f_n dM \quad , \quad (7.11-1)$$

where f_n is some function describing the form of the number probability distribution. The value $\Phi_M dM$ is the fraction of the total number of stars which have an initial mass in the range $[M, M+dM]$. As this is expressed as a fraction, we would wish that the integral of Φ_M over the full range of M be given by unity, so that

$$\int_0^{\infty} \Phi_M dM = 1 \quad . \quad (7.11-2)$$

The initial mass function describes the distribution of the total amount of mass in stars of varying masses. In a general form, we have that

$$\Psi_M dM = f_m dM \quad , \quad (7.11-3)$$

where f_m is some function which is used to describe the form of the initial mass function. The value $\Psi_M dM$ gives the fraction of the total mass available to form stars that is contained in stars of an initial mass in the interval $[M, M+dM]$. As the initial mass function is expressed as a fraction of some total mass, we would require that the integral of Ψ_M over the full range of M be given by unity, so that

$$\int_0^{\infty} \Psi_M dM = 1 \quad . \quad (7.11-4)$$

These forms of the number probability distribution and the initial mass function would require that $\Phi_M = \Psi_M = 0$ at $M = \infty$, in order to ensure that the integrals (7.11-2) and (7.11-4) can be finite. The initial mass function Ψ_M is related to the number probability distribution Φ_M by

$$\Psi_M \propto M \Phi_M \quad . \quad (7.11-5)$$

If we choose a general form for Φ_M , then we can determine the general form of the initial mass function Ψ_M from equations (7.11-4, 5). The specific form of Ψ_M can then be found from a consideration of the functions for dZ/dt given in the previous section. The reader may like to refer to the general analysis of the forms taken by Φ_M and Ψ_M which is given in appendix A.

We have chosen to use a number probability distribution given by

$$\Phi_M \propto M e^{-\alpha M} \quad , \quad (7.11-6)$$

where α is a parameter to be determined. Equation (7.11-6) is the $\beta=1$ case from the analysis given in appendix A. We can find normalise the function Φ_M and find the constant of proportionality from equation (7.11-2). If we integrate the function $Me^{-\alpha M}$ over the range of M we find that

$$\int_0^{\infty} Me^{-\alpha M} = \frac{1}{\alpha^2} \quad (7.11-7)$$

To normalise this to unity, we let Φ_M be given by

$$\Phi_M = \alpha^2 Me^{-\alpha M} \quad (7.11-8)$$

From equation (7.11-5) we then have that the initial mass function Ψ_M is given by

$$\Psi_M \propto \alpha^2 M^2 e^{-\alpha M} \quad (7.11-9)$$

which we can normalise from equation (7.11-4). If we integrate the function on the right-hand side of equation (7.11-9) over the range of M we find that

$$\int_0^{\infty} \alpha^2 M^2 e^{-\alpha M} dM = \frac{2}{\alpha} \quad (7.11-10)$$

so that in order for Ψ_M to be normalised, it must be given by

$$\Psi_M = \frac{\alpha^3}{2} M^2 e^{-\alpha M} \quad (7.11-11)$$

We can now consider the values taken by α for the different initial compositions. After some time t of the enrichment of the metallicity of the interstellar medium, we would wish to find that a population III has produced an average metallicity equal to that observed in the most metal-poor stars in our epoch. As discussed in appendix A, the average value of dZ/dt , the rate of change of the mass fraction Z , over an entire population III can be expressed as

$$\frac{\overline{dZ}}{dt} = \frac{\overline{Z}}{Z} = \frac{\int_0^{\infty} f_r(M) \zeta \dot{Z}_M M \Phi_M dM}{\int_0^{\infty} f_r(M) M \Phi_M dM} \quad (7.11-12)$$

where $f_r(M)$ is the fraction of mass returned to the interstellar medium by a star of mass M , and the parameter ζ gives the fraction of the total available mass which actually goes into making up stars. We can give the function $f_r(M)$ as

$$f_r(M) = \gamma M^\delta \quad , \quad (7.11-13)$$

where γ and δ are parameters to be chosen or fitted. We have chosen to take the parameter $\delta=0.05$ in order to avoid the function exceeding unity at too low a mass. It is acceptable for this function to exceed unity for high masses, as discussed in appendix A. The values for γ leading to the best fits of the function $f_r(M)$ are given by

Population IIIa,	$\gamma=0.8356,$
Population IIIb,	$\gamma=0.8411,$
Population IIIc,	$\gamma=0.8420.$

If we wish that a population III is to produce an average metallicity in the interstellar medium of $Z=10^{-6}$, comparable to values for the most metal-poor population II stars observed in our epoch, after a time t , then from equation (7.11-12) we may express this as

$$\bar{Z}t = \frac{t \int_0^\infty f_r(M) \zeta \dot{Z}_M M \Phi_M dM}{\int_0^\infty f_r(M) M \Phi_M dM} \quad , \quad (7.11-14)$$

where t would have to be given in years. Using the functions given in (7.9-3), (7.11-8) and (7.11-13) we may write equation (7.11-14) as

$$\frac{ta\gamma\zeta \int_0^\infty M^{6.05} e^{-\alpha M} dM}{\gamma \int_0^\infty M^{2.05} e^{-\alpha M} dM} = 10^{-6} \quad , \quad (7.11-15)$$

which may be integrated to give

$$\frac{ta\zeta}{\alpha^4} \frac{6.05!}{2.05!} = 10^{-6} \quad . \quad (7.11-16)$$

From the relation (7.11-16), we can determine values for the parameter α for different assumed values of t , ζ and different initial metallicities represented by the parameter a from equation (7.9-3). We should note that in obtaining the relation (7.11-16), we have assumed that stars forming

from material of a metallicity greater than the initial values of $Z=0$, 10^{-12} or 10^{-10} will have the same rates of nucleosynthesis dZ/dt as those determined for the stars formed initially. In addition, we have ignored the possibility that at lower masses there may be stars that will not contribute at all if the time t is less than their lifetimes. In the following table, we give the values taken by α at different values of t for $\zeta=1$, or all the material initially available went into the formation of stars.

t (yrs)	Population IIIa α	Population IIIb α	Population IIIc α
1×10^7	4.589	4.530	4.230
1×10^8	8.161	8.055	7.523
5×10^8	12.203	12.045	11.250
1×10^9	14.512	14.324	13.379
2×10^9	17.257	17.034	15.910
3×10^9	19.099	18.852	17.607

In the present epoch we expect there to be at least as much material in the interstellar medium as there is contained in stars. Accordingly, the next table gives the values of α at different values of t for a more conservative value of $\zeta=0.5$.

t (yrs)	Population IIIa α	Population IIIb α	Population IIIc α
1×10^7	3.859	3.809	3.557
1×10^8	6.863	6.773	6.326
5×10^8	10.261	10.129	9.460
1×10^9	12.203	12.045	11.250
2×10^9	14.511	14.324	13.379
3×10^9	16.060	15.853	14.806

Finally, we can give the same results for an extreme value of $\zeta=0.1$, where the mass contained in stars is only 10% of the total mass available to form stars.

t (yrs)	Population IIIa α	Population IIIb α	Population IIIc α
1×10^7	2.581	2.547	2.379
1×10^8	4.590	4.529	4.230
5×10^8	6.862	6.774	6.326
1×10^9	8.161	8.055	7.523
2×10^9	9.704	9.579	8.947
3×10^9	10.740	10.602	9.901

We should note that these values for α , in conjunction with the distribution function (7.11-8), indicate a distribution of masses very different to that observed in our epoch. The most common forms for the distribution function used for the solar neighbourhood are variations of

the function first described by Salpeter (1955). From the work of Salpeter, we can take a distribution function given by

$$\Phi_M = 0.03M^{-2.35} \quad , \quad (7.11-17)$$

where M is in solar units. We can determine from equation (7.11-17) the relative proportions of $1M_{\text{SUN}}$ and $10M_{\text{SUN}}$ stars, which is given by

$$\frac{\Phi_{10}}{\Phi_1} = \frac{1.340 \times 10^{-4}}{3.000 \times 10^{-2}} = 4.467 \times 10^{-3} \quad . \quad (7.11-18)$$

Returning to our population III distribution function, if we take the lowest value for α given above, $\alpha=2.379$ we can determine the relative proportions of $1M_{\text{SUN}}$ and $10M_{\text{SUN}}$ stars for this case. This is given by

$$\frac{\Phi_{10}}{\Phi_1} = \frac{2.636 \times 10^{-9}}{5.245 \times 10^{-1}} = 5.027 \times 10^{-9} \quad , \quad (7.11-19)$$

From an examination of the function Φ_M we can see that with increasing α , the value of Φ_{10}/Φ_1 will decrease rapidly, as the exponential part of the function dominates. Thus even for the lowest value of α from our results we find that more massive stars of a population III are more uncommon, by a factor of 10^6 in equations (7.11-18, 19), than would be expected in our epoch.

Now that we have determined the forms of the number probability distribution Φ_M and the initial mass function Ψ_M , we can comment on what these functions imply. We can find the most probable mass from considering the maximum value taken by Φ_M , or one of the values of M at which

$$\frac{d\Phi_M}{dM} = 0 \quad . \quad (7.11-18)$$

From equation (7.11-8) this differential is be given by

$$\frac{d\Phi_M}{dM} = \alpha^2(1 - \alpha M)e^{-\alpha M} \quad , \quad (7.11-19)$$

and so the differential (7.11-19) is zero when either $M \rightarrow \infty$ or M is the most probable mass M_p given by

$$M_p = \frac{1}{\alpha} \quad (7.11-20)$$

The most probable masses for different times t are given in the following table for $\zeta=1$.

t (yrs)	Population IIIa M_p	Population IIIb M_p	Population IIIc M_p
1×10^7	2.179×10^{-1}	2.208×10^{-1}	2.364×10^{-1}
1×10^8	1.225×10^{-1}	1.241×10^{-1}	1.329×10^{-1}
5×10^8	8.195×10^{-2}	8.302×10^{-2}	8.889×10^{-2}
1×10^9	6.891×10^{-2}	6.981×10^{-2}	7.474×10^{-2}
2×10^9	5.795×10^{-2}	5.871×10^{-2}	6.285×10^{-2}
3×10^9	5.236×10^{-2}	5.304×10^{-2}	5.680×10^{-2}

The next table contains the same results for a value of $\zeta=0.5$.

t (yrs)	Population IIIa M_p	Population IIIb M_p	Population IIIc M_p
1×10^7	2.591×10^{-1}	2.625×10^{-1}	2.811×10^{-1}
1×10^8	1.457×10^{-1}	1.476×10^{-1}	1.581×10^{-1}
5×10^8	9.746×10^{-2}	9.873×10^{-2}	1.057×10^{-1}
1×10^9	8.195×10^{-2}	8.302×10^{-2}	8.889×10^{-2}
2×10^9	6.891×10^{-2}	6.981×10^{-2}	7.474×10^{-2}
3×10^9	6.227×10^{-2}	6.308×10^{-2}	6.754×10^{-2}

The next table gives the same results for a value of $\zeta=0.1$.

t (yrs)	Population IIIa M_p	Population IIIb M_p	Population IIIc M_p
1×10^7	3.874×10^{-1}	3.926×10^{-1}	4.203×10^{-1}
1×10^8	2.179×10^{-1}	2.208×10^{-1}	2.364×10^{-1}
5×10^8	1.457×10^{-1}	1.476×10^{-1}	1.581×10^{-1}
1×10^9	1.225×10^{-1}	1.241×10^{-1}	1.329×10^{-1}
2×10^9	1.031×10^{-1}	1.044×10^{-1}	1.118×10^{-1}
3×10^9	9.311×10^{-2}	1.585×10^{-2}	1.010×10^{-1}

It is worth noting that some of these values for the most probable mass M_p lie below the minimum mass required for hydrogen ignition in a star.

The mean mass in our population III can be determined from the integral

$$\bar{M} = \int_0^{\infty} M \Phi_M dM \quad (7.11-21)$$

This leads to a mean mass given by

$$\bar{M} = \int_0^{\infty} \alpha^2 M^2 e^{-\alpha M} dM = \frac{2}{\alpha} \quad , \quad (7.11-22)$$

twice the most probable mass M_p . The values for the mean mass of a population III are tabled below for different times t and a value $\zeta=1$.

t (yrs)	Population IIIa Mean M	Population IIIb Mean M	Population IIIc Mean M
1×10^7	4.358×10^{-1}	4.415×10^{-1}	4.728×10^{-1}
1×10^8	2.450×10^{-1}	2.482×10^{-1}	2.658×10^{-1}
5×10^8	1.639×10^{-1}	1.660×10^{-1}	1.778×10^{-1}
1×10^9	1.378×10^{-1}	1.396×10^{-1}	1.495×10^{-1}
2×10^9	1.159×10^{-1}	1.174×10^{-1}	1.257×10^{-1}
3×10^9	1.047×10^{-1}	1.061×10^{-1}	1.136×10^{-1}

The next table contains the same results for a value of $\zeta=0.5$.

t (yrs)	Population IIIa Mean M	Population IIIb Mean M	Population IIIc Mean M
1×10^7	5.183×10^{-1}	5.251×10^{-1}	5.623×10^{-1}
1×10^8	2.914×10^{-1}	2.952×10^{-1}	3.162×10^{-1}
5×10^8	1.949×10^{-1}	1.975×10^{-1}	2.114×10^{-1}
1×10^9	1.639×10^{-1}	1.660×10^{-1}	1.778×10^{-1}
2×10^9	1.378×10^{-1}	1.396×10^{-1}	1.495×10^{-1}
3×10^9	1.245×10^{-1}	1.262×10^{-1}	1.351×10^{-1}

The next table gives the same results for a value of $\zeta=0.1$.

t (yrs)	Population IIIa Mean M	Population IIIb Mean M	Population IIIc Mean M
1×10^7	7.749×10^{-1}	7.852×10^{-1}	8.407×10^{-1}
1×10^8	4.358×10^{-1}	4.416×10^{-1}	4.728×10^{-1}
5×10^8	2.914×10^{-1}	2.952×10^{-1}	3.162×10^{-1}
1×10^9	2.450×10^{-1}	2.482×10^{-1}	2.658×10^{-1}
2×10^9	2.062×10^{-1}	2.088×10^{-1}	2.236×10^{-1}
3×10^9	1.862×10^{-1}	3.170×10^{-1}	2.020×10^{-1}

We can consider the minimum mass for hydrogen ignition to be $\sim 0.08 M_{\text{sun}}$. For a quantitative evaluation of this minimum mass, see Böhm-Vitense (1992). It would be interesting to determine what fraction of the total mass making up a population III lies in stars with masses too low to ignite their hydrogen, often referred to as 'brown dwarfs'. In the determination of the functions for dZ/dt and dY/dt we were assuming that these low mass stars were not present in significant numbers, so we could afford to use functions which gave small rates of change dZ/dt and dY/dt at these masses. Therefore if a significant proportion of the total mass is contained in these low mass stars, it could invalidate our choice of

functions for dZ/dt and dY/dt . The fraction of the mass contained in stars which lies in stars of masses $M \leq 0.08$, $f_{0.08}$, is given by the integral

$$f_{M_u} = \int_0^{M_u} \Psi_M dM, \quad (7.11-23)$$

where M_u is the upper limit of the integral, in this case $M_u=0.08$. From the form of Ψ_M given in equation (7.11-11), we can write equation (7.11-23) as

$$f_{M_u} = \frac{\alpha^3}{2} \int_0^{M_u} M^2 e^{-\alpha M} dM. \quad (7.11-24)$$

We can integrate equation (7.11-24) to give

$$f_{M_u} = 1 - \left(\frac{\alpha^2}{2} M_u^2 + \alpha M_u + 1 \right) e^{-\alpha M_u}. \quad (7.11-25)$$

To give the fraction of the total mass contained in stars of a mass less than $M=0.08$, then we would have to multiply $f_{0.08}$ by the fraction of material that goes into the formation of stars. This fraction would thus be given by $\zeta f_{0.08}$. The values of the fraction $\zeta f_{0.08}$ are given in the following table for $\zeta=1$.

t (yrs)	Population IIIa $f_{0.08}$	Population IIIb $f_{0.08}$	Population IIIc $f_{0.08}$
1×10^7	6.278×10^{-3}	6.060×10^{-3}	5.022×10^{-3}
1×10^8	2.866×10^{-2}	2.773×10^{-2}	2.330×10^{-2}
5×10^8	7.598×10^{-2}	7.373×10^{-2}	6.286×10^{-2}
1×10^9	1.122×10^{-1}	1.090×10^{-1}	9.368×10^{-2}
2×10^9	1.618×10^{-1}	1.576×10^{-1}	1.367×10^{-1}
3×10^9	1.982×10^{-1}	1.932×10^{-1}	1.686×10^{-1}

The same results are given in the following table for $\zeta=0.5$.

t (yrs)	Population IIIa $f_{0.08}$	Population IIIb $f_{0.08}$	Population IIIc $f_{0.08}$
1×10^7	3.897×10^{-3}	3.759×10^{-3}	3.107×10^{-3}
1×10^8	1.838×10^{-2}	1.776×10^{-2}	1.485×10^{-2}
5×10^8	5.047×10^{-2}	4.892×10^{-2}	4.141×10^{-2}
1×10^9	7.598×10^{-2}	7.373×10^{-2}	6.286×10^{-2}
2×10^9	1.121×10^{-1}	1.090×10^{-1}	9.367×10^{-2}
3×10^9	1.394×10^{-1}	1.356×10^{-1}	1.172×10^{-1}

The next table gives the same results for $\zeta=0.1$.

t (yrs)	Population IIIa $f_{0.08}$	Population IIIb $f_{0.08}$	Population IIIc $f_{0.08}$
1×10^7	1.258×10^{-3}	1.211×10^{-3}	9.968×10^{-4}
1×10^8	6.282×10^{-3}	6.056×10^{-3}	5.022×10^{-3}
5×10^8	1.812×10^{-2}	1.777×10^{-2}	1.485×10^{-2}
1×10^9	2.866×10^{-2}	2.773×10^{-2}	2.330×10^{-2}
2×10^9	4.408×10^{-2}	4.270×10^{-2}	3.609×10^{-2}
3×10^9	5.631×10^{-2}	5.459×10^{-2}	4.629×10^{-2}

We can see from the above tables, that unless we require a large time and a high value of the fraction ζ , there is not a significantly large fraction of the total mass in the form of brown dwarf stars of mass $M < 0.08 M_{\text{sun}}$. If a significant fraction $f_{0.08}$ of the mass in stars is in brown dwarfs, which will not produce any enrichment of the interstellar medium, then this consideration could affect the value of α . As we are taking functions for the rates of enrichment \dot{Y}_M and \dot{Z}_M of the form aM^b , there will be non-zero values at very low masses. However, providing that only a small fraction of the total mass in stars is contained in very low mass stars, and the rates \dot{Y}_M and \dot{Z}_M are small in this mass range, we can neglect this inconsistency.

7.12 Helium Enrichment Due To a Population III

From the initial mass function, Ψ_M , determined in the previous section and the functions for the rate of change of helium in the interstellar medium, dY/dt , we can determine the overall increase in the mass fraction of helium in the interstellar medium occurring over some time t . We have assumed that cosmological mechanisms produced helium to give a mass fraction $Y_i = 0.25$, and this is the initial mass fraction of helium in all the models presented in this work. So, following from the arguments leading to the equations (7.11-12, 14), we may write the mass fraction of helium Y_t after some time t as

$$Y_t = \frac{t\zeta \int_0^\infty f_r(M) \dot{Y}_M \Phi_M dM}{\int_0^\infty f_r(M)_r \Phi_M dM} + Y_i \quad , \quad (7.12-1)$$

where the function for dY/dt comes from equation (7.9-4). So we may write the change in the mass fraction Y due to a population III over some time t as

$$\Delta Y = \frac{ct\gamma\zeta \int_0^\infty M^{6.05} e^{-\alpha M} dM}{\gamma \int_0^\infty M^{2.05} e^{-\alpha M} dM} \quad . \quad (7.12-2)$$

Equation (7.12-2) integrates to form the relation

$$\Delta Y = \frac{ct\zeta}{\alpha^4} \frac{6.05!}{2.05!} \quad (7.12-3)$$

From the relation (7.12-3) we can determine the mass fraction of helium burned for the different values of t , ζ and α . We find that ΔY takes the same value at different values of t and ζ , as would be expected. As the rate of nucleosynthesis of helium is linked to the rate of nucleosynthesis of metals, we would not expect there to be different values of ΔY when the parameter α has been determined in each case from the assumption that a fixed value of Z has been reached. The values of ΔY for the three different initial compositions are given in the following table.

	Population IIIa	Population IIIb	Population IIIc
ΔY	6.240×10^{-7}	6.502×10^{-7}	1.136×10^{-6}

The values of ΔY tabled above are very much smaller than would be required to explain an initial mass fraction of helium in population II of $Y \sim 0.27$. Values in the range $0.23 < Y < 0.29$ are commonly used in works modelling population II stars. However, the above values for ΔY are only to be expected from the values of \dot{Y} determined from the models presented in this work, and the form of the function used to describe \dot{Y}_M .

7.13 The Mass Left As Remnants by a Population III

Having derived an initial mass function in section 7.11, we may now return to the consideration of stellar remnants. From the earlier discussion in section 7.10, and equation (7.10-2), we can approximate a value for M_t using the number distribution function. We can take M_t as being given by

$$M_t = t \int_0^\infty M \Phi_M \left(\frac{d(M_{\text{rem}} / M)_M}{dt} \right)_{t=0} dM \quad (7.13-1)$$

which from equation (7.10-3) and the number distribution function given in equation (7.11-8) becomes

$$M_t = t \alpha^2 d \int_0^\infty M^2 e^{-\alpha M} dM \quad (7.13-2)$$

Equation (7.14-2) will give the average initial value of $d(M_{\text{rem}}/M)/dt$ over an entire population III, multiplied by some time t . We should note

however, that as we are assuming the value of $d(M_{\text{rem}}/M)/dt$ to change over time, equation (7.14-2) is obviously only an approximation to the value of M_t . Equation (7.14-2) integrates to give us

$$M_t = \frac{2td}{\alpha} \quad (7.13-3)$$

We can now write equation (7.10-2) as

$$\left(\frac{d(M_{\text{rem}}/M)_M}{dt} \right)_t = \left(\frac{d(M_{\text{rem}}/M)_M}{dt} \right)_{t=0} \left(1 - \frac{2td}{\alpha} \right), \quad (7.13-4)$$

which from equation (7.10-3) becomes

$$\left(\frac{d(M_{\text{rem}}/M)_M}{dt} \right)_t = Md - \frac{2tMd^2}{\alpha} \quad (7.13-5)$$

We can use equation (7.14-5) and the number distribution function given in equation (7.11-8) to form an expression for the fraction of the total mass contained in stellar remnants after a time t . We will have to integrate the function given in equation (7.14-5) with the number distribution function over the range of masses, and then integrate the result over time to give the fraction of mass contained in stellar remnants at time t . We may write this fraction, $M_r(t)$, as

$$M_r(t) = \int_0^t \int_0^\infty M \Phi_M \left(\frac{d(M_{\text{rem}}/M)_M}{dt'} \right) dM dt', \quad (7.13-6)$$

where we have substituted a dummy variable t' for the time in the integral. Equation (7.14-6) may be written as

$$M_r(t) = \int_0^t \int_0^\infty (\alpha^2 d - 2\alpha d^2 t') M^2 e^{-\alpha M} dM dt', \quad (7.13-7)$$

and can be integrated to give

$$M_r(t) = \frac{2td}{\alpha} - \frac{2t^2 d^2}{\alpha^2} \quad (7.13-8)$$

We have assumed throughout these calculations that all of the initial material available to form stars will be used to do so. So the fraction $M_r(t)$ given in equation (7.14-8) is expressed as a fraction of the mass that went into the formation of stars. If ζ is the fraction of the total mass

available to form stars which does go into the formation of stars, then $\zeta M_{\text{r}}(t)$ will be the fraction of the total mass which is contained in stellar remnants at time t . The second term in equation (7.14-8) can be seen as a correction to take account of the continually decreasing mass available to form stars in each successive generation. As this term is small compared to the first term, we can assume that any errors introduced by the approximations used to determine M_{t} in equations (7.14-1, 3) will also be small. The values of $M_{\text{r}}(t)$ calculated from equation (7.14-8) are given for the three initial metallicities and various values of t in the following table, for a value of $\zeta=1$.

t (yrs)	Population IIIa $\zeta_{M_r}(t)$	Population IIIb $\zeta_{M_r}(t)$	Population IIIc $\zeta_{M_r}(t)$
1×10^7	5.134×10^{-4}	4.720×10^{-4}	5.678×10^{-4}
1×10^8	2.887×10^{-3}	2.654×10^{-3}	3.193×10^{-3}
5×10^8	9.653×10^{-3}	8.875×10^{-3}	1.068×10^{-2}
1×10^9	1.623×10^{-2}	1.493×10^{-2}	1.795×10^{-2}
2×10^9	2.730×10^{-2}	2.510×10^{-2}	3.019×10^{-2}
3×10^9	3.701×10^{-2}	3.402×10^{-2}	4.093×10^{-2}

The next table gives the same values for $\zeta=0.5$.

t (yrs)	Population IIIa $\zeta_{M_r}(t)$	Population IIIb $\zeta_{M_r}(t)$	Population IIIc $\zeta_{M_r}(t)$
1×10^7	3.053×10^{-4}	2.807×10^{-4}	3.376×10^{-4}
1×10^8	1.723×10^{-3}	1.578×10^{-3}	1.899×10^{-3}
5×10^8	5.740×10^{-3}	5.276×10^{-3}	6.348×10^{-3}
1×10^9	9.653×10^{-3}	8.875×10^{-3}	1.068×10^{-2}
2×10^9	1.624×10^{-2}	1.493×10^{-2}	1.795×10^{-2}
3×10^9	2.200×10^{-2}	2.023×10^{-2}	2.433×10^{-2}

The next table gives the same results for a value of $\zeta=0.1$.

t (yrs)	Population IIIa $\zeta_{M_r}(t)$	Population IIIb $\zeta_{M_r}(t)$	Population IIIc $\zeta_{M_r}(t)$
1×10^7	9.128×10^{-5}	8.394×10^{-5}	1.010×10^{-4}
1×10^8	5.133×10^{-4}	4.721×10^{-4}	5.678×10^{-4}
5×10^8	1.718×10^{-3}	1.578×10^{-3}	1.899×10^{-3}
1×10^9	2.887×10^{-3}	2.654×10^{-3}	3.193×10^{-3}
2×10^9	4.855×10^{-3}	4.464×10^{-3}	5.369×10^{-3}
3×10^9	6.581×10^{-3}	6.050×10^{-3}	7.278×10^{-3}

The fractions $\zeta_{M_r}(t)$ given in the above tables are small, but we should remember that these only represent the mass contained in stellar remnants. They do not take into account the mass contained in small stars such as brown dwarfs, or those which have lifetimes greater than the value of t .

7.14 Discussion and Conclusions

We have shown in this chapter that a population III could produce the mass fraction of metals observed in the photospheres of the most metal poor population II stars, while leaving only a small fraction of the total mass as remnants or brown dwarfs. It is important that not too much mass is 'removed' from the cycle of stellar evolution, otherwise we would expect to see more of these remnants in our epoch. The nucleosynthesis of the metals could occur in a time which is negligible compared to the lifetimes of the population II stars which will follow. The time that would have been taken to produce the required amount of metals depends upon

the form of the number distribution, and it is the mechanisms leading to the cooling of the collapsing gas clouds, as described in chapter one, that determine a characteristic (or most probable) mass and thence the number distribution. Unfortunately, there are no undisputed accounts of the mechanisms of stellar formation in the early universe to provide another means of determining a distribution function. For example, Silk (1977) suggests that the characteristic mass of stars forming in the primordial universe would be larger than that seen in our epoch. However, Palla, Stahler and Salpeter (1983) suggest that a lower characteristic mass is possible even in the complete absence of metals.

Perhaps the most surprising resulting from the calculations presented in this chapter is the very low increase ΔY in the mass fraction of helium due to the nuclear processing from a population III. From the literature, a median value for the initial value of Y in population II can be considered to be $Y \sim 0.27$. The amount of helium required to produce this value of Y cannot be produced by the population III considered in this work without producing an equivalent increase in metals, which obviously cannot be the case. To explain the mass fraction of helium we observe today, we must resort to a population II following on from a population III which has provided a significant enrichment of metals, but not of helium. A very general argument can be used to state that this population II would exist for a longer period of time, and have a more efficient CNO cycle due to the increased metal content. Thus this population II would have produced sufficient helium to explain the present values. For example, Dorman, Rood and O'Connell (1993) have presented models of population II stars that exhibit significant increases in Y in evolving between their zero-age and the horizontal branch stages. We should note, however, that the initial mass fraction Y_{popII} for a population II is by no means known to any great accuracy. The variation in the value of Y_{popII} between different authors is greater than the increase ΔY that we would require a population III to produce to reach $Y=0.27$.

The dividing of stars into populations when speaking of the evolution of the chemical composition of the interstellar medium is almost certainly misleading. This chemical evolution is a continuous process, and there is no clear cut point at which the stars cease to be 'population III' or become 'population II'. As the more massive stars evolve more rapidly, there can be several generations of these while the less massive stars have yet to reach an endstate. Thus stars that formed from material of different compositions could certainly exist at the same time. With increasing metallicity, however, the evolutionary behaviour of the forming stars will change as the composition of the interstellar medium becomes less 'population III' and more 'population II'. By assuming that the values \dot{Y} and \dot{Z} are constant with time, we have tried to approximate the effects of this continuous process as best we can. Of course, we have fitted the functions \dot{Y}_M and \dot{Z}_M to describe the rates at

which the material returned to the interstellar medium by stars varies with mass. This use of functions is an approximation made necessary by the small number of models evolved in this work. If a much larger number of models over a wider range of mass were evolved, then we could discard the use of functions and perform integrations such as (7.11-14) numerically, leading to results that better reflect the different behaviour of models at different masses.

The form of the number distribution function Φ_M that we are brought to by a consideration of the limit for the production of metals is different to that generally accepted for the solar neighbourhood in our epoch. This has been remarked upon earlier in this chapter, in that the more massive stars are much less common, by a factor of at least 10^6 , for our population III. Due to the way in which we have determined Φ_M , this is necessarily a consequence of the high values of \dot{Z}_M for $M=10M_{\text{sun}}$ and $M=15M_{\text{sun}}$, and to a lesser extent the chosen form of the functions \dot{Z}_M and Φ_M . However, it is worth remembering that as more observations have been made of lower mass (and hence less luminous) stars the distribution function for the solar neighbourhood is revised to give a lower proportion of more massive stars. It is still possible that we are overestimating the proportion of massive stars due to a lack of observations of very small, and hence less luminous, stars.

A final caution; on the basis of the few models examined in this work it is simply not possible to draw reliable conclusions on the evolution of the chemical composition of the early interstellar medium. All of the conclusions reached in this chapter, on the basis of data presented in chapter six or elsewhere, should be regarded as tentative at best. However, it was deemed appropriate to begin to attempt this task - and the fitting of distribution functions to the mass of heavy elements released by population III stars to the interstellar medium can be a first step only. As remarked earlier, a grid of models would ideally be used to replace the fitting of functions performed in this chapter.

Chapter Eight

Summary

8.1 Summary

We have modelled the evolution of population III stars at masses of $15M_{\text{sun}}$, $10M_{\text{sun}}$, $5M_{\text{sun}}$ and $2M_{\text{sun}}$, three initial metallicities of $Z=10^{-10}$, 10^{-12} and zero and the initial value of the mass fraction of helium in all models is taken to be $Y=0.25$. We have demonstrated that there are significant differences in the zero-age parameters and evolutionary behaviour of our models due to differences in initial metallicity in the range $0 \leq Z \leq 10^{-10}$. Such differences become more significant with increasing mass, and we have ascribed these differences to the contribution of the CNO-cycle to the total energy production rate of the models. We have also shown that, by the Nieuwenhuijzen and de Jager formula for mass loss rates, no loss of processed material occurs over the time our models were evolved. This is due to the low rate of mass loss in these models, in conjunction with the absence of the 'first dredge-up' of enriched material that would be expected in stars of population I or II.

By identifying the hydrogen-depleted cores of our evolved models with studies of the bulk yields of helium stars we have been able to determine approximate rates at which our models will return material to the interstellar medium in supernova events. We have used this information, together with an upper limit on the value of Z we would expect a population III to produce, in order to determine distribution functions for a population III initially of metallicity $Z=10^{-10}$, 10^{-12} and zero. The distribution functions lead to the helium production ΔY for a population III, which we have found to be surprising low. The values we have determined for ΔY , $n \times 10^{-4}$, are far too low to produce the median value in the literature of the population II initial helium content, and in fact are negligible compared to the mass fraction of helium thought to be produced by cosmological mechanisms.

We can state that, from our results, the existence of a population III is a plausible mechanism for the prompt enrichment of metals in the early universe. The amount of metals returned to the interstellar medium by the more massive stars is high enough to guarantee a significant level of enrichment in a very short time, provided the distribution function indicates that there are enough massive stars. Such a population III would have a mass distribution significantly different to that derived for the solar neighbourhood in our epoch. From our results, the proportion of massive stars is lower, by a factor of at least 10^6 , for our population III than for the observed population I or population II in our epoch.

Appendix A

A Discussion of the Functions Presented in Chapter Seven

A.1 A Consideration of the Distribution Functions

We must consider carefully when selecting the functions to describe both the distribution of stars in mass and the rates of nucleosynthesis in stars of masses. The constraints placed on our choice of functions come as much from the mathematics underlying the use of these functions as they do from physical considerations. In this appendix, we shall detail more general cases for the functions than were presented in chapter seven.

A.2 The Distribution of Stars with Initial Mass

We have no initial clues as to the form of the function describing the distribution of stars with mass. Bearing this in mind, it is best to take a family of functions in the hope that at least one will prove productive. We can use a family of functions Φ_M , where $\Phi_M dM$ gives the fraction of the total number of stars that have an initial mass in the range $[M, M+dM]$. As this function is expressed as a fraction, we can already set some limits on the values Φ_M must take. For example, Φ_M must be positive over the whole range of M , and must be normalised, that is

$$\int_0^{\infty} \Phi_M dM = 1 \quad . \quad (A.2-1)$$

We have selected the family of functions given by

$$\Phi_M \propto M^\beta e^{-\alpha M} \quad , \quad \beta=0, 1, 2, \dots \quad (A.2-2)$$

where β is zero or some positive integer and α is a parameter to be determined. By varying the value of the parameter β , we can select functions of different forms. These functions are easily normalised, as we find that

$$\int_0^{\infty} M^\beta e^{-\alpha M} dM = \frac{\beta!}{\alpha^{\beta+1}} \quad , \quad (A.2-3)$$

and so normalising the functions (A.2-2) gives us

$$\Phi_M = \frac{\alpha^{\beta+1}}{\beta!} M^\beta e^{-\alpha M} \quad . \quad (A.2-4)$$

It is worth noting that the normalisation process imposes the values $\beta \geq 0$. If β took a negative value, then we would have that $\beta! = \infty$ and normalisation of the function would clearly not be possible.

From the normalised functions (A.2-4) it is possible to find a most probable mass M_p . This occurs at the maxima of Φ_M , or one of the values of M leading to

$$\frac{d\Phi_M}{dM} = \frac{\alpha^{\beta+1}}{\beta!} (\beta M^{\beta-1} - \alpha M^\beta) e^{-\alpha M} = 0. \quad (\text{A.2-5})$$

This occurs either as $M \rightarrow \infty$, the minimum, or at a value for M_p given by

$$M_p = \frac{\beta}{\alpha}. \quad (\text{A.2-6})$$

We should note that for the $\beta=0$ function, this leads to a most probable mass $M_p=0$.

We can also determine an average mass from the normalised functions (A.2-4). This value can be determined from

$$\bar{M} = \int_0^\infty M \Phi_M dM, \quad (\text{A.2-7})$$

which can be written as

$$\bar{M} = \frac{\alpha^{\beta+1}}{\beta!} \int_0^\infty M^{\beta+1} e^{-\alpha M} dM. \quad (\text{A.2-8})$$

Equation (A.2-8) integrates to give us the average mass

$$\bar{M} = \frac{\beta+1}{\alpha}. \quad (\text{A.2-9})$$

From equation (A.2-4), the number distribution Φ_M , we can find a mass distribution Ψ_M . The value $\Psi_M dM$ gives the fraction of the total mass that is contained in stars of an initial mass in the range $[M, M+dM]$. We can define Ψ_M as

$$\Psi_M \propto M \Phi_M = M^{\beta+1} e^{-\alpha M}. \quad (\text{A.2-10})$$

As Ψ_M is expressed as a fraction, we would wish this function to be normalised so that

$$\int_0^\infty \Psi_M dM = 1. \quad (\text{A.2-11})$$

In conjunction with equation (A.2-10) this leads to Ψ_M given as

$$\Psi_M = \frac{\alpha^{\beta+2}}{(\beta+1)!} M^{\beta+1} e^{-\alpha M} \quad . \quad (\text{A.2-12})$$

A.3 Describing the Nucleosynthesis Rates with Mass

We cannot consider functions describing the rate of change \dot{X}_M of the mass fractions of elements X completely separately from the distribution functions Φ_M discussed in the preceding section. Our only constraints on the precise form of these distribution functions, in other words the values of the parameters α and β , come partially from the functions describing \dot{X}_M . To illustrate this, we can see that the average rate of change of the mass fraction X over an entire population of stars is given by

$$\frac{\overline{dX}}{dt} = \overline{\dot{X}} = \frac{\int_0^{\infty} M \dot{X}_M \Phi_M dM}{\int_0^{\infty} M \Phi_M dM} \quad . \quad (\text{A.3-1})$$

We can place a limit on this average rate from a consideration of the highest value of X we wish to produce after some time t . If we state values for this highest value X_{\max} and for t then we have that

$$X_{\max} = t \overline{\dot{X}} \quad , \quad (\text{A.3-2})$$

and we can choose a value for β and then determine the value of α for this given t and X_{\max} . The mean values are not quite as they are described in equation (A.3-1), but it illustrates the nature of the problem. We would obviously wish to select a function for \dot{X}_M which will not overly complicate the integrals while still satisfying our requirements in other ways.

Before writing down any forms taken by the function \dot{X}_M , we can place a few simple limitations on the way in which this function must behave. Firstly, we would wish that $\dot{X}_M \rightarrow 0$ as $M \rightarrow 0$ because there is little or no nuclear burning in very small objects. Secondly, the function must not get too large too quickly. It is acceptable for $\dot{X}_M \rightarrow \infty$ as $M \rightarrow \infty$, as we are combining \dot{X}_M with one of the functions Φ_M which decrease to zero as $M \rightarrow \infty$. In addition, the function should be positive or zero at all points. If we wish to fit a function exactly, then it must not have more than two

parameters, as these must be determined from only two points of data (those rates determined for the models of $15M_{\text{sun}}$ and $10M_{\text{sun}}$ in chapter seven). Accordingly, the first function we could consider is of the form

$$\dot{X}_M = aM + bM^2 \quad , \quad (\text{A.3-3})$$

where a and b are parameters to be determined from the data, and which will integrate easily in conjunction with the distribution functions (A.2-4). If we have the values of \dot{X}_M at two masses M_1 and M_2 then the parameters a and b are given by

$$\begin{aligned} a &= \frac{M_2^2 \dot{X}_1 - M_1^2 \dot{X}_2}{M_2^2 M_1 - M_1^2 M_2} \\ b &= \frac{M_2 \dot{X}_1 - M_1 \dot{X}_2}{M_2 M_1^2 - M_1 M_2^2} \end{aligned} \quad . \quad (\text{A.3-4})$$

This function (A.3-3) has the desired properties, in that it gives $\dot{X}_M=0$ at $M=0$ and will not increase too quickly with increasing M . In fitting this function, we may find that one of the parameters is negative, which would lead to unphysical results as the function \dot{X}_M would be negative over some of the range of M . If this is the case then this function cannot be used.

Instead of the power law form for \dot{X}_M given in equation (A.3-3) we could use an exponential form of the function. For example, we could take

$$\dot{X}_M = a(1 - e^{-bM}) \quad , \quad (\text{A.3-5})$$

where again a and b are parameters to be determined. This form of \dot{X}_M will integrate easily in conjunction with the distribution Φ_M , and satisfies the criterion that $\dot{X}_M=0$ at $M=0$. In addition, this function approaches a finite value as $M \rightarrow \infty$, meaning that it will not get too large with increasing mass as will most power law functions. If we are fitting the function (A.3-5) to data at two masses M_1 and M_2 with rates \dot{X}_1 and \dot{X}_2 respectively, when $M_1 < M_2$ and $\dot{X}_1 < \dot{X}_2$ then we would want the parameter $b > 0$. To find b we can take equation (A.3-5) at the two masses to give

$$\begin{aligned} \dot{X}_1 &= a(1 - e^{-bM_1}) \\ \dot{X}_2 &= a(1 - e^{-bM_2}) \end{aligned} \quad , \quad (\text{A.3-6})$$

and then divide one by the other to give

$$c = \frac{\dot{X}_2}{\dot{X}_1} = \frac{1 - e^{-bM_2}}{1 - e^{-bM_1}} \quad . \quad (\text{A.3-7})$$

The two masses are given by $M_1=10$ and $M_2=15$, and so if we make the substitution $x=e^{-5b}$ then equation (A.3-7) can be seen to be a cubic equation in x ;

$$x^3 - cx + (c - 1) = 0 \quad . \quad (\text{A.3-8})$$

We can not that $x=1$, or $b=0$, satisfies the equation but this is a degenerate case and of little use to us. We can factorize this solution out to give us

$$(x - 1)(x^2 - (c - 1)x - (c - 1)) = 0 \quad . \quad (\text{A.3-9})$$

As $\dot{X}_1 < \dot{X}_2$, we must have that $c > 1$, and so $(c-1) > 0$. Therefore it is the positive root of the quadratic term in equation (A.3-9) that we want, or

$$x = \frac{1}{2} \left[(c - 1) + \sqrt{(c - 1)^2 + 4(c - 1)} \right] \quad . \quad (\text{A.3-10})$$

We can note that as

$$b = -\frac{1}{5} \ln x \quad , \quad (\text{A.3-11})$$

for b to be positive, we must have that $0 < x < 1$, for which we require that $c = \dot{X}_2 / \dot{X}_1 < 3/2$. If this is not the case, then (A.3-5) would be an inappropriate function to choose.

A third possible choice of function for \dot{X}_M would be a simple power law of the form

$$\dot{X}_M = aM^b \quad , \quad (\text{A.3-11})$$

where a and b are the two parameters to be chosen or determined. We can fit the function (A.3-11) to the two points of data, or we can choose the value of b and then find a best fit of the function to the data by varying the parameter a . We can determine the parameters from the two points at masses of M_1 and M_2 ;

$$\begin{aligned}\dot{X}_1 &= aM_1^b \\ \dot{X}_2 &= aM_2^b\end{aligned}\quad (\text{A.3-12})$$

which leads to a value for b given by

$$b = \frac{\log \dot{X}_1 - \log \dot{X}_2}{\log M_2 - \log M_1} \quad (\text{A.3-13})$$

We can then find a by simple substitution.

We would wish to choose b rather than fit a value for this parameter in order to ensure that the lower mass stars in a population have appropriate values of \dot{X}_M . Certainly those stars below the mass required for hydrogen ignition should have no significant nucleosynthesis rates. By selecting a high value for b we can ensure that these lower mass stars will have low or near-zero values of \dot{X}_M . The disadvantage of selecting a high value for the parameter b is that mentioned earlier - above the point at which we are fitting the function, \dot{X}_M may increase too quickly with increasing mass. The choice of a large b is only justifiable if the final form of the distribution function (A.2-4) determined using the function (A.3-11) shows that the fraction of stars with large masses - above the fitting point for the function \dot{X}_M - is small or negligible.

A.4 The Fraction of Material Ejected

In the previous section, we stated that the rate of nucleosynthesis averaged over an entire population of stars could be given by equation (A.3-1). This equation in fact neglects an important weighting term. The rates \dot{X}_M are derived from the mass fractions of elements present in the ejected material of the models presented in this work. Therefore we would need to weight the mass in equation (A.3-1) by a term describing the fraction of the initial mass of a star that is returned to the interstellar medium over its lifetime. If we include this term, the resulting average rates can be seen to describe the chemical evolution of the interstellar medium rather than the stars. We can rewrite equation (A.3-1) with this additional weighting term as

$$\frac{\overline{dX}}{dt} = \overline{\dot{X}} = \frac{\int_0^{\infty} f_{r(M)} M \dot{X}_M \Phi_M dM}{\int_0^{\infty} f_{r(M)} M \Phi_M dM} \quad (\text{A.4-1})$$

where $f_{r(M)}$ is the fraction of a star of initial mass M that is returned to the interstellar medium over its lifetime. What function should we choose

for this fraction $f_r(M)$? We can state a few criteria for the form of the function immediately. As $f_r(M)$ is expressed as a fraction of some mass M , it should be positive over the range of M , and we would wish that $f_r(M) \rightarrow 0$ as $M \rightarrow 0$ as stars of a very low mass will not return any significant amount of their material to the interstellar medium. We would normally not wish this function to exceed unity at any point over the range of M , but if we have chosen a function for Φ_M that decreases rapidly with increasing M , then we can tolerate values of $f_r(M) > 1$ for large M . This is a similar argument to that for our tolerance of a rapidly increasing value of \dot{X}_M above the mass at which we fit the function. It is also obviously desirable to take a form of $f_r(M)$ that will not overly complicate that integral (A.2-1). Accordingly, we have taken a function for $f_r(M)$ of the form

$$f_{r(M)} = \gamma M^\delta \quad , \quad (\text{A.4-2})$$

where γ and δ are parameters to be fitted or chosen. This function will clearly exceed unity if both γ and δ are positive, as we would wish them to be. As this is so, it is best if we choose a value of δ to ensure that the function does not exceed unity at too low a mass. Once we have a value for δ , we can determine γ to best fit the data we have for the fraction of material returned.

A.5 The Fraction of Material Forming Stars

So far in our formulations, we have assumed that all the material available to form stars will be used to form stars. This is false assumption in our epoch, and a significant fraction of the total amount of mass in the solar neighbourhood is thought to be in the interstellar medium rather than stars. There is no reason to think that this would not be the case for the formation of a primordial population III. We can represent the fraction of the total mass that actually forms stars by the parameter ζ . We can assume that the rate \dot{X}_M for any given mass of star will be effectively 'diluted' by mixing the ejecta from the star with material that has not been processed in stars. So the true rate of change of the mass fraction of element X for stars of mass M becomes $\zeta \dot{X}_M$.

A.6 The Average Rates of Nucleosynthesis

In section A.3 we described several functions for \dot{X}_M along with reasons as to why they might not be suitable for use. In fact, the form of the data prevents the use of the functions described in equations (A.3-3) and (A.3-5) for the reasons touched upon in that section. We have therefore selected the form given by equation (A.3-11). If we use this form of \dot{X}_M in conjunction with the form for $f_r(M)$ given in equation (A.4-2)

and the form of Φ_M given in equation (A.2-4), then equation (A.4-1) can be written as

$$\bar{X} = \frac{a\gamma \int_0^{\infty} M^{\beta+b+\delta+1} e^{-\alpha M} dM}{\gamma \int_0^{\infty} M^{\beta+\delta+1} e^{-\alpha M} dM} \quad (\text{A.6-1})$$

If we then consider, as described in section A.5, that only a fraction ζ of the total amount of material goes into the formation of stars, then equation (A.6-1) becomes

$$\bar{X} = \frac{a\gamma\zeta \int_0^{\infty} M^{\beta+b+\delta+1} e^{-\alpha M} dM}{\gamma \int_0^{\infty} M^{\beta+\delta+1} e^{-\alpha M} dM} \quad (\text{A.6-2})$$

Equation (A.6-2) integrates to give

$$\bar{X} = \frac{a\zeta (\beta + b + \delta + 1)!}{\alpha^b (\beta + \delta + 1)!} \quad (\text{A.6-3})$$

It is interesting to note that the parameter γ does not appear in the expression (A.6-2). The only difficulty in evaluating this expression would occur if one or more of the parameters β , b or δ took non-integer values. In this case, we would make use of the equivalence

$$n! = \Gamma(n + 1) \quad , \quad (\text{A.6-4})$$

and the value of the gamma function in the range $0 < n < 1$ can be found in most tables of evaluations.

It can be seen from chapter seven that the values adopted for the parameters contained in the functions that were chosen rather than fitted are ;

$$b=4, \quad \beta=1, \quad \delta=0.05.$$

The high value of b ensures that the values of \dot{X}_M for the low mass range are not unreasonably large, and the rapid increase in \dot{X}_M for high masses is balanced by the decrease in Φ_M with $\beta=1$ over the same range. The small non-integer value for δ ensures that the function $f_T(M)$ only exceeds unity in a range of M where Φ_M is small.

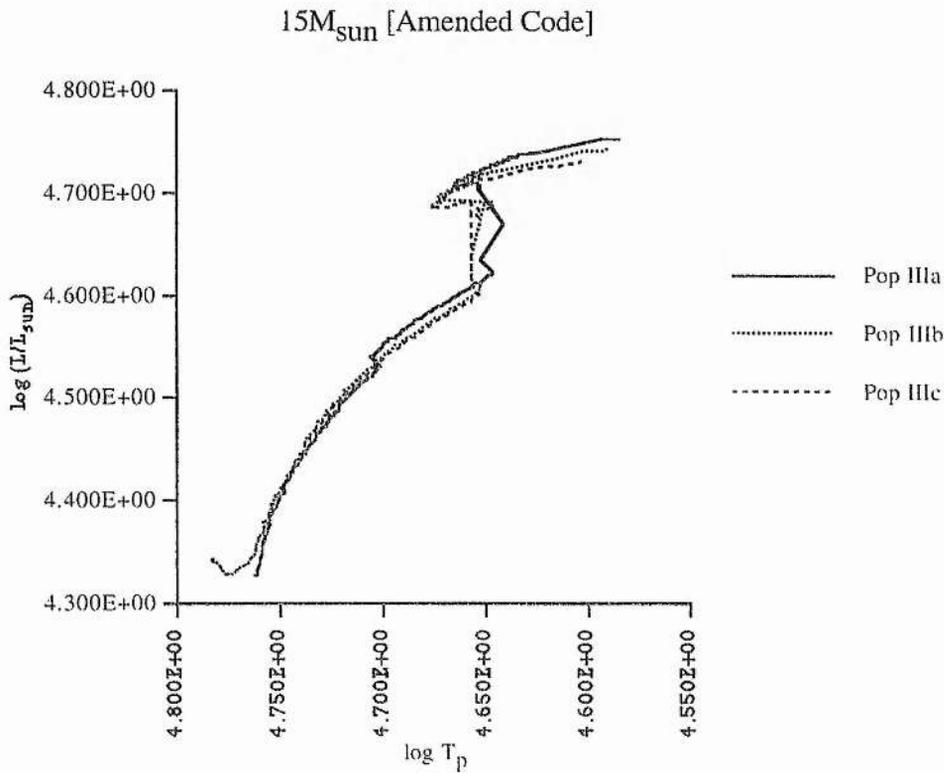
Appendix B

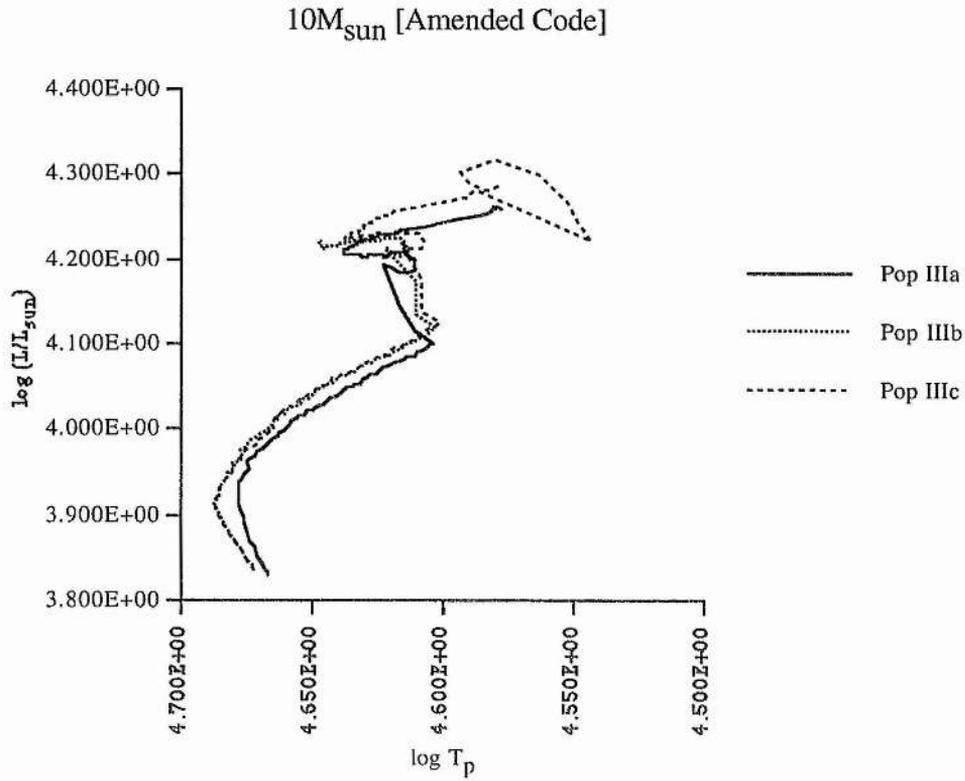
Composite Diagrams of Evolution for the Amended Code

This second appendix contains the diagrams of data produced by the amended code which were not presented in chapter six for reasons of clarity. Broadly, they illustrate the same trends as those given in chapter six from data produced by the initial code. Any differences between the initial and amended codes have either already been remarked upon, or will be remarked upon in this appendix.

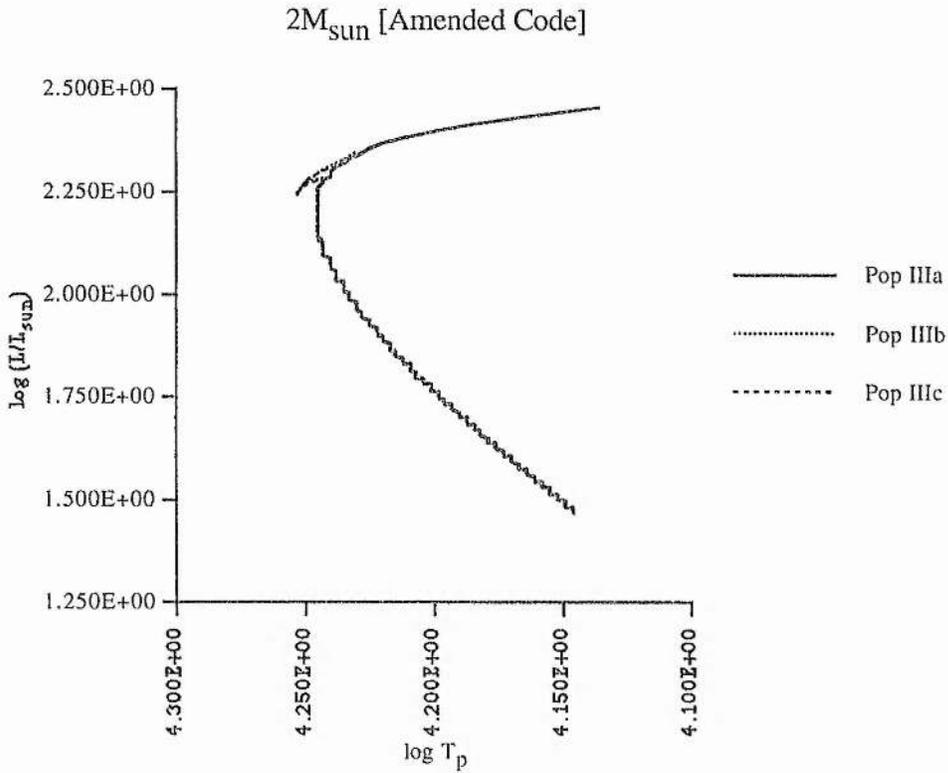
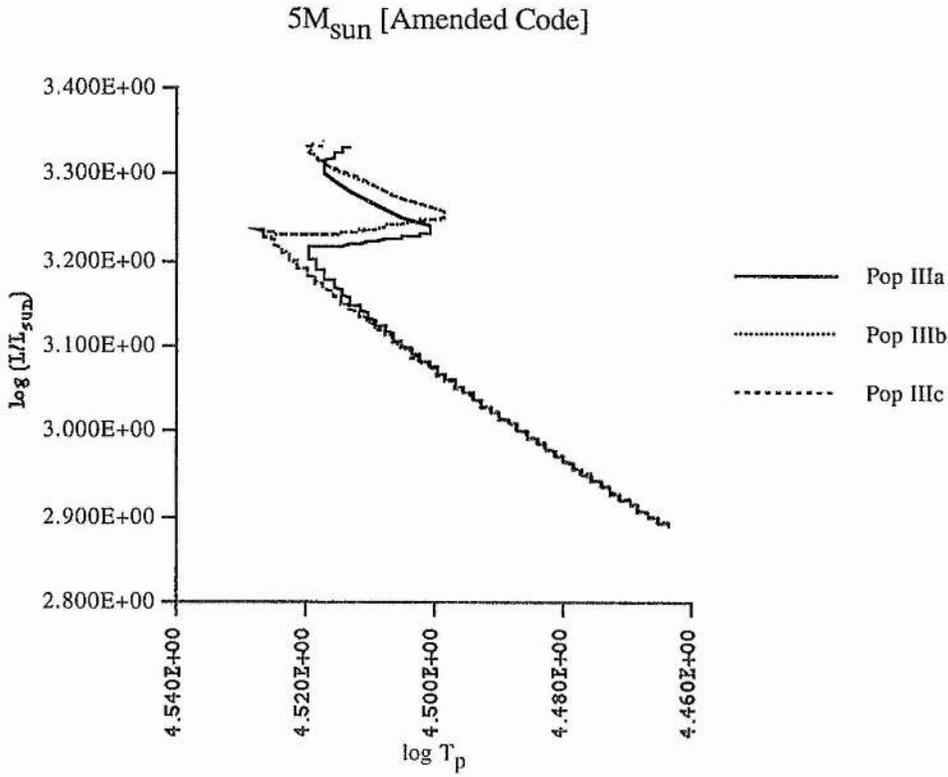
B.1 H-R Diagrams for the Amended Code

Firstly we can show the HR diagrams for those models evolved using the amended code. As for the diagrams presented in chapter six, each shows evolutionary tracks for the three different initial compositions (populations IIIa, IIIb and IIIc) at a specific mass.





These more massive models show differences with metallicity that are broadly similar to those shown by the set of models generated by the initial code. These differences, as before, are especially pronounced after the stage where the core hydrogen is nearing exhaustion.

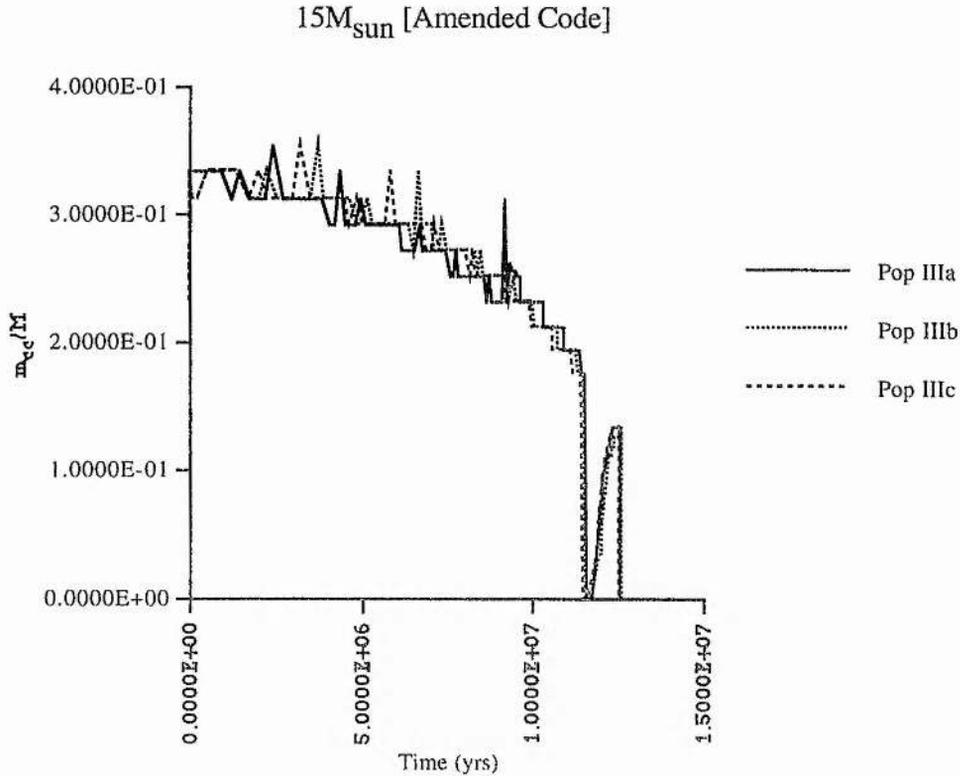


We can see the same trends in the less massive models evolved using the amended code as we could for those evolved using the initial code. That is, that any differences in the evolutionary tracks are not noticeable until the later stages of evolution. In the case of the $2M_{\text{sun}}$

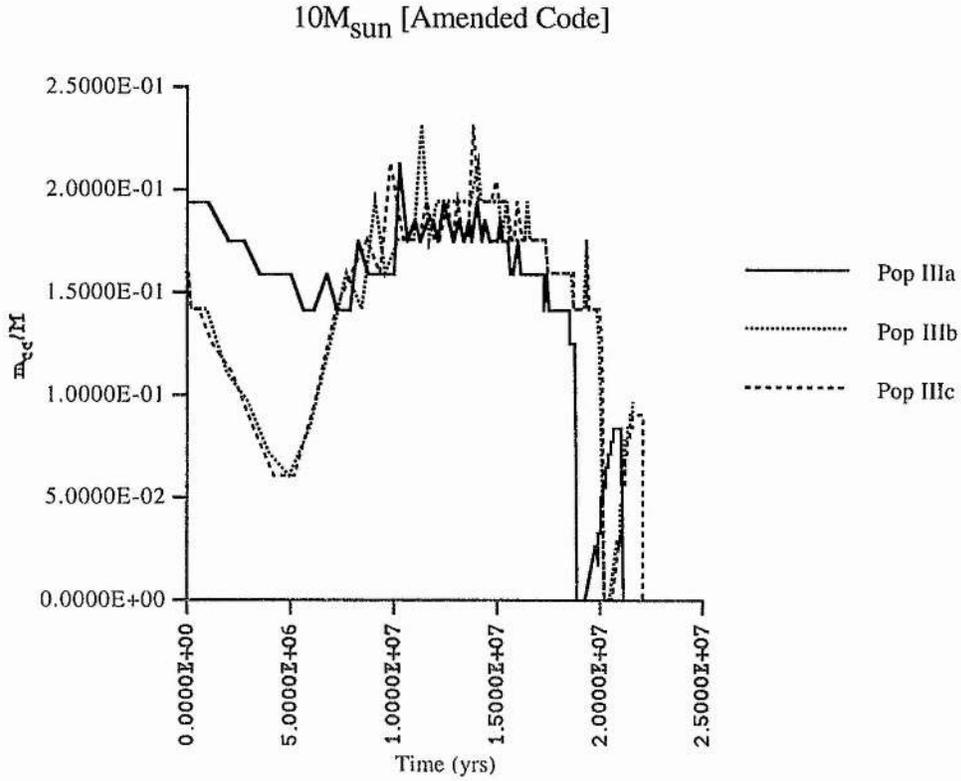
model of population IIIa, we can see that it does not follow the same evolutionary track at the point at which core hydrogen is exhausted or nearing exhaustion. This could be due to its failure to form a convective core at this stage of evolution.

B.2 Convective Core Data for the Amended Code

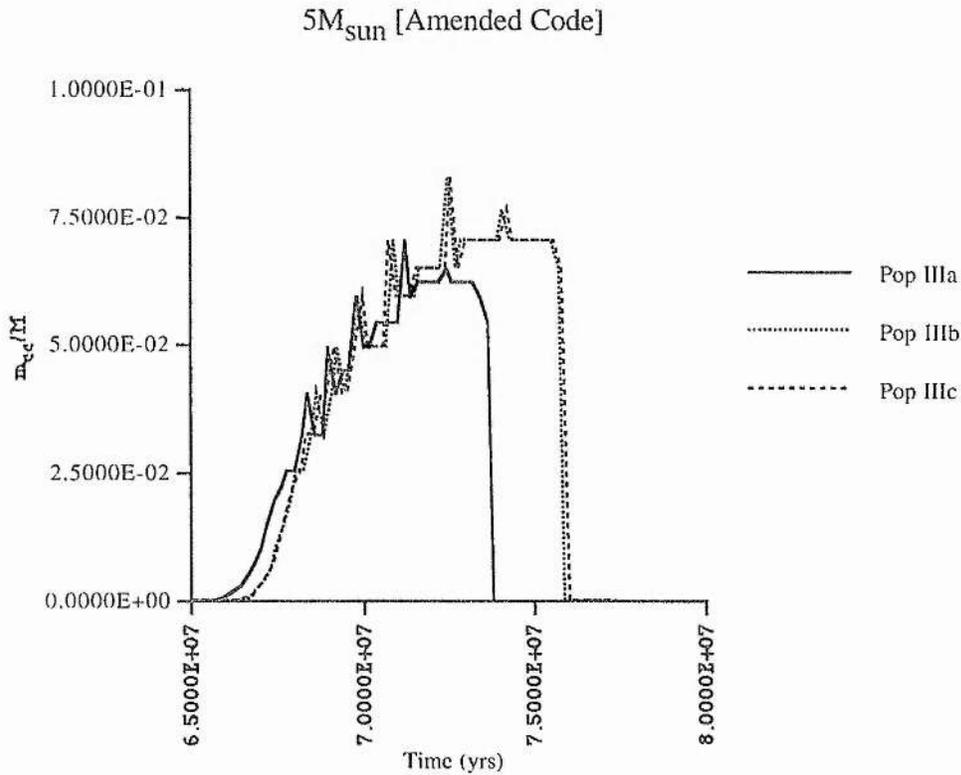
The next set of diagrams give data for the convective cores of models generated using the amended code.

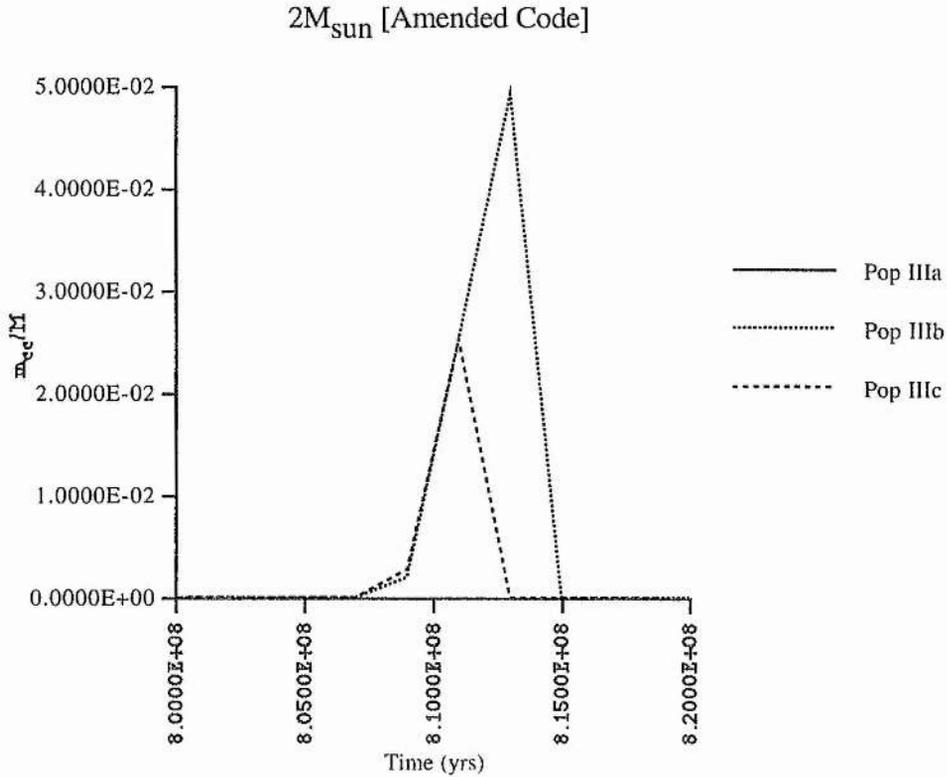


We can see in the $15M_{\text{sun}}$ diagram for the mass of the convective core that the differences between the three models in the times at which the convective core appears and disappears are very much less than those of the models evolved using the initial code.



The $10M_{\text{sun}}$ diagram shows that there is less difference in the behaviour of the convective core between the models of population IIIb and IIIc than there is when they are evolved using the initial code.

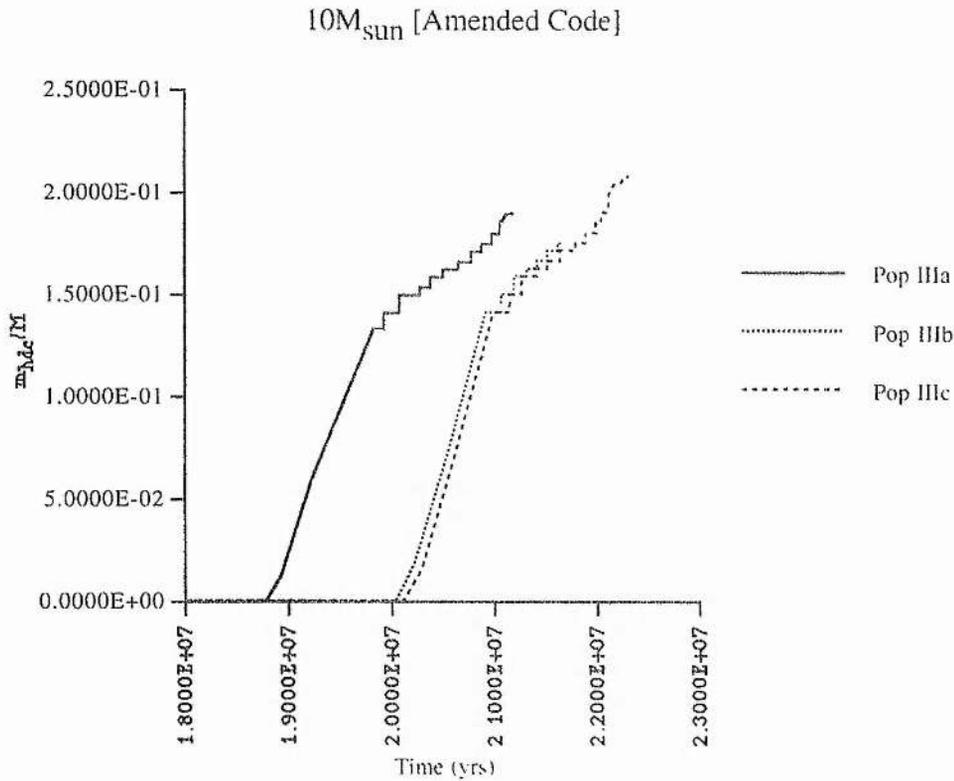
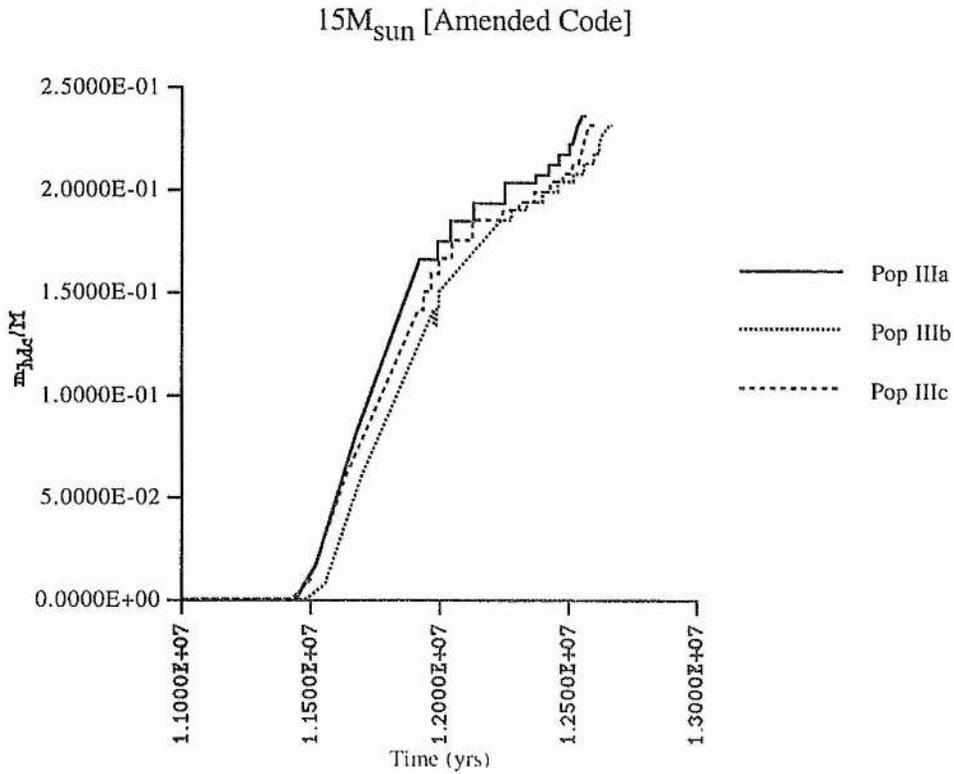




This last diagram shows that, as for the initial code, the $2M_{\text{sun}}$ model of population IIIa did not experience a second period of core convection. However, unlike the same model evolved under the initial code, a hydrogen-depleted core was evolved. This is shown in the next section. This difference aside, we can see that these diagrams demonstrate much the same relation between populations IIIa, IIIb and IIIc as was demonstrated by those presented in chapter six.

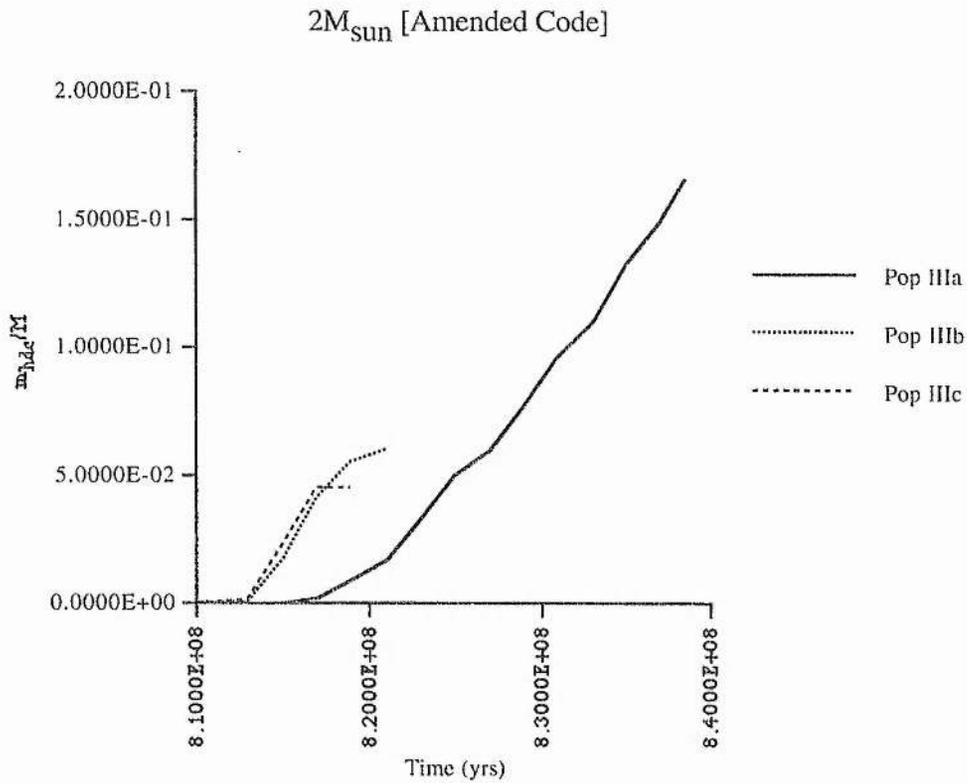
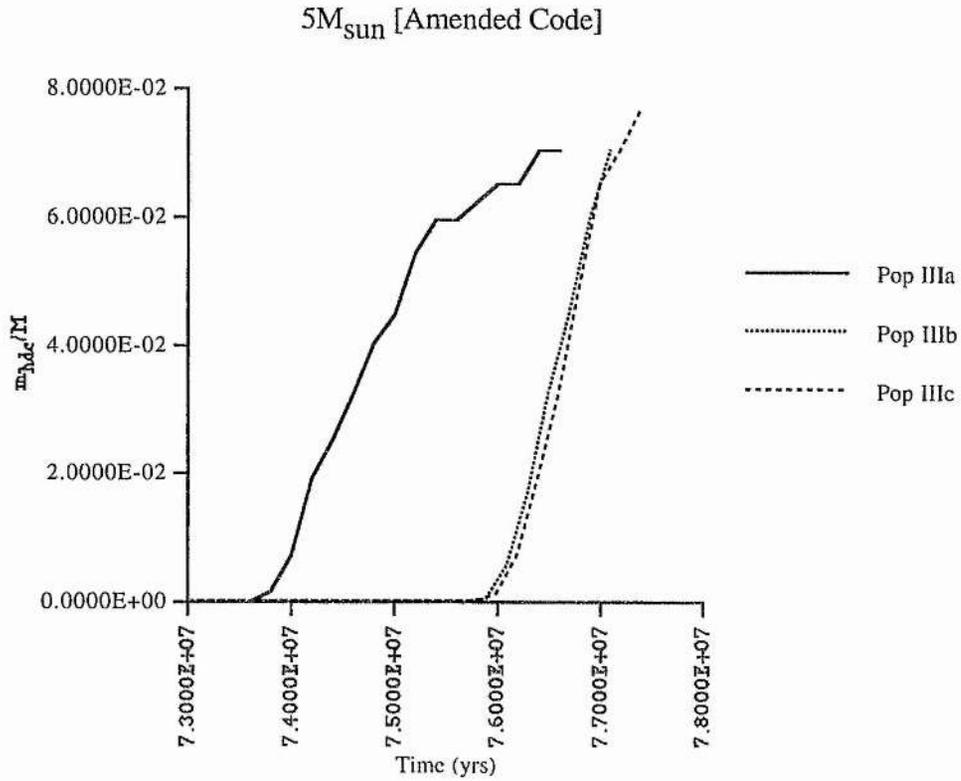
B.3 Hydrogen-Depleted Core Data for the Amended Code

The next set of diagrams present data for the hydrogen-depleted cores of models evolved using the amended code.



Unlike the $10M_{\text{sun}}$ models evolved using the initial code, the order in which the three models evolve their hydrogen-depleted cores is not very different from the behaviour of the models at other masses. The model of

population IIIc does not evolve its hydrogen-depleted core earlier than the population IIIa model in this case.



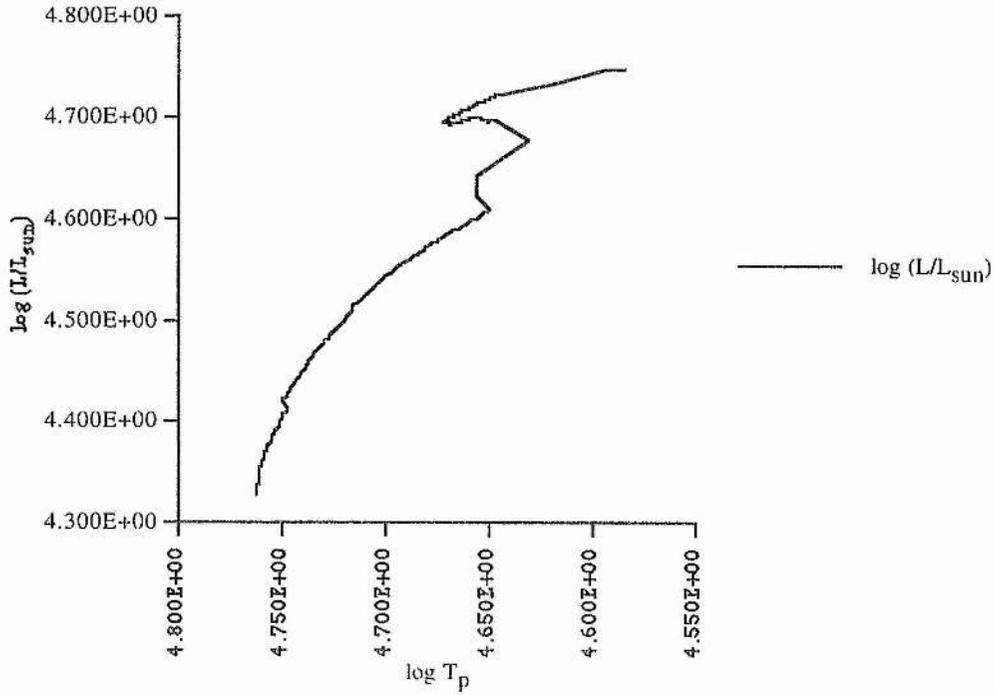
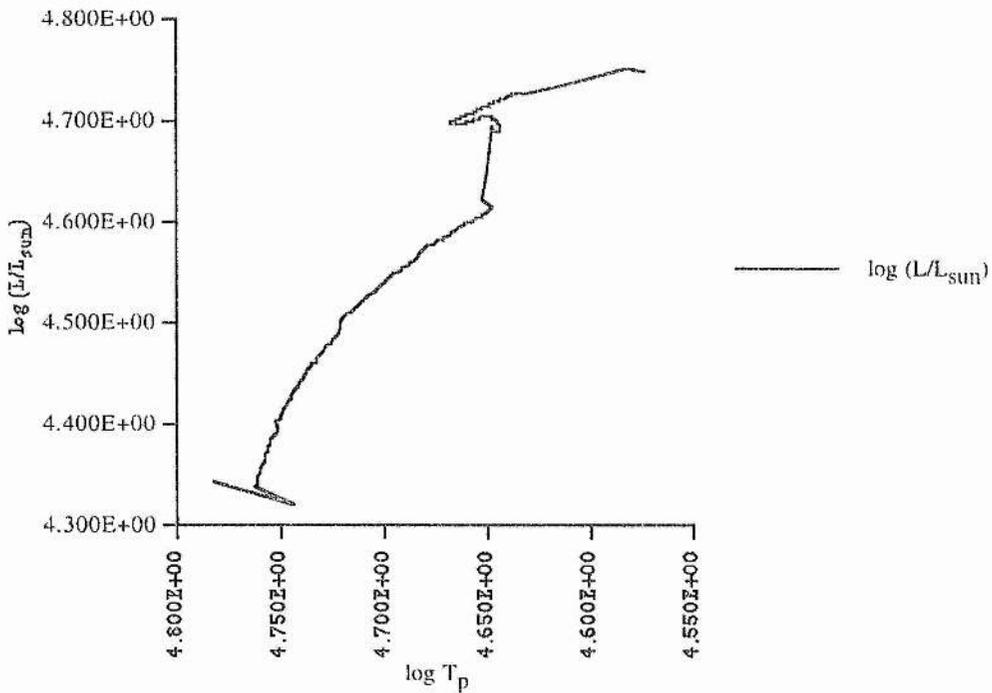
We can see that for these models, there is a more pronounced difference in the time of core formation between population IIIa and the other two

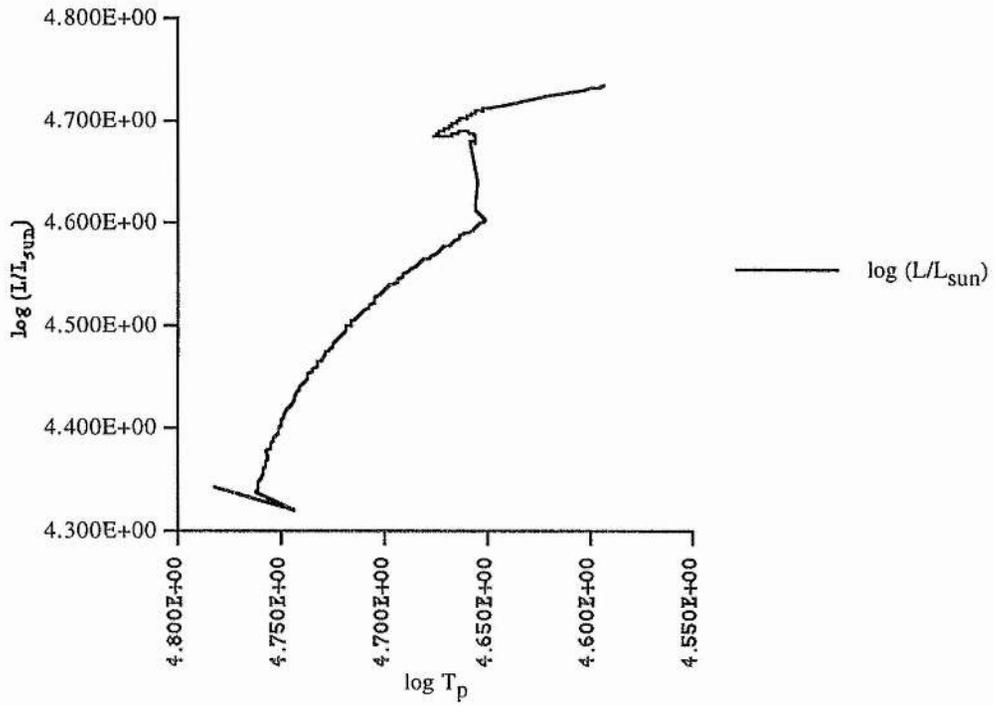
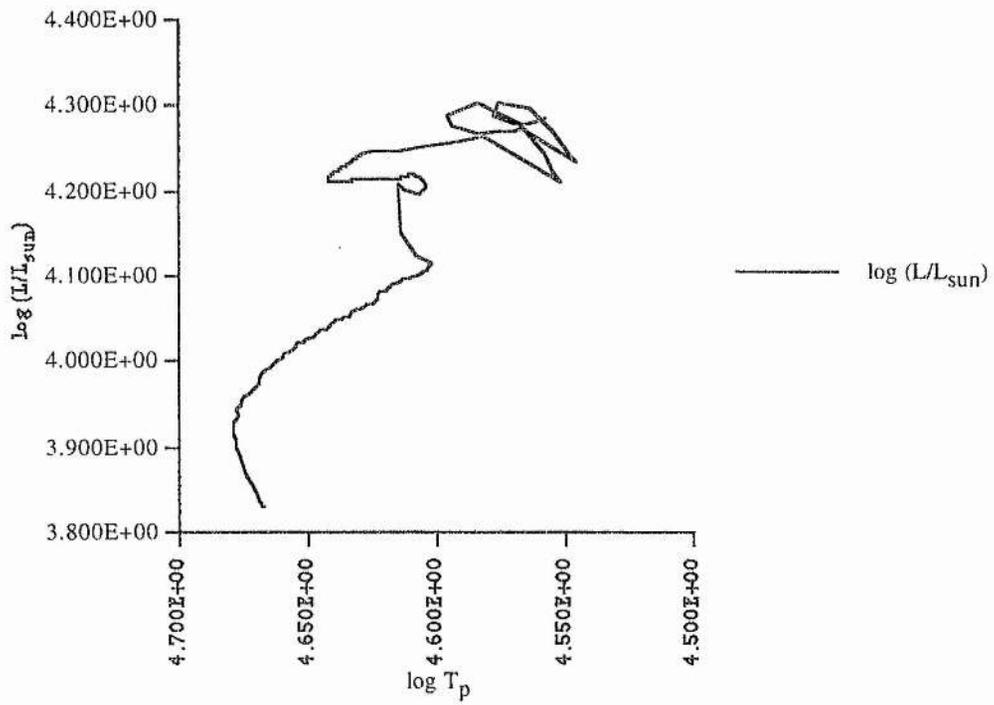
populations. The $2M_{\text{sun}}$ models of population IIIa form hydrogen-depleted cores more slowly than models of population IIIb or IIIc, a mirror of the behaviour of the more massive models. The failure of the same model generated by the initial code to form a hydrogen-depleted zone at all could be seen to be a continuation of this trend, bearing in mind the explanation given for this behaviour in chapter six.

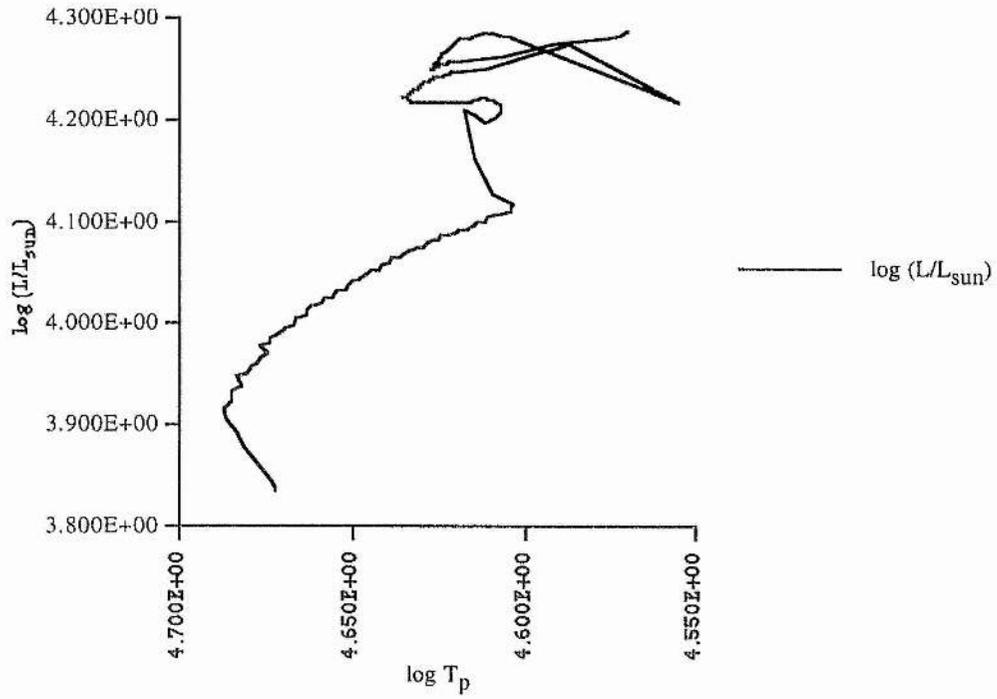
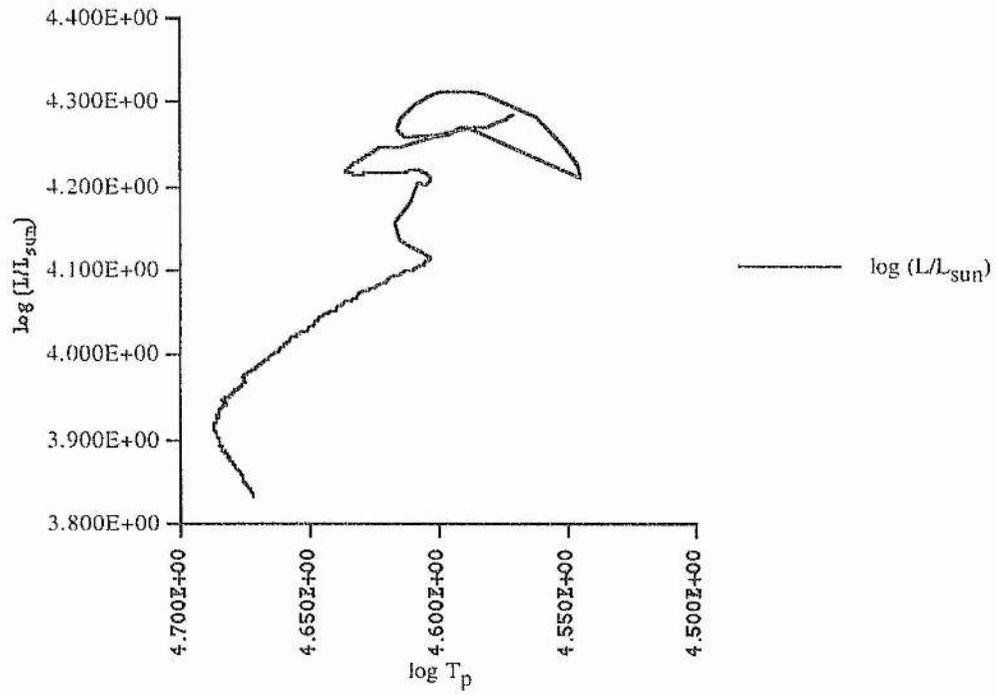
Appendix C

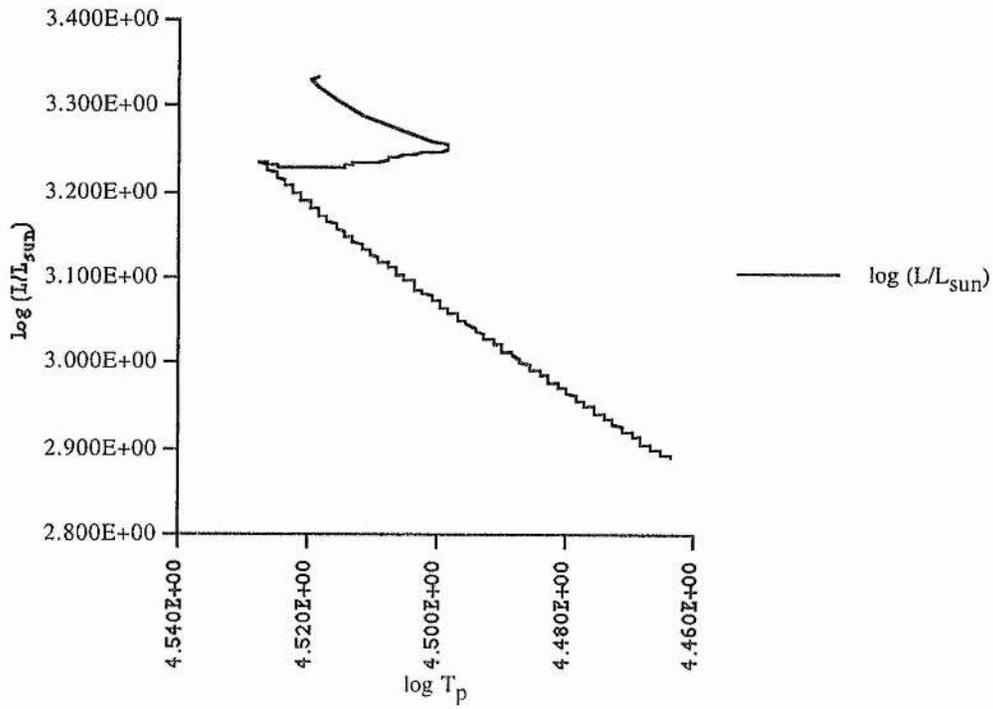
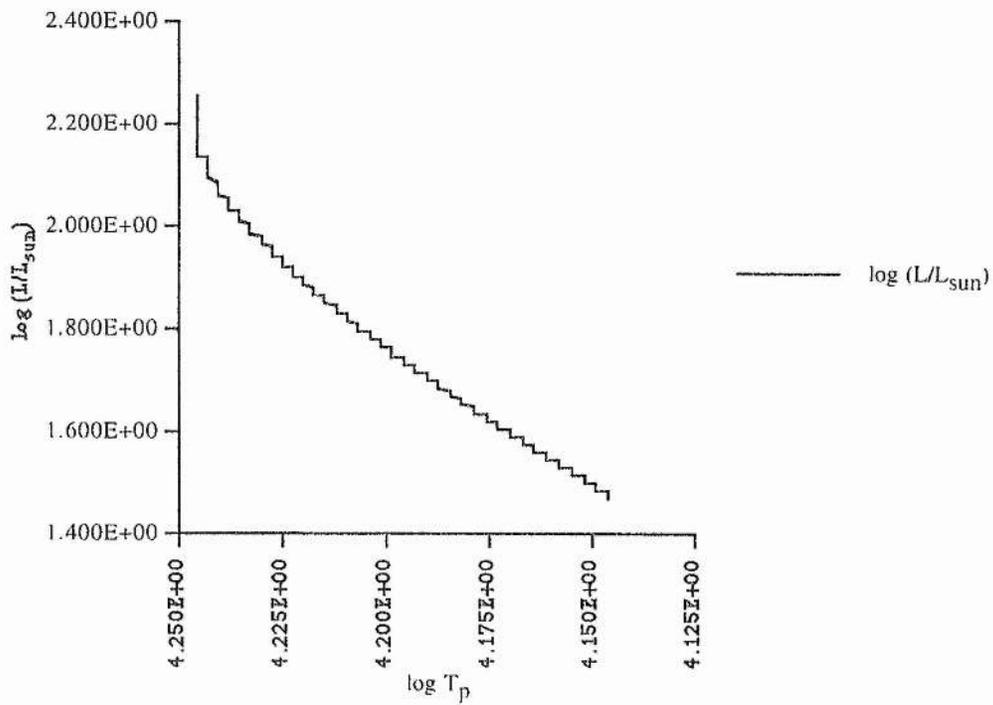
Diagrams of Evolution for All Models

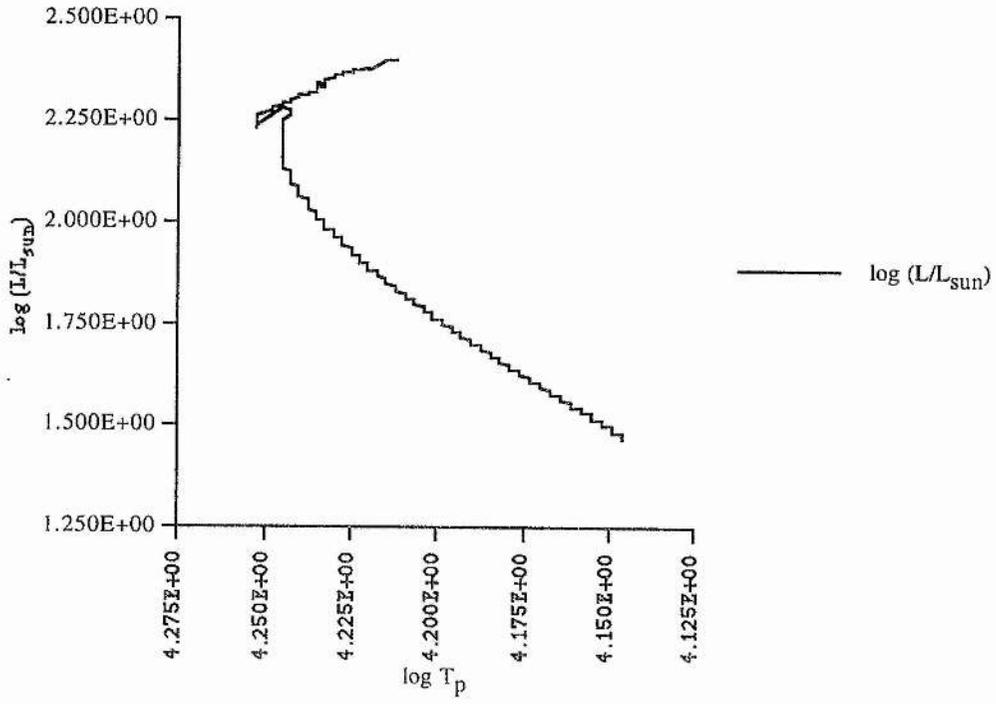
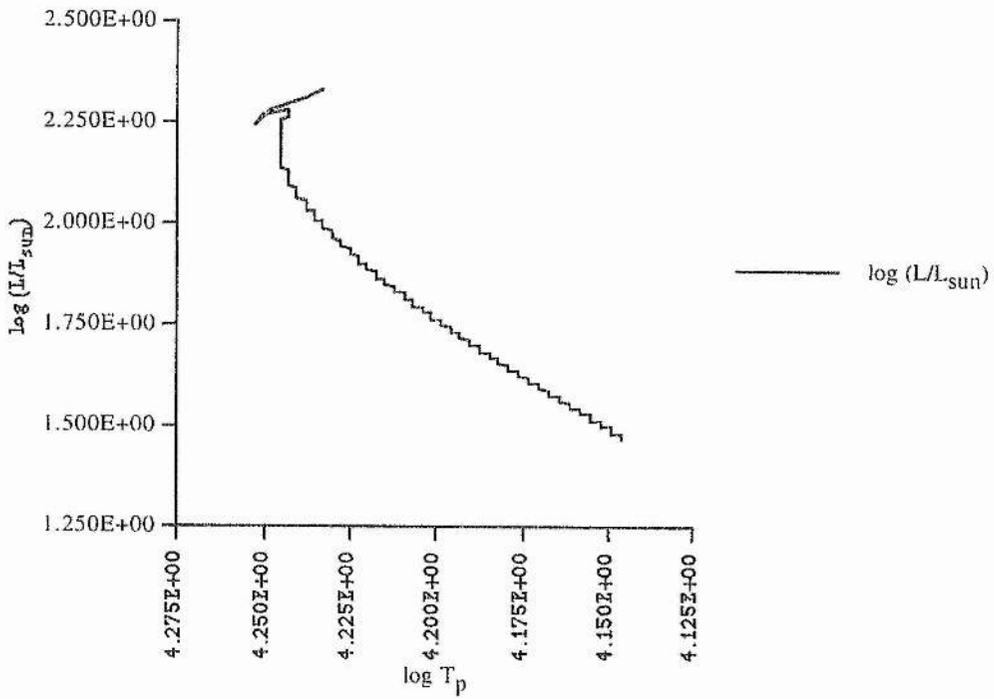
C.1 H-R Diagrams, Initial Code

C.1.1 $15M_{\text{sun}}$ Pop IIIa, $15M_{\text{sun}}$ Pop IIIb, $15M_{\text{sun}}$ 

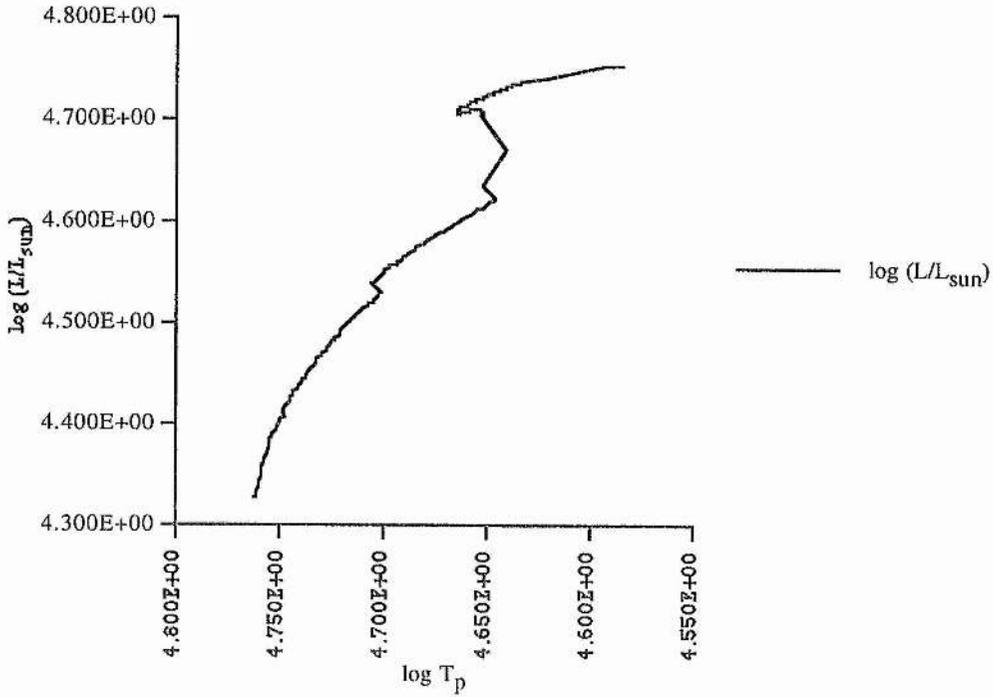
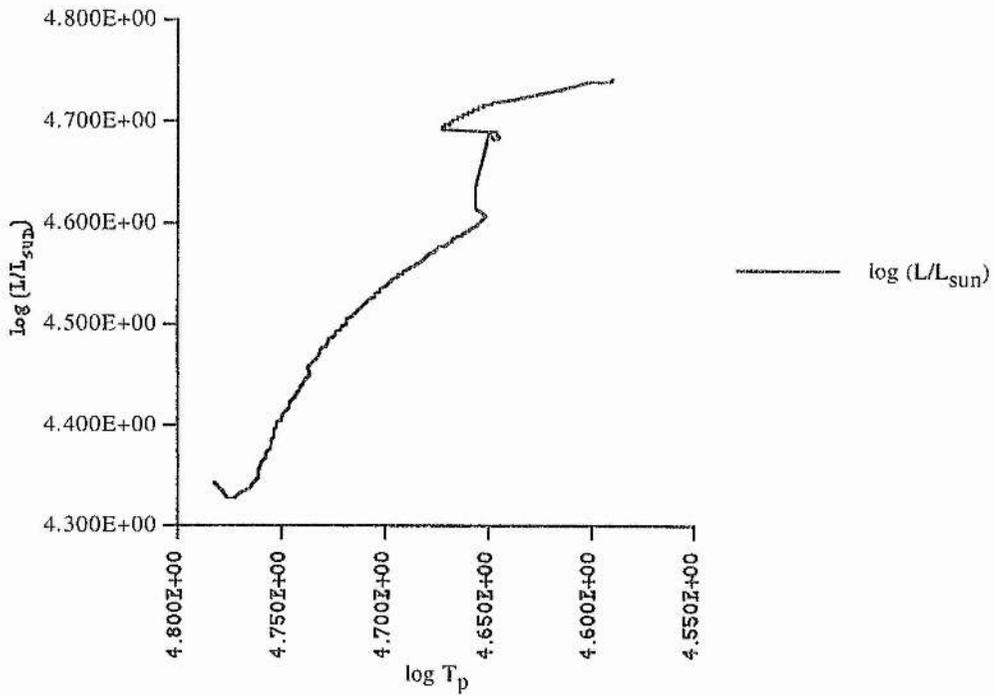
Pop IIIc, $15M_{\text{sun}}$ C.1.2 $10M_{\text{sun}}$ Pop IIIa, $10M_{\text{sun}}$ 

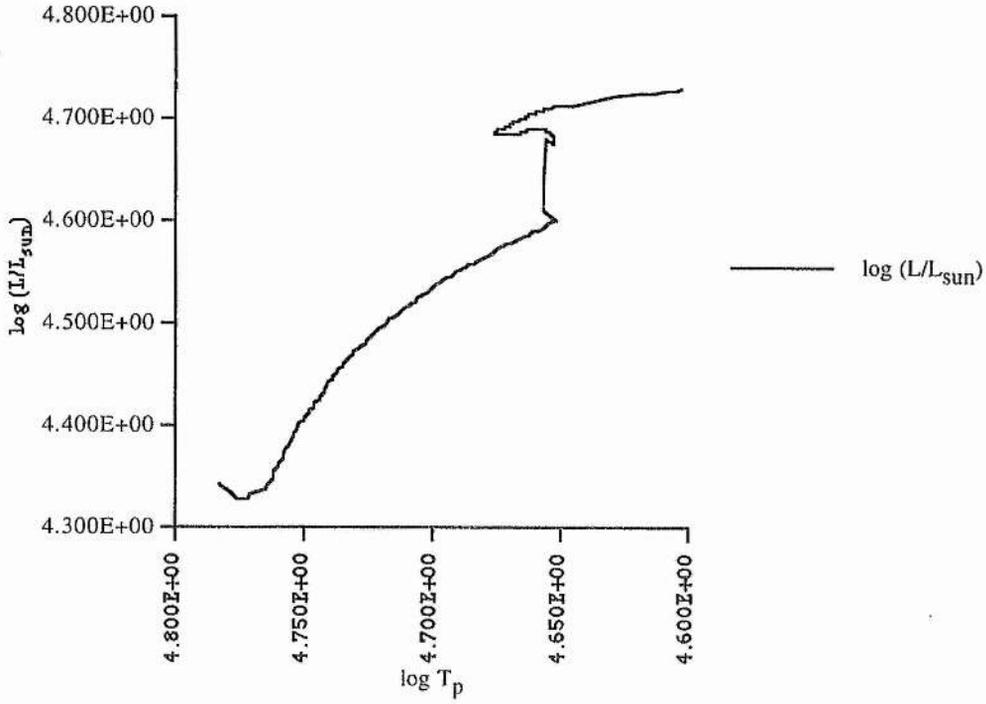
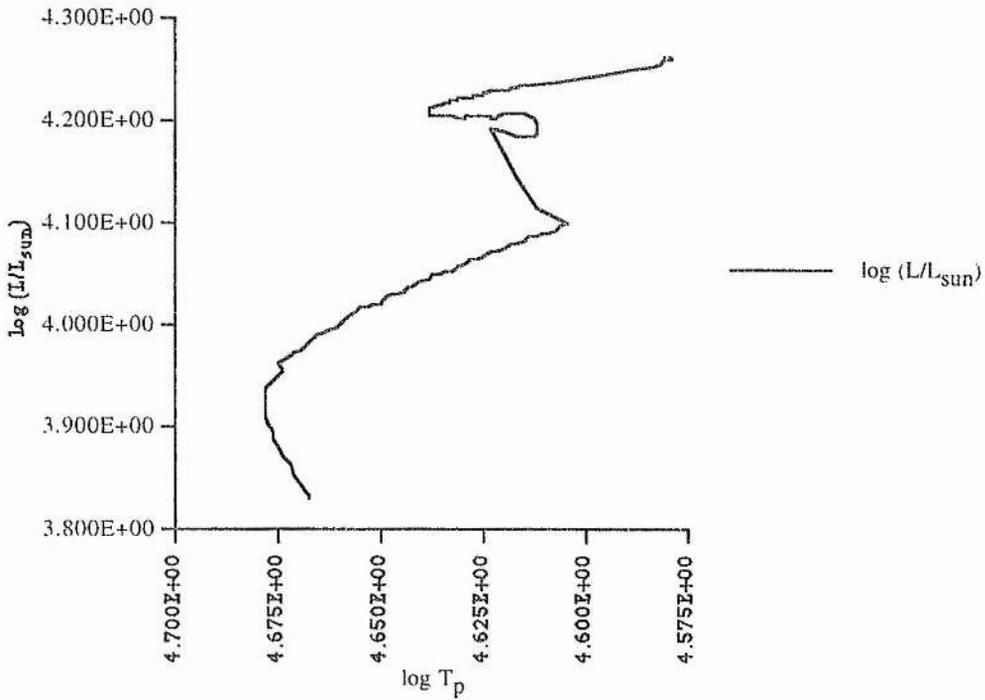
Pop IIIb, $10M_{\text{sun}}$ Pop IIIc, $10M_{\text{sun}}$ 

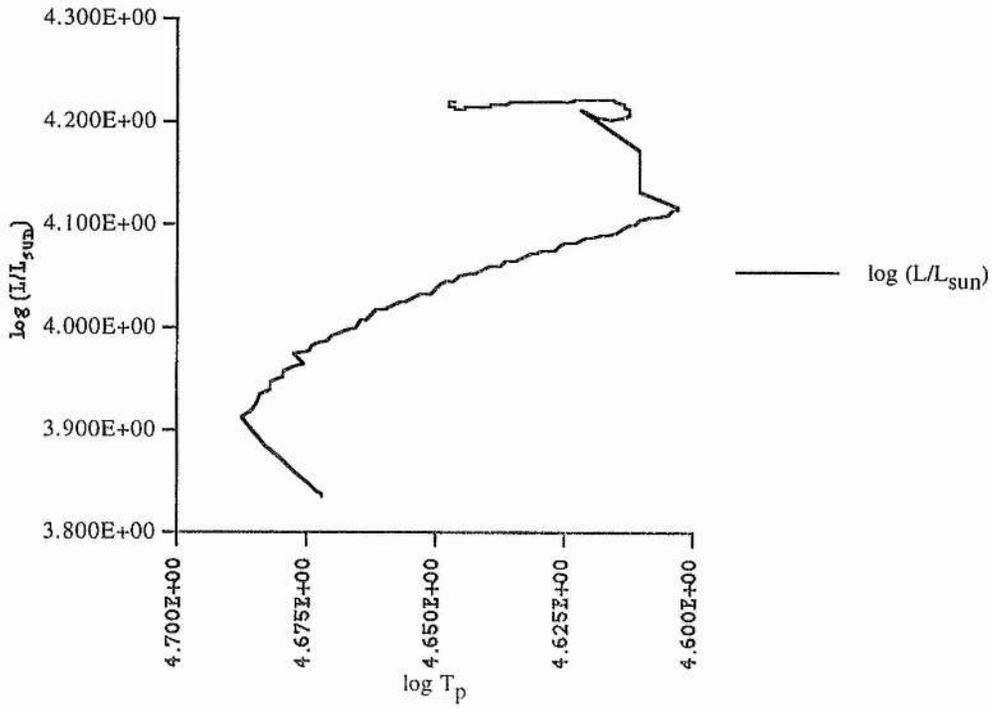
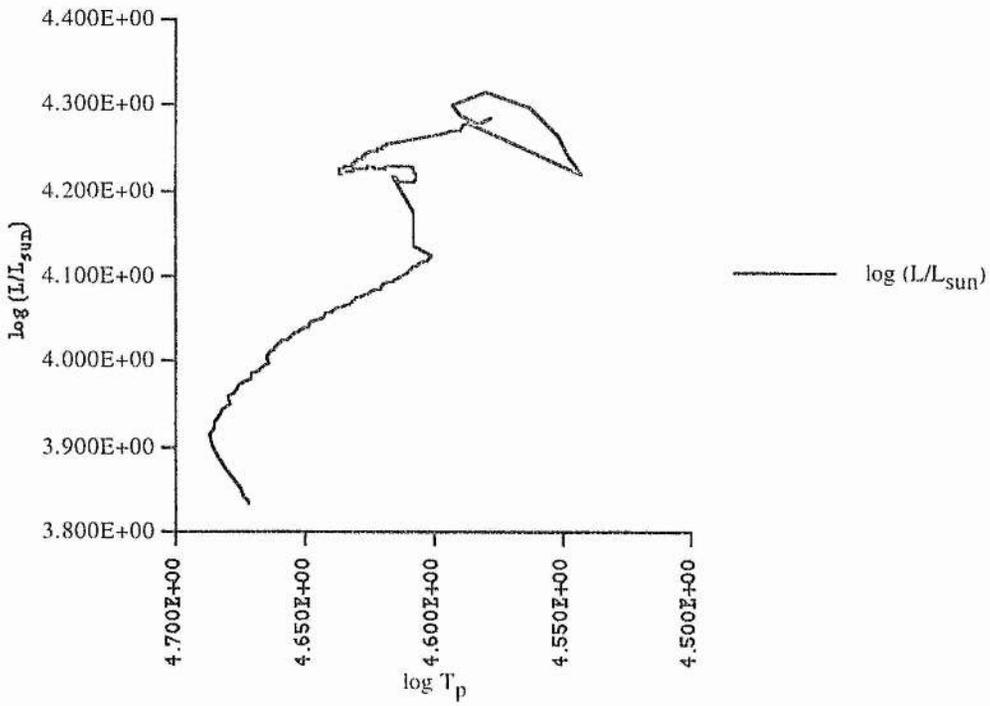
Pop IIIc, $5M_{\text{sun}}$ C.1.4 $2M_{\text{sun}}$ Pop IIIa, $2M_{\text{sun}}$ 

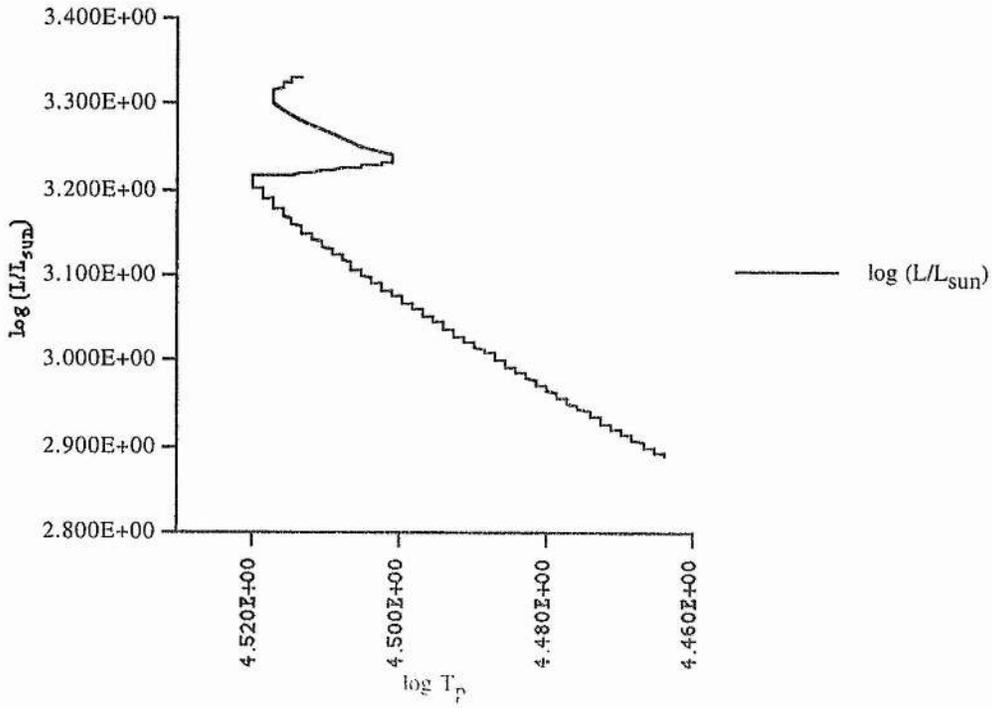
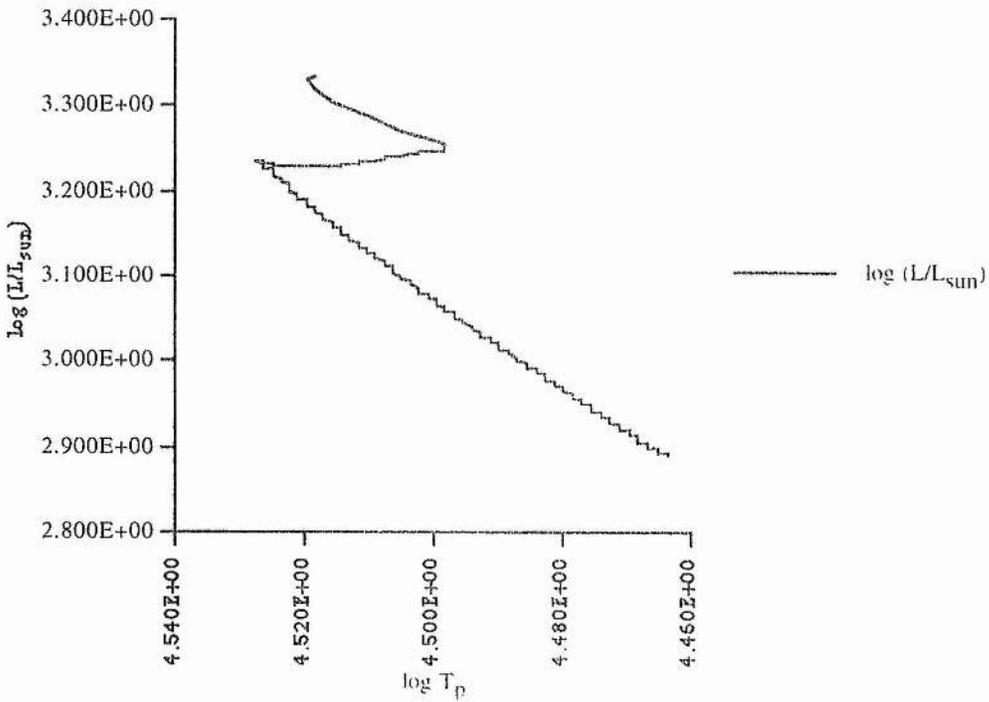
Pop IIIb, $2M_{\text{sun}}$ Pop IIIc, $2M_{\text{sun}}$ 

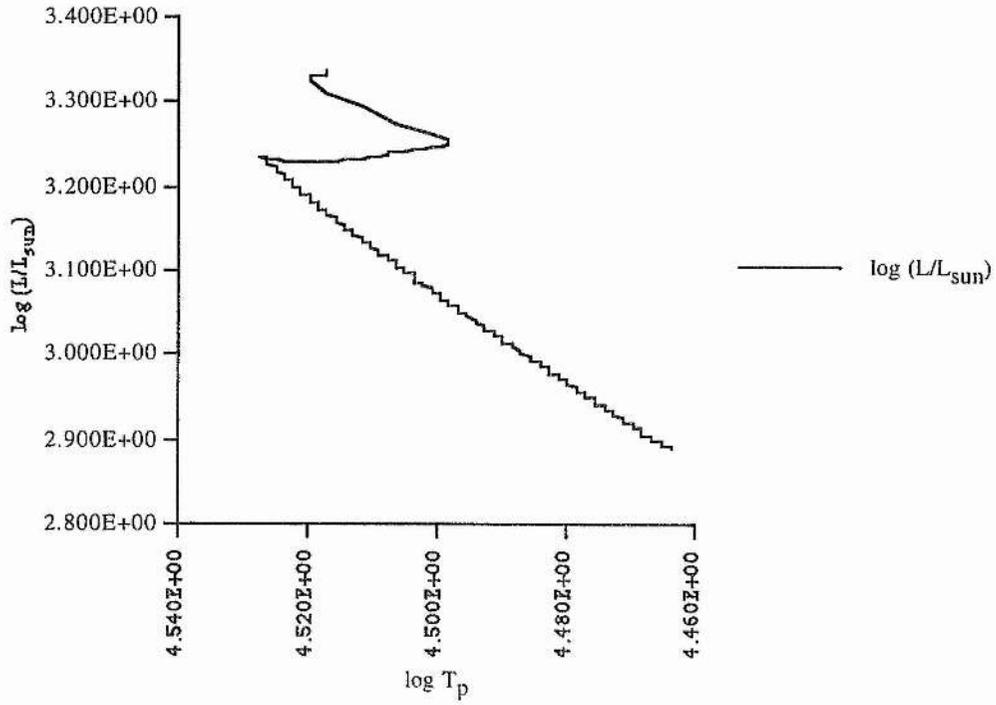
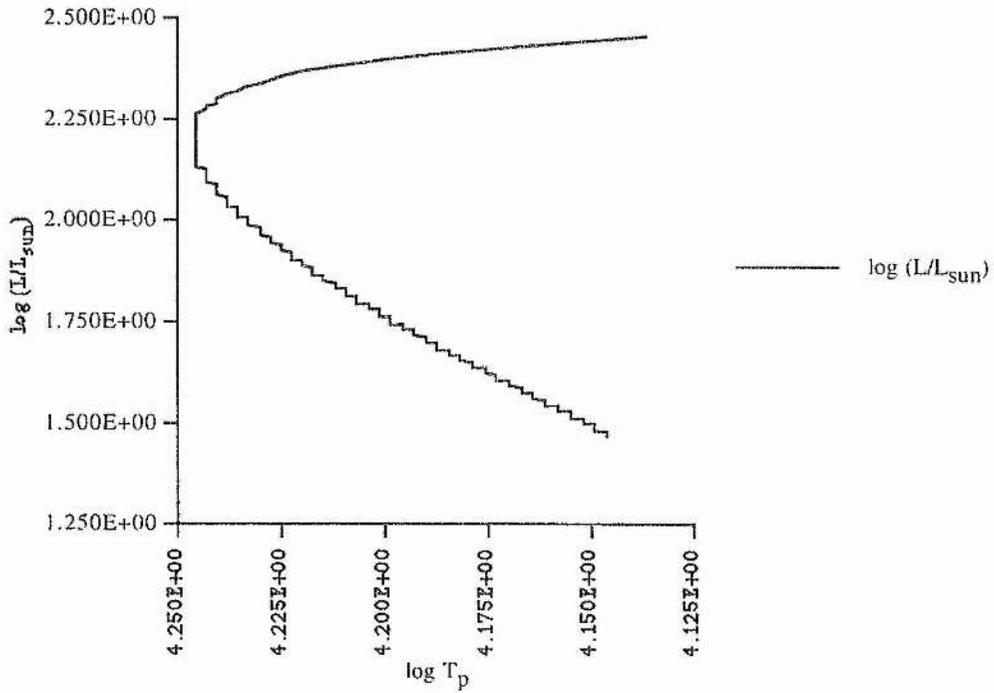
C.2 H-R Diagrams, Amended Code

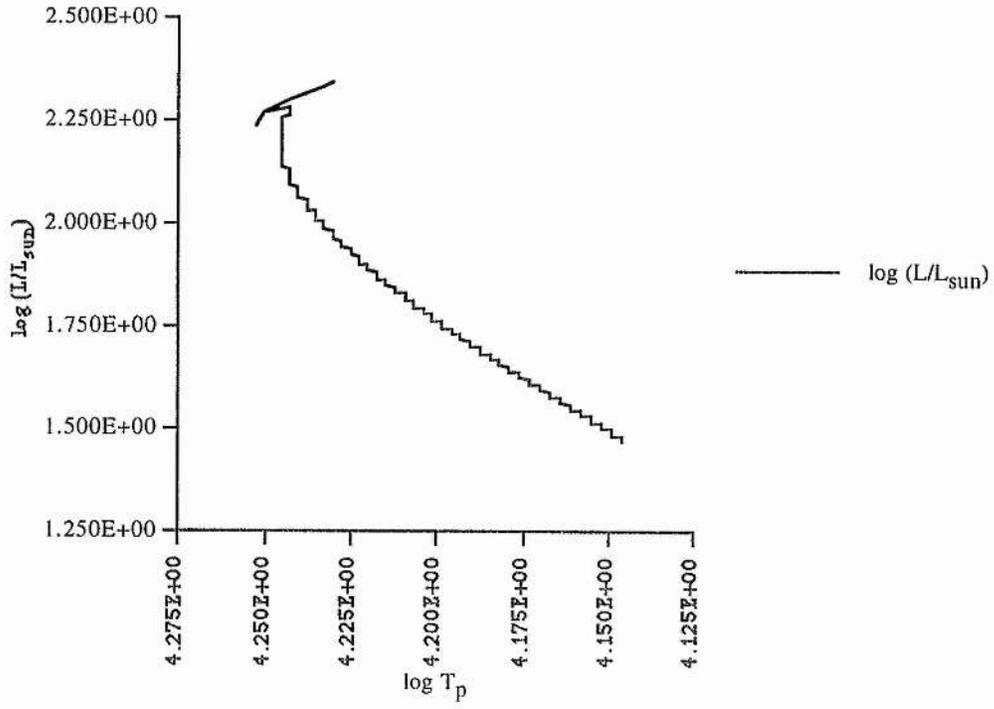
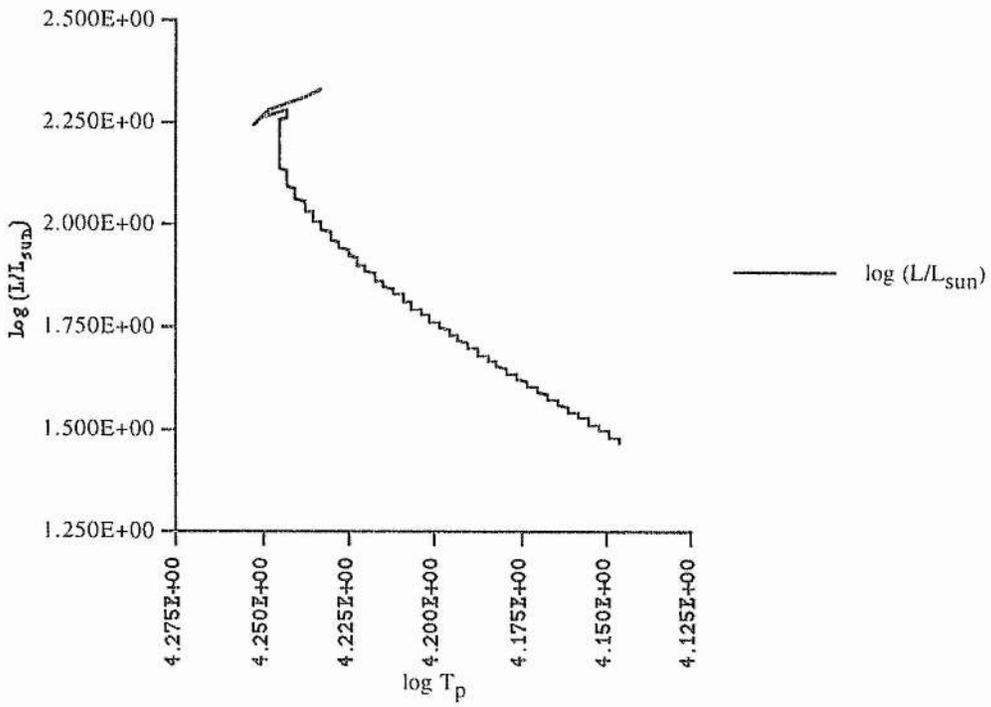
C.2.1 $15M_{\text{sun}}$ Pop IIIa, $15M_{\text{sun}}$ [Amended Code]Pop IIIb, $15M_{\text{sun}}$ [Amended Code]

Pop IIIc, $15M_{\text{sun}}$ [Amended Code]C.2.2 $10M_{\text{sun}}$ Pop IIIa, $10M_{\text{sun}}$ [Amended Code]

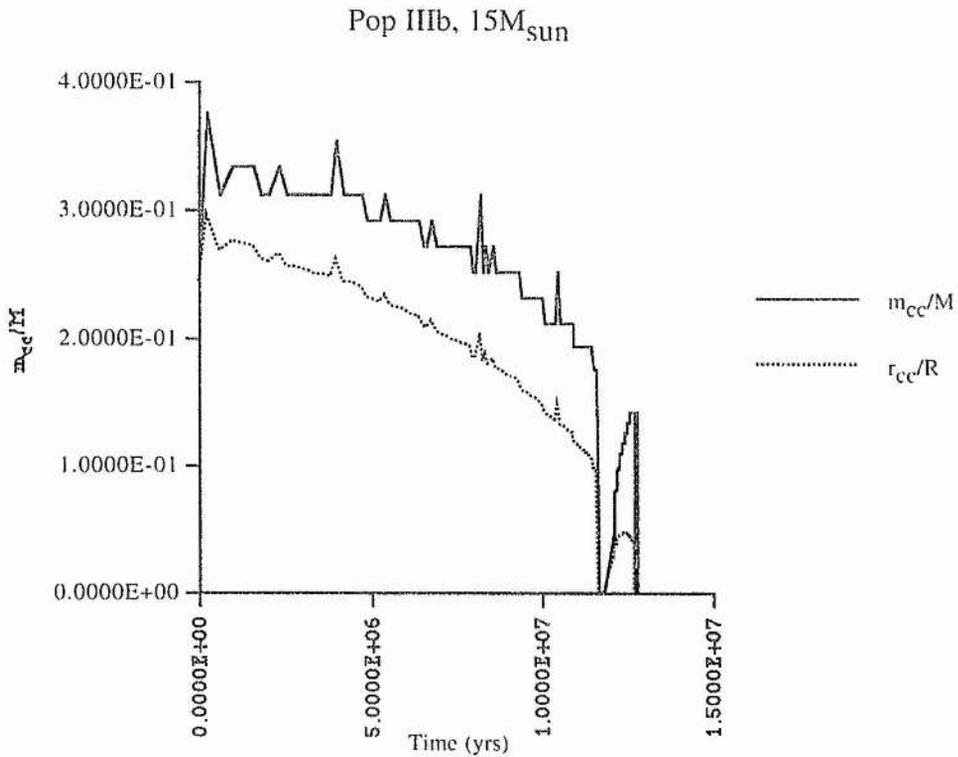
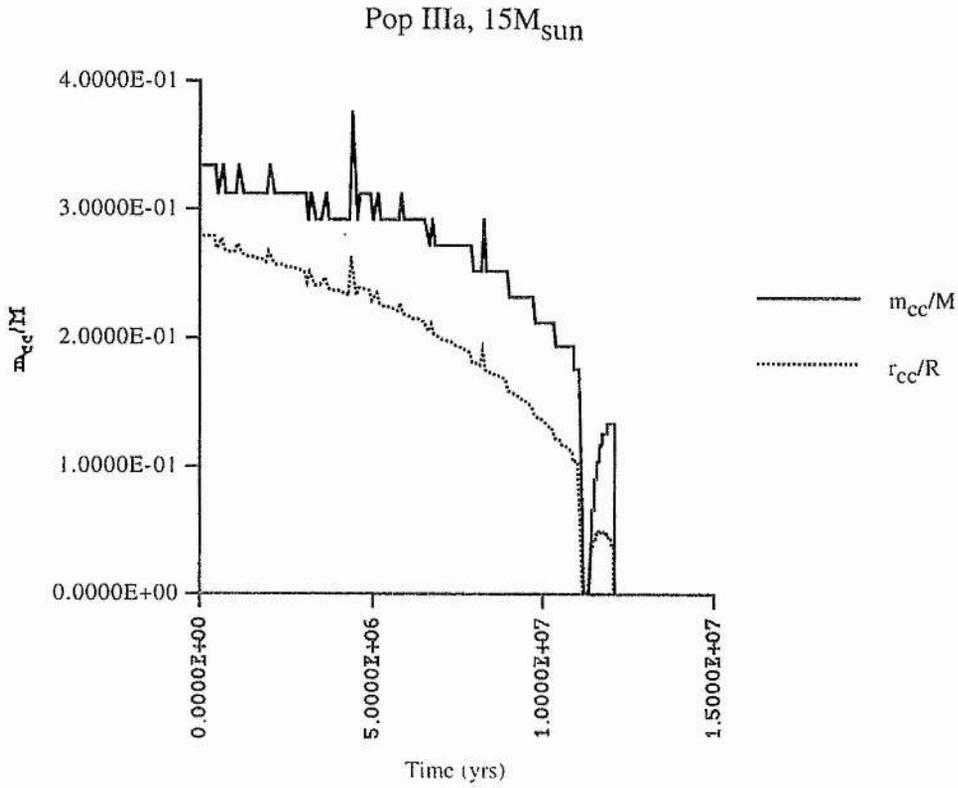
Pop IIIb, $10M_{\text{sun}}$ [Amended Code]Pop IIIc, $10M_{\text{sun}}$ [Amended Code]

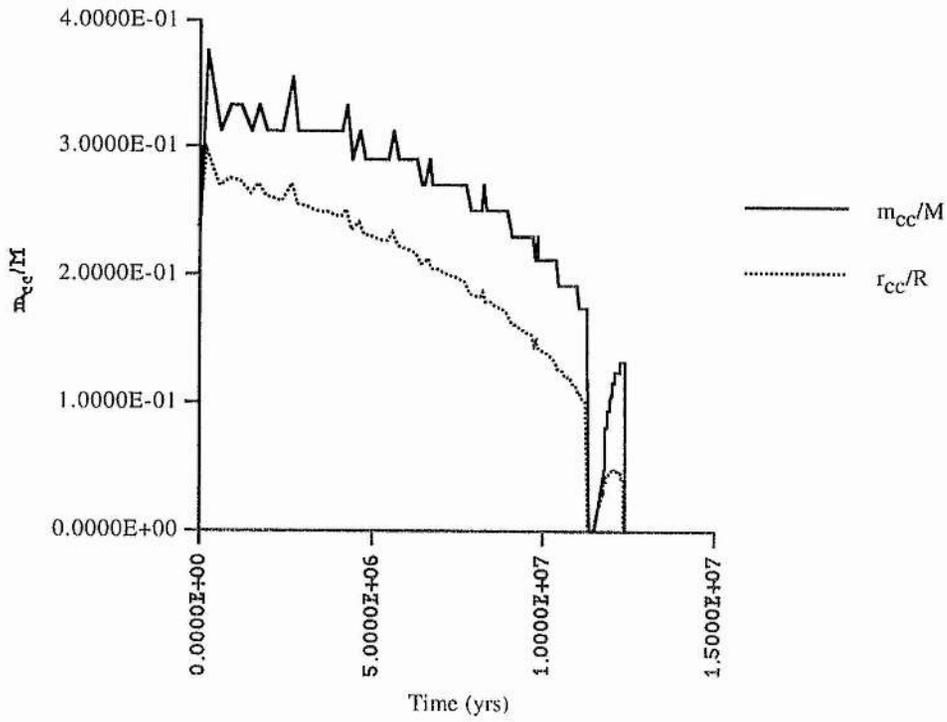
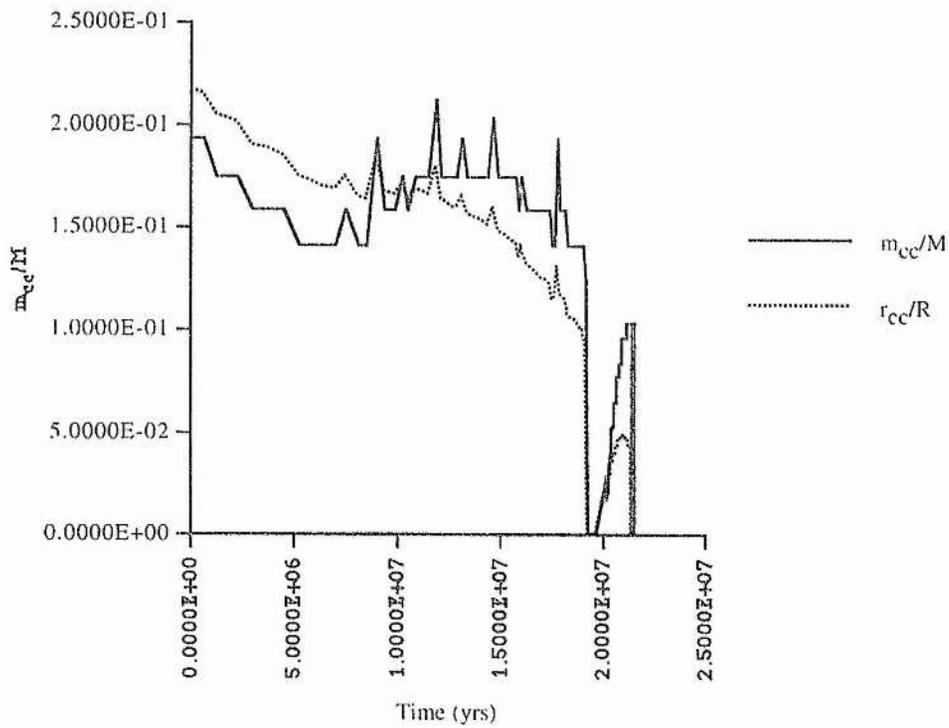
C.2.3 $5M_{\text{sun}}$ Pop IIIa, $5M_{\text{sun}}$ [Amended Code]Pop IIIb, $5M_{\text{sun}}$ [Amended Code]

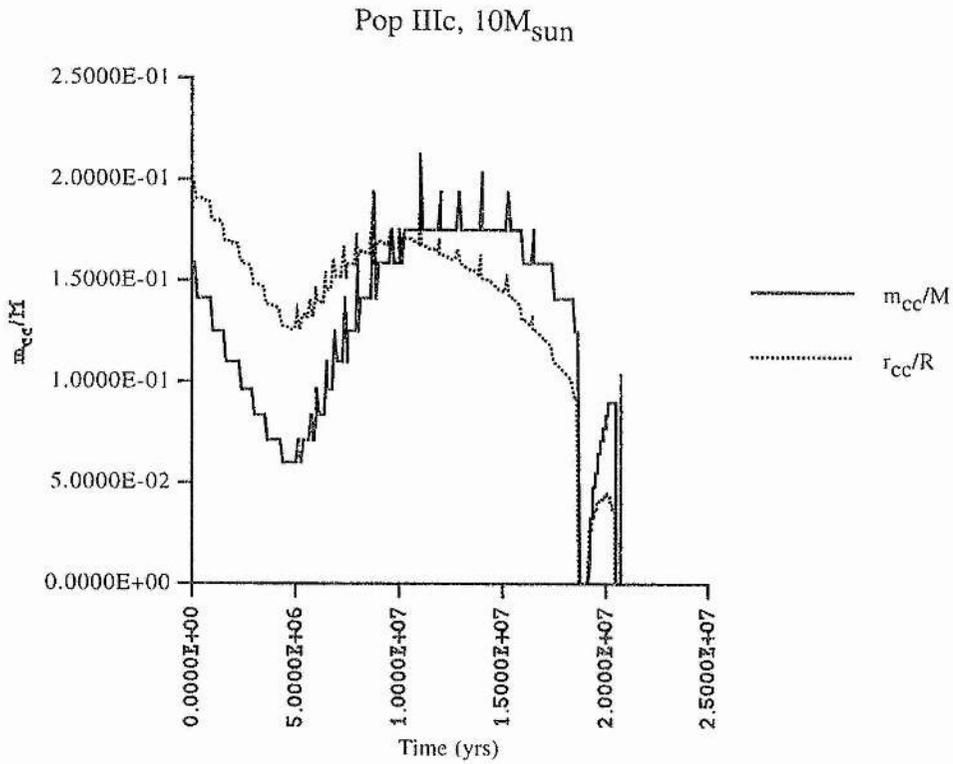
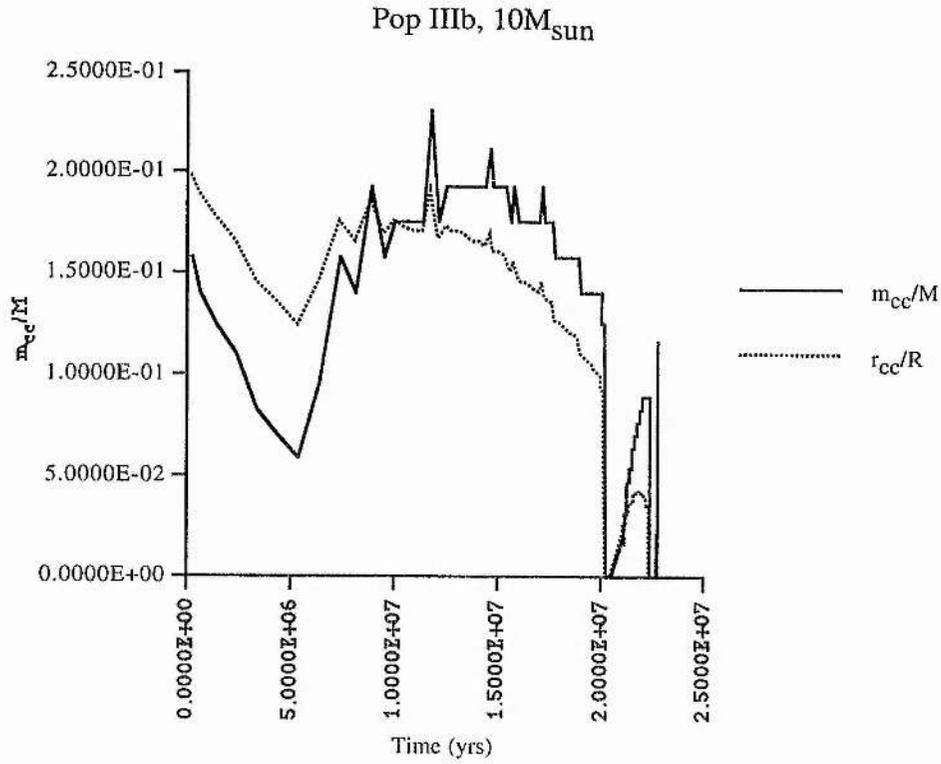
Pop IIIc, $5M_{\text{sun}}$ [Amended Code]C.2.4 $2M_{\text{sun}}$ Pop IIIa, $2M_{\text{sun}}$ [Amended Code]

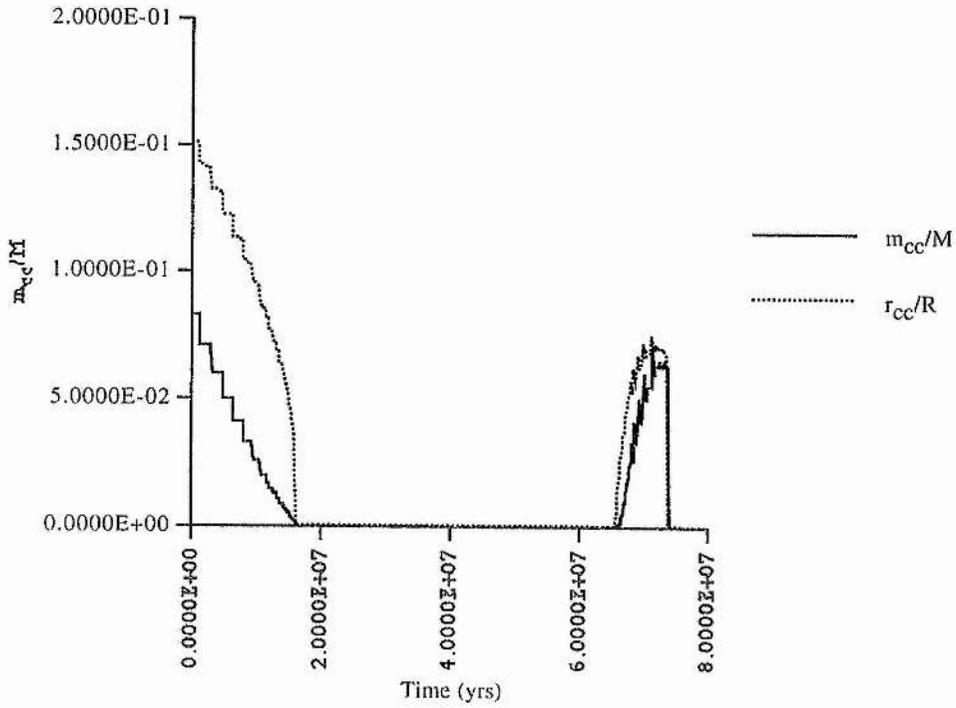
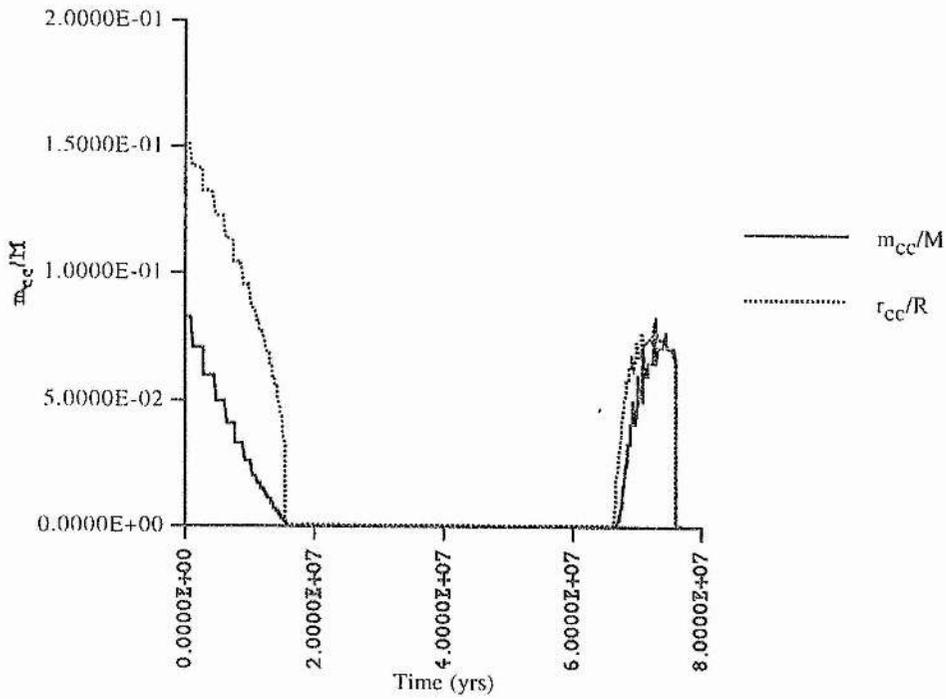
Pop IIIb, $2M_{\text{sun}}$ [Amended Code]Pop IIIc, $2M_{\text{sun}}$ [Amended Code]

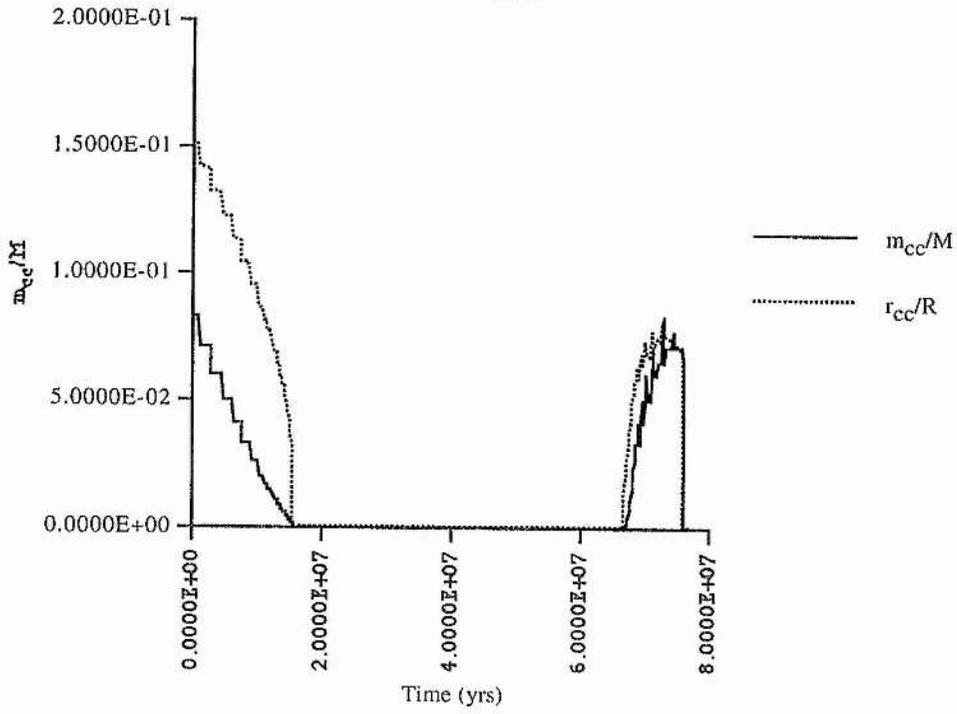
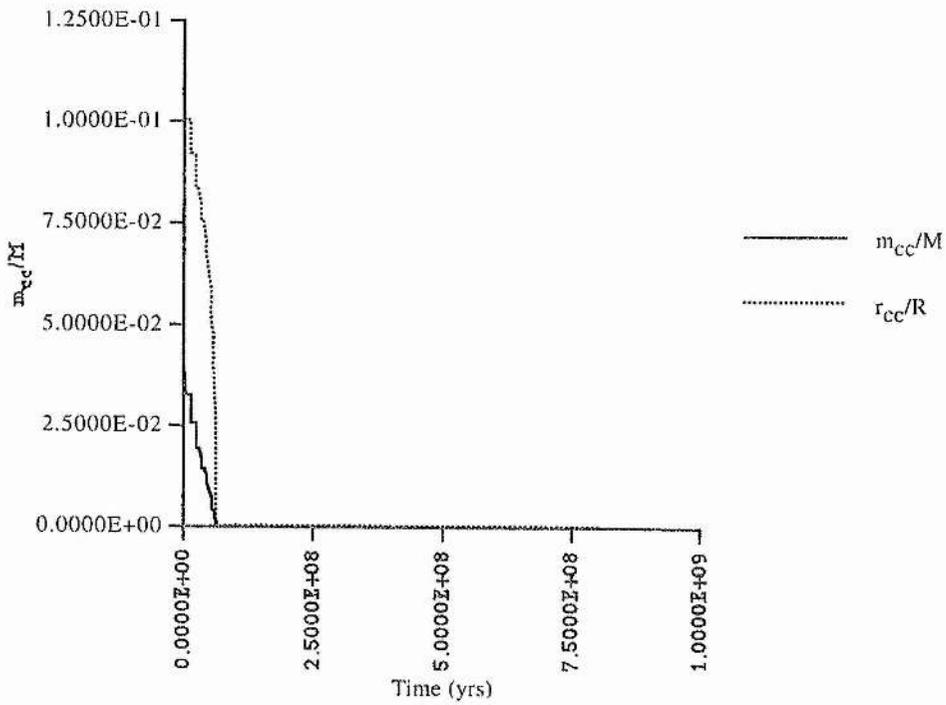
C.3 Convective Core Data, Initial Code

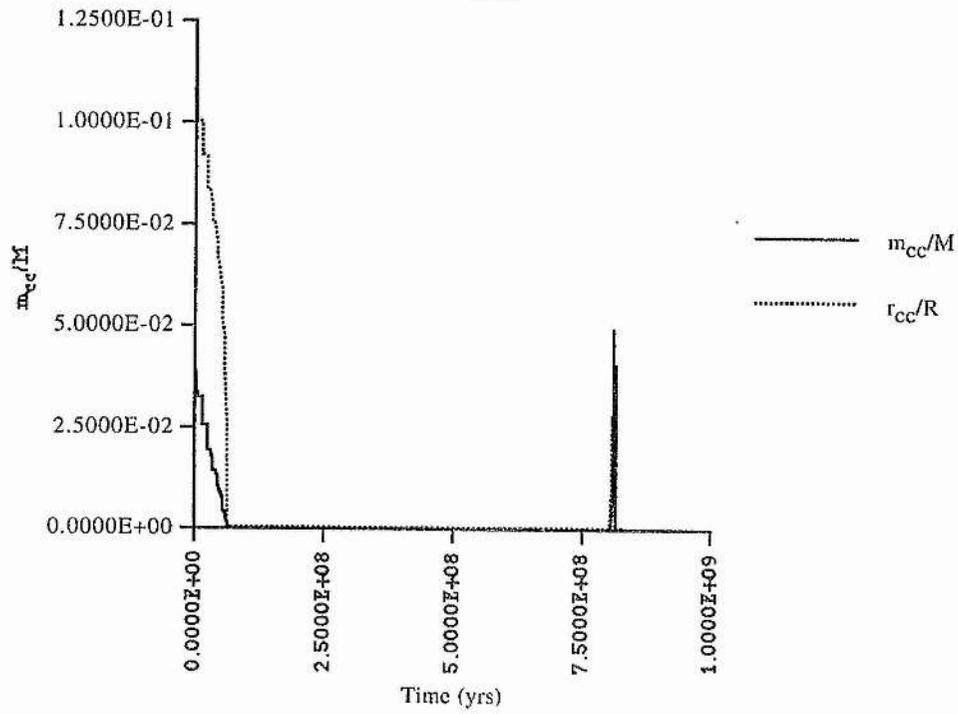
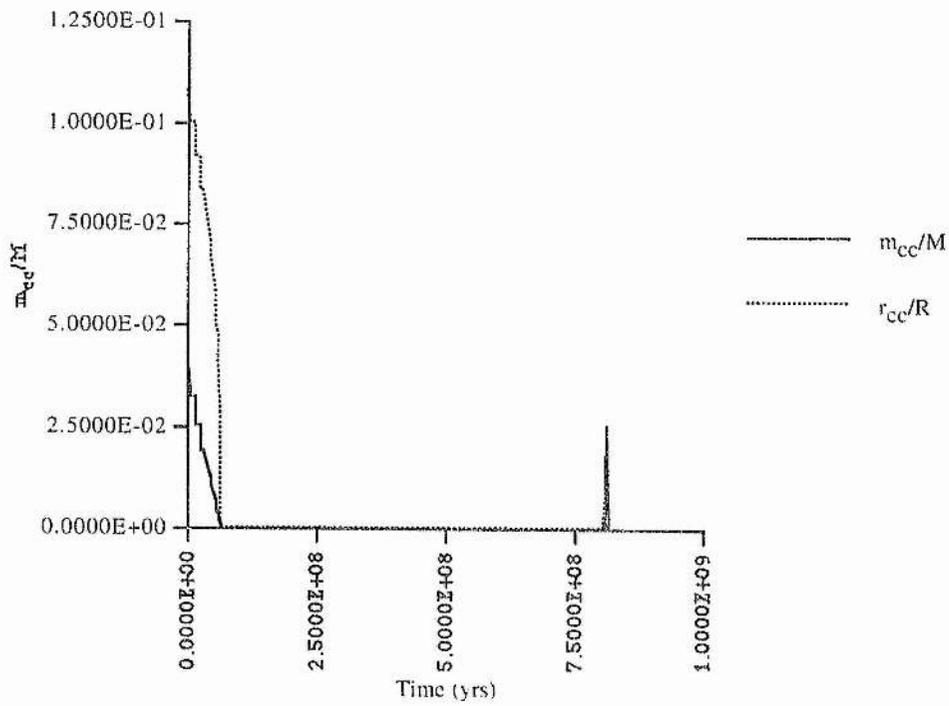
C.3.1 $15M_{\text{sun}}$ 

Pop IIIc, $15M_{\text{sun}}$ C.3.2 $10M_{\text{sun}}$ Pop IIIa, $10M_{\text{sun}}$ 

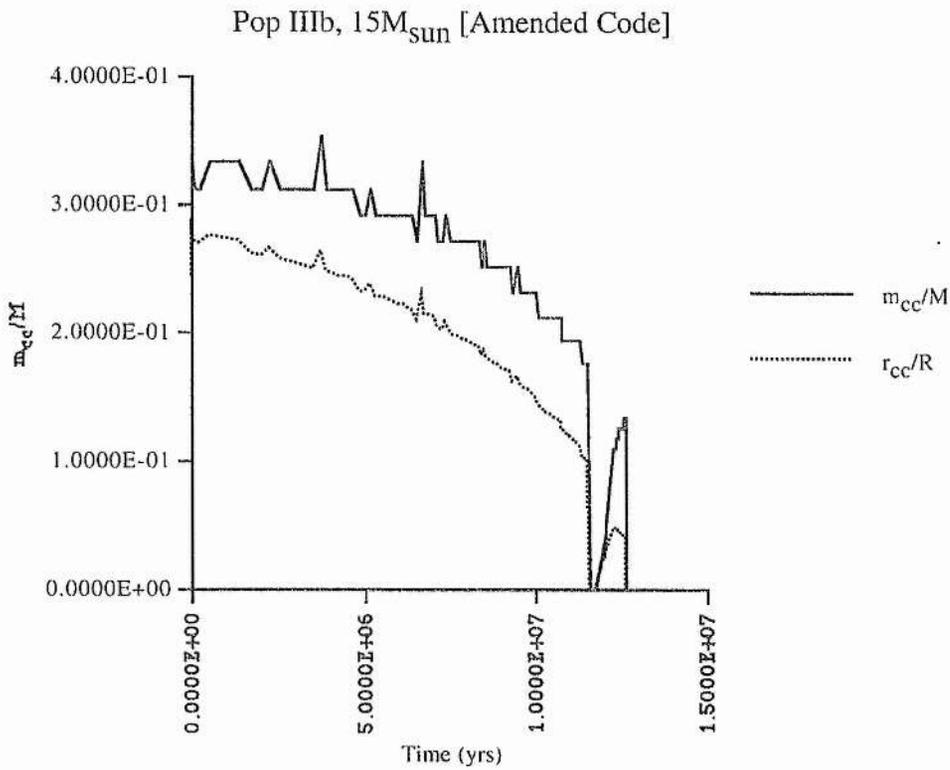
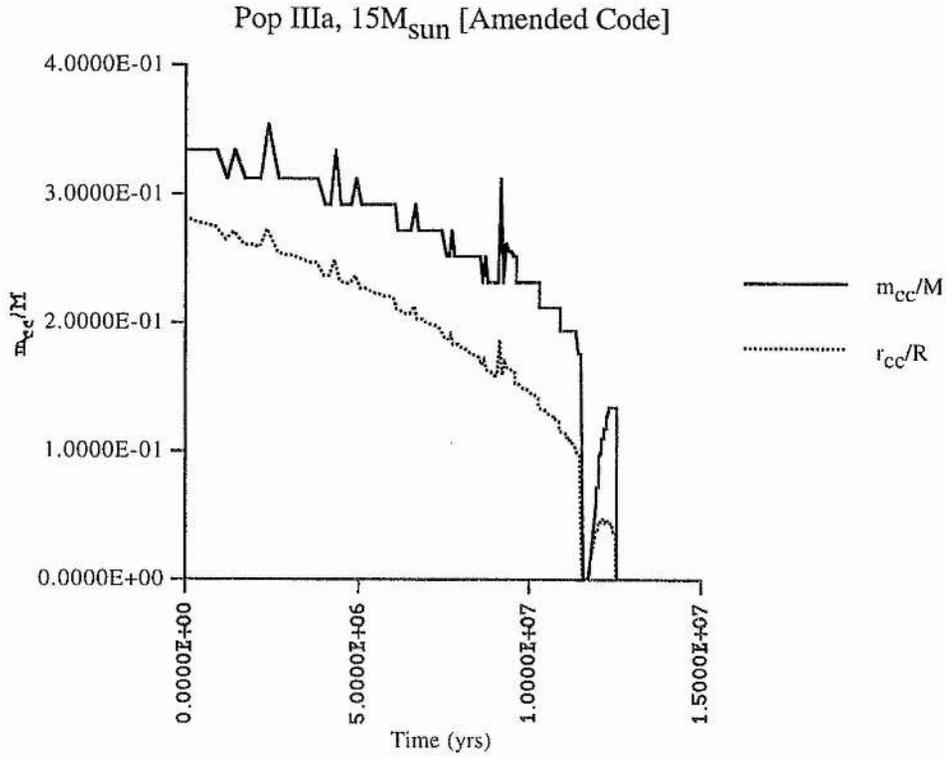


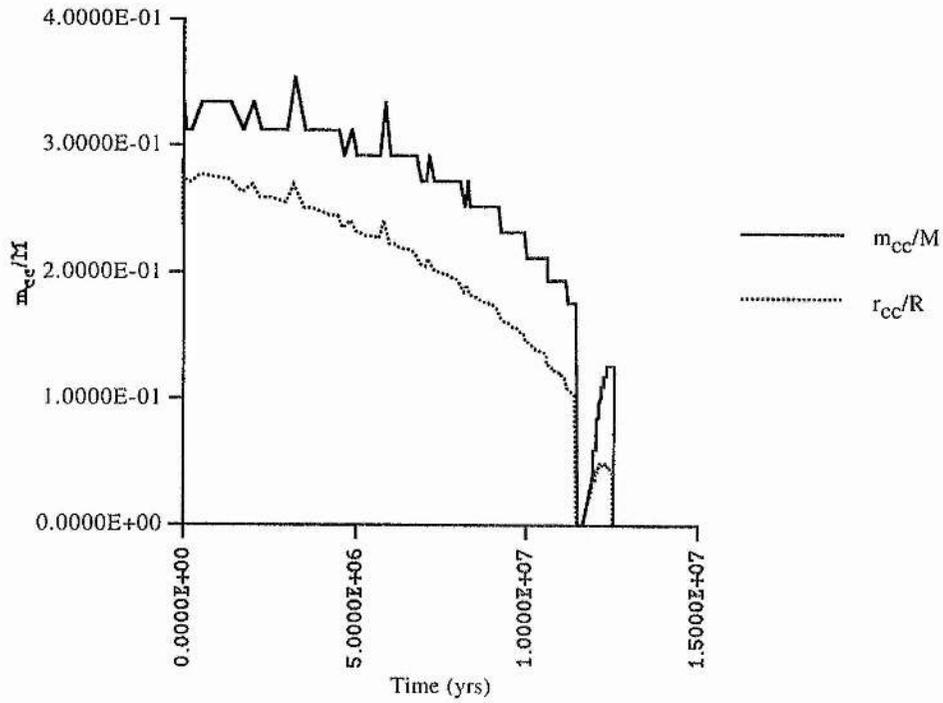
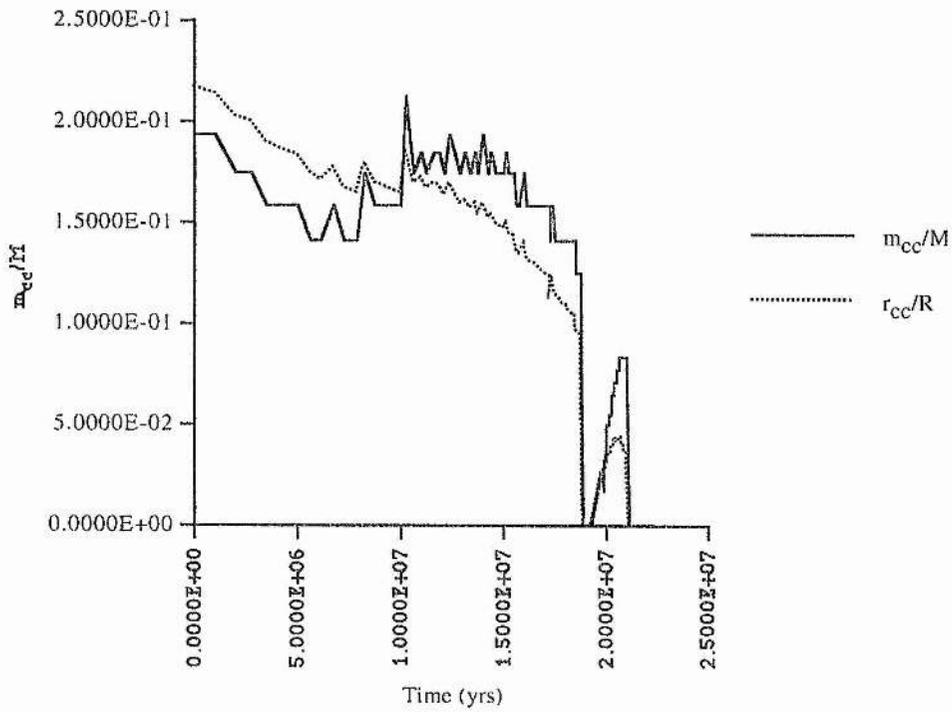
C.3.3 $5M_{\text{sun}}$ Pop IIIa, $5M_{\text{sun}}$ Pop IIIb, $5M_{\text{sun}}$ 

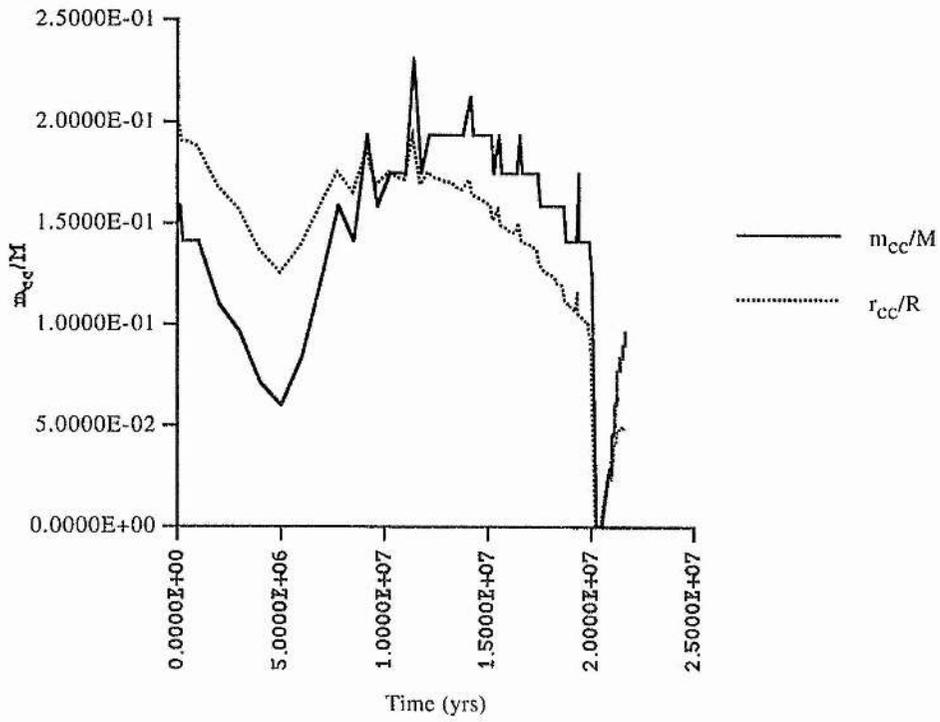
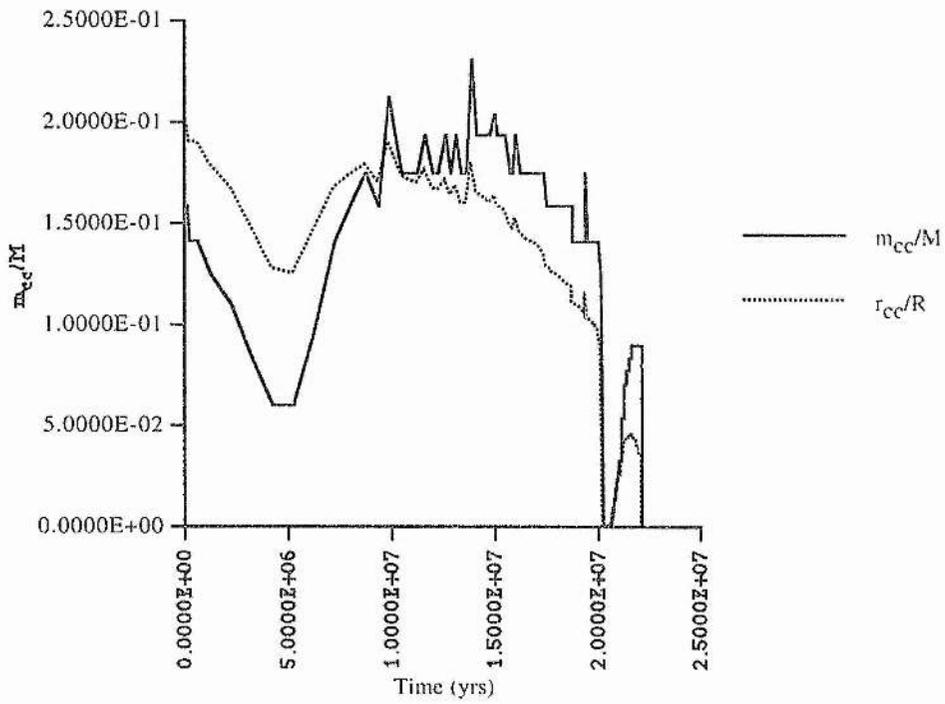
Pop IIIc, $5M_{\text{sun}}$ C.3.4 $2M_{\text{sun}}$ Pop IIIa, $2M_{\text{sun}}$ 

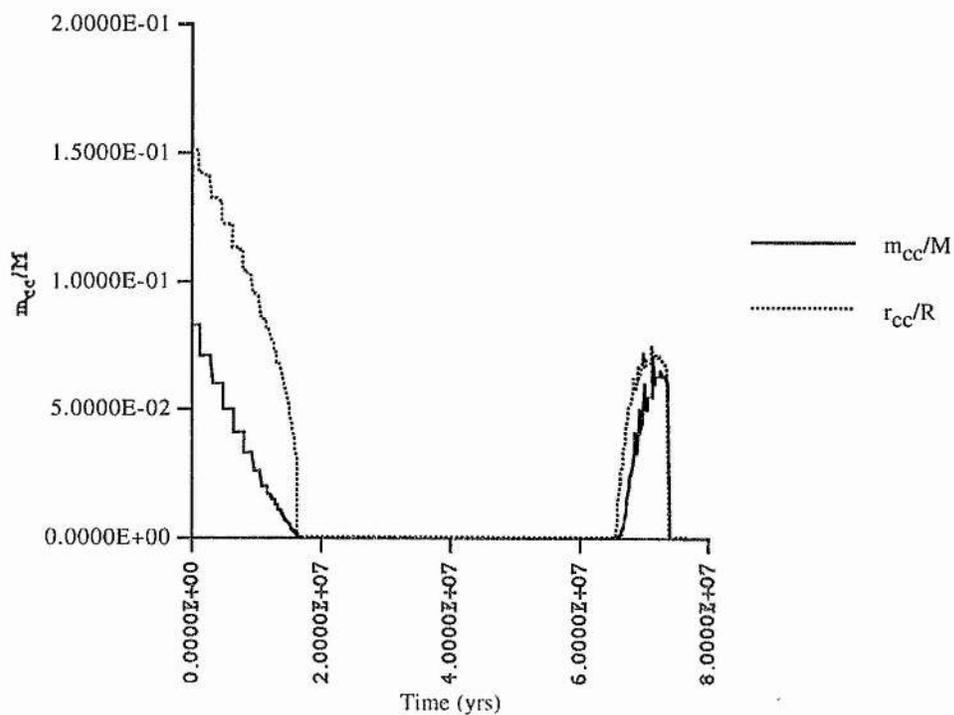
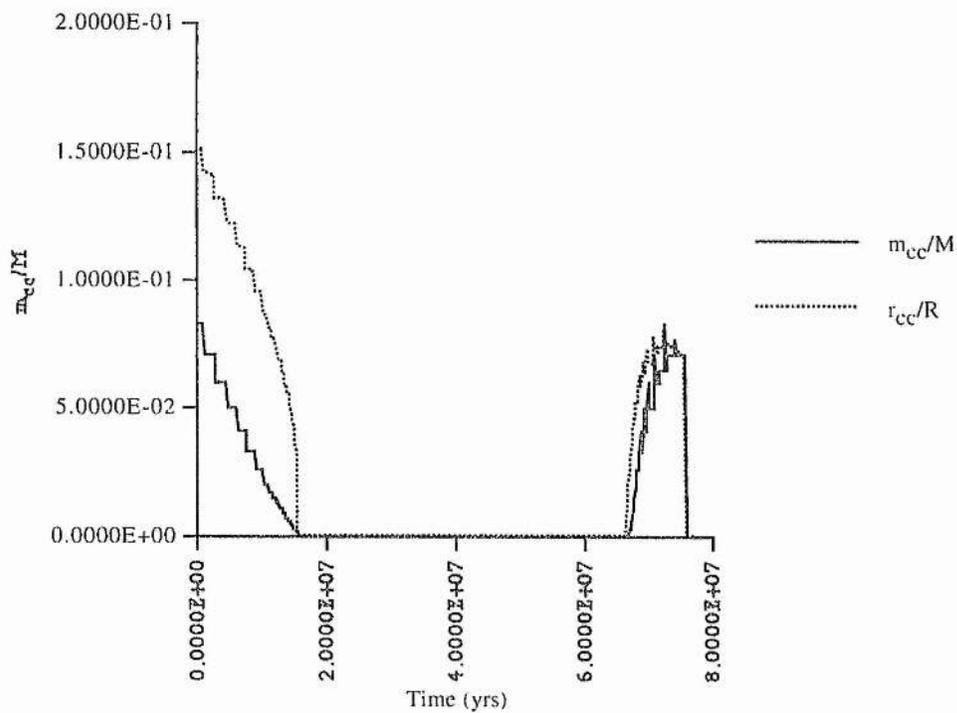
Pop IIIb, $2M_{\text{sun}}$ Pop IIIc, $2M_{\text{sun}}$ 

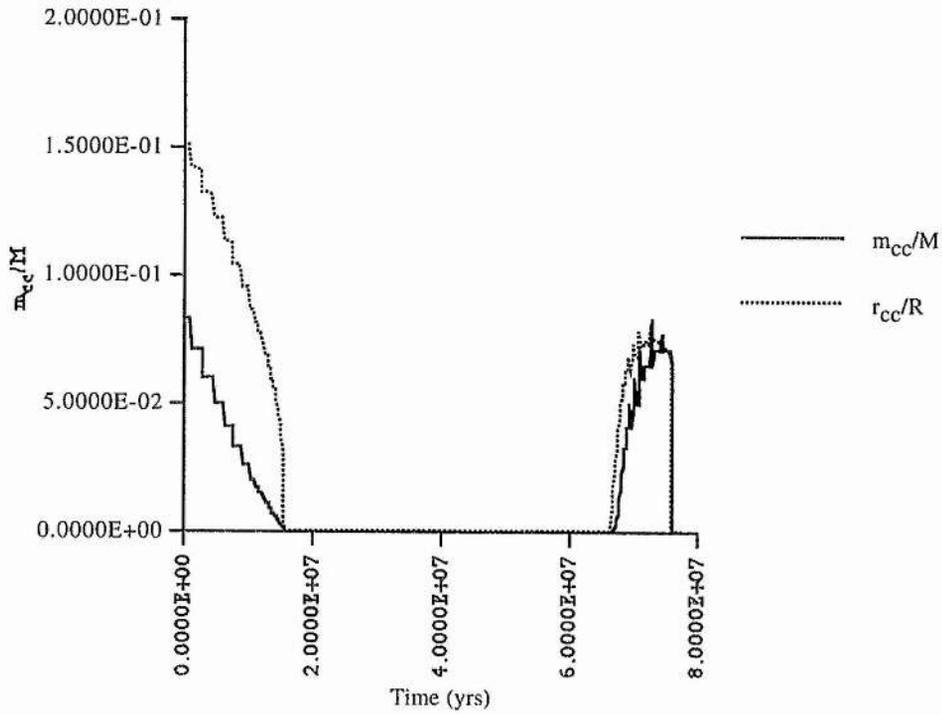
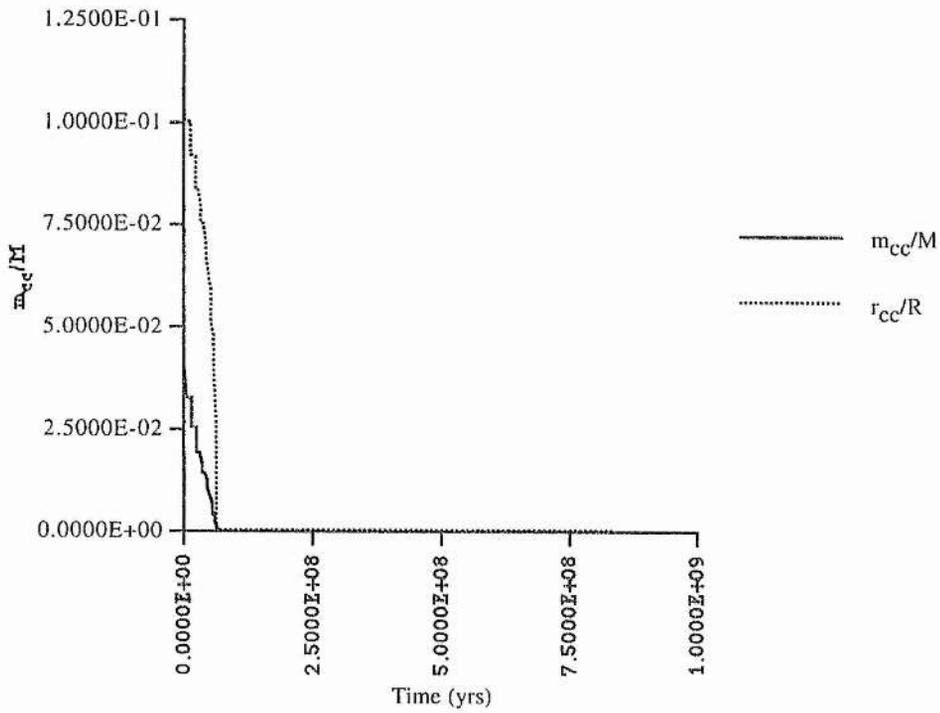
C.4 Convective Core Data, Amended Code

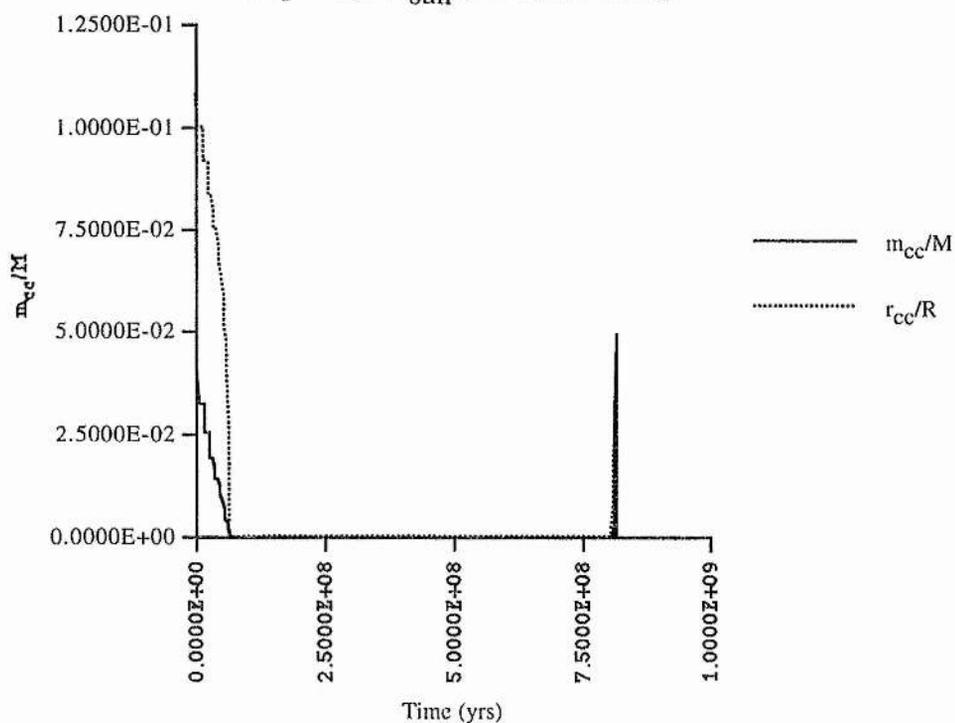
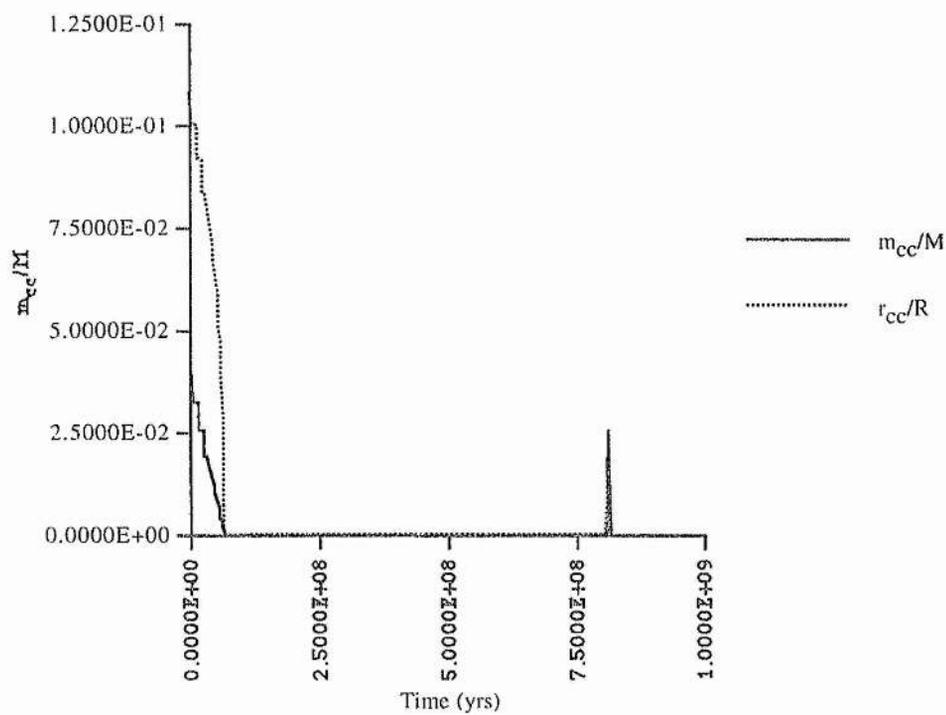
C.4.1 $15M_{\text{sun}}$ 

Pop IIIc, $15M_{\text{sun}}$ [Amended Code]C.4.2 $10M_{\text{sun}}$ Pop IIIa, $10M_{\text{sun}}$ [Amended Code]

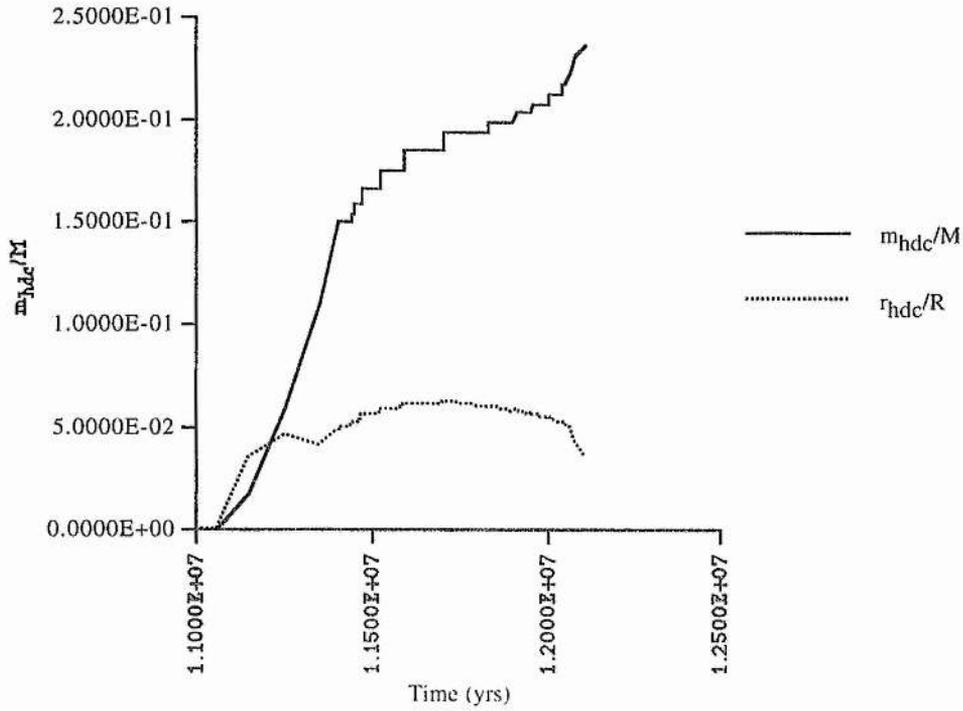
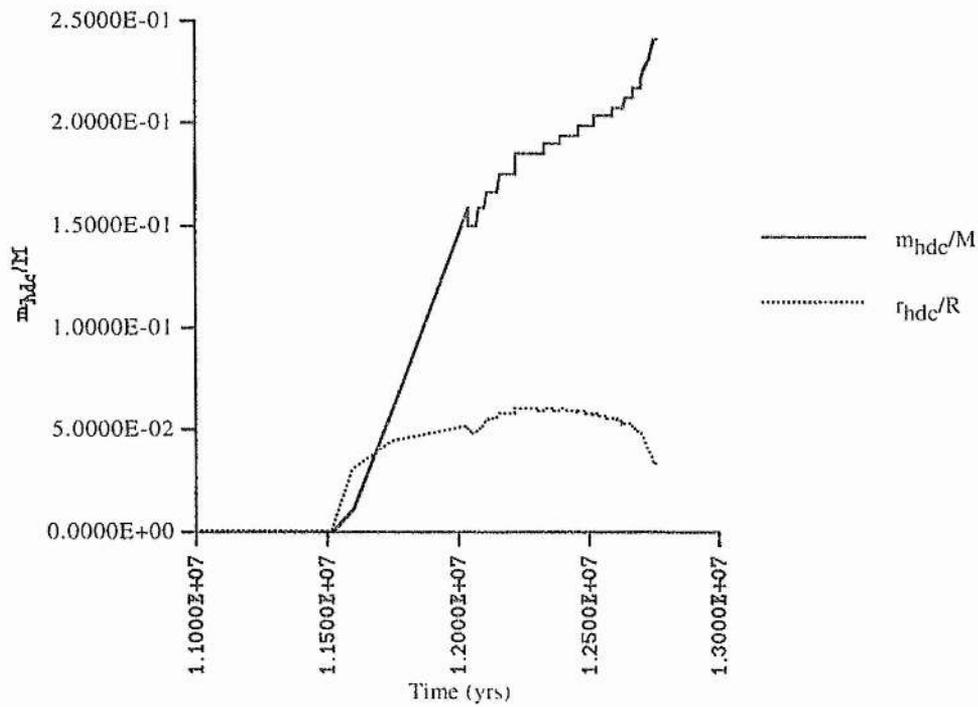
Pop IIIb, $10M_{\text{sun}}$ [Amended Code]Pop IIIc, $10M_{\text{sun}}$ [Amended Code]

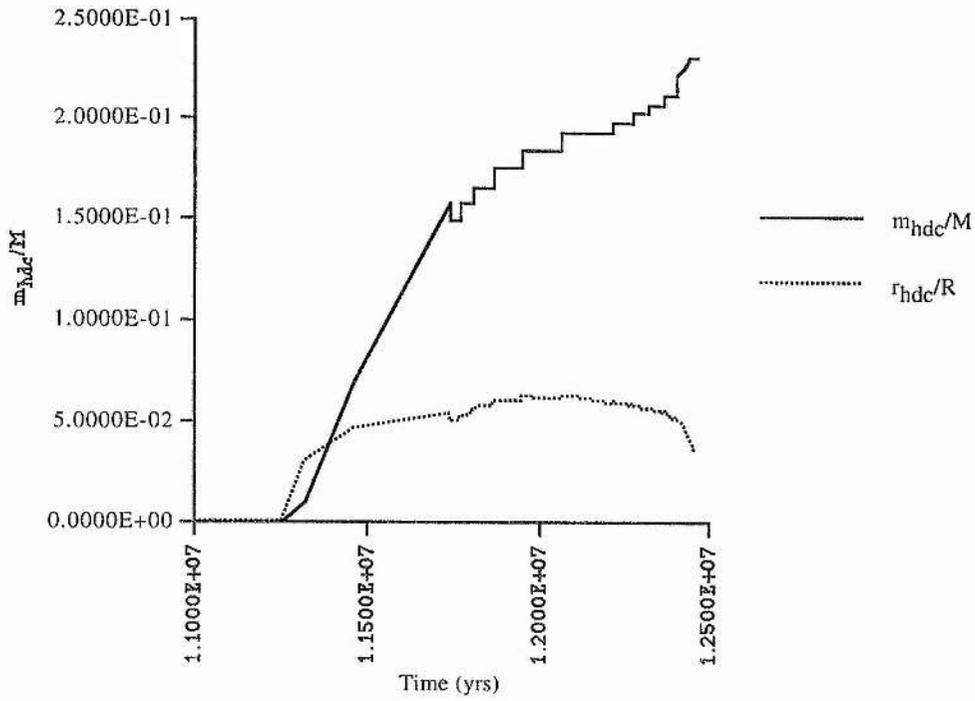
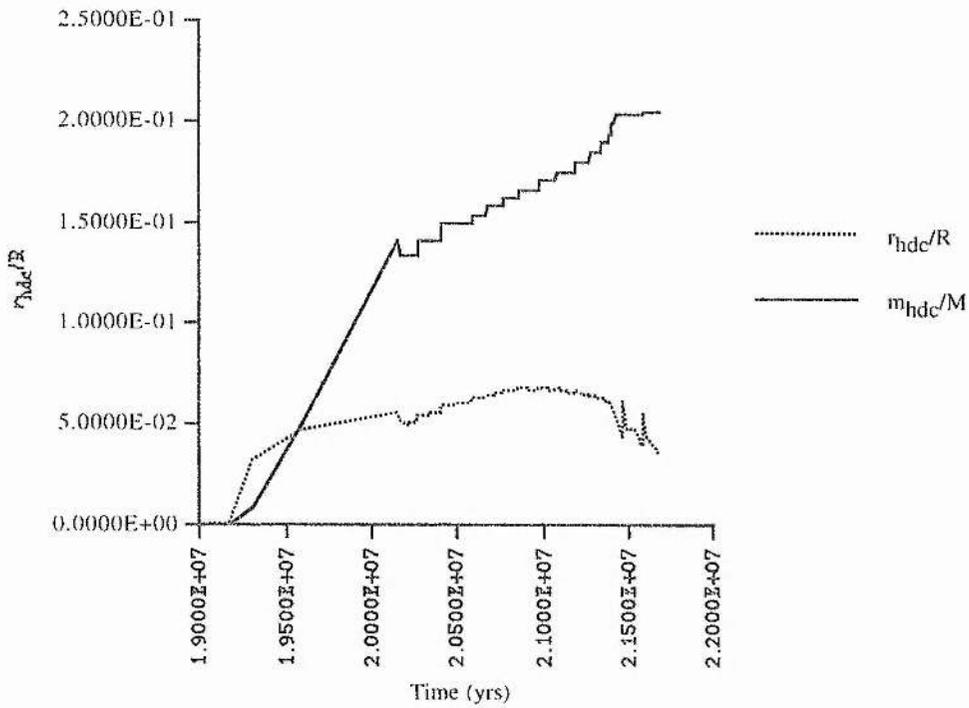
C.4.3 $5M_{\text{sun}}$ Pop IIIa $5M_{\text{sun}}$ [Amended Code]Pop IIIb, $5M_{\text{sun}}$ [Amended Code]

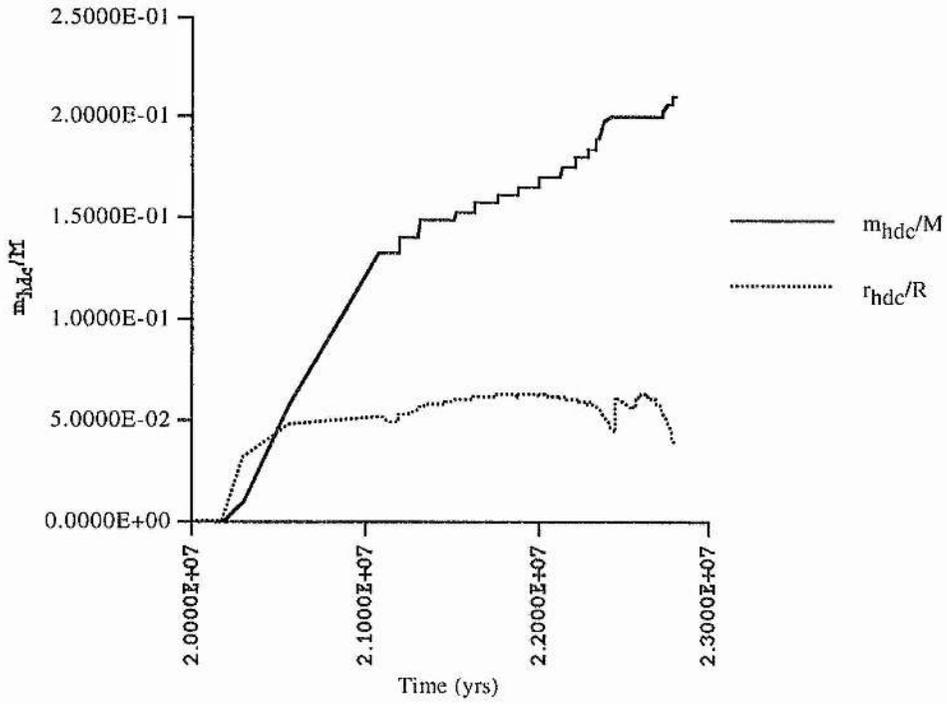
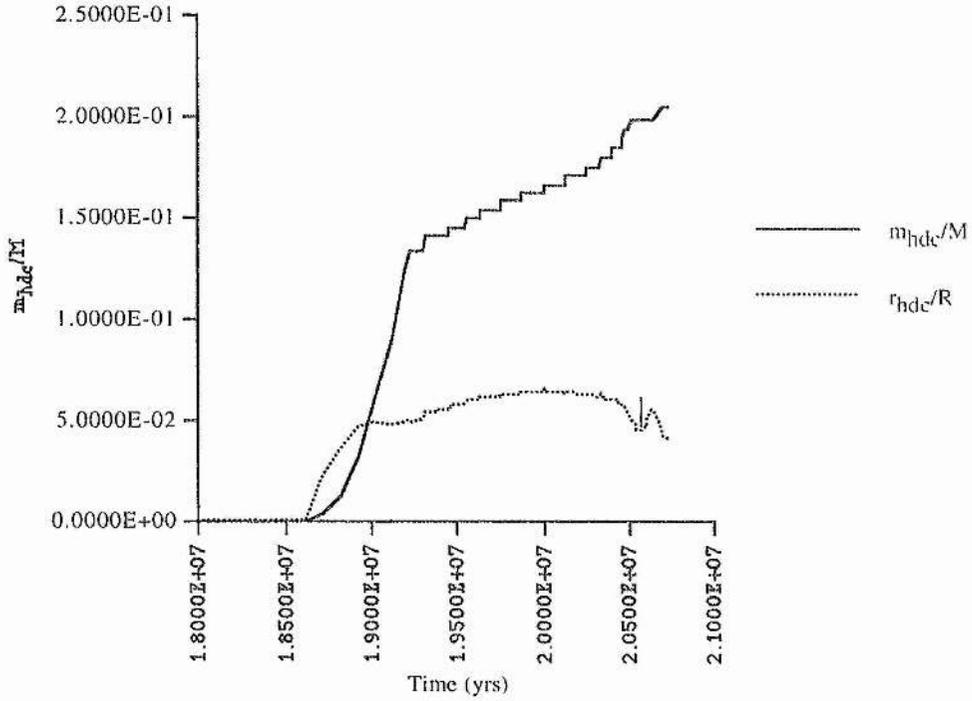
Pop IIIc, $5M_{\text{sun}}$ [Amended Code]C.4.4 $2M_{\text{sun}}$ Pop IIIa, $2M_{\text{sun}}$ [Amended Code]

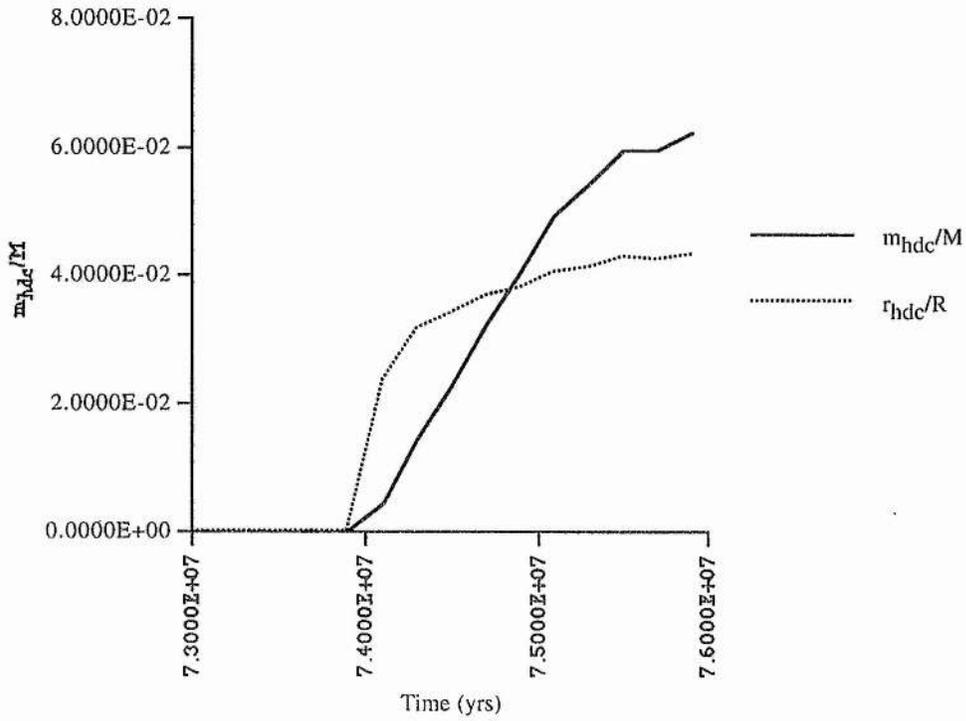
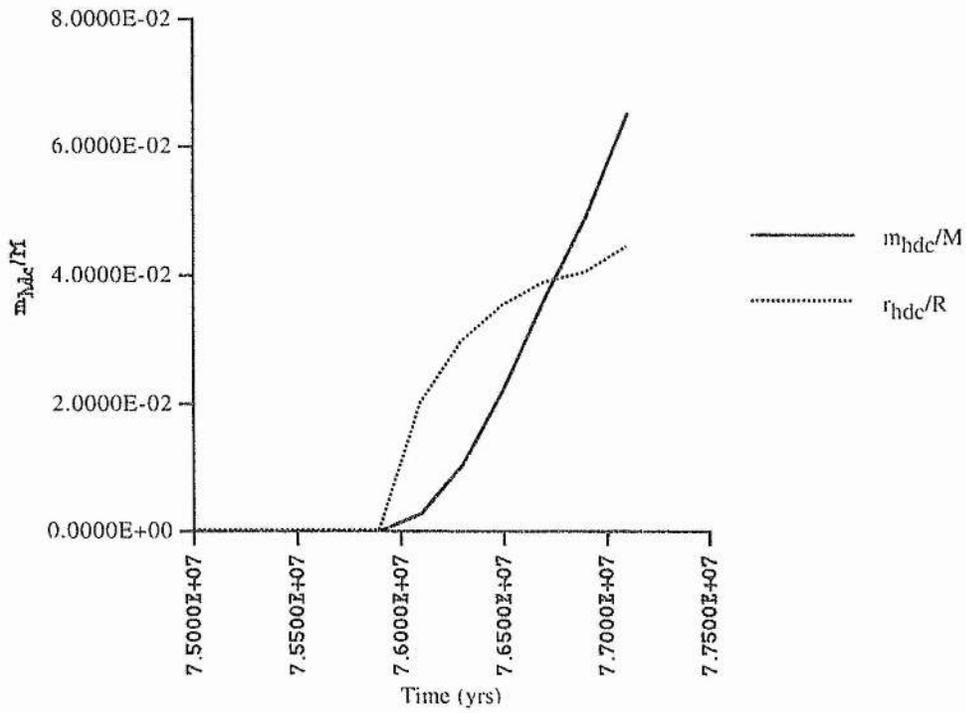
Pop IIIb, $2M_{\text{sun}}$ [Amended Code]Pop IIIc, $2M_{\text{sun}}$ [Amended Code]

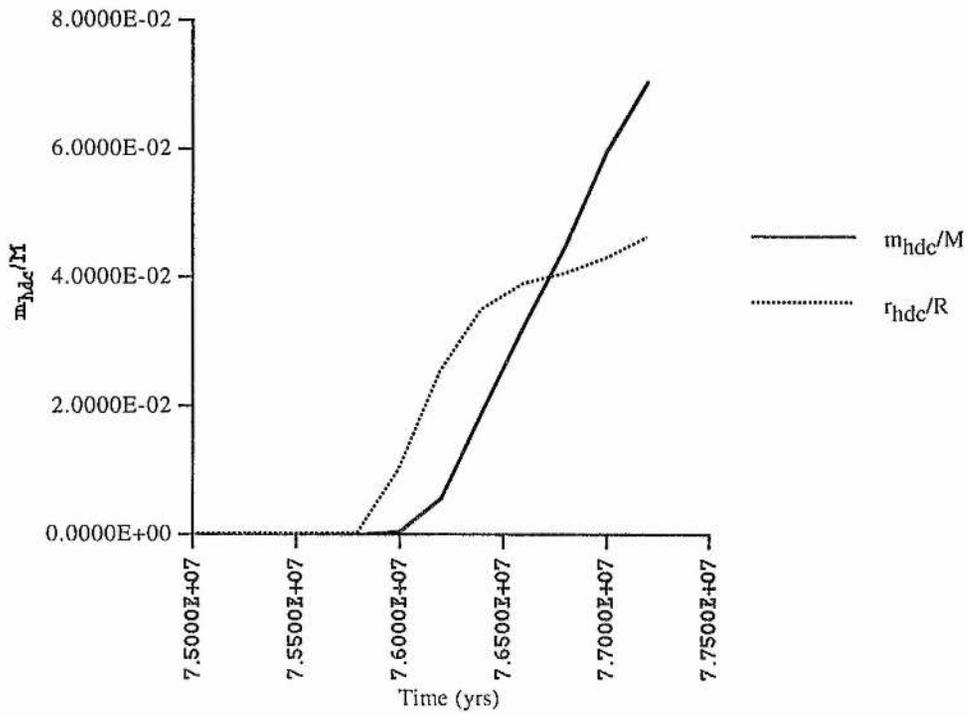
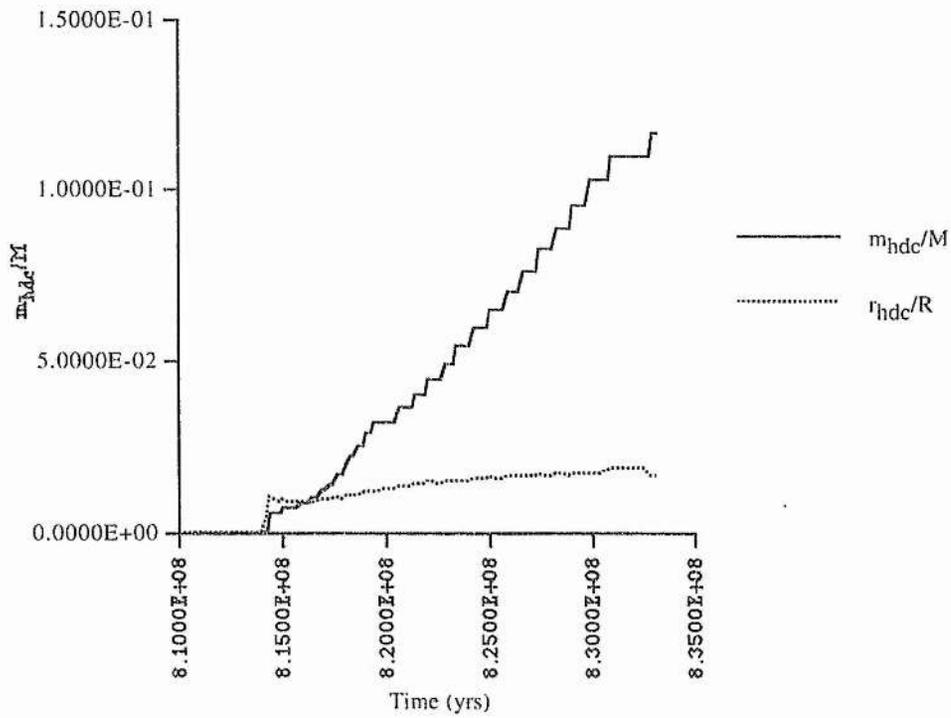
C.5 Hydrogen-Depleted Core Data, Initial Code

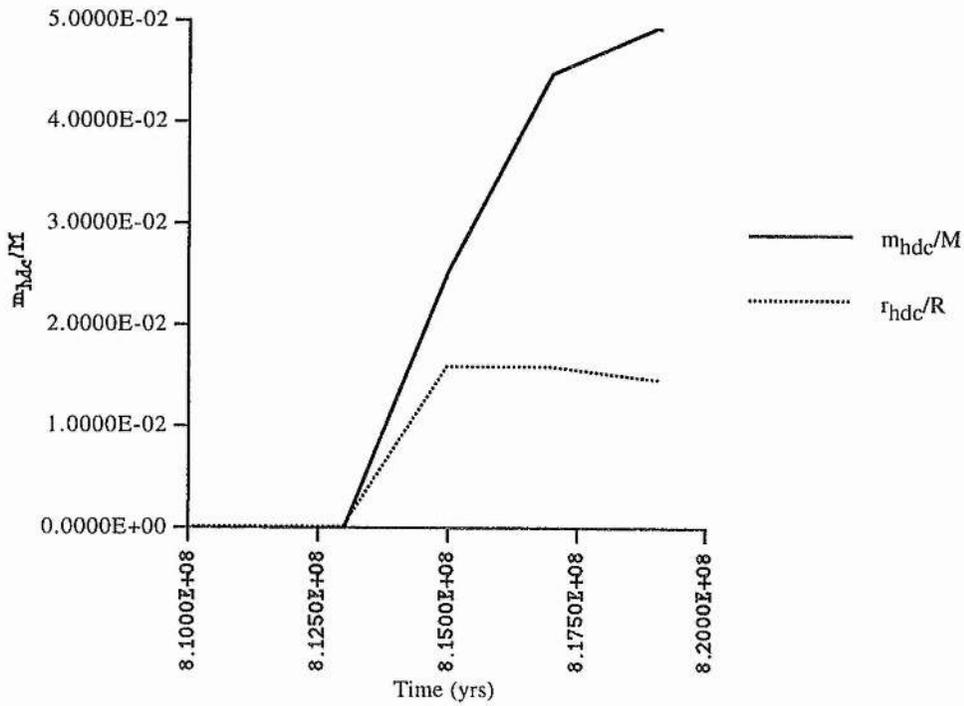
C.5.1 $15M_{\text{sun}}$ Pop IIIa, $15M_{\text{sun}}$ Pop IIIb, $15M_{\text{sun}}$ 

Pop IIIc, $15M_{\text{sun}}$ C.5.2 $10M_{\text{sun}}$ Pop IIIa, $10M_{\text{sun}}$ 

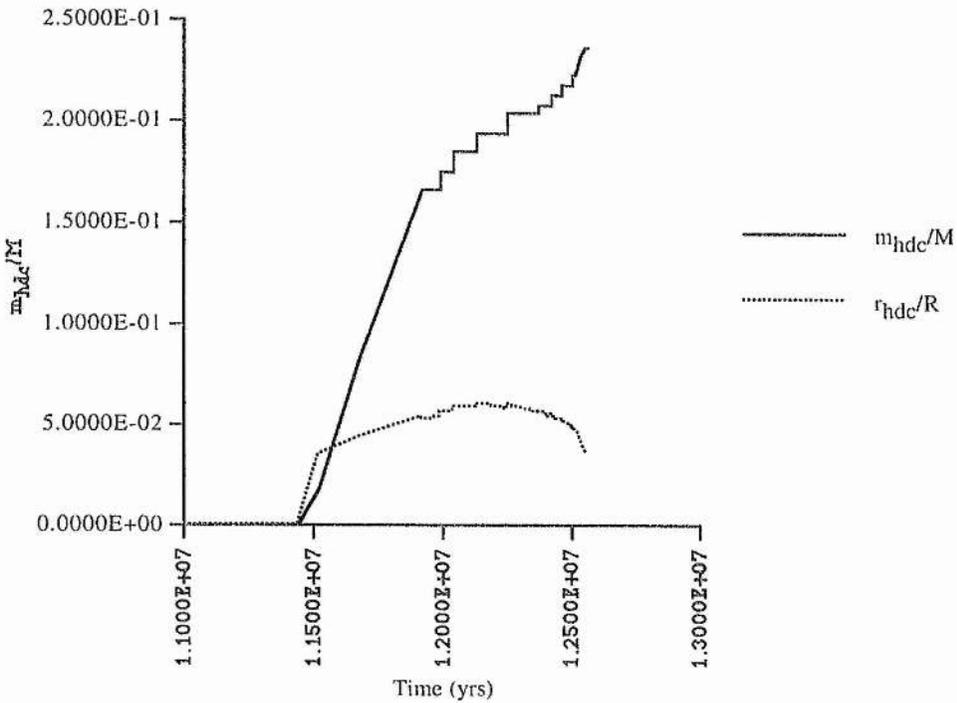
Pop IIIb, $10M_{\text{sun}}$ Pop IIIc, $10M_{\text{sun}}$ 

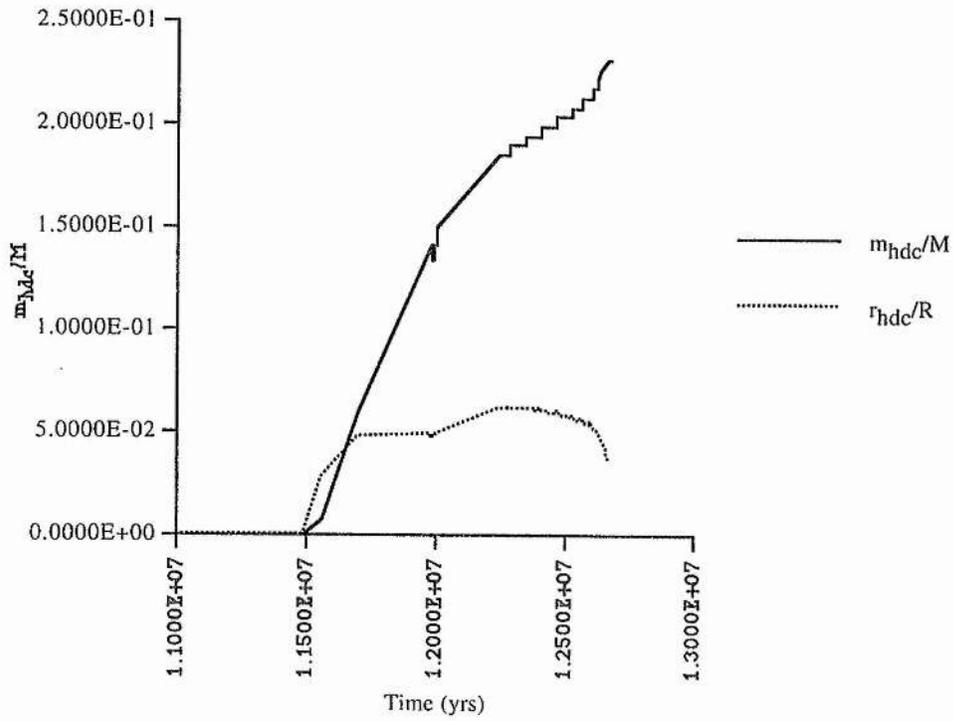
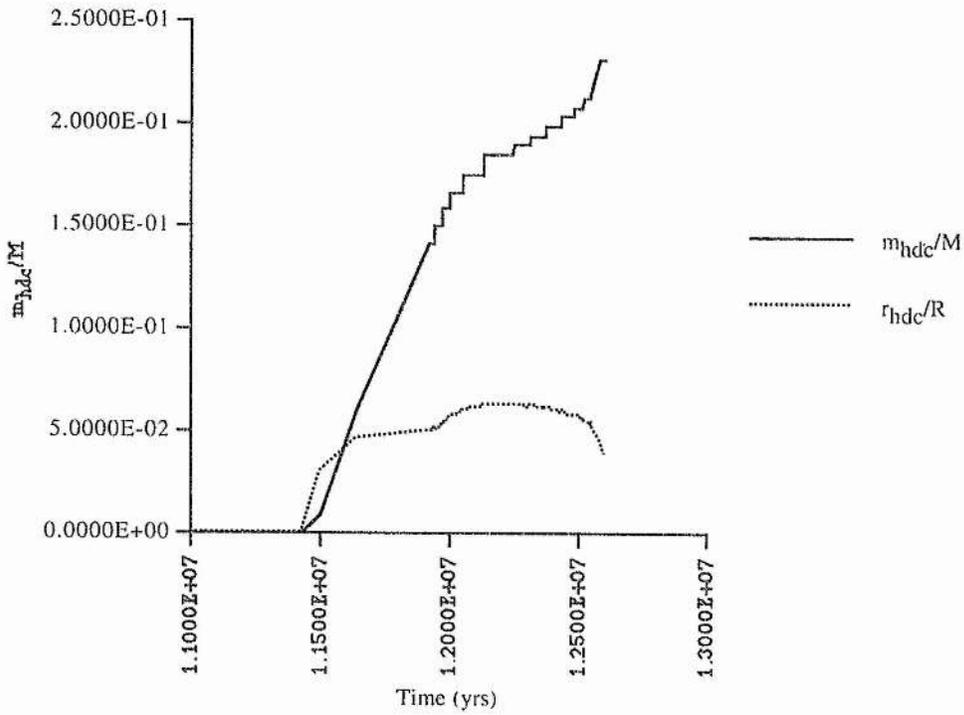
C.5.3 $5M_{\text{sun}}$ Pop IIIa, $5M_{\text{sun}}$ Pop IIIb, $5M_{\text{sun}}$ 

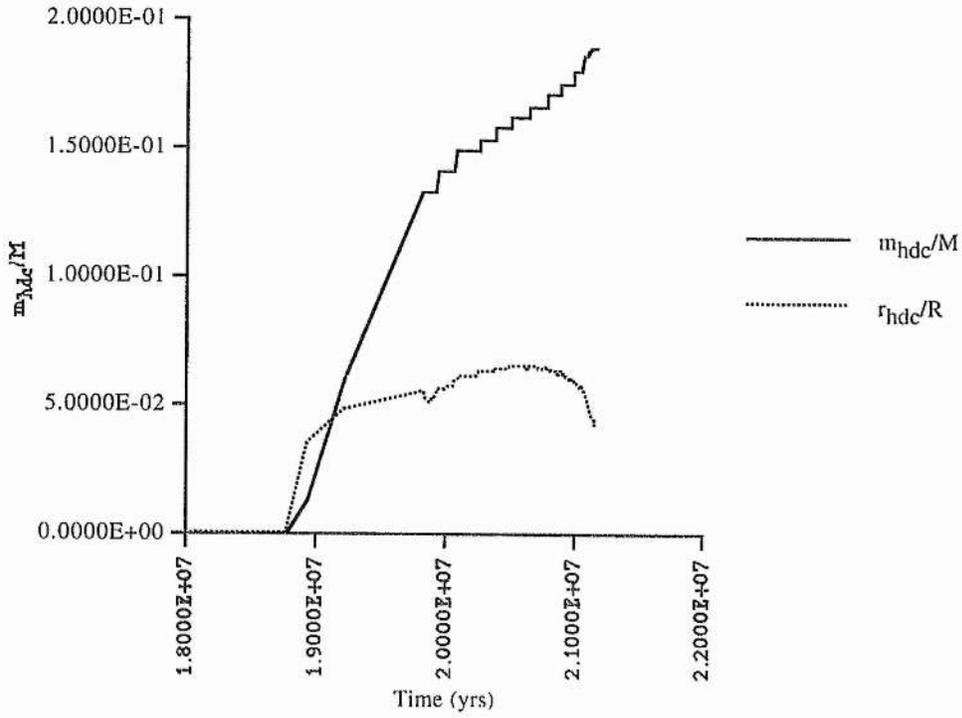
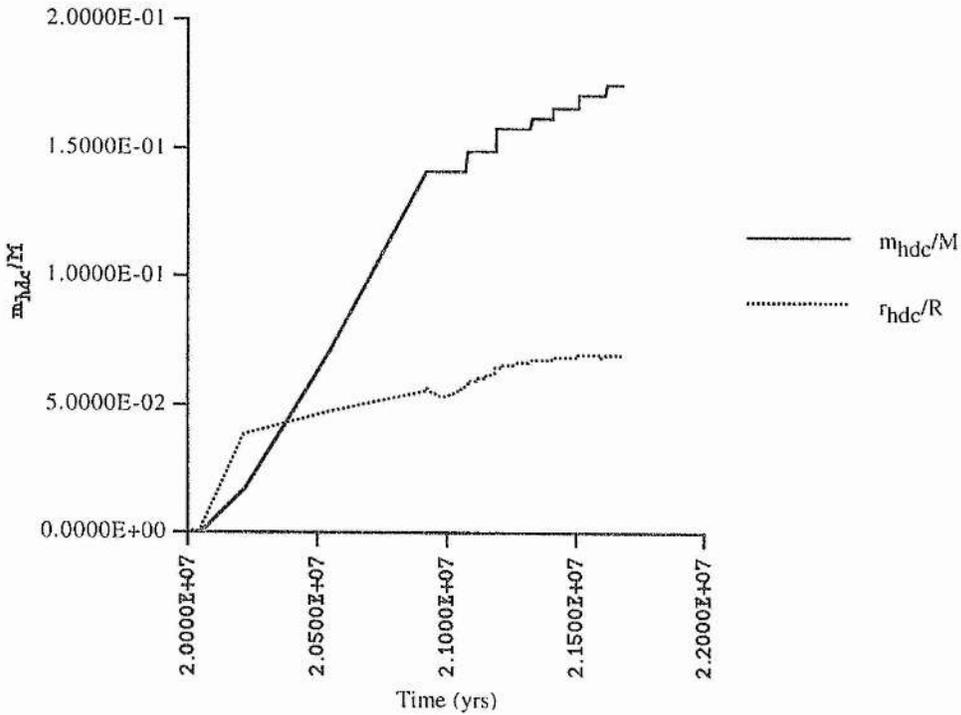
Pop IIIc, $5M_{\text{sun}}$ C.5.4 $2M_{\text{sun}}$ Pop IIIb, $2M_{\text{sun}}$ 

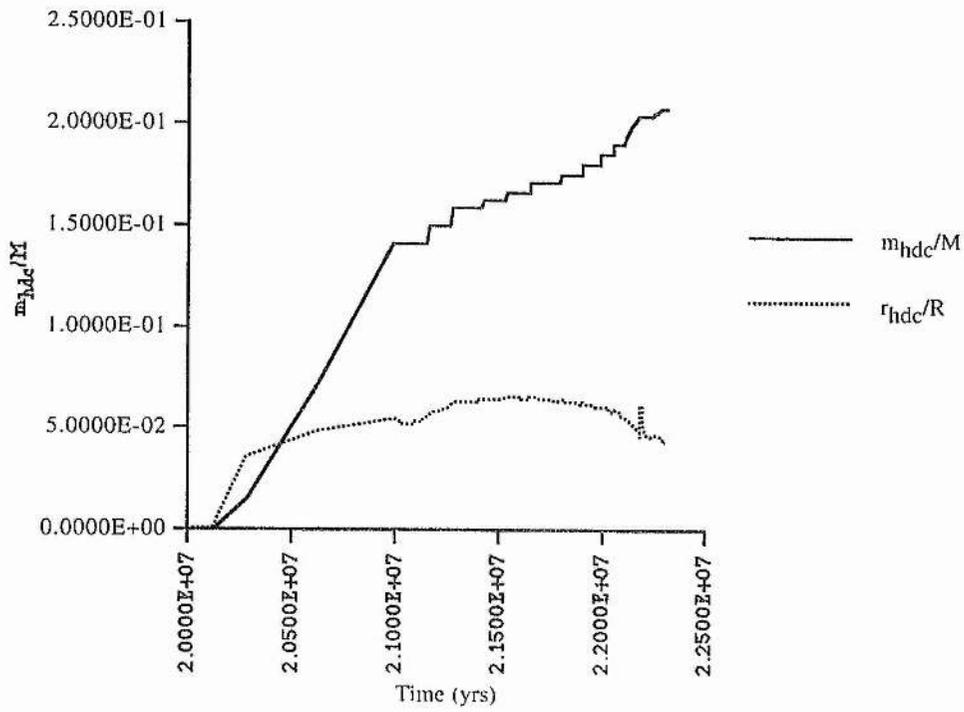
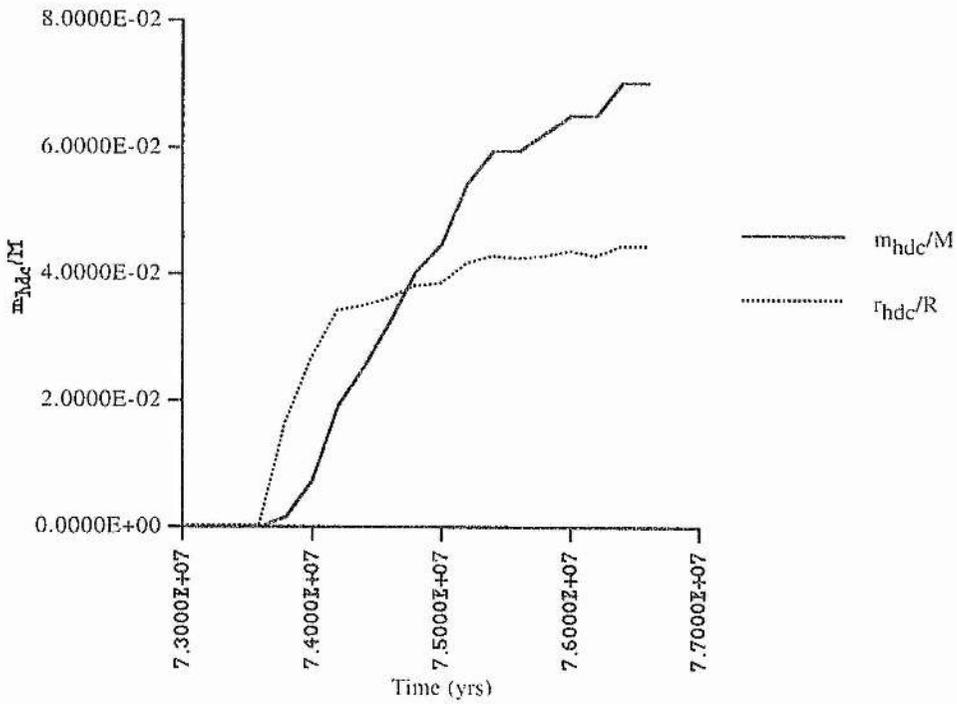
Pop IIIc, $2M_{\text{sun}}$ 

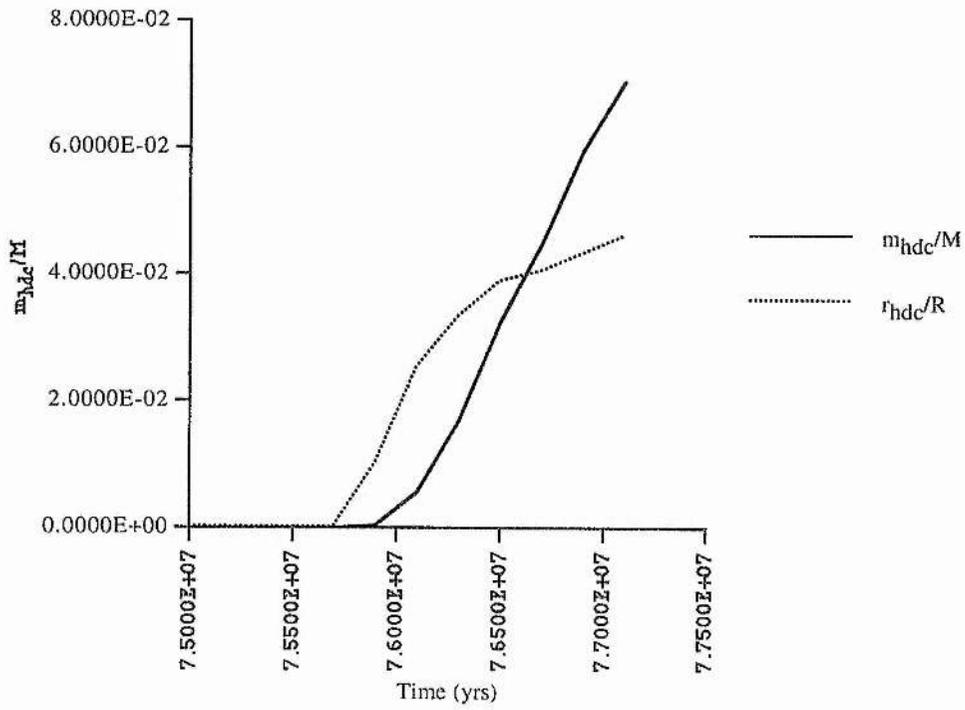
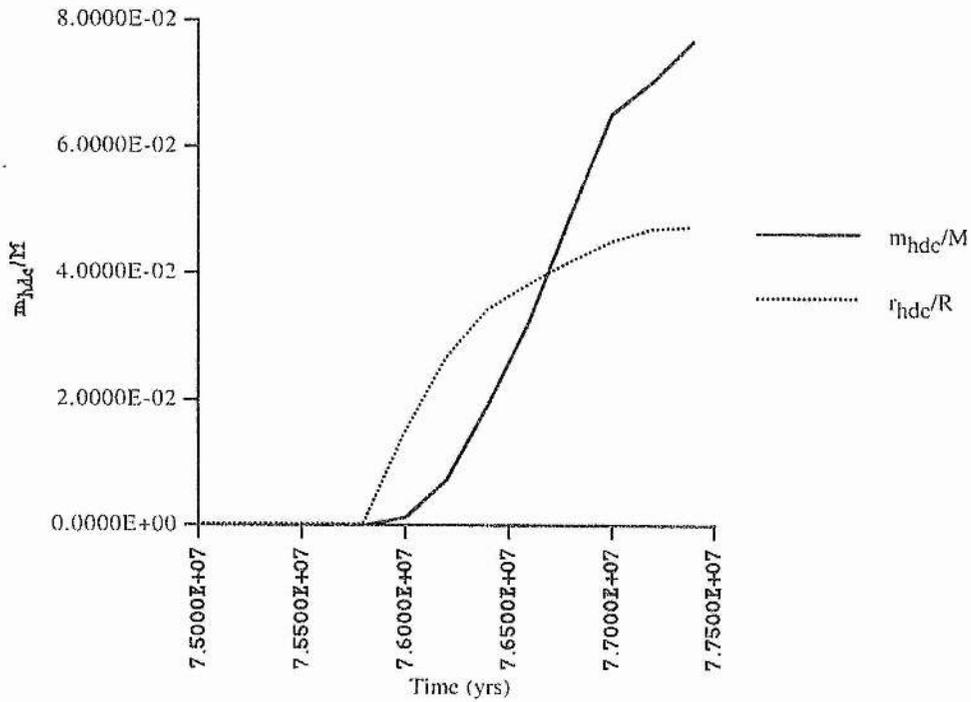
C.6 Hydrogen-Depleted Core Data, Amended Code

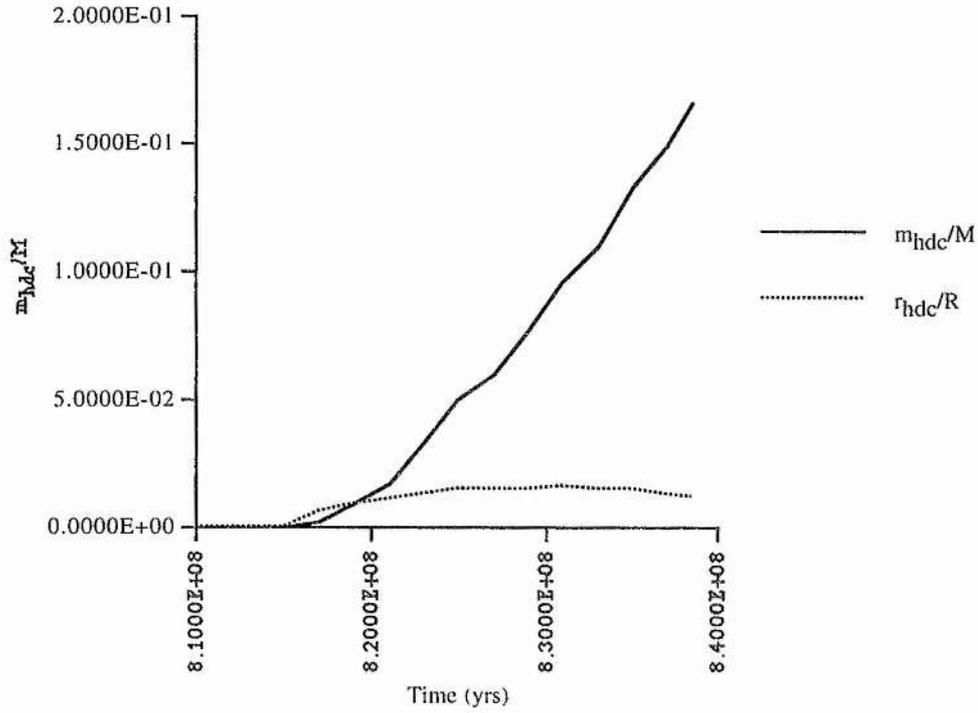
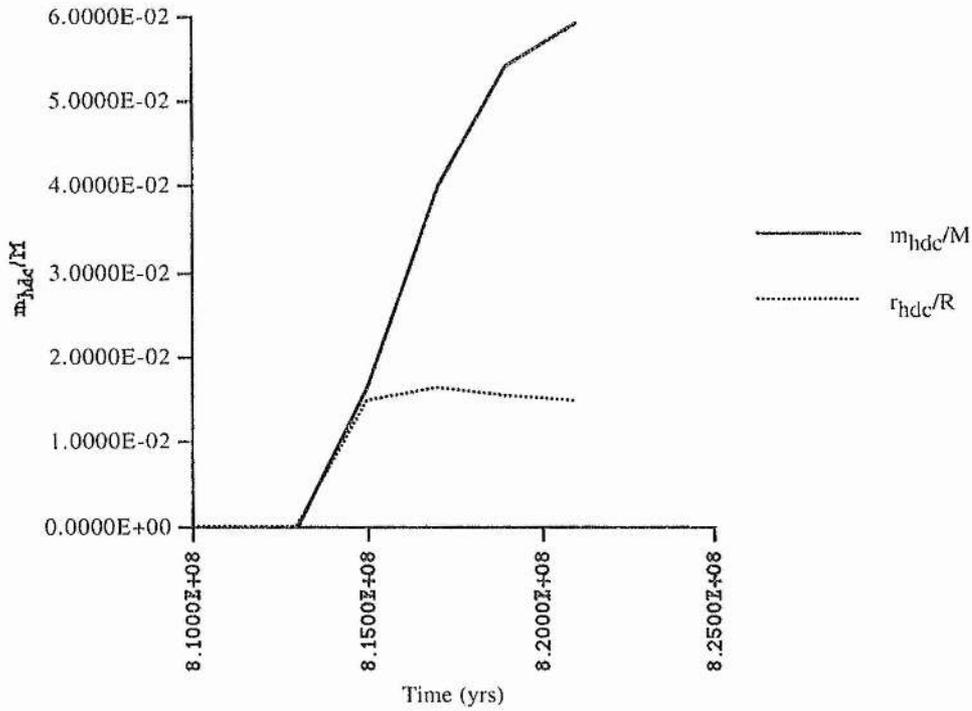
C.6.1 $15M_{\text{sun}}$ Pop IIIa, $15M_{\text{sun}}$ [Amended Code]

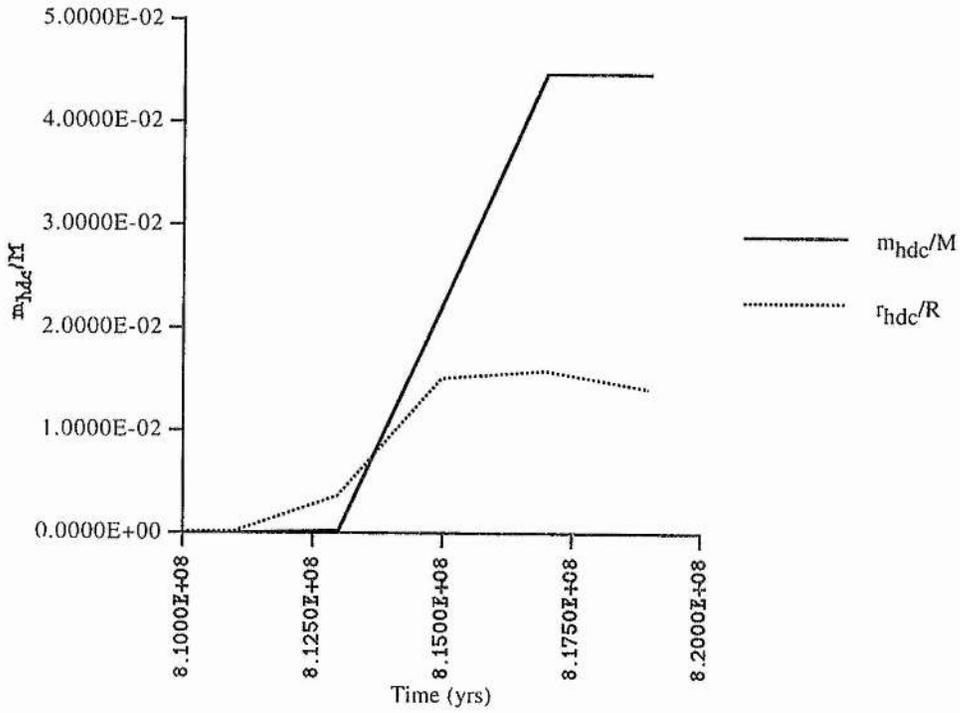
Pop IIIb, $15M_{\text{sun}}$ [Amended Code]Pop IIIc, $15M_{\text{sun}}$ [Amended Code]

C.6.2 $10M_{\text{sun}}$ Pop IIIa, $10M_{\text{sun}}$ [Amended Code]Pop IIIb, $10M_{\text{sun}}$ [Amended Code]

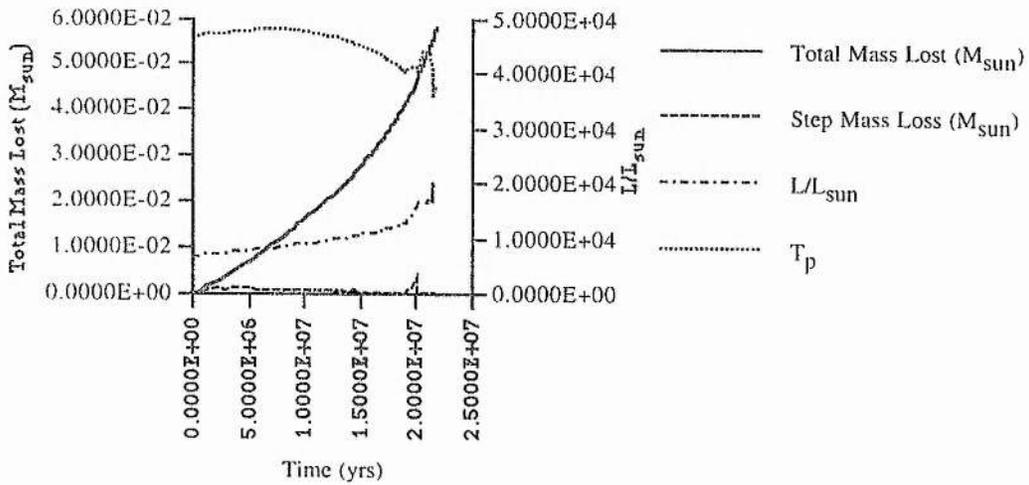
Pop IIIc, $10M_{\text{sun}}$ [Amended Code]C.6.3 $5M_{\text{sun}}$ Pop IIIa, $5M_{\text{sun}}$ [Amended Code]

Pop IIIb, $5M_{\text{sun}}$ [Amended Code]Pop IIIc, $5M_{\text{sun}}$ [Amended Code]

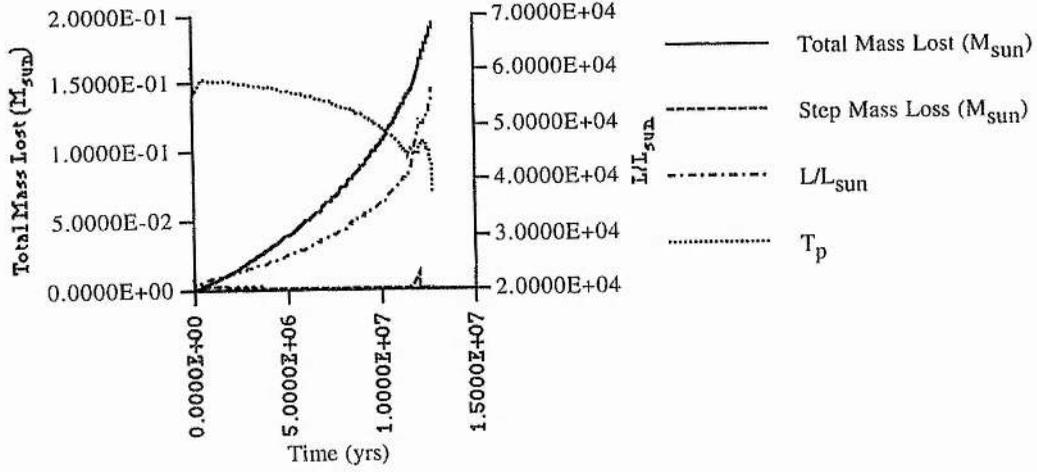
C.6.4 $2M_{\text{sun}}$ Pop IIIa, $2M_{\text{sun}}$ [Amended Code]Pop IIIb, $2M_{\text{sun}}$ [Amended Code]

Pop IIIc, $2M_{\text{sun}}$ [Amended Code]

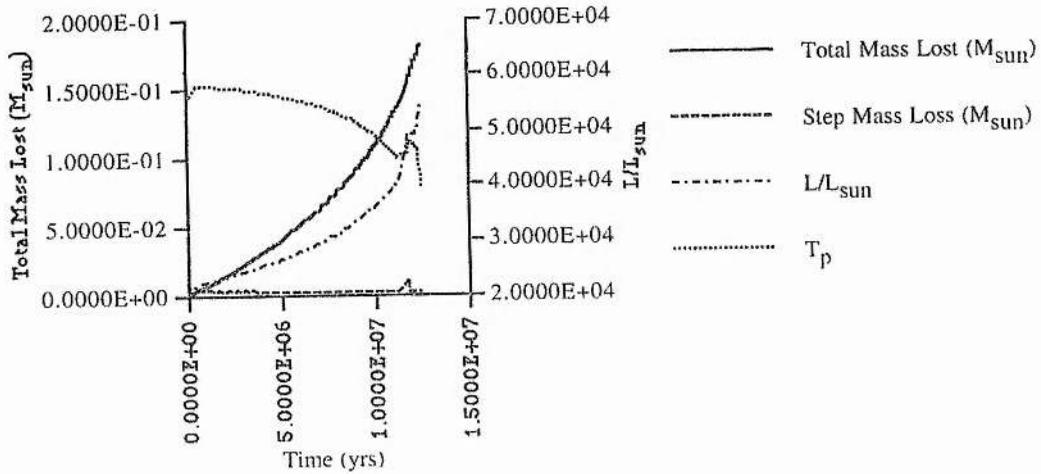
C.7 Mass Loss, Luminosity and Temperature Data, Initial Code

C.7.1 $15M_{\text{sun}}$ Pop IIIa, $15M_{\text{sun}}$ 

Pop IIIb, 15M_{sun}

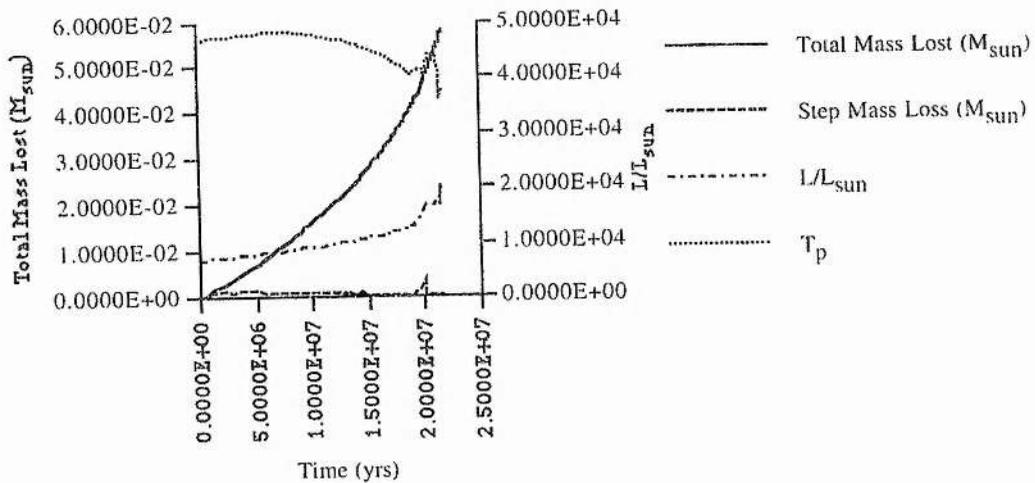


Pop IIIc, 15M_{sun}

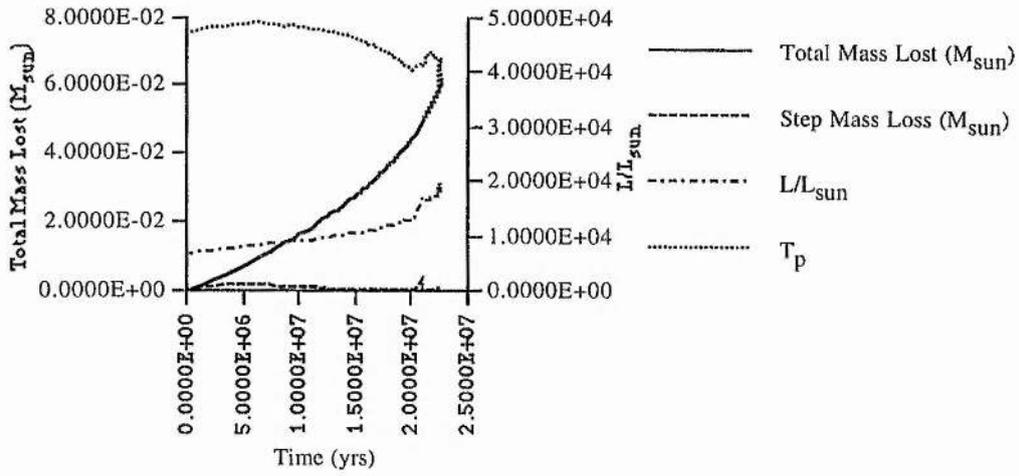


C.7.2 10M_{sun}

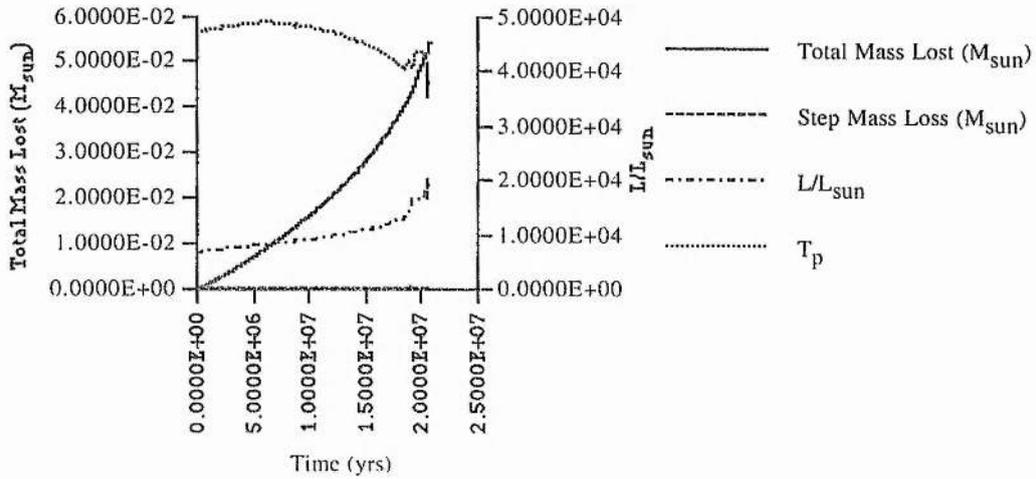
Pop IIIa, 10M_{sun}



Pop IIIb, $10M_{\text{sun}}$

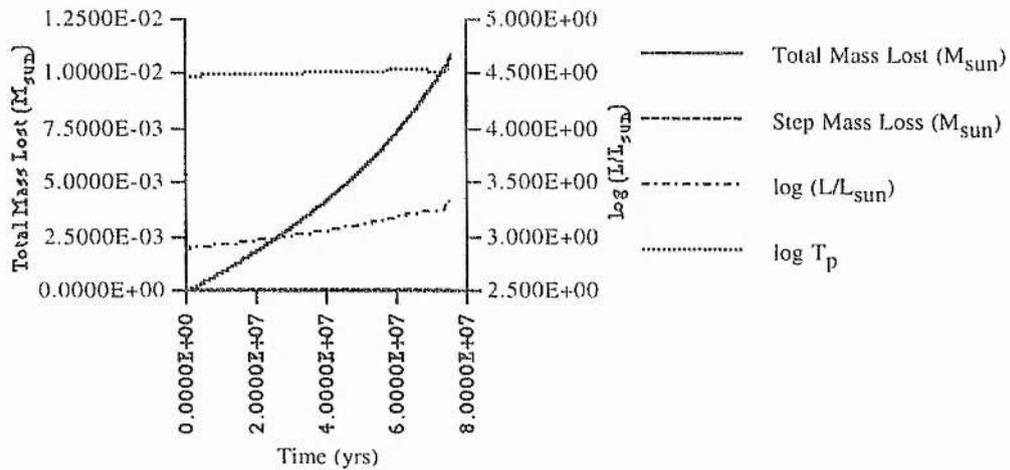


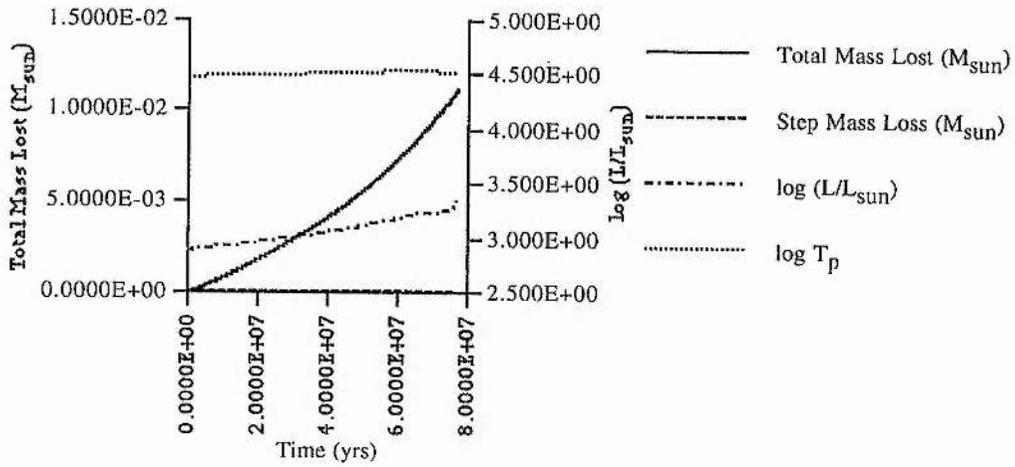
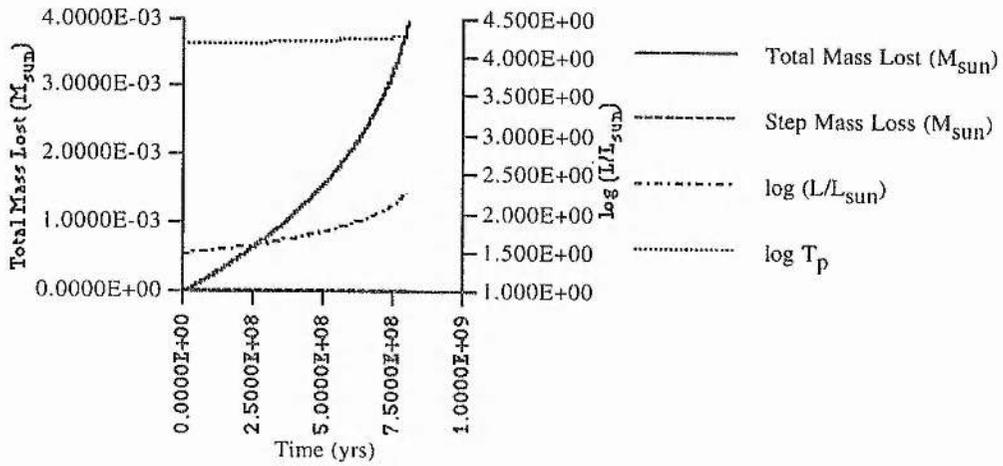
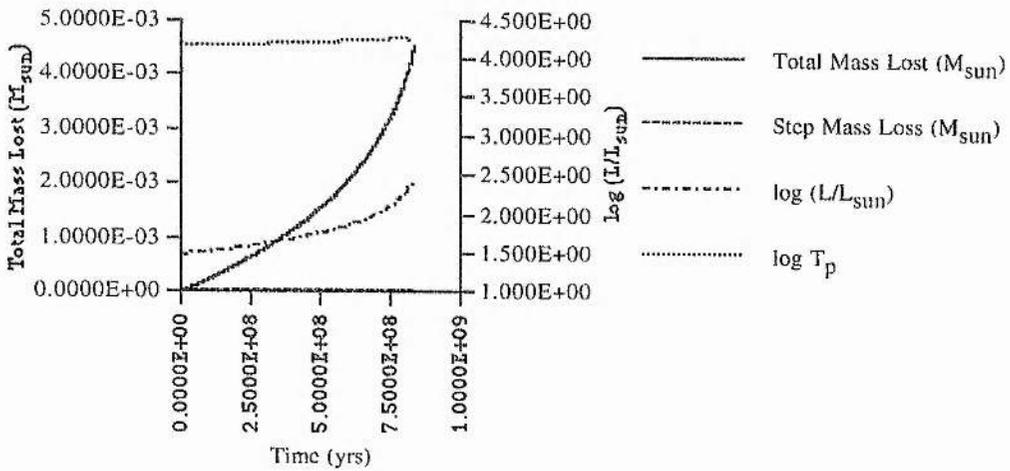
Pop IIIc, $10M_{\text{sun}}$

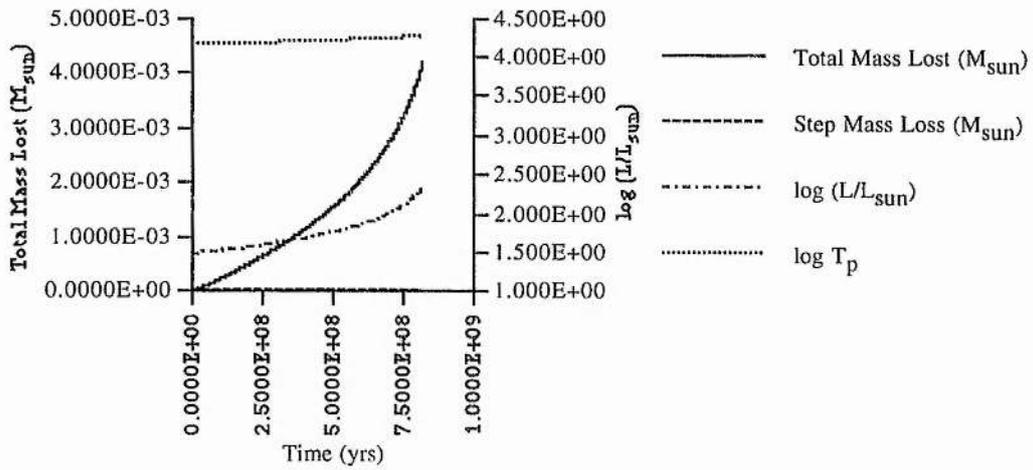


C.7.3 $5M_{\text{sun}}$

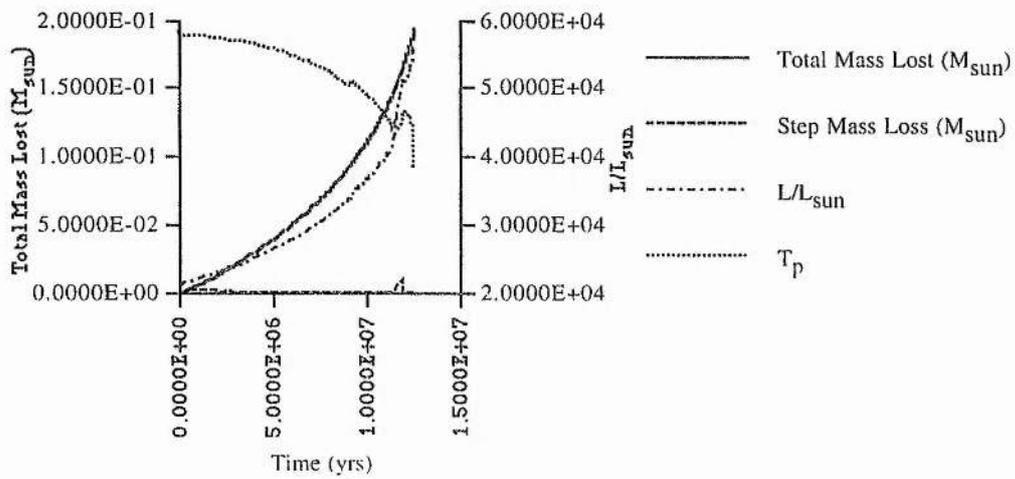
Pop IIIa, $5M_{\text{sun}}$

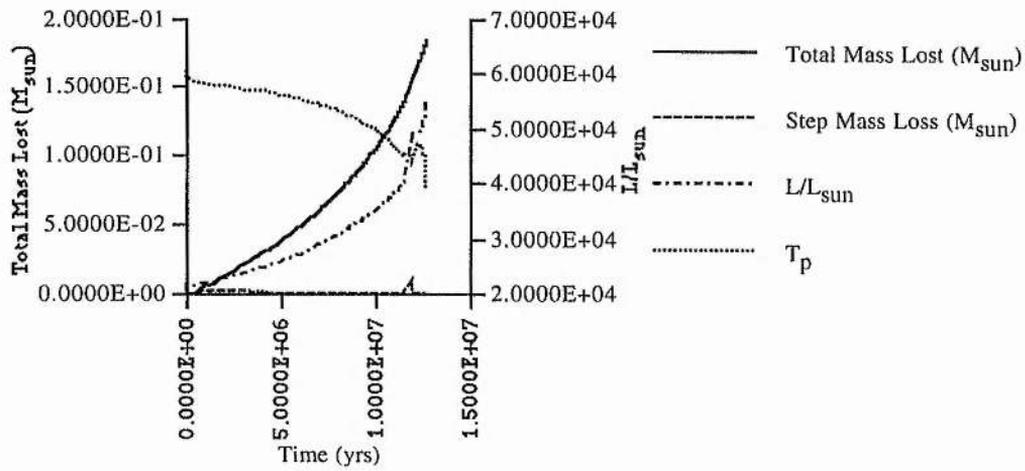
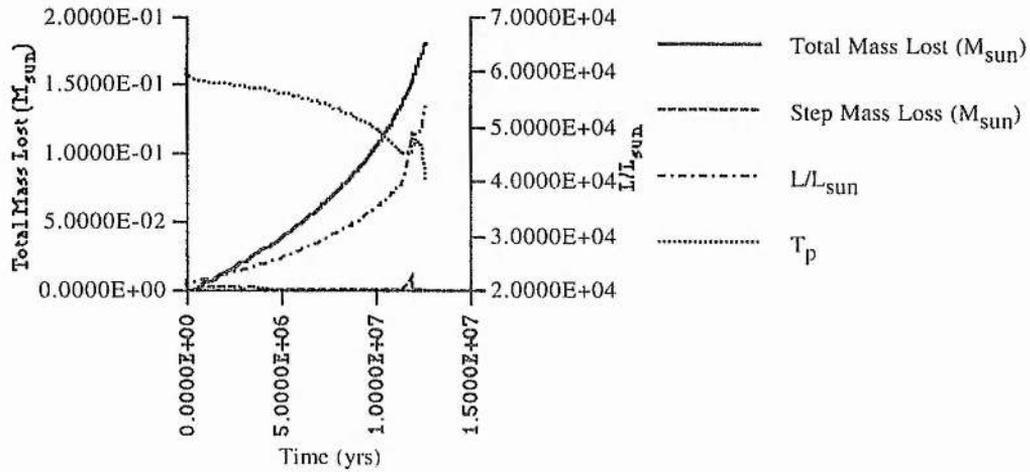
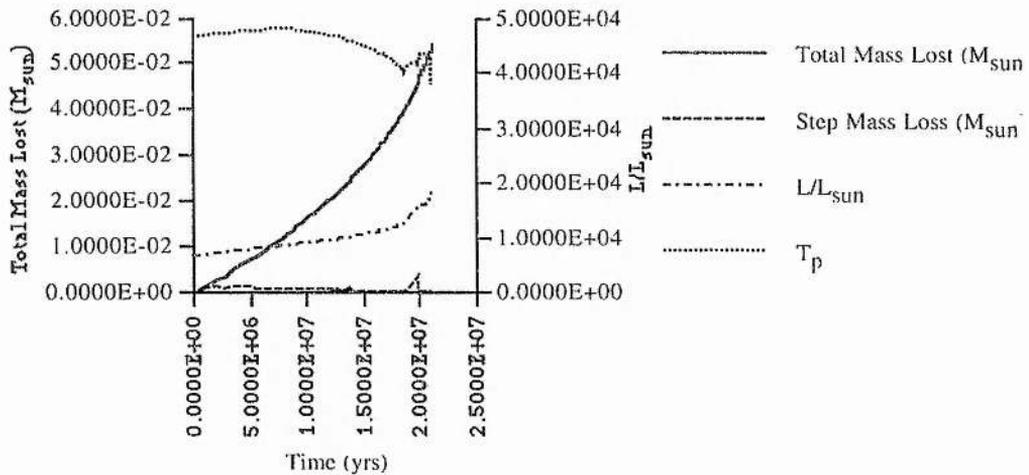


Pop IIIb, $5M_{\text{sun}}$ C.7.4 $2M_{\text{sun}}$ Pop IIIa, $2M_{\text{sun}}$ Pop IIIb, $2M_{\text{sun}}$ 

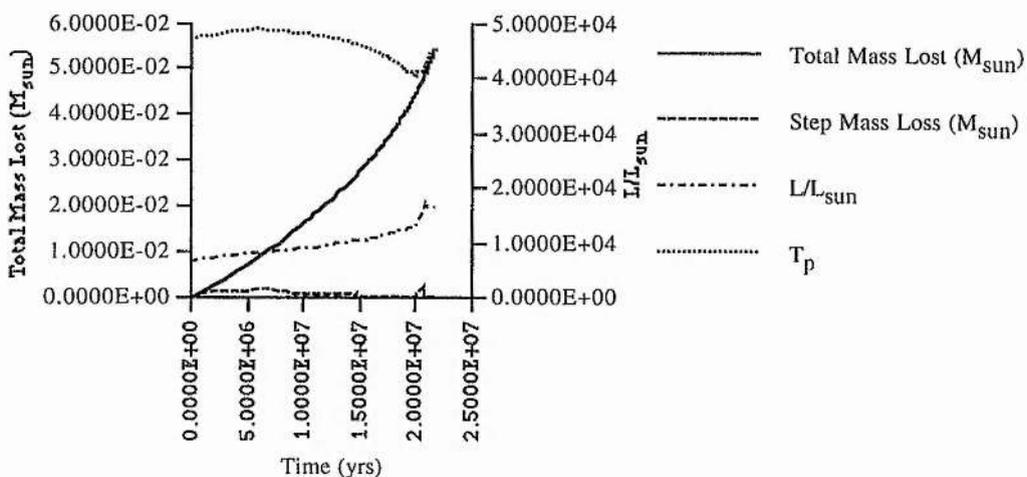
Pop IIIc, $2M_{\text{sun}}$ 

C.8 Mass Loss, Luminosity and Temperature Data, Amended Code

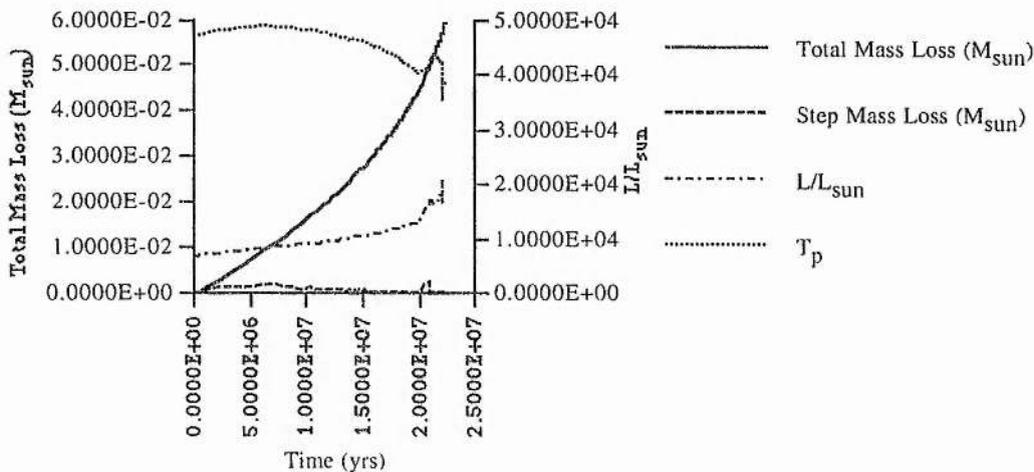
C.8.1 $15M_{\text{sun}}$ Pop IIIa, $15M_{\text{sun}}$ [Amended Code]

Pop IIIb, $15M_{\text{sun}}$ [Amended Code]Pop IIIc, $15M_{\text{sun}}$ [Amended Code]C.8.2 $10M_{\text{sun}}$ Pop IIIa, $10M_{\text{sun}}$ [Amended Code]

Pop IIIb, $10M_{\text{sun}}$ [Amended Code]

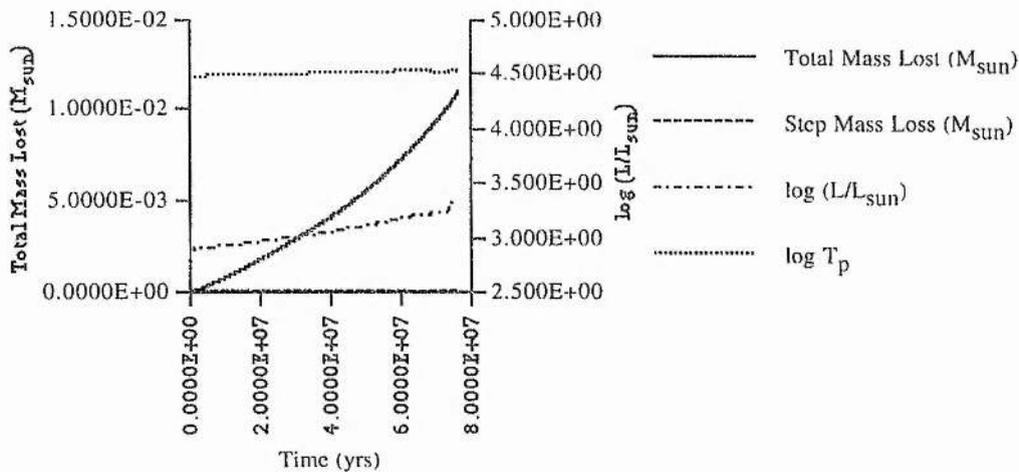


Pop IIIc, $10M_{\text{sun}}$ [Amended Code]

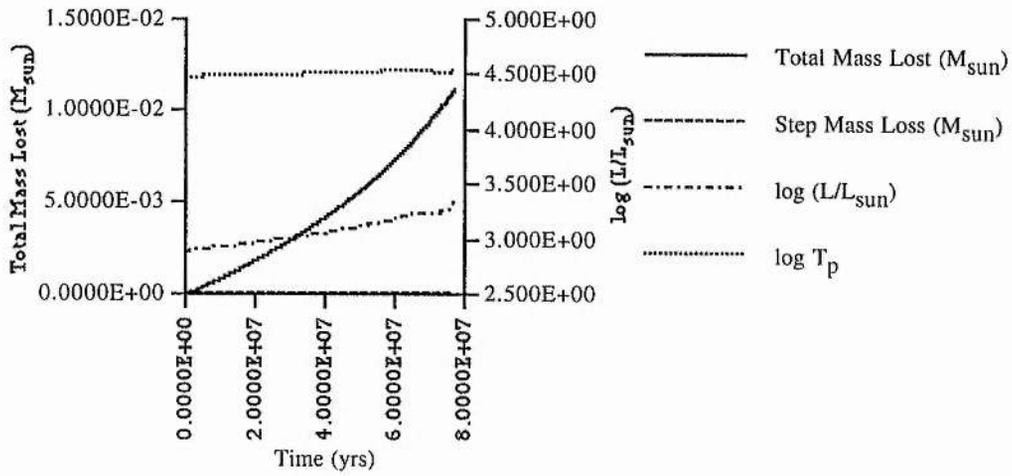


C.8.3 $5M_{\text{sun}}$

Pop IIIa, $5M_{\text{sun}}$ [Amended Code]

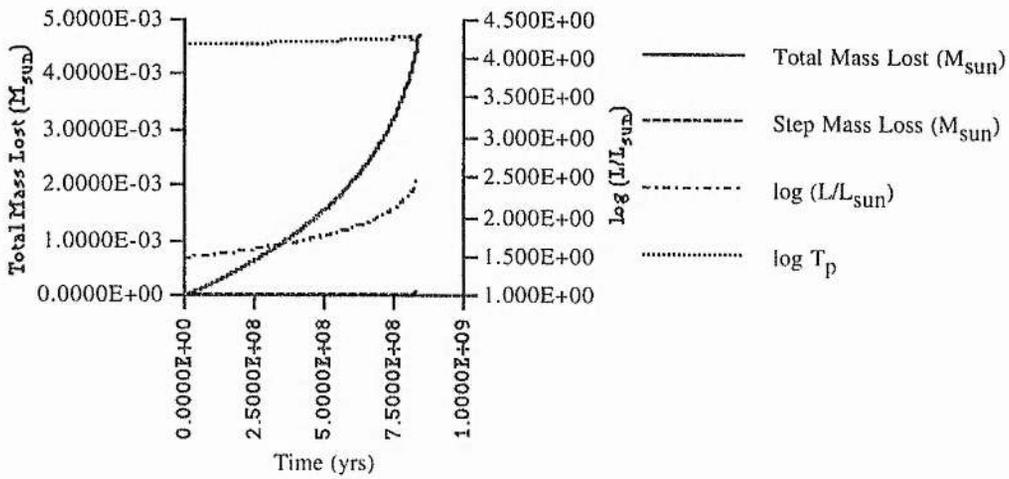


Pop IIIb, $5M_{\text{sun}}$ [Amended Code]

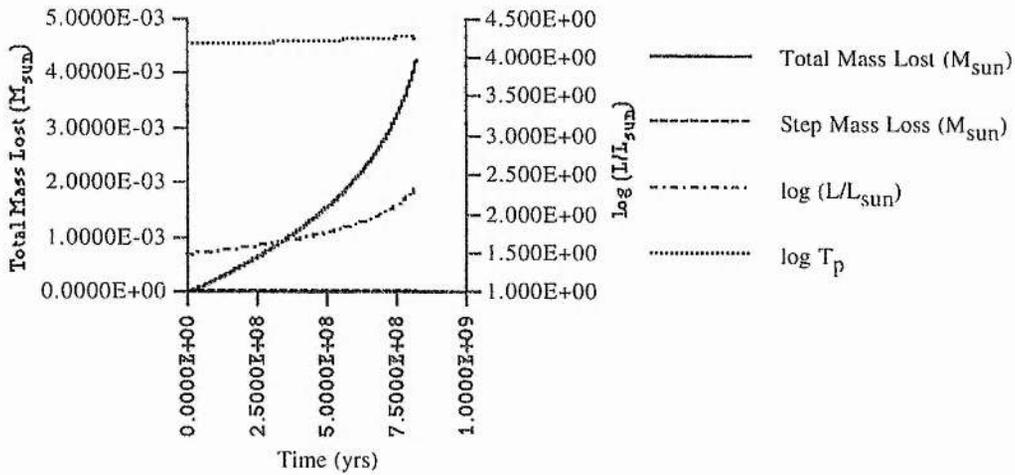


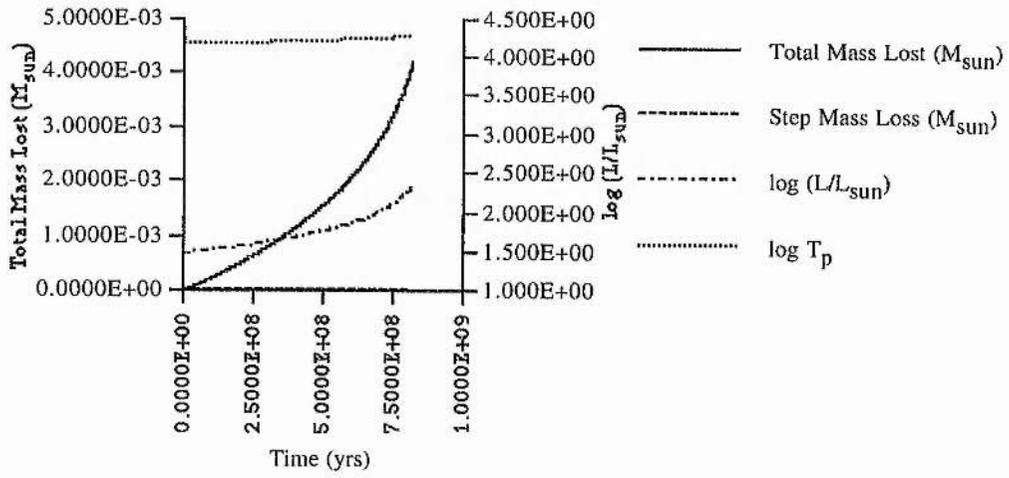
C.8.4 $2M_{\text{sun}}$

Pop IIIa, $2M_{\text{sun}}$ [Amended Code]



Pop IIIb, $2M_{\text{sun}}$ [Amended Code]



Pop IIIc, $2M_{\text{sun}}$ [Amended Code]

Appendix D

Listings of the TRC Code and Others

D.1 The TRC Code

'mlevel.f'

```

C HENYEV MAIN PROGRAM
  IMPLICIT REAL*8 (A-H,O-Z)
  WRITE(6,60)
60 FORMAT(' STELLAR STRUCTURE AND EVOLUTION - TRC/KWH CODE',/)
  READ (5,55)KIN,KCM,KRZ,KDE,KSC,KPR,NDS1,NDS2,NDS3,KEV
  WRITE(6,61)
61 FORMAT(' KIN KCM KRZ KDE KSC KPR NDS1 NDS2 NDS3 KEV')
  WRITE(6,55)KIN,KCM,KRZ,KDE,KSC,KPR,NDS1,NDS2,NDS3,KEV
55 FORMAT(10I5)
  READ (5,56)STARMS,FSCM,FSCL,FDVM,EPSD
  WRITE(6,62)
62 FORMAT(' STARMS FSCM FSCL FDVM EPSD')
  WRITE(6,56)STARMS,FSCM,FSCL,FDVM,EPSD
  READ (5,56)DELTY,XIN,YIN,ZIN,RMIX
  WRITE(6,63)
63 FORMAT(' DELTY XIN YIN ZIN RMIX')
  WRITE(6,56)DELTY,XIN,YIN,ZIN,RMIX
56 FORMAT(1P8E10.3)
C EXTRA PARAMETERS INCLUDED BY PDT
  READ(5,6100)TMAX,MLOSS
  WRITE(6,6110)
6110 FORMAT(' TMAX MLOSS')
  WRITE(6,6100)TMAX,MLOSS
6100 FORMAT(1PE10.3,2I5)
  STARMO=STARMS
  DO 1 I=1,1
    STARMS=STARMO*(1.0+0.25*(I-1))
    WRITE(6,64)STARMS
64 FORMAT('/ MODEL STAR MASS = ',1PE10.3/)
C OPEN STATEMENTS FOR PORTABLE OUTPUT FILES
  OPEN(7,FILE="fort.7",FORM="FORMATTED")
  OPEN(8,FILE="fort.8",FORM="FORMATTED")
  CALL HENYEV(KIN,KCM,KRZ,KDE,KSC,KPR,NDS1,NDS2,NDS3,KEV,
  1STARMS,FSCM,FSCL,FDVM,EPSD,DELTY,XIN,YIN,ZIN,RMIX,
  2TMAX,MLOSS)
1 CONTINUE
  STOP
  END
C HENYEV STELLAR EVOLUTION CODE (KWH)
  SUBROUTINE HENYEV(KIN,KCM,KRZ,KDE,KSC,KPR,NDS1,NDS2,NDS3,KEV,
  1 STARMS,FSCM,FSCL,FDVM,EPSD,DELTY,XIN,YIN,ZIN,RMIX,
  2 TMAX,MLOSS)
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION SC(2),DSC(4,2),CC(4),DCC(6,4),CS(5,2),DEL(512)
  DIMENSION DDELJ(4),DDELK(4),DV(4),DVMIN(8),DVMAX(8),IDEA(4,4)
  DIMENSION HJ(4),DHJ(4,4),HK(4),DHK(4,4),GJK(4),DGJ(4,4),DGK(4,4)
  DIMENSION U(2048),V(2048),W(2048),AM(10,10),BM(10,10),SV(512,20)
  DIMENSION VMC(512),VRC(512),VLC(512),VPC(512),VTC(512)
  DIMENSION VDC(512),VUC(512),VSC(512),VKC(512),VEC(512)
  DIMENSION VGC(512),VVC(512),VXC(512),VYC(512),VZC(512)
  DIMENSION AZC(512,8),EC1(512),EC2(512),EC3(512),AEC(512,8)
  DIMENSION PCM(512),TCM(512),DCM(512),VCM(512),ECM(512)
  DIMENSION VCP(512),VDT(512),VDA(512)
  DIMENSION VMB(512),VRB(512),VLB(512),VPB(512),VTB(512)
  DIMENSION VDB(512),VUB(512),VSB(512),VKB(512),VEB(512)
  DIMENSION VGB(512),VVB(512),VXB(512),VYB(512),VZB(512)
  DIMENSION AZB(512,8),EB1(512),EB2(512),EB3(512),AEB(512,8)
  DIMENSION PBM(512),TBM(512),DBM(512),VBM(512),EBM(512)
  DIMENSION DVR(512),DVL(512),DVP(512),DVT(512)
  DIMENSION BMM(512),BRM(512),BAM(512),BVM(512),BPM(512),BGM(512)
  DIMENSION ZVM(512),ZPM(512),ZGM(512),ZDM(512),BDM(512),BEF(512)
  DIMENSION KAT(3),VMS(3),VRS(3),VLS(3),VPS(3),VTS(3),VDS(3)
  DIMENSION SP(512,20),DUCON(4),ZAC(8),AZM(8),AMM(8),QAL(8),EAC(8)
  EQUIVALENCE (SP(1, 1),VMC(1)),(SP(1, 2),VRC(1)),(SP(1, 3),VLC(1))
  EQUIVALENCE (SP(1, 4),VPC(1)),(SP(1, 5),VTC(1)),(SP(1,13),VDC(1))
  EQUIVALENCE (SP(1,14),VUC(1)),(SP(1,15),VSC(1)),(SP(1, 9),VGC(1))
  EQUIVALENCE (SP(1,10),VEC(1)),(SP(1,11),VKC(1)),(SP(1,12),DEL(1))
  EQUIVALENCE (SP(1, 6),VXC(1)),(SP(1, 7),VYC(1)),(SP(1, 8),VZC(1))
  EQUIVALENCE (SP(1,16),EC1(1)),(SP(1,17),EC2(1)),(SP(1,18),EC3(1))
  EQUIVALENCE (SP(1,19),VVC(1))

```

```

EQUIVALENCE (AZM(1),AZC),(AZM(2),AZO),(AZM(3),AZNE),(AZM(4),AZMG)
EQUIVALENCE (AZM(5),AZSI),(AZM(6),AZS),(AZM(7),AZAR),(AZM(8),AZCA)
EQUIVALENCE (AMM(1),AMC),(AMM(2),AMO),(AMM(3),AMNE),(AMM(4),AMMG)
EQUIVALENCE (AMM(5),AMSI),(AMM(6),AMS),(AMM(7),AMAR),(AMM(8),AMCA)
EQUIVALENCE (QAL(1),QCAO),(QAL(2),QOANE),(QAL(3),QNEMG),(QAL(4),QMGS1)
1QMGS1),(QAL(5),QSIAS),(QAL(6),QSAAR),(QAL(7),QARCA),(QAL(8),QCAXX)
C EQUIVALENCE (QAL(5),QSIAS),(QAL(6),QSAAR),(QAL(7),QARCA),(QAL(8),QCAXX)
EQUIVALENCE (EAC(1),ECAO),(EAC(2),EOANE),(EAC(3),ENEMG),(EAC(4),EMGSI)
1EMGSI),(EAC(5),ESIAS),(EAC(6),ESAAR),(EAC(7),EARCA),(EAC(8),ECAXX)
C EQUIVALENCE (EAC(5),ESIAS),(EAC(6),ESAAR),(EAC(7),EARCA),(EAC(8),ECAXX)
C ENUC RETURNS ECCMG,ECOSI,EOOS IN EAC(6),EAC(7),EAC(8)
EQUIVALENCE (EAC(6),ECCMG),(EAC(7),ECOSI),(EAC(8),EOOS)
ABS(X)=DABS(X)
EXP(X)=DEXP(X)
ALOG(X)=DLOG(X)
ALOG10(X)=DLOG10(X)
SQRT(X)=DSQRT(X)
AMAX1(X1,X2)=DMAX1(X1,X2)
  FG(K,L,M,N)=K*((VLS(M)-VLS(N))*(VTK-VTS(M))-
1 (VTS(M)-VTS(N))*(VLK-VLS(M)))
DATA PIE/3.1415926536/,ETA/1.0E-3/,YEARS/3.155693E7/
DATA GCM/6.670E-8/,CVL/2.997925E10/,RCA/7.56495E-15/
DATA AVN/6.0222E+23/,EV6/1.6022E-06/
DATA QPPC/13.811/,QCNO/24.970/,QHEC/7.274/
DATA AHY/1.007825/,AHE/4.002603/,AMZ/12.0/
DATA AZM/6.0,8.0,10.0,12.0,14.0,16.0,18.0,20.0/
DATA AMM/12.0,16.0,20.0,24.0,28.0,32.0,36.0,40.0/
DATA QAL/7.162,4.730,9.315,9.986,4*0.000/
DATA QCCMG/13.931/,QCOSI/16.754/,QOOS/16.541/
DATA SUNM/1.989E33/,SUNR/6.960E10/,SUNL/3.900E33/
DATA DVMIN/0.020,0.020,0.020,0.020,0.020,0.020,0.020,0.020/
DATA DVMAX/0.100,0.100,0.100,0.150,0.025,0.100,0.100,0.100/
DATA INIT/1/,IDEA/1,1,1,1,2,2,2,2,3,3,3,3,2,1,2,2/
DATA NV/8/,JTS/1/,MAT/64/,IRP/1/,KSR/8/,KSM/20/,INAT/0/,IMP/0/
DATA ZERO/0.0/,ONE/1.0/,TWO/2.0/,THREE/3.0/,FOUR/4.0/,TEN/10.0/
DATA ADG/0.4/,ALPHA/1.0/,DVC/0.25/,DELT/0.1/,DELL/0.1/
DATA D9/9.0/,D19/19.0/,D27/27.0/,D368/368.0/,D729/729.0/
DATA D4672/4672.0/,D19683/19683.0/
DATA XYZ/0.05/,DCP/0.0/,DCT/0.0/,DDP/0.0/,DDT/0.0/,DDELA/0.0/
DATA FDT/1.0/,ALE/1.0/,ALP/1.0/,ALT/1.0/
DATA LMV/0/,NDV/2/,KSP/-1/,MNZ/512/,NMZ/8/
DATA ZFC12/0.4/,ZFCO16/0.6/

C VERSION II
C DATA ZFC12/0.3/,ZFO16/0.6/,ZFN14/0.1/

KKP=KPR/10
KPR=MOD(KPR,10)
NDS4=NDS3+1
NDS5=NDS3+2
NDSO=NDS3
EPSDI=EPSD
GSUN=GCM*SUNM/SUNR/SUNR
CC1=FOUR/D9
CC2=D19/D27
CC3=D368/D729
CC4=D4672/D19683
UCONC=TWO*THREE*SQRT(TWO)*CVL*RCA
KSUV=MOD(KEV,10)
INEV=KEV/10
IF(KEV.GT.9)THEN
KEV=KEV-10
END IF
ALOG2=ALOG(TWO)
ALOG3=ALOG(THREE)
TBSL=ALOG3-FOUR*ALOG2
ALOGP=ALOG(PIE)
FPL=TWO*ALOG2+ALOGP
FPBTL=FPL-ALOG3
GCL=ALOG(GCM)
GFPL=GCL-FPL
RCAL=ALOG(RCA)
PRCL=RCAL-ALOG3
ACM=RCA*CVL
ACL=ALOG(ACM)
PACL=ALOG(PIE*RCA*CVL)
STARM=STARMS*SUNM

```

```

STARML=ALOG(STARM)
TTGL=ALOG2-ALOG3+GCL
SCMM=FSCM*STARM
SCML=ALOG(SCMM)
SMPM=(ONE+ETA)*SCMM
SMPL=ALOG(SMPM)
SCLM=FSCL*SUNL
SCLL=ALOG(SCLM)
DELA=ADG
DTYMAX=TMAX
MDS1=IABS(NDS1)
IF(KIN-1)100,100,200
100 CONTINUE
IF(MDS1-10)101,101,102
101 CONTINUE
IF(NDS1)103,103,104
103 CONTINUE
C START BY MAKING A SIMPLE MODEL
WRITE(6,*)' HENYEV : MAKE A SIMPLE MODEL WITH PARAMETER ',MDS1
SMS=STARMS
SRS=0.0
NP=MDS1
CALL MODEL(NP,NB,SMS,SRS,SV)
NA=0
C I SUSPECT THIS VARIABLE NAME SHOULD BE CHANGED TO AVOID
C PROBLEMS LATER ON WITH THE OTHER USE OF SMASS
SMASS=SMS*SUNM
GO TO 105
104 CONTINUE
C START BY MAKING A LANE-EMDEN MODEL
WRITE(6,*)' HENYEV : INTEGRATE A LANE-EMDEN MODEL OF INDEX ',MDS1
SMS=STARMS
SRS=0.0
NP=MDS1
NP=MIN(4,NP)
CALL LEMOD(NP,NB,SMS,SRS,SV)
NA=0
SMASS=SMS*SUNM
GO TO 105
102 CONTINUE
IF(NDS1)116,117,118
117 STOP 117
116 CONTINUE
NDS1=MDS1
C START BY INTEGRATING AN ATMOSPHERE/ENVELOPE/CORE
WRITE(6,*)' HENYEV : INTEGRATE ATMOSPHERE/ENVELOPE/CORE'
STARMSL=LOG10(STARMS)
STARLSL=3.0*STARMSL-0.3
TEFFL=0.5*STARMSL+3.7
C CALL MODENV(NDS1,NDS2,KPR,STARMS,STARLSL,TEFFL,XIN,YIN,ZIN)
KRAD=1
CALL MODENV(KRAD,NDS1,NDS2,KPR,STARMS,STARLSL,TEFFL,XIN,YIN,ZIN)
118 CONTINUE
WRITE(6,*)' HENYEV : GET MODEL FROM DATA FILE ',NDS1
READ (NDS1 )NB,NA,SMASS,STARRS,STARL,GP,TP,DP,PP,TC,DC,PC,
1X,Y,Z
WRITE( 6,160)NB,NA,SMASS,STARRS,STARL,GP,TP,DP,PP,TC,DC,PC,
1X,Y,Z
105 CONTINUE
JB=0
DO 110 IB=1,NB
IF(MDS1-10)111,111,112
112 CONTINUE
READ (NDS1 )LB,KB,(SV(IB,J),J=1,NV)
IF(KPR-1)113,113,114
114 CONTINUE
WRITE(6,160)LB,KB,(SV(IB,J),J=1,NV)
113 CONTINUE
160 FORMAT(2I5,(T11,1P10E10.3))
111 CONTINUE
IF(IB-NA)110,115,115
C105 GO TO 110
115 CONTINUE
JB=JB+1
KB=JB
IF(IB-NB)120,130,110
130 VMC(KB)=ZERO

```

```

VRC(KB)=-TEN*TEN
VLC(KB)=ZERO
GO TO 125
120 VMC(KB)=ALOG(ONE-SV(IB,1)/SMASS/(ONE+ETA))
C120 VMC(KB)=ALOG(ONE-SV(IB,1)*(STARMS/SMASS)/SMPM)
VRC(KB)=ALOG(SV(IB,2))
VLC(KB)=ALOG(ONE+SV(IB,3)/SCLM)
125 VTC(KB)=ALOG(SV(IB,4))
VPC(KB)=ALOG(SV(IB,5))
VDC(KB)=ALOG(SV(IB,6))
C VKC(KB)=ALOG(SV(IB,7))
C VEC(KB)=ALOG(SV(IB,8))
VXC(KB)=XIN
VYC(KB)=YIN
VZC(KB)=ZIN
AZC(KB,1)=ZIN*ZFC12
AZC(KB,2)=ZIN*ZFO16
DO 126 K=3,NMZ
AZC(KB,K)=0.0
126 CONTINUE
IF(KPR-1)121,121,122
122 CONTINUE
WRITE(6,160)IB,KB,VMC(KB),VRC(KB),VLC(KB),VTC(KB),VPC(KB),VDC(KB)
121 CONTINUE
110 CONTINUE
IF(NDS1-10)131,131,132
132 CONTINUE
REWIND NDS1
131 CONTINUE
C VEC(1)=VEC(2)
MB=JB
IST=0
ITS=0
DTY=ZERO
TTY=ZERO
RDT=ZERO
IRZ=1
C SET IRZ=1 TO FORCE REZONING WHEN KIN=1
GO TO 300
200 IF(NDS2)260,260,265
265 WRITE(6,*) INPUT MODEL FROM DATA SET NUMBER ',NDS2
READ ( NDS2)MB,ITS,DYT,TFY,SMASSI,SMASS,STARRS,STARL,
1 TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
2 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,
3 (ZAC(J),J=1,NMZ)
WRITE(6,163)
WRITE(6,161)ITS,MB,DYT,TTY,SMASSI,SMASS,STARRS,STARL,
1 TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
2 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,
3 (ZAC(J),J=1, 6)
DO 270 IB=1,MB
READ ( NDS2) VMC(IB),VRC(IB),VLC(IB),VPC(IB),VTC(IB),
1 VDC(IB),VUC(IB),VSC(IB),VKC(IB),VEC(IB),VGC(IB),VVC(IB),
2 VXC(IB),VYC(IB),VZC(IB),(AZC(IB,JB),JB=1,NMZ)
IF(KEV)275,275,270
275 VXC(IB)=XIN
VYC(IB)=YIN
VZC(IB)=ZIN
AZC(IB,1)=ZIN*ZFC12
AZC(IB,2)=ZIN*ZFO16
DO 246 JB=3,NMZ
AZC(IB,JB)=0.0
246 CONTINUE
270 CONTINUE
IF(KEV.EQ.0) THEN
SMASS=STARMS
END IF
STARM=SMASS*SUNM
STARML=ALOG(STARM)
SCMM=FSCM*STARM
SCML=ALOG(SCMM)
SMPM=(ONE+ETA)*SCMM
SMPL=ALOG(SMPM)
TMLOST=SMASSI-SMASS
C VRC(MB)=-TEN*TEN
REWIND NDS2
IRZ=MOD(KRZ,10)

```

```

C   FOR KIN=2 SUBSEQUENT REZONING DEPENDS ON KRZ
C   IF(STARMS-SMASS)285,280,285
  IF(ITS)280,280,285
280 CONTINUE
260 CONTINUE
  IST=0
  RDT=ZERO
C   GO TO 300
  GO TO 295
285 CONTINUE
  IST=1
C   DYT IS THE LAST TIME STEP USED
  IF(DYT)295,295,290
290 RDT=ONE/DELTY/YEARS
295 CONTINUE
300 CONTINUE
  ITC=0
  RDTP=RDT
C   USE KCM TO ADD ADDITIONAL MASS ZONES AT CENTRE
  K=0
910 IF(KCM)915,925,920
915 IF((ONE-EXP(VMC(MB-1)))/(ONE-EXP(VMC(1)))-TEN**KCM)925,925,930
920 IF(K-KCM)930,925,925
930 IF(K)935,940,935
940 WRITE(6,*) HENYEY : ADDING ZONES AT CENTRE '
935 K=K+1
  VMC(MB+1)=VMC(MB)
  VRC(MB+1)=VRC(MB)
  VLC(MB+1)=VLC(MB)
  VPC(MB+1)=VPC(MB)
  VTC(MB+1)=VTC(MB)
  VDC(MB+1)=VDC(MB)
  VXC(MB+1)=VXC(MB)
  VYC(MB+1)=VYC(MB)
  VZC(MB+1)=VZC(MB)
  VMC(MB)=VMC(MB-1)/TWO
  VRC(MB)=VRC(MB-1)-ALOG2/THREE
  VLC(MB)=VLC(MB-1)/TWO
  VPC(MB)=(VPC(MB-1)+VPC(MB+1))/TWO
  VTC(MB)=(VTC(MB-1)+VTC(MB+1))/TWO
  VDC(MB)=(VDC(MB-1)+VDC(MB+1))/TWO
  VXC(MB)=(VXC(MB-1)+VXC(MB+1))/TWO
  VYC(MB)=(VYC(MB-1)+VYC(MB+1))/TWO
  VZC(MB)=(VZC(MB-1)+VZC(MB+1))/TWO
DO 936 J=1,NMZ
  AZC(MB+1,J)=AZC(MB,J)
  AZC(MB,J)=(AZC(MB-1,J)+AZC(MB+1,J))/TWO
936 CONTINUE
  WRITE(6,160)K,MB,VMC(MB),VRC(MB),VLC(MB),
  1VTC(MB),VPC(MB),VDC(MB),VXC(MB),VYC(MB),VZC(MB)
  MB=MB+1
  GO TO 910
925 CONTINUE
C   WRITE(6,160)K,MB,VMC(MB),VRC(MB),VLC(MB),
C   1VTC(MB),VPC(MB),VDC(MB),VXC(MB),VYC(MB),VZC(MB)
C   INITIAL MODEL - STATIC OR EVOLVING - NOW AVAILABLE
C   REZONE MODEL ACCORDING TO VALUE OF FDVM AND LIMITS DVMIN, DVMAX
  MBP=MB
C   CALL REZONE(KRZ,MB,SP,SV)
  MTOT=0
  IF(IRZ)999,999,994
994 CONTINUE
  WRITE(6,*) HENYEY : REZONE MODEL WITH FACTOR ',FDVM
998 JB=0
  NTOT=0
  NMIN=0
  NMAX=0
  NWGS=0
  DO 950 IB=1,MB
  JB=JB+1
  DO 951 J=1,KSM
951 SV(JB,J)=SP(IB,J)
  DO 961 J=1,NMZ
961 AZB(JB,J)=AZC(IB,J)
  IF(IB-MB)990,950,999
990 CONTINUE
  IF(IB-1)950,950,952

```

```

952 KMIN=0
    KMAX=0
    DO 953 J=1,KSR
        ADSV=ABS(SV(JB,J))-SV(JB-1,J)
        IF(ADSV-FDVM*DVMIN(J))954,954,955
954 KMIN=KMIN+1
    GO TO 953
955 IF(ADSV-FDVM*DVMAX(J))953,956,956
956 KMAX=KMAX+1
953 CONTINUE
    IF(KMIN-KSR)960,965,999
965 NTOT=NTOT+1
    NMIN=NMIN+1
    JB=JB-1
    GO TO 950
960 IF(KMAX-1)950,950,970
970 IF(MB+JB-1B-MNZ)985,975,975
975 NWGS=NWGS+1
    GO TO 950
985 NTOT=NTOT+1
    NMAX=NMAX+1
    DO 980 J=1,KSM
        SV(JB+1,J)=SV(JB,J)
980 SV(JB,J)=(SV(JB-1,J)+SV(JB+1,J))/TWO
    DO 981 J=1,NMZ
        AZB(JB+1,J)=AZB(JB,J)
981 AZB(JB,J)=(AZB(JB-1,J)+AZB(JB+1,J))/TWO
    JB=JB+1
950 CONTINUE
    MB=JB
    WRITE(6,165)MBP,NMIN,NMAX,NTOT,MB,NWGS
165 FORMAT(8I10)
    DO 995 IB=1,MB
    DO 991 J=1,KSM
991 SP(IB,J)=SV(IB,J)
    DO 992 J=1,NMZ
992 AZC(IB,J)=AZB(IB,J)
    IF(NTOT)995,997,995
997 IF(KPR-2)995,995,996
996 WRITE(6,160)MB,IB,VMC(IB),VRC(IB),VLC(IB),VPC(IB),VTC(IB)
995 CONTINUE
    MTOT=MTOT+NTOT
    IF(NTOT)998,999,998
999 CONTINUE
    IRZ=MOD(KRZ,10)
C DURING EVOLUTION SUBSEQUENT REZONING DEPENDS ON KRZ
    IF(IST)310,310,320
310 CONTINUE
    IST=1
    RDT=ZERO
    GO TO 390
320 ITS=ITS+1
C BEGIN NEW TIME STEP BY BACKING UP MODEL
    DO 340 IB=1,MB
        VMB(IB)=VMC(IB)
        VRB(IB)=VRC(IB)
        VLB(IB)=VLC(IB)
        VPB(IB)=VPC(IB)
        VTB(IB)=VTC(IB)
        VDB(IB)=VDC(IB)
        VUB(IB)=VUC(IB)
        VSB(IB)=VSC(IB)
C K,E,G,V ARE NOT BACKED UP
C VKB(IB)=VKC(IB)
C VEB(IB)=VEC(IB)
C VGB(IB)=VGC(IB)
C VVB(IB)=VVC(IB)
C IN CASE MIXING HAS OCCURRED
C OR MASS LOSS
C OR REZONING
    DO 335 JB=1,NMZ
335 ZAC(JB)=AZC(IB,JB)
    CALL ENUCZT(VTC(IB),VDC(IB),VXC(IB),VYC(IB),VZC(IB),ZAC,
    1EC1(IB),EC2(IB),EC3(IB),EAC,VEC(IB),DET,DED)
    DO 339 JB=1,NMZ
339 AEC(IB,JB)=EAC(JB)
    ECM(IB)=EXP(VEC(IB))

```

```

C   E NOW BACKED UP
   VEB(IB)=VEC(IB)
C   IF NO PREVIOUS MODEL IN MEMORY AND/OR NEW ZONES ADDED
C   THE MAGNITUDES OF CERTAIN VARIABLES MUST BE EVALUATED
   IF(INIT+MTOT)340,350,360
360 CONTINUE
C   COMPUTE MAGNITUDES FOR P,T,D,V,E
   PCM(IB)=EXP(VPC(IB))
   TCM(IB)=EXP(VTC(IB))
   DCM(IB)=EXP(VDC(IB))
   VCM(IB)=ONE/DCM(IB)
   ECM(IB)=EXP(VEC(IB))
350 CONTINUE
   VXB(IB)=VXC(IB)
   VYB(IB)=VYC(IB)
   VZB(IB)=VZC(IB)
   EB1(IB)=EC1(IB)
   EB2(IB)=EC2(IB)
   EB3(IB)=EC3(IB)
   DO 345 JB=1,NMZ
   AZB(IB,JB)=AZC(IB,JB)
   AEB(IB,JB)=AEC(IB,JB)
345 CONTINUE
   PBM(IB)=PCM(IB)
   TBM(IB)=TCM(IB)
   DBM(IB)=DCM(IB)
   VBM(IB)=VCM(IB)
   EBM(IB)=ECM(IB)
340 CONTINUE
   INIT=0
   GO TO 355
C   RECOVER BACKED UP MODEL WITH VIEW TO CHANGING TIME STEP
330 DO 380 IB=1,MB
   VMC(IB)=VMB(IB)
   VRC(IB)=VRB(IB)
   VLC(IB)=VLB(IB)
   VPC(IB)=VPB(IB)
   VTC(IB)=VTB(IB)
   VDC(IB)=VDB(IB)
   VUC(IB)=VUB(IB)
   VSC(IB)=VSB(IB)
C   VKC(IB)=VKB(IB)
   VEC(IB)=VEB(IB)
C   VGC(IB)=VGB(IB)
   VVC(IB)=VVB(IB)
   VXC(IB)=VXB(IB)
   VYC(IB)=VYB(IB)
   VZC(IB)=VZB(IB)
   EC1(IB)=EB1(IB)
   EC2(IB)=EB2(IB)
   EC3(IB)=EB3(IB)
   DO 385 JB=1,NMZ
   AZC(IB,JB)=AZB(IB,JB)
   AEC(IB,JB)=AEB(IB,JB)
385 CONTINUE
C   P,T,D,V,E MAGNITUDES NOW RECOVERED FROM BACK-UP
   PCM(IB)=PBM(IB)
   TCM(IB)=TBM(IB)
   DCM(IB)=DBM(IB)
   VCM(IB)=VBM(IB)
   ECM(IB)=EBM(IB)
380 CONTINUE
355 CONTINUE
C   DTY=FDI*DELTY
C   DTS=DTY*YEARS
C   RDT=ONE/DTS
C   DTEAVN=DTS/EV6/AVN
390 CONTINUE
499 CONTINUE
C   BEGIN NEW CYCLE OF ITERATIONS
   ITC=ITC+1
   IF(ITC-10)394,394,396
394 EPSD=EPSDI
   GO TO 395
396 IF(EPSD-8.0*EPSDI)397,397,395
397 EPSD=2.0*EPSD
   WRITE(6,*) RELAX CONVERGENCE, CONTINUE EXECUTION : EPSD =,EPSD

```

```

GO TO 395
399 STOP 399
395 CONTINUE
C BEGIN SERIES OF ITERATIONS
  ITN=0
  THETA=ONE
400 CONTINUE
C START A NEW ITERATION
  ITN=ITN+1
  IF(ITS)1400,1400,1401
C1401 IF(IMP)1400,1402,1403
1401 IF(ITN-1)1403,1403,1400
C1402 IF(ITN-1)1400,1403,1400
1403 IF(KEV)1400,1400,1405
1405 DXYZ=ZERO
  DTY=FDT*DELTY
  DTY=MIN(DTY,DTYMAX)
  FDT=DTY/DELTY
  DTS=DTY*YEARS
  RDT=ONE/DTS
  DTEA=DTS/EV6/AVN
C USE NUCLEAR REACTION RATES TO EVALUATE NEW ELEMENT ABUNDANCES
DO 405 IB=1,MB
  EPPC=EB1(IB)
  ECNO=EB2(IB)
  EHEC=EB3(IB)
  DO 2405 JB=1,NMZ
    EAC(JB)=AEB(IB,JB)
2405 CONTINUE
  XNEW=VXB(IB)-DTEA*AHY*(2.0*EPPC/QPPC+4.0*ECNO/QCNO)
  YNEW=VYB(IB)+DTEA*AHE*(0.5*EPPC/QPPC+ ECNO/QCNO-3.0*EHEC/QHEC)
  ZNEW=VZB(IB)+DTEA*AMZ*( EHEC/QHEC)
C ALPHA BURNING
  YNEW=YNEW-DTEA*AHE*
  I(ECAO/QCAO+EOANE/QOANE+ENEMG/QNEMG+EMGSI/QMGSI)
  XNEW=AMAX1(ZERO,XNEW)
  YNEW=AMAX1(ZERO,YNEW)
C ZNEW=AMAX1(ZERO,ZNEW)
C SNEW=XNEW+YNEW+ZNEW
  AZC(IB,1)=AZB(IB,1)+DTEA*AMC*(EHEC/QHEC-ECAO/QCAO)
  AZC(IB,1)=AZC(IB,1)-DTEA*AMC*(2.0*ECCMG/QCCMG)
  AZC(IB,2)=AZB(IB,2)+DTEA*AMO*(ECAO/QCAO+EOANE/QOANE)
  AZC(IB,2)=AZC(IB,2)-DTEA*AMO*(ECOSI/QCOSI+2.0*EOOS/QOOS)
  AZC(IB,3)=AZB(IB,3)+DTEA*AMNE*(EOANE/QOANE-ENEMG/QNEMG)
  AZC(IB,4)=AZB(IB,4)+DTEA*AMMG*(ENEMG/QNEMG-EMGSI/QMGSI)
  AZC(IB,4)=AZC(IB,4)+DTEA*AMMG*ECCMG/QCCMG
  AZC(IB,5)=AZB(IB,5)+DTEA*AMSI*(EMGSI/QMGSI)
  AZC(IB,5)=AZC(IB,5)+DTEA*AMSI*ECOSI/QCOSI
C AZC(IB,5)=AZB(IB,5)+DTEA*AMSI*(EMGSI/QMGSI-ESIAS/QSIAS)
C AZC(IB,6)=AZB(IB,6)+DTEA*AMS*(ESIAS/QSIAS-ESAAR/QSAAR)
  AZC(IB,6)=AZB(IB,6)+DTEA*AMS*EOOS/QOOS
C AZC(IB,7)=AZB(IB,7)+DTEA*AMAR*(ESAAR/QSAAR-EARCA/QARCA)
C AZC(IB,8)=AZB(IB,8)+DTEA*AMCA*(EARCA/QARCA)
C AZC(IB,8)=AZB(IB,8)+DTEA*AMCA*(EARCA/QARCA-ECAXX/QCAXX)
  SUMZ=0.0
  DO 3405 J=1,NMZ
    AZC(IB,J)=AMAX1(ZERO,AZC(IB,J))
  SUMZ=SUMZ+AZC(IB,J)
3405 CONTINUE

C VERSION II
C THIS LINE ADDED IN TO KEEP N14 INCLUDED
SUMZ=SUMZ+ZFN14*VZB(IB)

  ZNEW=SUMZ
  SNEW=XNEW+YNEW+ZNEW
  VXC(IB)=XNEW/SNEW
  VYC(IB)=YNEW/SNEW
  VZC(IB)=ZNEW/SNEW
  DO 4405 J=1,NMZ
    AZC(IB,J)=AZC(IB,J)/SNEW
4405 CONTINUE
  DXYZ=AMAX1(DXYZ,ABS(VXC(IB)-VXB(IB)))
  DXYZ=AMAX1(DXYZ,ABS(VYC(IB)-VYB(IB)))
  DXYZ=AMAX1(DXYZ,ABS(VZC(IB)-VZB(IB)))
C LATE BURNING STILL TO BE IMPLEMENTED
405 CONTINUE

```

```

C   KEEP CUTTING TIMESTEP VIA FDT UNTIL BURNING DXYZ DOES NOT EXCEED CXYZ
      KXYZ=0
1408 IF(DXYZ-CXYZ)1406,1406,1407
1407 DXYZ=DXYZ*0.75
      FDT=FDT*0.75
      KXYZ=KXYZ+1
      GO TO 1408
1406 IF(KXYZ)1400,1400,1409
1409 WRITE(6,1410)KXYZ
1410 FORMAT(' BURN TIME CONTROL : KXYZ = ',I2)
      GO TO 330
1400 CONTINUE
      DO 490 IB=1,MB
      IF(IB-1)490,410,420
420 CONTINUE
      VMJ=VMK
      VRJ=VRK
      VLJ=VLK
      VPJ=VPK
      VTJ=VTK
      VDj=VDK
      VUJ=VUK
      VSJ=VSK
      VKJ=VKK
      VEJ=VEK
      DELJ=DELK
      VGJ=VGK
      VXJ=VXK
      VYJ=VYK
      VZJ=VZK
      DO 421 I=1,4
      HJ(I)=HK(I)
      DDELJ(I)=DDELK(I)
      DO 422 L=1,4
      DHJ(L,I)=DHK(L,I)
422 CONTINUE
421 CONTINUE
410 CONTINUE
      JB=IB
      VMK=VMC(JB)
      VRK=VRC(JB)
      VLK=VLC(JB)
      VPK=VPC(JB)
      VTK=VTC(JB)
      VDK=VDC(JB)
C   VUK=VUC(JB)
C   VSK=VSC(JB)
C   VKK=VKC(JB)
C   VEK=VEC(JB)
C   VGK=VGC(JB)
      VXK=VXC(JB)
      VYK=VYC(JB)
      VZK=VZC(JB)
      TCM(JB)=EXP(VTK)
      PCM(JB)=EXP(VPK)
C   COMPUTE ALL PHYSICS AND DERIVATIVES
C   COMPUTE AVERAGE NUCLEAR CHARGES AND MASSES
      AVZM=0.0
      AVMM=0.0
      IF(VZK)2409,2409,3409
3409 DO 409 KB=1,NMZ
      ZAC(KB)=AZC(JB,KB)
      AVZM=AVZM+ZAC(KB)*AZM(KB)/AMM(KB)
      AVMM=AVMM+ZAC(KB)/AMM(KB)
409 CONTINUE
2409 CONTINUE
C   EOS SHOULD HAVE AVZM AND AVMM
      CALL PANDER(LMV,NDV,KSP,VXK,VYK,VZK,VTK,VDK,VPK,VUK,VSK,VKK,VEK,
1DPT,DPD,DUT,DUD,DST,DSD,DKT,DKD,DET,DED,
2DVDTT,DVDTD,SHCPT,SHCPD,DADAT,DADAD)

C   VERSION II
      2DVDTT,DVDTD,SHCPT,SHCPD,DADAT,DADAD,ZAC)

C   COMPUTE NUCLEAR REACTION RATES AND DERIVATIVES
C   ENUCZT TO HAVE INPUT ZAC AND OUTPUT AEC IN 1-D VECTORS ZAC AND EAC
      CALL ENUCZT(VTK,VDK,VXK,VYK,VZK,ZAC,EPPC,ECNO,EHEC,EAC,

```

```

1VEK,DET,DED)
EC1(IB)=EPPC
EC2(IB)=ECNO
EC3(IB)=EHEC
DO 4409 KB=1,NMZ
AEC(JB,KB)=EAC(KB)
4409 CONTINUE
C WRITE(6,166)IB,JB,VMK,VRK,VLK,VTk,VPK,VDK,VEK,VKK,VUK,VSK
C WRITE(6,166)IB,JB,DPT,DPD,DET,DED,DKT,DKD,DUT,DUD,DST,DSD
166 FORMAT(2I5,10F12.5)
VDC(JB)=VDK
VUC(JB)=VUK
VSC(JB)=VSK
VKC(JB)=VKK
VEC(JB)=VEK
DCM(JB)=EXP(VDK)
VCM(JB)=ONE/DCM(JB)
ECM(JB)=EXP(VEK)
DDDPK=ONE/DPD
DDDTK=-DPT/DPD
DVDPK=-DDDPK
DVDTK=-DDDTK
DEDPK=DED/DPD
DEDTK=DET+DED*DDDTK
DKDPK=DKD/DPD
DKDTK=DKT+DKD*DDDTK
DUDPK=DUD/DPD
DUDTK=DUT+DUD*DDDTK
DSDPK=DSD/DPD
DSDTK=DST+DSD*DDDTK
SHCV=VUK/TCM(JB)*DUT
SHCP=SHCV+PCM(JB)/DCM(JB)/TCM(JB)*DPT*DPT/DPD
AG0=SHCP/SHCV
AG1=AG0*DPD
AG2=AG0/(AG0-(AG0-ONE)/DPT)
AG3=ONE+(AG0-ONE)*DPD/DPT
DAD=(AG2-ONE)/AG2
C HOLD CP, DVDT AND DAD FIXED FOR CURRENT CYCLE OF ITERATIONS
IF(ITN-1)424,424,423
424 CONTINUE
VCP(JB)=SHCP
VDT(JB)=DVDTK
VDA(JB)=DAD
423 CONTINUE
VGC(JB)=AG1
C WRITE(6,160)IB,IB,DDDTK,DDDPK,DVDTK,DVDPK,DEDTK,DEDPK
C WRITE(6,160)IB,IB,DKDTK,DKDPK,DUDTK,DUDPK,DSDTK,DSDPK
C WRITE(6,160)IB,IB,SHCV,SHCP,AG0,AG1,AG2,AG3,DAD
DELA=VDA(JB)
DELC=DELA
DELX=DELA
AVE=ALE*ECM(IB)+(ONE-ALE)*EBM(IB)
EMDQDT=AVE
DEMDQP=ALE*ECM(IB)*DEDPK
DEMDQT=ALE*ECM(IB)*DEDTK
IF(ITS)5180,5180,5170
5170 CONTINUE
IF(KSUV-1)5180,5181,5182
5181 CONTINUE
AVT=ALT*TCM(IB)+(ONE-ALT)*TBM(IB)
DELS=VSC(IB)-VSB(IB)
EMDQDT=EMDQDT-RDT*AVT*DELS
DEMDQP=DEMDQP-RDT*(AVT*VSC(IB)*DSDPK)
DEMDQT=DEMDQT-RDT*(AVT*VSC(IB)*DSDTK+ALT*TCM(IB)*DELS)
C WRITE(6,160)IB,KSUV,AVE,AVT,DELS,TCM(IB),VSC(IB),DSDPK,DSDTK,
C 1 EMDQDT,DEMDQP,DEMDQT
GO TO 5180
5182 AVP=ALP*PCM(IB)+(ONE-ALP)*PBM(IB)
DELV=VCM(IB)-VBM(IB)
DELU=VUC(IB)-VUB(IB)
EMDQDT=EMDQDT-RDT*(DELU+AVP*DELV)
DEMDQP=DEMDQP-RDT*(VUC(IB)*DUDPK+AVP*VCM(IB)*DVDPK+
1 ALP*PCM(IB)*DELV)
DEMDQT=DEMDQT-RDT*(VUC(IB)*DUDTK+AVP*VCM(IB)*DVDTK)
5180 CONTINUE
DELZ=TBSL-PACL-GCL+VKK+VPK-FOUR*VTK
BMK=EXP(VMK)

```

```

CBMK=ONE-BMK
BLK=EXP(VLK)
EBLK=BLK-ONE
IF(IB-MB)401,402,490
401 CONTINUE
  DELR=EXP(DE LZ+SCLL-SMPL)*EBLK/CBMK
  GAK=CBMK*EXP(GCL+SMPL-TWO*VRK)
  GO TO 403
402 DELR=EXP(DE LZ+VEK)
  GAK=ZERO
403 CONTINUE
  DEL(IB)=DELR
  DELK=DELR
  DDELK(1)=ZERO
  AD=DELK
  IF(IB-MB)406,407,490
406 CONTINUE
  DDELK(2)=AD*BLK/EBLK
  DDELK(3)=AD*(DKDPK+ONE)
  DDELK(4)=AD*(DKDTK-FOUR)
  GO TO 408
407 DDELK(2)=ZERO
  DDELK(3)=AD*(DKDPK+DEDPK+ONE)
  DDELK(4)=AD*(DKDTK+DEDTK-FOUR)
408 CONTINUE
C  WRITE(6,160)MB,IB,DELX,DELR,DELK,DDELK
  IF(DELK-DE LX)411,412,412
412 IF(RMIX)1412,411,1413
1412 DELK=DELA
  DO 1414 I=1,4
1414 DDELK(I)=ZERO
  GO TO 411
C  HERE BEGINS MIXING LENGTH THEORY OF CONVECTION FOR DELK AND DDELK
1413 CONTINUE
  UCON=UCONC/RMIX/RMIX*GAK*EXP(THREE*(VTK-VPK/TWO)-VDK/TWO-VKK)
  UCON=UCON/VCP(IB)/SQRT(VDT(IB))
  DUCON(1)=-TWO*UCON
  DUCON(2)=ZERO
  DUCON(3)=(-THREE/TWO-DKDPK-DDDPK/TWO-DCP-DDP)*UCON
  DUCON(4)=(THREE -DKDTK-DDDTK/TWO-DCT-DDT)*UCON
  FCON=CC3*UCON*UCON
  BCON=(CC1*(DELR-DELA)+CC4*UCON*UCON)*UCON
  DCON=BCON**2+FCON**3
  WCON=BCON+SQRT(DCON)
  WCON=WCON**(ONE/THREE)
  VCON=(WCON*(WCON+CC2*UCON)-FCON)
  DELK=DELA
  DO 413 I=1,4
  IF(UCON)415,415,414
415 DDELK(I)=ZERO
  GO TO 413
414 IF(WCON)1415,415,1415
1415 CONTINUE
  DELK=DELA+(VCON/WCON)**2-UCON*UCON
  DFCON=TWO*CC3*UCON*DUCON(I)
  DBCON=(CC1*(DDELK(I)-DDELA)+TWO*CC4*UCON*DUCON(I))*UCON
  1 +BCON/UCON*DUCON(I)
  DDCON=TWO*BCON*DBCON+THREE*FCON*FCON*DFCON
  DWCON=(DBCON+DDCON/SQRT(DCON)/TWO)/WCON/WCON/THREE
  DVCON=TWO*WCON*DWCON+CC2*(WCON*DUCON(I)+UCON*DWCON)-DFCON
  DDELK(I)=DDELA+TWO*(VCON/WCON)*(DVCON-DWCON*(VCON/WCON))/WCON
  1 -TWO*UCON*DUCON(I)
413 CONTINUE
  IF(KPR-2)411,1416,1416
1416 CONTINUE
  WRITE(6,160)MB,IB,UCON,FCON,BCON,DCON,XCON,WCON,VCON
  WRITE(6,160)MB,IB,DELX,DELR,DELK,DDELK
411 CONTINUE
C  VGC(JB)=DELK
  IF(IB-1)490,416,417
416 CONTINUE
C  APPLY SURFACE BOUNDARY CONDITIONS ACCORDING TO VALUE OF KSC
  IF(KSC-1)1420,1420,1422
1420 CONTINUE
  BP=EXP(PACL-SCLL+TWO*VRK+FOUR*VTK)
  SC(1)=VLK-ALOG(ONE+BP)
  DSC(1,1)=-TWO*BP/(ONE+BP)

```

```

DSC(2,1)=ONE
DSC(3,1)=ZERO
DSC(4,1)=+TWO*DSC(1,1)
IF(KSC-1)2420,1421,1422
2420 CONTINUE
  SC(2)=VFK+TWO*VRK+VKK-TTGL-STARML
  DSC(1,2)=TWO
  DSC(2,2)=ZERO
  DSC(3,2)=(DKDPK+ONE)
  DSC(4,2)=DKDTK
  GO TO 425
1421 DELRL=ALOG(DELR)
  CZEROL=(DELRL-((DKDTK-FOUR)*VTK+(DKDPK+ONE)*VFK))
  PRSM1L=(DKDPK+ONE)*(RCAL-ALOG2-ALOG3+FOUR*VTK)
  CZPRS=EXP(CZEROL+PRSM1L)
  PTPM1L=(VFK*(ONE+DKDPK))
  CZPTP=EXP(CZEROL+PTPM1L)
  APPM1=(ALPHA*PRSM1-PTPM1)/(ONE+DKDPK)
  CZAPP1=(ALPHA*CZPRS-CZPTP)/(DKDPK+ONE)
  DSC(1,2)=ZERO
  DSC(2,2)=+CZAPP1*(BLK/EBLK)
  DSC(3,2)=CZPTP
  IF(DKDTK-4)2421,4421,2421
4421 SC(2)=(ONE-ALPHA)*VTK+ALPHA*ALOG(TWO)/FOUR+CZAPP1
  DSC(4,2)=(ONE-ALPHA)+FOUR*ALPHA*CZPRS)
  GO TO 425
2421 TFMNA=(ONE-ALPHA*TWO**((DKDTK/FOUR-ONE))*EXP(VTK*(FOUR-DKDTK))
  SC(2)=TFMNA/(FOUR-DKDTK)+CZAPP1
  DSC(4,2)=(TFMNA+FOUR*ALPHA*CZPRS)
C  WRITE(6,106)DKDTK,DKDPK,DELR,DELRL,CZEROL,PRSM1L,PTPM1L,
C  1CZPRS,CZPTP,TFMNA
  GO TO 425
1422 CONTINUE
C  FOR KSC=2 INITIATE ATMOSPHERE CALCULATIONS
  IF(INAT)1430,1431,1432
1431 KS=+1
  VTS(1)=VTK-DELT/THREE
  VLS(1)=VLK-DELL/THREE
  VTS(2)=VTS(1)+DELT
  VLS(2)=VLS(1)
  VTS(3)=VTS(1)
  VLS(3)=VLS(1)+DELL
  KAT(1)=1
  KAT(2)=1
  KAT(3)=1
  GO TO 1438
1432 IF(FG(KS,1,2,3))1433,1434,1434
1433 VLS(1)=VLS(3)+VLS(2)-VLS(1)
  VTS(1)=VTS(3)+VTS(2)-VTS(1)
  KS=-KS
  KAT(1)=1
  GO TO 1432
1434 IF(FG(KS,2,3,1))1435,1436,1436
1435 VLS(2)=VLS(1)+VLS(1)-VLS(2)
  VTS(2)=VTS(1)+VTS(1)-VTS(2)
  KS=-KS
  KAT(2)=1
  GO TO 1432
1436 IF(FG(KS,3,1,2))1437,1438,1438
1437 VLS(3)=VLS(1)+VLS(1)-VLS(3)
  VTS(3)=VTS(1)+VTS(1)-VTS(3)
  KS=-KS
  KAT(3)=1
  GO TO 1432
1438 IF(KAT(1)+KAT(2)+KAT(3))1430,1430,1439
1439 DO 1440 IAT=1,3
  IF(KAT(IAT))1440,1440,1441
1441 CURL=SCLM*(EXP(VLS(IAT))-ONE)
  CURT= EXP(VTS(IAT))
  CURR=SQRT(CURL/CURT**4/PIE/ACM)
  CALL ATMOSP(MAT,IRP,STARM,CURL,CURT,CURP,GRV,KPR,SV,
  1 VVK,VYK,VZK)
  KAT(IAT)=0
  VRS(IAT)=ALOG(CURR)
  VPS(IAT)=ALOG(CURP)
1440 CONTINUE
  DO 1443 IS=1,3

```

```

AM(1,IS)=VPS(IS)
AM(2,IS)=VTS(IS)
AM(3,IS)=ONE
BM(1,IS)=VRS(IS)
1443 BM(2,IS)=VLS(IS)
CALL SLINEQ(3,2,AM,BM)
DO 1444 I=1,2
DO 1445 J=1,2
1445 CS(J,I)=DELTAK(I,J)
DO 1446 J=3,5
1446 CS(J,I)=-AM(I,J-2)
1444 CONTINUE
1430 INAT=1
DO 1447 I=1,2
SC(I)=CS(1,I)*VRK+CS(2,I)*VLK+CS(3,I)*VPK+CS(4,I)*VTK+CS(5,I)
DO 1448 J=1,4
1448 DSC(J,I)=CS(J,I)
1447 CONTINUE
425 CONTINUE
IF(KPR-2)417,1417,2417
2417 WRITE(6,106)VMK,VRK,VLK,VPK,VTK,VDK
DO 2418 I=1,3
2418 WRITE(6,160)KAT(I),KS,VMS(I),VRS(I),VLS(I),VPS(I),VTS(I),VDS(I)
1417 DO 2419 I=1,2
WRITE(6,160)KSC,I,SC(I),(CS(J,I),J=1,5)
2419 WRITE(6,160)KSC,I,SC(I),(DSC(J,I),J=1,4)
417 CONTINUE
IF(IB-MB)418,419,490
419 CONTINUE
C APPLY CENTRAL BOUNDARY CONDITIONS
BMJ=EXP(VMJ)
CBMJL=ALOG(ONE-BMJ)
SMCB=SMPL+CBMJL
VRI=(SMCB-VDK-FPBT)/THREE
CC(1)=VRJ-VRI
VLI=ALOG(ONE+EXP(SMCB-SCLL+VEK))
CC(2)=VLJ-VLI
GMDP=EXP(GCL+(FPBTL+TWO*SMCB+FOUR*VDK)/THREE-VPK)/TWO
IF(GMDP-DVC)428,670,670
428 CC(3)=VPJ-VPK-ALOG(ONE-GMDP )
CC(4)=VTJ-VTK-ALOG(ONE-GMDP*DELK)
DO 426 I=1,4
DO 427 L=1,4
DCC(L,I)=ZERO
427 CONTINUE
DCC(I,I)=ONE
426 CONTINUE
DCC(5,1)=DDDPK/THREE
DCC(6,1)=DDDTK/THREE
DCC2=EXP(SMCB-SCLL-VLI+VEK)
DCC(5,2)=-DCC2*DEDPK
DCC(6,2)=-DCC2*DEDTK
IF(KEV)4260,4260,4261
4261 CONTINUE
DCC1=EXP(SMCB-SCLL-VLI)
DCC(5,2)=-DCC1*DEMDQP
DCC(6,2)=-DCC1*DEMDQT
4260 CONTINUE
DCC0=EXP(SMCB-SCLL)
VLI=ALOG(ONE+DCC0*EMDQDT)
CC(2)=VLJ-VLI
GMDPC=GMDP/(ONE-GMDP)
DCC(5,3)=-ONE+GMDPC*(FOUR*DDDPK/THREE-ONE)
DCC(6,3)= GMDPC*(FOUR*DDDTK/THREE )
GMDPD=GMDP/(ONE-GMDP*DELK)
DCC(5,4)= GMDPD*(DELK*(FOUR*DDDPK/THREE-ONE)+DDELK(3))
DCC(6,4)=-ONE+GMDPD*(DELK*(FOUR*DDDTK/THREE )+DDELK(4))
IF(KPR-1)1426,1426,1427
1427 DO 1428 I=1,4
1428 WRITE(6,160)ITN,I,CC(I),(DCC(J,I),J=1,6)
1426 CONTINUE
GO TO 460
418 CONTINUE
HKR=-EXP(SMPL-FPL+VMK-THREE*VRK-VDK)
HK(1)=HKR
DHK(1,1)=-THREE*HK(1)
DHK(2,1)=ZERO

```

```

DHK(3,1)=-HK(1)*DDDPK
DHK(4,1)=-HK(1)*DDDTK
CHKL=SMPL+VMK-SCLL-VLK
ECHKL=EXP(CHKL)
HKE =-EXP((CHKL+VEK))
HK(2)=HKE
DHK(1,2)=ZERO
DHK(2,2)=-HK(2)
DHK(3,2)=-DHK(2,2)*DEDPK
DHK(4,2)=-DHK(2,2)*DEDTK
IF(ITS)4180,4180,4170
4170 CONTINUE
IF(KSUV-1)4180,4181,4181
4181 HK(2)=-ECHKL*EMDQDT
DHK(1,2)=ZERO
DHK(2,2)=-HK(2)
DHK(3,2)=-ECHKL*DEMDQP
DHK(4,2)=-ECHKL*DEMDQT
4180 CONTINUE
HKPN=+EXP(TWO*SMPL+GFPL+VMK-FOUR*VRK-VPK)*CBMK
C INEV CONTROLS INCLUSION OF RADIUS-DERIVED
C INERTIAL TERMS IN EVOLUTION
HKIC=+EXP(SMPL-FPL+VMK-TWO*VRK-VPK)*INEV
RCM=EXP(VRK)
VVC(IB)=RDT*(RCM-EXP(VRB(IB)))
RDTP=INEV*RDTP
RDTA=SQRT(RDT*RDTP)
RDD=RDTA*(VVC(IB)-VVB(IB))
HKPI=HKIC*RDD
C HKPI=ZERO
HK(3)=HKPN+HKPI
DHK(1,3)=-FOUR*HKPN-TWO*HKPI+HKIC*RDT*RDTA*RCM
C DHK(1,3)=-FOUR*HKPN
DHK(2,3)=ZERO
DHK(3,3)=-HK(3)
DHK(4,3)=ZERO
HK(4)=HK(3)*DELK
DHK(1,4)=HK(3)*DDELK(1)-FOUR*HK(4)
DHK(2,4)=HK(3)*DDELK(2)
DHK(3,4)=HK(3)*DDELK(3)-HK(4)
DHK(4,4)=HK(3)*DDELK(4)
108 FORMAT(8E15.6)
C WRITE(6,108)DELA,DEL R,DEL K,DEL K,DDELK,I,H,HK,DHJ,DHK
IF(IB-2)490,430,430
430 CONTINUE
DX =VMK-VMJ
DV(1)=VRK-VRJ
DV(2)=VLK-VLJ
DV(3)=VPK-VPJ
DV(4)=VTK-VTJ
DO 431 I=1,4
IDE=IDEA(I,KDE)
GO TO (4311,4312,4313),IDE
4311 CONTINUE
GJK(I)=DV(I)-DX/TWO*(HJ(I)+HK(I))
GO TO 4310
4312 CONTINUE
EID=2*I-5
FID=EID/ABS(EID)
GJK(I)=DV(I)-FID*DX*SQRT((HJ(I)+HK(I)))
GO TO 4310
4313 CONTINUE
GJK(I)=DV(I)-DX*(HJ(I)*HK(I))/(HJ(I)+HK(I))
4310 CONTINUE
DO 432 L=1,4
GO TO (4321,4322,4323),IDE
4321 CONTINUE
DGJ(L,I)=-DELTAK(L,L)-DX/TWO*DHI(L,I)
DGL(L,I)=+DELTAK(L,L)-DX/TWO*DHK(L,I)
GO TO 4320
4322 CONTINUE
DGJ(L,I)=-DELTAK(L,L)-DX/TWO*SQRT((HK(I)+HJ(I))*DHI(L,I))
DGL(L,I)=+DELTAK(L,L)-DX/TWO*SQRT((HJ(I)+HK(I))*DHK(L,I))
GO TO 4320
4323 CONTINUE
DGJ(L,I)=-DELTAK(L,L)-DX*(HK(I)/(HJ(I)+HK(I)))*2*DHI(L,I)
DGL(L,I)=+DELTAK(L,L)-DX*(HJ(I)/(HJ(I)+HK(I)))*2*DHK(L,I)

```

```

4320 CONTINUE
432 CONTINUE
431 CONTINUE
C WRITE(6,160)MB,IB,DX,DV,GJK
C WRITE(6,108)DGJ,DGK
  IF(IB-2)490,440,450
440 CONTINUE
C SOLVE EQUATIONS FOR SURFACE CONDITIONS
  DO 441 I=1,4
  DO 442 L=1,2
    AM(I ,L )=DSC(I,L)
    AM(L+4,I+2)=DGK(L,I)
442 CONTINUE
  DO 443 L=1,4
    AM(I ,L+2)=DGJ(I,L)
443 CONTINUE
    BM(1,I+2)=-DGK(3,I)
    BM(2,I+2)=-DGK(4,I)
    BM(3,I+2)=-GJK(I)
441 CONTINUE
  DO 444 I=1,2
  DO 445 L=1,2
    AM(L+4, I)=ZERO
    BM( L, I)=ZERO
445 CONTINUE
    BM(3,I)=-SC(I)
444 CONTINUE
106 FORMAT(1P10E12.5)
C WRITE(6,106)AM,BM
  CALL SLINEQ(6,3,AM,BM)
C WRITE(6,106)AM,BM
  DO 446 L=1,6
  U(L)=AM(1,L)
  V(L)=AM(2,L)
  W(L)=AM(3,L)
C WRITE(6,160)LB,L,U(L),V(L),W(L)
446 CONTINUE
  GO TO 490
450 CONTINUE
C SOLVE EQUATIONS FOR INTERMEDIATE ZONES
  N=IB-1
  DO 451 I=1,4
  DO 452 L=1,2
    AM( L,I)=DGJ(L+2,I)
    AM(L+2,I)=DGK(L ,I)
452 CONTINUE
    BM( 1, I)=-DGK(3,I)
    BM( 2, I)=-DGK(4,I)
    BM( 3, I)=-GJK(I)
    AM(1,I)=AM(1,I)+DGJ(1,I)*U(4*N-3)+DGJ(2,I)*U(4*N-2)
    AM(2,I)=AM(2,I)+DGJ(1,I)*V(4*N-3)+DGJ(2,I)*V(4*N-2)
    BM(3,I)=BM(3,I)-DGJ(1,I)*W(4*N-3)-DGJ(2,I)*W(4*N-2)
451 CONTINUE
C WRITE(6,106)AM,BM
  CALL SLINEQ(4,3,AM,BM)
C WRITE(6,106)AM,BM
  DO 453 I=1,4
  L=4*N-2+I
  U(L)=AM(1,I)
  V(L)=AM(2,I)
  W(L)=AM(3,I)
C WRITE(6,160)IB,L,U(L),V(L),W(L)
453 CONTINUE
  IF(IB-MB)490,460,490
460 CONTINUE
C WRITE(6,108)DELA,DELRL,DELK,DELK,DDELK
C SOLVE EQUATIONS FOR CENTRE CONDITIONS
  M=IB
  DO 461 I=1,4
  DO 462 L=1,4
    AM(L,I)=DCC(L+2,I)
462 CONTINUE
    AM(1,I)=+AM(1,I)+DCC(1,I)*U(4*M-7)+DCC(2,I)*U(4*M-6)
    AM(2,I)=+AM(2,I)+DCC(1,I)*V(4*M-7)+DCC(2,I)*V(4*M-6)
    BM(1,I)=-CC (I)-DCC(1,I)*W(4*M-7)-DCC(2,I)*W(4*M-6)
461 CONTINUE
C WRITE(6,106)AM,BM

```

```

CALL SLINEQ(4,1,AM,BM)
C WRITE(6,106)AM,BM
DPL=AM(1,1)
DTL=AM(1,2)
DPM=AM(1,3)
DTM=AM(1,4)
490 CONTINUE
M=MB
DVR(M)=ZERO
DVL(M)=ZERO
DVP(M)=DPM
DVT(M)=DTM
L=M-1
DVP(L)=DPL
DVT(L)=DTL
DMAX=AMAX1(AMAX1(ABS(DPM),ABS(DTM)),AMAX1(ABS(DPL),ABS(DTL)))
KB=MB-2
SUMS=DVP(M)**2+DVT(M)**2+DVP(L)**2+DVT(L)**2
DO 500 IB=1,KB
N=KB+1-IB
N4=4*N
DVL(N+1)=U(N4+2)*DVP(N+1)+V(N4+2)*DVT(N+1)+W(N4+2)
DVR(N+1)=U(N4+1)*DVP(N+1)+V(N4+1)*DVT(N+1)+W(N4+1)
DMAX=AMAX1(DMAX,AMAX1(ABS(DVL(N+1)),ABS(DVR(N+1))))
DVT(N )=U(N4 )*DVP(N+1)+V(N4 )*DVT(N+1)+W(N4 )
DVP(N )=U(N4-1)*DVP(N+1)+V(N4-1)*DVT(N+1)+W(N4-1)
DMAX=AMAX1(DMAX,AMAX1(ABS(DVT(N)),ABS(DVP(N))))
C WRITE(6,160)N,N4,DVL(N+1),DVR(N+1),DVT(N),DVP(N)
SUMS=SUMS+DVL(N+1)**2+DVR(N+1)**2+DVT(N)**2+DVP(N)**2
500 CONTINUE
DVL( 1)=U( 2)*DVP( 2)+V( 2)*DVT( 2)+W( 2)
DVR( 1)=U( 1)*DVP( 2)+V( 1)*DVT( 2)+W( 1)
DMAX=AMAX1(DMAX,AMAX1(ABS(DVL(1)),ABS(DVR(1))))
SUMS=SUMS+DVL(1)**2+DVR(1)**2
RMSE=SQRT(SUMS/(4*MB-2))
C WRITE(6,160)MB,MB,DVL(1),DVR(1)
510 IF(THETA*RMSE-DVC)520,520,530
C510 IF(THETA*DMAX-DVC)520,520,530
530 THETA=THETA/TWO
GO TO 510
520 CONTINUE
C THETA=MAX(0.5,THETA)
NPP=KPR-1
DO 600 IB=1,MB
IF(MIN0(IABS(IB-1),IABS(IB-MB))-NPP)610,620,620
610 CONTINUE
WRITE(6,160)MB,IB,VMC(IB),
1 VRC(IB),DVR(IB),VLC(IB),DVL(IB),VPC(IB),DVP(IB),VTC(IB),DVT(IB)
620 CONTINUE
VRC(IB)=VRC(IB)+DVR(IB)*THETA
VLC(IB)=VLC(IB)+DVL(IB)*THETA
VPC(IB)=VPC(IB)+DVP(IB)*THETA
VTC(IB)=VTC(IB)+DVT(IB)*THETA
VPR=PRCL+4.0*VTC(IB)
VTR=(VPC(IB)-PRCL)/4.0
IF(VPC(IB)-VPR)601,601,600
601 WRITE(6,602)IB,VTC(IB),VPC(IB),VDC(IB)
602 FORMAT(' PRESSURE WARNING : ZONE,I,P,C',I10,1P3E10.3)
VPC(IB)=VPR+0.25*THETA*ABS(DVP(IB))
VTC(IB)=VTR-0.25*THETA*ABS(DVT(IB))
600 CONTINUE
STARRS= EXP(VRC( 1))/SUNR
STARLC=(EXP(VLC( 1))-ONE)*(SCLM/SUNL)
STARTC=EXP(VTC(1))
IF(KPR)630,630,640
640 WRITE(6,160)ITC,ITN,FDT,DTY,RMSE,THETA,STARRS,STARLC,STARTC
630 THETA=SQRT(THETA)
IF(RMSE-EP5D)700,650,650
650 CONTINUE
C
C DROPS TIMESTEP IF RMSE IS NOT DECREASING
IF(KFO.GE.5)GOTO 8190
IF(RMSEB.LE.RMSE) THEN
DTY=DTY*0.75
KFO=KFO+1
END IF
ABSV=ABS(RMSEB-RMSE)

```

```

IF(ABSV.LE.EPSD) THEN
DTY=DTY*0.75
KFO=KFO+1
END IF
8190 CONTINUE
RMSEB=RMSE
C
C650 IF(ITS)400,400,660
C EXPERIMENTAL RECYCLE
C650 IF(ITS)655,655,660
655 IF(RMSE-TEN**(-ITC))499,499,400
660 IF(RMSE-TEN**(5-ITN))665,670,670
C660 GO TO 665
665 IF(ITN-10)400,670,670
670 FDT=FDT/TWO
GO TO 330
700 CONTINUE
C
C MODIFIED BY PDT TO INCLUDE MASS LOSS BY ZONE STRIPPING
IF(MLOSS-1)6210,6200,6210
6200 IF(ITS-1)6210,6220,6220
6220 CONTINUE
SLUMN=(EXP(VLC(1))-ONE)*(SCLM/SUNL)
TPHOT=EXP(VTC(1))
C FORMULA FOR MASS LOSS FROM CASINELLI & LAMERS 1991
C EMPIRICALLY DERIVED FOR O-STARS
C STPML=EXP(-5.5+1.6*ALOG(SLUMN/1E6)-ALOG(TPHOT/TPHOTZ))*DTY
C WRITE(6,7010)
7010 FORMAT('MASS LOSS BY CASINELLI & LAMERS')
C FORMULA FOR MASS LOSS FROM NIEUWENHUIJZEN AND DE JAGER
C ASTRON.ASTROPHYS. 231,134 1990
STPML=(9.631**(-15))*(SLUMN**1.42)*(SMASS**0.16)*(STARRS**0.81)*DTY
WRITE(6,7020)
7020 FORMAT('MASS LOSS BY NIEUWENHUIJZEN AND DE JAGER')
TMLOST=TMLOST+STPML
JX=1
6800 JX=JX+1
6060 DIF=(EXP(VMC(JX))-EXP(VMC(1)))*STARM*(ONE+ETA)/SUNM
IF(STPML-DIF)6240,6800,6800
6240 CONTINUE
C DO-LOOP TO SMEAR COMPOSITION ACROSS THE "REMOVED" ZONES
MMX=1
7110 MMX=MMX+1
TEMPC=(EXP(VMC(MMX))-EXP(VMC(1)))*STARM*(ONE+ETA)/SUNM
IF(TMLOST*SMASS/SMASSI-TEMPC)7120,7120,7110
7120 CONTINUE
LMX=MMX
7040 LMX=LMX+1
TEMPC=(EXP(VMC(LMX))-EXP(VMC(MMX)))*STARM*(ONE+ETA)/SUNM
IF(TMLOST*SMASS/SMASSI-TEMPC)7050,7050,7040
7050 CONTINUE
VXC(1)=VXC(MMX)
VYC(1)=VYC(MMX)
VZC(1)=VZC(MMX)
DO 6712 J=1,NMZ
AZC(1,J)=AZC(MMX,J)
6712 CONTINUE
DO 6710 K=2,LMX-1
VXC(K)=VXC(1)+K*(VXC(LMX)-VXC(1))/LMX
VYC(K)=VYC(1)+K*(VYC(LMX)-VYC(1))/LMX
VZC(K)=VZC(1)+K*(VZC(LMX)-VZC(1))/LMX
DO 6714 J=1,NMZ
AZC(K,J)=AZC(1,J)+K*(AZC(LMX,J)-AZC(1,J))/LMX
6714 CONTINUE
6710 CONTINUE
C ADDITIONAL MASS LOSS DUE TO TRANSMUTATION OF ELEMENTS -
C THIS MASS LOSS HAS NO EFFECT ON THE COMPOSITION
C VARIABLES ABOVE.
TRANML=(SUNL/SUNM)*YEARS*SLUMN*DTY/CVL**2
STPML=STPML+TRANML
TMLOST=TMLOST+TRANML
C CHANGING THE MASS AND RADIUS VARIABLES
STARMX=STARM
STARRX=STARRS
STARM=STARM-STPML*SUNM
STARMS=STARMS-STPML
SMASS=SMASS-STPML

```

```

SMPM=(ONE+ETA)*FSCM*STARM
SMPL=ALOG(SMPM)
SCMM=FSCM*STARM
SCML=ALOG(SCMM)
STARRS=STARRS*((0-(0-((STARM/STARMX)-1))**(1.0/3.0))+1)
DO 6700 K=1,MB
  VMB(K)=VMC(K)
  VMC(K)=ALOG(1-STARM*(1-EXP(VMC(K)))/STARMX)
IF(K.EQ.MB)GOTO 6700
  TEMPR=(EXP(VRC(K)))*((0-(0-((1-EXP(VMC(K)))
  1/(1-EXP(VMB(K))-1))**(1.0/3.0))+1)
  VRC(K)=ALOG(TEMPR)
6700 CONTINUE
  WRITE(6,6080)STPML
6080 FORMAT(' STEP MASS LOSS = ',1PE10.3)
  WRITE(6,6070)JX-2
6070 FORMAT(' NUMBER OF ZONES "REMOVED" = ',I5)
  WRITE(6,6090)TMLOST
6090 FORMAT(' TOTAL MASS LOST' = ',1PE10.3)
6210 CONTINUE
C
  IF(KRZ-10)740,750,750
740 CONTINUE
C  DETERMINE CONVECTION ZONES
  CALL CONV(IRZ,MB,NX,NY,SMPM,VDA,DEL,
  1VMC,VRC,VTC,VDC,VXC,VYC,VZC,TTY,DTY,AZC)
750 CONTINUE
  STARRS= EXP(VRC( 1))/SUNR
  STARLS=(EXP(VLC( 1))-ONE)*(SCLM/SUNL)
  TPHOT = EXP(VTC( 1))
  IF(ITS.EQ.ZERO) SMASSI=SMASS
  PPHOT = EXP(VPC( 1))
  DPHOT = EXP(VDC( 1))
  TCEN = EXP(VTC(MB))
  PCEN = EXP(VPC(MB))
  DCEN = EXP(VDC(MB))
  XCEN = VXC(MB)
  YCEN = VYC(MB)
  ZCEN = VZC(MB)
  XPHOT=VXC(1)
  YPHOT=VYC(1)
  ZPHOT=VZC(1)
  GRAVS=GSUN*STARMS/STARRS/STARRS
  BMAGS=-2.5*LOG10(STARLS)
  TTY=TTY+DTY
  KFO=ZERO
  IF(KPR-IST)710,720,720
720 CONTINUE
  WRITE(6,760)(VEC(MB+1-IB),IB=1,32)
  IF(NX)731,721,721
721 IF(NX-MB)713,731,731
713 WRITE(6,760)(VEC(NX+1-IB),IB=1,32)
731 IF(NY)732,722,722
722 IF(NY-MB)723,732,732
723 WRITE(6,760)(VEC(NY+1-IB),IB=1,32)
732 CONTINUE
760 FORMAT(8F10.3)
  WRITE(6,162)
162 FORMAT(' STEP', ' ITNS', ' TIME STEP', ' TOTL TIME',
  1 5X, 'IMASS',5X, 'CMASS',4X, 'RADIUS', ' LUMINOSITY',
  2  /,T11,8X, 'TP',8X, 'PP',8X, 'DP',8X, 'XP',8X, 'YP',8X, 'ZP',
  3  /,T11,8X, 'TC',8X, 'PC',8X, 'DC',8X, 'XC',8X, 'YC',8X, 'ZC',
  4  /,T11,5X, 'ZC12P',5X, 'ZO16P',4X, 'ZNE20P',4X, 'ZMG24P',
  54X, 'ZSI28P',5X, 'ZS32P', /, T11,5X, 'ZC12C',5X, 'ZO16C',4X,
  6'ZNE20C',4X, 'ZMG24C',4X, 'ZSI28C',5X, 'ZS32C')
163 FORMAT(' STEP', ' GRID', ' TIME STEP', ' TOTL TIME',
  1 5X, 'IMASS',5X, 'CMASS',4X, 'RADIUS', ' LUMNOSITY',
  2  /,T11,8X, 'TP',8X, 'PP',8X, 'DP',8X, 'XP',8X, 'YP',8X, 'ZP',
  3  /,T11,8X, 'TC',8X, 'PC',8X, 'DC',8X, 'XC',8X, 'YC',8X, 'ZC',
  4  /,T11,5X, 'ZC12P',5X, 'ZO16P',4X, 'ZNE20P',4X, 'ZMG24P',
  54X, 'ZSI28P',5X, 'ZS32P', /, T11,5X, 'ZC12C',5X, 'ZO16C',4X,
  6'ZNE20C',4X, 'ZMG24C',4X, 'ZSI28C',5X, 'ZS32C')
710 IST=1
  ICN=100*ITC+ITN
  WRITE(6,161)ITS,ICN,DTY,TTY,SMASSI,SMASS,STARRS,STARLS,
  1 TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
  2 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,

```

```

3      (AZC(1,JB),JB=1,6),
4      (AZC(MB,JB),JB=1,6)
161  FORMAT(2I5,(T11,1P6E10.3))
C    WRITE(6,*)
C    PREPARE TO SAVE/OUTPUT CONVERGED MODEL
      KTS=MOD(ITS,JTS)
      IF(KTS)800,810,800
810  IF(NDSO)800,800,820
820  WRITE(6,*)' OUTPUT MODEL TO DATA SET NUMBER ',NDSO
      WRITE(NDSO)MB,ITS,DTY,TTY,SMASSI,SMASS,STARRS,STARLS,
1      TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
2      TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,
3      (AZC(MB,JB),JB=1,NMZ)
C    1 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN
      MA=1
      DO 830 IB=1,MB
      IF(KPR-1)835,845,840
840  WRITE(6,160)ITS,IB,VMC(IB),VRC(IB),VLC(IB),VPC(IB),VTC(IB),
1      VDC(IB),VUC(IB),VSC(IB),VEC(IB),VXC(IB),VYC(IB),VZC(IB)
845  CONTINUE
C    SMM=(ONE+ETA)*FSCM*STARM*(ONE-EXP(VMC(IB)))
C    SMR=EXP(VRC(IB))
C    SML=(SCLM/SUNL)*(EXP(VLC(IB))-ONE)
C    SMT=EXP(VTC(IB))
C    SMP=EXP(VPC(IB))
C    SMD=EXP(VDC(IB))
C    SME=EXP(VEC(IB))
C    SMK=EXP(VKC(IB))
C    SMG=VGC(IB)
835  CONTINUE
      WRITE(NDSO) VMC(IB),VRC(IB),VLC(IB),VPC(IB),VTC(IB),
C    1 VDC(IB),VUC(IB),VSC(IB),VEC(IB),VXC(IB),VYC(IB),VZC(IB),
C    2 EC1(IB),EC2(IB),EC3(IB),VVC(IB)
1      VDC(IB),VUC(IB),VSC(IB),VXC(IB),VEC(IB),VGC(IB),VVC(IB),
2      VXC(IB),VYC(IB),VZC(IB),(AZC(IB,JB),JB=1,NMZ)
830  CONTINUE
      REWIND NDSO
      NDSO=NDS3+NDS4-NDSO
800  CONTINUE
      AGEMY=1.0E-06*TTY
      TEFFL=ALOG10(TPHOT)
      SLOGL=ALOG10(STARLS)
      GRAVL=ALOG10(GRAVS)
      WRITE(6,*)' H-R DATA: AGE6 LOGT LOGL BMAG
1 LOGG'
      WRITE(6,164)ITS,MB,AGEMY,TEFFL,SLOGL,BMAGS,GRAVL
164  FORMAT(2I5,1P7E10.3)
      WRITE(7,8100)BMAGS,TTY,TEFFL,SLOGL,GRAVL
      WRITE(8,8000)TTY,STPML,STARLS,TPHOT,YPHOT,ZPHOT,TMLOST
8000  FORMAT(E11.4,E10.3,E11.3E3,E10.3)
8100  FORMAT(E10.3,E11.4,3E10.3)
C    FIND APSIDAL MOTION CONSTANTS
      DO 801 IB=1,MB
      JB=MB+1-IB
      BMM(IB)=(ONE+ETA)*FSCM*STARM*(1.0-EXP(VMC(JB)))
      BRM(IB)=EXP(VRC(JB))
      BAM(IB)=4.0*PIE*BRM(IB)**2
      ZDM(IB)=EXP(VDC(JB))
      BVM(IB)=1.0/ZDM(IB)
      BPM(IB)=EXP(VPC(JB))
      BGM(IB)=VGC(JB)
801  CONTINUE
      JIB=MB/100+1
      MIB=MB/JIB-1
      MIB=MB/JIB
      DO 802 IB=1,MIB
      JB=(IB-1)*JIB+1
      BMM(IB)=BMM(JB)
      BRM(IB)=BRM(JB)
      BAM(IB)=BAM(JB)
      ZDM(IB)=ZDM(JB)
      BVM(IB)=BVM(JB)
      BPM(IB)=BPM(JB)
      BGM(IB)=BGM(JB)
802  CONTINUE
      APKM=CAPSI(MIB,BMM,BRM,ZDM)
      APKL=ALOG10(APKM)

```

```

WRITE(6,*)' APSIDAL MOTION: K LOGK'
WRITE(6,160)ITS,MIB,APKM,APKL
IF(KKP)805,805,808
808 CONTINUE
C FIND ADIABATIC EIGENVALUES AND EIGENFUNCTIONS
DO 803 IB=1,MIB-1
ZDM(IB)=BMM(IB+1)-BMM(IB)
ZPM(IB)=(BPM(IB)+BPM(IB+1))/2.0
ZVM(IB)=(BVM(IB)+BVM(IB+1))/2.0
ZGM(IB)=(BGM(IB)+BGM(IB+1))/2.0
803 CONTINUE
DO 804 IB=2,MIB-1
BDM(IB)=(ZDM(IB)+ZDM(IB-1))/2.0
C BDM(IB)=(BMM(IB+1)+ZMM(IB-1))/2.0
804 CONTINUE
MODE=KKP
IF(MODE)805,805,806
806 CONTINUE
WRITE(6,*)' RADIAL PULSATION: LINEAR ADIABATIC'
CALL SEIGEN(MIB,MODE,BMM,BDM,BRM,BAM,ZDM,ZVM,ZPM,ZGM,BEF,WS)
WR=SQRT(WS)
WC=WR/PIE/2.0
PSEC=1.0/WC
PMIN=PSEC/60.0
PHR=PMIN/60.0
PDAY=PHR/24.0
WRITE(6,*)' MODE, GRID, WR, WC, PSEC, PMIN, PHR, PDAY'
WRITE(6,160)MODE,MIB,WR,WC,PSEC,PMIN,PHR,PDAY
C WRITE(6,*)' EIGENFUNCTION'
C WRITE(6,167)(BEF(IB),IB=1,MIB)
167 FORMAT(1P8E10.3)
805 CONTINUE
WRITE(6,*)' *****'
C DURING FURTHER EVOLUTION CONTROL TIME STEP VIA ITERATION COUNT
IF(KEV)850,850,860
860 IF(ITN-5)870,870,875
870 FDT=TWO*FDT
GO TO 880
875 IF(ITN-10)880,885,885
885 FDT=FDT/TWO
880 IF(KTS)320,300,300
850 CONTINUE
RETURN
END
C KRONECKERS DELTA - INTEGER
FUNCTION KDELTA(I,J)
IMPLICIT REAL*8 (A-H,O-Z)
KDELTA=0
IF(I-J)1,2,1
2 KDELTA=1
1 RETURN
END
C KRONECKERS DELTA - REAL
FUNCTION DELTAK(I,J)
IMPLICIT REAL*8 (A-H,O-Z)
DATA ZERO/0.0/,ONE/1.0/
DELTAK=ZERO
IF(I-J)1,2,1
2 DELTAK=ONE
1 RETURN
END

```

'sutils.f'

```

C SUTILS - SUITE OF UTILITIES FOR STELLAR STRUCTURE AND EVOLUTION
C CONTAINING: LEMOD,XYZATM,SLINEQ,CONV,CAPIS,SEIGEN,MODENV,MODEL
C CONSTRUCT LANE-EMDEN STELLAR MODEL
SUBROUTINE LEMOD(N,K,SMS,SRS,SV)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION X(512),Y(512),Z(512)
DIMENSION VM(512),VR(512),VP(512),VL(512),VT(512),VD(512)
DIMENSION SV(512,20)
DATA PIE/3.1415926536/,GCG/6.670E-8/,CVL/2.997925E+10/
DATA AVN/6.02350E23/,BCK/1.38054E-16/,ARC/7.56495E-15/
DATA SUNM/1.989E+33/,SUNR/6.960E+10/,SUNL/3.900E+33/
OPEN(10,FILE="fort.10",FORM="FORMATTED")

```

```

SIG=ARC*CVL/4
B=1.0
W=0.5
H=0.01*(N+2)
SM=SMS*SUNM
IF(SRS)1,1,2
1 SRS=SQRT(SMS)
2 SR=1.00*SRS*SUNR
DA=0.75*SM/SR**3/PIE
SLS=SMS**3
SL=0.50*SLS*SUNL
TE4=SL/SR/SR/SIC/PIE/4
PRE=ARC*TE4/3
TE=SQRT(SQRT(TE4))
CALL LANEMD(N,H,X,Y,Z,K,X1,W1)
WRITE(6,*)'LANE-EMDEN MODEL'
WRITE(6,6)N,H,X1,W1
A=SR/X1
D0=DA/3.0*X1**3/W1
DE=D0*Y(K)**N
PGE=AVN*BCK*DE
PE=PRE+PGE
PHI0=GCG*X1/W1*SM/SR
P0=PHI0/(N+1)*D0
T0=B*W/AVN/BCK*PHI0/(N+1)
WRITE(6,6)K,D0,P0,T0,PHI0
DO 3 I=1,K
VM(I)=-X(I)**2*Z(I)
VM(I)=VM(I)*SM/W1
VR(I)=A*X(I)
C VP(I)=PE+P0*(Y(I)**(N+1)-Y(K)**(N+1))
VP(I)=P0*Y(I)**(N+1)
VL(I)=SL
VT(I)=TE+T0*(Y(I)-Y(K))
VD(I)=D0*Y(I)**N
C WRITE(6,6)I,VM(I),VR(I),VP(I),VL(I),VT(I),VD(I)
J=K+1-I
SV(J,1)=VM(I)
SV(J,2)=VR(I)
SV(J,3)=VL(I)
SV(J,4)=VT(I)
SV(J,5)=VP(I)
SV(J,6)=VD(I)
3 CONTINUE
SV(K-1,3)=0.25*SV(K-1,3)
SV(K-2,3)=0.50*SV(K-2,3)
SV(K-3,3)=0.75*SV(K-3,3)
RETURN
6 FORMAT(15,1P6E10.3)
END
SUBROUTINE LANEMD(N,H,XA,YA,ZA,K,X1,W1)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XA(K),YA(K),ZA(K)
F(X,Y,Z)=-2/X*Z-Y**N
X=0.0
Y=1.0
YP=0.0
G1=-H/3
K=1
10 CONTINUE
C WRITE(6,6)X,Y,YP
XA(K)=X
YA(K)=Y
ZA(K)=YP
XC=X+H/2
YC=Y+H/2*YP+H/8*G1
IF(YC)20,20,21
21 CONTINUE
ZC=YP+G1/2
G2=H*F(XC,YC,ZC)
ZC=YP+G2/2
G3=H*F(XC,YC,ZC)
XC=X+H
YC=Y+H*YP+H/2*G3
IF(YC)20,20,22
22 CONTINUE
ZC=YP+G3

```

```

G4=H*F(XC,YC,ZC)
DY=H*(YP+(G1+G2+G3)/6)
DYP=(G1+2*(G2+G3)+G4)/6
X=XC
Y=Y+DY
IF(Y)20,20,23
23 CONTINUE
YP=YP+DYP
G1=H*F(X,Y,YP)
K=K+1
GO TO 10
20 DX=-Y/YP
X1=X+DX
WRITE(6,6)X,DX,X1
YP1=YP+DX*G1/H
W1=-X1**2*YP1
WRITE(6,6)YP,YP1,W1
RETURN
6 FORMAT(3F10.5)
END
C  ATMOSP
SUBROUTINE ATMOSP(NSA,IRP,STARM,STARR,STARL,TEFF,PEFF,GRAV,KP,SVS,
1 X,Y,Z)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION SVS(16,512)
DIMENSION VM(5),ZM(5)
EQUIVALENCE (VM(1),DMASS),(VM(2),DRAD),(VM(3),DLUM),
1(VM(4),TEMP),(VM(5),PTOT)
EQUIVALENCE (ZM(1),RHO),(ZM(2),ZERO),(ZM(3),ROSSK),
1(ZM(4),W),(ZM(5),TAU)
SQRT(X)=DSQRT(X)
ABS(X)=DABS(X)
DATA VLC/2.997925E10/,RCA/7.56495E-15/,GCC/6.67000E-8/
DATA PIE/3.1415926536/,AVN/6.02252E23/,BCK/1.38054E-16/
DATA ZERO/0.0/
GRAV=GCC*STARM/STARR**2
FPS=4*PIE*RCA*VLC/4
TEFF= SQRT(SQRT(STARL/STARR/STARR/FPS))
E=1.0E-5
DLUM=0.0
RHOS=1.0E-10
IT=0
30 IT=IT+1
ND=1
NA=NSA
1 NA=NA/ND
IF(NA-16)2,3,3
2 IF(IT-10)31,32,32
32 STOP 32
31 RHOS=RHOS/2
GO TO 30
3 DTAU=2.0/3.0/NA
PTOTP=RCA*TEFF**4/6
DPDTP=0.0
TAU=0.0
RHO=RHOS
DO 4 I=1,NA
TAU=TAU+DTAU
TEMP= SQRT(SQRT((1+3*TAU/2)/2))*TEFF
PRAD=RCA/3*TEMP**4
C5 CALL CONEQS(1,0,1,X,Y,Z,TEMP,RHO,P,AN,EN,AM,W,EG,ROSSK,Q,S)
5 CALL PVECTR(+1, 1, 0,X,Y,Z,VM,ZM)
W=AVN*BCK*RHO*TEMP/(PTOT-PRAD)
DPDT=GRAV/ROSSK
PTOT=PTOTP+(DPDTP+DPDT)/2*DTAU
PGAS=PTOT-IRP*PRAD
IF(PGAS)6,6,7
6 ND=2
J=IT
WRITE(6,66)J,GRAV,DPDT,PRAD,TEMP,PTOT,RHO,PGAS,ROSSK,W,TAU
GO TO 1
7 RHOP=RHO
RHO=(W*PGAS/AVN/BCK/TEMP+RHOP)/2
IF(ABS(RHOP/RHO-1)-E)8,5,5
8 CONTINUE
RRK=1/RHO/ROSSK
IF(I-1)4,9,10

```

```

9  DRAD=0.0
   DMASS=0.0
   GO TO 11
10  DRAD=DTAU*(RRK+RRKP)/2
   DMASS=4*PIE*STARR**2*(RHO+RHOP)/2*DRAD
11  RRKP=RRK
   IF(I-NA)20,21,21
20  IF(KP-2)22,22,21
21  CONTINUE
   WRITE(6,66)I,DMASS,DRAD,DLUM,TEMP,PTOT,RHO,PRAD,ROSSK,W,TAU
22  CONTINUE
66  FORMAT(15,1P10E11.4)
   PTOT=PTOT
   DPDP=DPDT
   DO 15 J=1,5
     SVS(J ,I)=VM(J)
15  SVS(J+5,I)=ZM(J)
4   CONTINUE
   DO 16 I=2,NA
     DO 17 J=1,3
17  SVS(J,I-1)=SVS(J,I)
16  CONTINUE
   SVS(1,NA)=STARM
   SVS(2,NA)=STARR
   SVS(3,NA)=STARL
   DO 18 I=2,NA
     DO 19 J=1,3
19  SVS(J,NA+1-I)=SVS(J,NA+2-I)+SVS(J,NA+1-I)
18  CONTINUE
   PEFF=PTOT
   RETURN
   END
C   SLINEQ
   SUBROUTINE SLINEQ(N,M,A,B)
   IMPLICIT REAL*8 (A-H,O-Z)
   DIMENSION A(10,10),B(10,10),LK(10),KL(10)
   ABS(X)=DABS(X)
   DO 21 I=1,N
21  LK(I)=I
   DO 1 I=1,N
     X=0.0
     K=I
     DO 22 J=I,N
C    Y=DABS(A(J,I))
     Y=ABS(A(J,I))
     IF(Y-X)23,23,24
24  X=Y
     K=J
23  CONTINUE
22  CONTINUE
     AII=A(K,I)
     DO 25 IJ=1,N
       ZA=A(K,IJ)
       A(K,IJ)=A(I,IJ)
       A(I,IJ)=ZA
25  CONTINUE
     L=LK(I)
     LK(I)=LK(K)
     LK(K)=L
     X=0.0
     K=I
     DO 2 J=I,N
C    Y=DABS(A(L,I))
     Y=ABS(A(L,I))
     IF(Y-X)3,3,4
4   X=Y
     K=J
3   CONTINUE
2   CONTINUE
     AII=A(I,K)
     DO 5 JI=I,N
       ZA=A(JI,K)
       A(JI,K)=A(JI,I)
       A(JI,I)=ZA/AII
5   CONTINUE
     DO 6 JI=1,M
       ZB=B(JI,K)

```

```

      B(JI,K)=B(JI,I)
      B(JI,I)=ZB/AII
6     CONTINUE
      IF(I-N)11,10,10
11    CONTINUE
      IP=I+1
      DO 7 JJ=IP,N
      D=A(I,JJ)
      DO 8 JI=I,N
      A(JI,JJ)=A(JI,JJ)-D*A(JI,I)
8     CONTINUE
      DO 9 JI=I,M
      B(JI,JJ)=B(JI,JJ)-D*B(JI,I)
9     CONTINUE
C     WRITE(6,100)((A(I1,I2),I1=1,5),I2=1,5),(B(1,I1),I1=1,5)
C100  FORMAT(5E20.5)
7     CONTINUE
10    CONTINUE
C     WRITE(6,100)((A(I1,I2),I1=1,5),I2=1,5),(B(1,I1),I1=1,5)
1     CONTINUE
      NM=N-1
      DO 12 I=1,N
      II=N+1-I
      IIP=II+1
      DO 13 J=1,M
      D=0.0
      IF(IIP-N)15,15,13
15    DO 14 K=IIP,N
14    D=D+A(K,II)*B(J,K)
13    B(J,II)=B(J,II)-D
12    CONTINUE
C     WRITE(6,99)LK
99    FORMAT(25I4)
      DO 27 I=1,N
      DO 28 J=1,M
      II=LK(I)
      A(J,II)=B(J, I)
28    CONTINUE
27    CONTINUE
      RETURN
      END
C     CHECK FOR CONVECTION
      SUBROUTINE CONV(IRZ,MB,IX,IY,SMPM,VDA,DEL,
1VMC,VRC,VTC,VDC,VXC,VYC,VZC,TTY,DTY,AZC)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION VDA(MB),DEL(MB),VMC(MB),VRC(MB),VTC(MB),VDC(MB)
      DIMENSION VXC(MB),VYC(MB),VZC(MB)
      DIMENSION AZC(512,8),SAZ(8),AAZ(8)
      DIMENSION MC(2),ICI(2,8),VM(2),VR(2),VT(2),VD(2)
      DATA NOMX/'H '/,NOMY/'HE '/,NOMZ/'Z '/
      DATA NZ/8/
      VMT=0.0
      VRT=0.0
      VMX=0.0
      VRX=0.0
      VMY=0.0
      VRY=0.0
      NCZ=0
      IC=0
      KC=0
      DO 100 IB=1,MB
      JC=0
      IF(DEL(IB)-VDA(IB))101,102,102
102  JC=1
101  IF(JC-IC)103,104,103
103  KC=KC+1
      MC(KC)=IB-IC
      IC=JC
104  IF(KC-1)100,105,106
105  IF(IB-MB)100,107,100
107  MC(2)=MB
106  CONTINUE
      NCZ=NCZ+1
      ICI(1,NCZ)=MC(1)
      ICI(2,NCZ)=MC(2)
      DO 201 II=1,2
      IJ=MC(II)

```

```

      VM(IJ)=(1.0-EXP(VMC(IJ)))/(1.0-EXP(VMC(1)))
      VR(IJ)=EXP(VRC(IJ)-VRC(1))
      VT(IJ)=EXP(VTC(IJ))
      VD(IJ)=EXP(VDC(IJ))
201 CONTINUE
C
C   MODIFICATION BY PDT TO WRITE TO FILE
      IF(MB.EQ.MC(2)) THEN
          VMT=VM(1)
          VRT=VR(1)
          END IF
C
      WRITE(6,116)MC(1),VM(1),VR(1),VT(1),VD(1),
1          MC(2),VM(2),VR(2),VT(2),VD(2)
116 FORMAT(5X,'CONVECTION MRTD ZONE',I5,1P4E10.3,
1          /T2L, ' ZONE',I5,1P4E10.3)
C116 FORMAT(5X,'CONVECTION ZONE MRTD',I5,1P4E10.3, ' TO',I5,1P4E10.3)
      DO 202 J=1,2
          IBC=MC(J)
          IF(IBC-MB)203,202,202
203 IF(IBC-1)202,202,204
204 IF(J-1)202,205,206
205 IBL=IBC+1
          IBR=IBC
          GO TO 207
206 IF(J-2)202,208,202
208 IBL=IBC
          IBR=IBC-1
207 DVMC=VMC(IBR)-VMC(IBL)
          DDEL=DEL(IBR)-DEL(IBL)
          DVDA=VDA(IBR)-VDA(IBL)
          VMX=VMC(IBL)+DVMC*(VDA(IBL)-DEL(IBL))/(DDEL-DVDA)
          FL=(VMC(IBR)-VMX)/DVMC
          FR=(VMX-VMC(IBL))/DVMC
C   WRITE(6,216)IBL,IBR,VMX,FL,FR
216 FORMAT(2I5,1P3E10.3)
202 CONTINUE
      IC=0
      KC=0
100 CONTINUE
C   WRITE(6,226)NCZ
226 FORMAT(I5,' CONVECTION ZONE(S) ')
      IF(NCZ)360,360,310
310 DO 320 NC=1,NCZ
          N1=ICI(1,NC)
          N2=ICI(2,NC)
          IF(N1-N2)370,320,320
370 SM=0.0
          SX=0.0
          SY=0.0
          SZ=0.0
          DO 375 M=1,NZ
              SAZ(M)=0.0
375 CONTINUE
          VM2=0.0
          DO 330 IB=N1,N2
              VMX=VM2
              VMC2=VMC(IB)
              VM2=SMPM*(1.0-EXP(VMC2))
              IF(IB-N1)330,330,340
340 VM1=VMX
              DM=VM2-VM1
              SM=SM+DM
              SX=SX+DM*(VXC(IB)+VXC(IB-1))/2.0
              SY=SY+DM*(VYC(IB)+VYC(IB-1))/2.0
              SZ=SZ+DM*(VZC(IB)+VZC(IB-1))/2.0
              DO 345 M=1,NZ
                  SAZ(M)=SAZ(M)+DM*(AZC(IB,M)+AZC(IB-1,M))/2.0
345 CONTINUE
              DO 335 M=1,NZ
                  AAZ(M)=SAZ(M)/SM
335 CONTINUE
330 CONTINUE
          AX=SX/SM
          AY=SY/SM
          AZ=SZ/SM
C   WRITE(6,216)N1,N2,VXC(N1),VYC(N1),VZC(N1)

```

```

C   WRITE(6,216)N1,N2,VXC(N2),VYC(N2),VZC(N2)
    WRITE(6,216)N1,N2,AX,AY,AZ
    DO 350 IB=N1,N2
      VXC(IB)=AX
      VYC(IB)=AY
      VZC(IB)=AZ
    DO 355 M=1,NZ
      AZC(IB,M)=AAZ(M)
355  CONTINUE
350  CONTINUE
320  CONTINUE
360  CONTINUE
    DO 410 IB=1,MB
      IF(VXC(IB))420,420,425
425  IX=IB
420  IF(VYC(IB))430,430,435
435  IY=IB
430  CONTINUE
410  CONTINUE
      IF(IX-MB)460,465,465
460  CONTINUE
      VMX=(1.0-EXP(VMC(IX)))/(1.0-EXP(VMC(1)))
      VRX=EXP(VRC(IX)-VRC(1))
      VTX=EXP(VTC(IX))
      VDX=EXP(VDC(IX))
      WRITE(6,416)NOMX,IX,VMX,VRX,VTX,VDX
465  CONTINUE
      IF(IY-MB)470,475,475
470  CONTINUE
      VMY=(1.0-EXP(VMC(IY)))/(1.0-EXP(VMC(1)))
      VRY=EXP(VRC(IY)-VRC(1))
      VTY=EXP(VTC(IY))
      VDY=EXP(VDC(IY))
      WRITE(6,416)NOMY,IY,VMY,VRY,VTY,VDY
475  CONTINUE
416  FORMAT(5X,A4,'DEPLETED TO',I5,' AT M,R,T,D',IP4E10.3)
    WRITE(10,9000)(TTY+DTY),VMT,VRT,VMX,VRX,VMY,VRY
9000  FORMAT(E11.4,6E10.3)
    RETURN
    END
C   CAPSIS : CALCULATE APSIDAL MOTION CONSTANTS
C   PROGRAM TCAPSIS
    SUBROUTINE TCAPSIS
    IMPLICIT REAL*8 (A-H,O-Z)
    DIMENSION VMC(512),VRC(512),VDC(512)
    DATA PIE/3.1415926536/
    DO 10 IM=1,8
      M=2**IM+1
      DO 1 I=1,M
        VRC(I)=1.0*(I-1)/(M-1)
        VRC(I)=(I-1)*1.0/(M-1)
        VDC(I)=(1.0-VRC(I))
        VMC(I)=4.0/3.0*PIE*VRC(I)**3*(1.0-3.0*VRC(I)/4.0)
C   WRITE(6,6)I,VRC(I),VDC(I),VMC(I)
1    CONTINUE
      C=CAPSIS(M,VMC,VRC,VDC)
      WRITE(6,6)M,C
10   CONTINUE
    STOP
6    FORMAT(I10,1P3E10.3)
    END
    FUNCTION CAPSIS(M,VMC,VRC,VDC)
    IMPLICIT REAL*8 (A-H,O-Z)
    DIMENSION VMC(M),VRC(M),VDC(M)
    DIMENSION XA(512),UA(512),GA(512),HA(512)
    DIMENSION YA(512),ZA(512)
    DATA PIE/3.1415926536/
    DO 1 I=1,M
C   J=M+1-I
      J=I
C   IF(I-M)2,3,4
      IF(I-1)4,3,2
2    XA(J)=VRC(I)/VRC(M)
      UA(J)=4.0*PIE*VRC(I)**3/VMC(I)*VDC(I)
      GA(J)=2.0*UA(J)/XA(J)
      HA(J)=2.0*(UA(J)-3.0)/XA(J)**2
      GO TO 4

```

```

3  XA(J)=0.0
   UA(J)=3.0
   GA(J)=0.0
   HA(J)=0.0
4  CONTINUE
C  WRITE(6,16)J,XA(J),UA(J),GA(J),HA(J)
1  CONTINUE
C  HA(1)=HA(2)
   YA(1)=1.0
   ZA(1)=0.0
   DO 5 I=1,M-1
     J=I+1
     H=XA(J)-XA(I)
     Y0=YA(I)
     Z0=ZA(I)
     IF(I-1)5,6,7
6    W1=-H*Y0*HA(1)/7.0
     GO TO 8
7    Y=Y0
     Z=Z0
     W1=-H*(Z*GA(I)+Y*HA(I))
8    Y=Y0+H/2.0*Z0+H/8.0*W1
     Z=Z0+W1/2.0
     W2=-H*(Z*(GA(I)+GA(J))+Y*(HA(I)+HA(J)))/2.0
     Y=Y0+H/2.0*Z0+H/8.0*W2
     Z=Z0+W2/2.0
     W3=-H*(Z*(GA(I)+GA(J))+Y*(HA(I)+HA(J)))/2.0
     Y=Y0+H*Z0+H/2.0*W3
     Z=Z0+W3
     W4=-H*(Z*GA(J)+Y*HA(J))
     DZ=(W1+2.0*(W2+W3)+W4)/6.0
     DY=H*(Z0+(W1+W2+W3)/6.0)
     ZA(J)=Z0+DZ
     YA(J)=Y0+DY
C  WRITE(6,16)J,XA(J),UA(J),GA(J),HA(J),YA(J),ZA(J)
5  CONTINUE
   CAPSIS=(3.0*YA(M)-ZA(M))/(4.0*YA(M)+2.0*ZA(M))
C  WRITE(6,16)M,CAPSIS
   RETURN
16 FORMAT(I10,1P7E10.3)
END
C  SUBROUTINE FOR STELLAR EIGENVALUES AND EIGENFUNCTIONS
   SUBROUTINE SEIGEN(NB,NM,BMM,BDM,BRN,BAN,ZDM,ZVN,ZPN,ZGN,EFN,W2)
   IMPLICIT REAL*8 (A-H,O-Z)
   DIMENSION BMM(NB),BDM(NB),BRN(NB),BAN(NB)
   DIMENSION ZDM(NB),ZVN(NB),ZPN(NB),ZGN(NB),EFN(NB)
   DIMENSION A(512),B(512),C(512),D(512),E(512)
   DATA PIE/3.1415926536/,GCCG/6.670E-08/
   IF(NM)10,20,30
10  NS=-NM
     N=50
     DO 25 I=1,N
       A(I)=-1
       B(I)=2+1.0/I
       C(I)=A(I)
25  CONTINUE
     GO TO 40
30  CONTINUE
     N=NB-4
     NS=NM
     DO 1 J=1,N
       I=J+1
       IB=I+1
       A(J)=0.0
       B(J)=0.0
       C(J)=0.0
       ARM=BAN(IB)/SQRT(BDM(IB))
       DRM1=+BAN(IB-1)/ZVN(IB-1)/ZDM(IB-1)/SQRT(BDM(IB-1))
       A(J)=-ARM*ZGN(IB-1)*ZPN(IB-1)*DRM1
       DR1=+BAN(IB)/ZVN(IB)/ZDM(IB)/SQRT(BDM(IB))
       DRM2=-BAN(IB)/ZVN(IB-1)/ZDM(IB-1)/SQRT(BDM(IB))
       B(J)=-4.0*GCCG*BMM(IB)/BRN(IB)/BRN(IB)/BRN(IB)
       *+ARM*(ZGN(IB)*ZPN(IB)*DR1-ZGN(IB-1)*ZPN(IB-1)*DRM2)
       DR2=-BAN(IB+1)/ZVN(IB)/ZDM(IB)/SQRT(BDM(IB+1))
       C(J)=+ARM*ZGN(IB)*ZPN(IB)*DR2
1    CONTINUE
40  CONTINUE

```

```

C  WRITE(6,*)' DIAGONAL MATRIX ELEMENTS - A,B,C'
C  WRITE(6,16)(A(J),J=1,N)
C  WRITE(6,16)(B(J),J=1,N)
C  WRITE(6,16)(C(J),J=1,N)
16  FORMAT(5E15.6)
    ABC=0.0
    DO 7 J=1,N
      ABC=MAX(ABC,ABS(A(J)),ABS(B(J)),ABS(C(J)))
7  CONTINUE
    DO 8 J=1,N
      A(J)=A(J)/ABC
      B(J)=B(J)/ABC
      C(J)=C(J)/ABC
8  CONTINUE
C  WRITE(6,*)' NORMALIZED DIAGONAL MATRIX ELEMENTS - ABC,A,B,C'
C  WRITE(6,16)ABC
C  WRITE(6,16)(A(J),J=1,N)
C  WRITE(6,16)(B(J),J=1,N)
C  WRITE(6,16)(C(J),J=1,N)
    NZ=N
    CALL EIGENS(NZ,NS,ABC,A,B,C,D,E,EV)
    IF(NM)60,60,50
50  CONTINUE
    DO 9 J=1,N
      I=J+1
      IB=I+1
      E(J)=E(J)*SQRT(BDM(N+2)/BDM(IB))
9  CONTINUE
60  CONTINUE
    W2=EV*ABC
    W1=SQRT(W2)
    PS=2*PIE/W1
    PM=PS/60
    PH=PM/60
    PD=PH/24
C  WRITE(6,*)' NORM.FACTOR, EIGENVALUE, ANG.FREQ. W**2 AND W'
C  WRITE(6,16)ABC,EV,W2,W1
C  WRITE(6,*)' PERIOD IN SECONDS, MINUTES, HOURS, DAYS'
C  WRITE(6,16)PS,PM,PH,PD
C  WRITE(6,*)' EIGENFUNCTION'
C  WRITE(6,16)(E(J),J=1,N)
    EFN(1)=0.0
    EFN(2)=0.0
    DO 70 J=1,N
      EFN(J+2)=E(J)
70  CONTINUE
    M=N+2
    DO 80 J=M,NB
      EFN(J)=EFN(M)
80  CONTINUE
20  CONTINUE
    RETURN
    END
C  EIGENS - SUBROUTINE USING STURM METHOD
SUBROUTINE EIGENS(NZ,NS,ABC,A,B,C,D,E,EV)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(NZ),B(NZ),C(NZ),D(NZ),E(NZ)
DIMENSION F(512),G(512)
DATA PIE/3.1415926536/,EPS/1.0E-6/
    N=NZ
    A(1)=0
    C(N)=0
    EV=0
    DD=1
3  CALL STURMS(N,M,EV,A,B,C,D,E,F,G)
C  WRITE(6,16)N,M,EV,F(N),G(N)
16  FORMAT(2I5,5E14.5)
    IF(M-NS)1,1,2
1  EV=EV+DD
    GO TO 3
2  CONTINUE
    DDD=EV/NS/10
    EV=0
    DO 4 I=1,NS
      K=I-1
      DD=DDD
8  CALL STURMS(N,M,EV,A,B,C,D,E,F,G)

```

```

C  WRITE(6,16)I,M,EV,F(N),G(N),DE,DD
   IF(M-K)5,6,7
5  CONTINUE
   EV=EV+DD
   GO TO 8
7  DD=DD/2
   EV=EV-DD
   GO TO 8
6  CONTINUE
   IF(EV)50,50,60
50 CONTINUE
   DD=DD/2.0
   GO TO 5
60 CONTINUE
   DE=-F(N)/G(N)
   IF(ABS(DE)-EPS*ABS(EV))11,9,9
9  IF(ABS(DE)-DD)12,12,10
10 DE=DD
12 EV=EV+DE
   GO TO 8
11 CONTINUE
   W2=ABC*EV
   W1=SQRT(W2)
   PS=2*PIE/W1
   PM=PS/60
   PH=PM/60
   PD=PH/24
   WRITE(6,*)' I,M,ABC,EV,W2,W1'
   WRITE(6,71)I,M,ABC,EV,W2,W1
   WRITE(6,*)' I,M,PS,PM,PH,PD'
   WRITE(6,71)I,M,PS,PM,PH,PD
71 FORMAT(2I5,4E15.5)
   CALL TRIDDEF(N,EV,A,B,C,D,E)
C  WRITE(6,*)' EFN'
C  WRITE(6,26)(E(I),J=1,N)
26 FORMAT(5E15.6)
4  CONTINUE
   RETURN
   END
SUBROUTINE STURMS(N,M,EV,A,B,C,D,E,F,G)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(N),B(N),C(N),D(N),E(N)
DIMENSION F(N),G(N)
M=0
FF=1
DO 10 I=1,N
IF(I-2)1,2,3
1  F(I)=B(I)-EV
   G(I)=-1
   GO TO 4
2  F(I)=(B(I)-EV)*F(I-1)-A(I)*C(I-1)
   G(I)=(B(I)-EV)*G(I-1)-F(I-1)
   GO TO 4
3  F(I)=(B(I)-EV)*F(I-1)-A(I)*C(I-1)*F(I-2)
   G(I)=(B(I)-EV)*G(I-1)-A(I)*C(I-1)*G(I-2)-F(I-1)
4  IF(I-N)8,6,6
8  IF(F(I))5,6,5
5  IF(F(I)*FF)7,6,6
7  M=M+1
   FF=-FF
6  CONTINUE
10 CONTINUE
   RETURN
   END
C  SOLVE TRIDIAGONAL MATRIX EQUATION
SUBROUTINE TRIDDEF(N,EV,A,B,C,D,E)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(N),B(N),C(N),D(N),E(N)
DIMENSION BB(512),CC(512),DD(512)
DO 1 I=1,N
BB(I)=B(I)-EV
D(I)=0
1  CONTINUE
CC(1)=C(1)/BB(1)
DD(1)=D(1)/BB(1)
DO 2 I=2,N
BB(I)=BB(I)-A(I)*CC(I-1)

```

```

      CC(I)=C(I)/BB(I)
      DD(I)=(D(I)-A(I)*DD(I))/BB(I)
2    CONTINUE
      E(N)=1
      DO 3 J=1,N-1
        I=N-J
        E(I)=DD(I)-CC(I)*E(I+1)
3    CONTINUE
      RETURN
      END
C    MATENCOS
C    MODEL STELLAR ATMOSPHERE, ENVELOPE AND CORE SUBROUTINE
C    USES EDDINGTON APPROXIMATION IN ATMOSPHERE
C    RUNGE-KUTTA FOURTH ORDER INTEGRATION INWARDS
C    INTEGRATION TERMINATES IF MASS, RADIUS OR LUMINOSITY NEGATIVE
C    ALL PHYSICS THROUGH PANDER SUBROUTINE
C    KRAD=0 FOR RADIATIVE, KRAD>0 FOR CONVECTIVE EQUILIBRIUM
C    T. R. CARSON : 6 JULY 1994
      SUBROUTINE MODENV(KRAD,NDS1,NDS2,KPRT,STARMS,STARLL,TEFFL,X,Y,Z)
C    SUBROUTINE MODENV(NDS1,NDS2,KPRT,STARMS,STARLL,TEFFL,X,Y,Z)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SVS(8,1024)
      DIMENSION RKF(4),RKW(4)
      DATA VLC/2.997925E10/,RCA/7.56495E-15/,GCG/6.67000E-8/
      DATA AVN/6.02252E23/,BCK/1.38054E-16/,PIE/3.1415926536/
      DATA SUNM/1.989E33/,SUNL/3.900E33/,SUNR/6.960E10/
C    DATA STARMS/1.0/,STARLL/0.0/,TEFFL/3.800/
C    DATA X/0.700/,Y/0.280/,Z/0.020/
      DATA NA/64/,NE/128/,LE/512/
      DATA RKF/0.0,0.5,0.5,1.0/,RKW/1.0,2.0,2.0,1.0/
C    KRAD=1
      WRITE(6,*) 'MODENV : BEGIN MODEL ATMOSPHERE/ENVELOPE'
      KPMOD=MAX(1,KPRT)
      OS=0.4
      AK=AVN*BCK
      SIGMA=RCA*VLC/4.0
      DTPC=3.0/64.0/PIE/SIGMA/GCG
      STARM=STARMS*SUNM
      STARLS=10.0**STARLL
      STARL=STARLS*SUNL
      TEFFM=10.0**TEFFL
      STARR=SQRT(STARL/TEFFM**4/4/PIE/SIGMA)
      STARRS=STARR/SUNR
      GRAVP=GCG*STARM/STARR**2
      WRITE(6,161)NA,STARM,STARL,TEFFM,STARR
      TAUP=2.0/3.0
      DTAU=TAUP/NA
      TAU=0.0
      T0=TEFFM/SQRT(SQRT(2.0))
      D0=0.0
      PR=RCA*T0**4/3.0
      P0=PR
      O0=OS*(1.0+X)/2
      RD=0.0
      RM=0.0
      T=T0
      D=D0
      P=P0
      O=O0
      R=STARR
      FM=STARM
      FL=STARL
      DO 1 IA=1,NA
        TI=T
        PI=P
        RI=R
        FMI=FM
        FLI=FL
        TAU1=TAU
        DPI=0.0
        DP=0.0
        DRD=0.0
        DRM=0.0
        DO 2 K=1,4
          IF(K-1)3,3,4
4    CONTINUE
        TAU=TAU1+RKF(K)*DTAU

```

```

T=SQRT(SQRT((1.0+3.0*TAU/2.0)/2.0))*TEFFM
P=PI+RKF(K)*DPI
C  CALL PANDER(1,0,-1,1,0,T,D,P,U,S,O,E,
CALL PANDER(1,0,-1,X,Y,Z,T,D,P,U,S,O,E,
1 DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,
2 DD1,DD2,DD3,DD4,DD5,DD6)
3  CONTINUE
DPI=DTAU*GRAVP/O
DP=DP+RKW(K)*DPI/6.0
C  WRITE(6,161)K,TAU,T,P,D,O,DPI,DP
2  CONTINUE
P=PI+DP
C  CALL PANDER(1,0,-1,1,0,T,D,P,U,S,O,E,
CALL PANDER(1,0,-1,X,Y,Z,T,D,P,U,S,O,E,
1 DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,
2 DD1,DD2,DD3,DD4,DD5,DD6)
DRD=DTAU/D/O
RD=RD+DRD
C  R=STARR
DRM=4.0*PIE*R**2*DTAU/O
RM=RM+DRM
C  WRITE(6,161)IA,T,D,P,O,RD,RM,TAU
FM=FM-DRM
R=R-DRD
SVS(1,IA)=FM
SVS(2,IA)=R
SVS(3,IA)=STARL
SVS(4,IA)=T
SVS(5,IA)=P
SVS(6,IA)=D
SVS(7,IA)=O
SVS(8,IA)=E
IF(KPRT)1,1,5
5  CONTINUE
WRITE(6,161)IA,T,D,P,R,FM,FL,O,E
1  CONTINUE
C  WRITE(6,161)NA,T,D,P,O,RD,RM,TAU
TP=T
DP=D
PP=P
OP=O
RP=R
FMP=STARM-RM
FLP=STARL
PPL=LOG(PP)
PCF=100.0
PCE=PCF*3.0/8.0/PIE*GRAVP**2/GCG
PCL=LOG(PCE)
DPL=(PCL-PPL)/NE
C  WRITE(6,161)NE,PP,PCE,PPL,PCL,DPL
PL=PPL
FM=FMP
FL=FLP
R=RP
T=TP
E=0.0
GRAVR=GRAVP
DTPR=DTPC*O*P/T**4*(FL/FM)
DTPA=DTPR
DAD=DTPA
DO 11 IE=1,LE
FMI=FM
FLI=FL
RI=R
TI=T
PLI=PL
PI=P
DFM=0.0
DFL=0.0
DR=0.0
DT=0.0
DFMI=0.0
DFLI=0.0
DRI=0.0
DTI=0.0
DO 12 K=1,4
IF(K-1)13,13,14

```

```

14 CONTINUE
  FM=FMI+RKF(K)*DFMI
  IF(FM)16,16,114
114 CONTINUE
  FL=FLI+RKF(K)*DFLI
  IF(FL)16,16,214
214 CONTINUE
  R=RI+RKF(K)*DRI
  IF(R)16,16,314
314 CONTINUE
  T=TI+RKF(K)*DTI
  PL=PLI+RKF(K)*DPL
  P=EXP(PL)
  NDV=0
  IF(KRAD)140,140,141
141 NDV=2
140 CONTINUE
C CALL PANDER(1,0,-1,1,1,T,D,P,U,S,O,E,
C CALL PANDER(1,0,-1,X,Y,Z,T,D,P,U,S,O,E,
  CALL PANDER(1,NDV,-1,X,Y,Z,T,D,P,U,S,O,E,
  1 DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,
  2 DD1,DD2,DD3,DD4,DD5,DD6)
  DAD=0.0
  IF(KRAD)130,130,131
131 CONTINUE
  DDDP=1.0/DPD
  DDDT=-DPT/DPD
  DVDP=-DDDP
  DVDT=-DDDT
  DEDP=DED/DPD
  DEDT=DET+DED*DDDT
  DODP=DOD/DPD
  DODT=DOT+DOD*DDDT
  DUDP=DUD/DPD
  DUDT=DUT+DUD*DDDT
  DSDP=DSD/DPD
  DSdT=DST+DSD*DDDT
  CV=U/T*DUT
  CP=CV+P/D/T*DPT*DPT/DPD
  G0=CP/CV
  G1=G0*DPD
  G2=G0/(G0-(G0-1.0)/DPT)
  G3=1.0+(G0-1.0)*DPD/DPT
  DAD=(G2-1.0)/G2
  DTPA=DAD
130 CONTINUE
13 CONTINUE
  GRAVR=CCG*FM/R**2
  DFMI=-4.0*PIE/GRAVR*R*R*DPL
  DFLI=DFMI*E
  DRI=-P*DPL/D/GRAVR
  DTPR=DTPC*O*P/T**4*(FL/FM)
  DTP=DTPR
  DTPAZ=0.4
  IF(KRAD)120,120,121
C121 DTP=MIN(DTPR,DTPA)
121 DTP=MIN(DTPR,DTPAZ)
120 CONTINUE
  DTI=T*DTP*DPL
  DFM=DFM+RKW(K)*DFMI/6.0
  DFL=DFL+RKW(K)*DFLI/6.0
  DR=DR+RKW(K)*DRI/6.0
  DT=DT+RKW(K)*DTI/6.0
12 CONTINUE
  T=T+DT
C PL=PLI+DPL
C P=EXP(PL)
  NDV=0
  IF(KRAD)240,240,241
241 NDV=2
240 CONTINUE
C CALL PANDER(1,0,-1,1,1,T,D,P,U,S,O,E,
C CALL PANDER(1,0,-1,X,Y,Z,T,D,P,U,S,O,E,
  CALL PANDER(1,NDV,-1,X,Y,Z,T,D,P,U,S,O,E,
  1 DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,
  2 DD1,DD2,DD3,DD4,DD5,DD6)
  DAD=0.0

```

```

IF(KRAD)230,230,231
231 CONTINUE
DDDP=1.0/DPD
DDDT=-DPT/DPD
DVDP=-DDDP
DVDT=-DDDT
DEDP=DED/DPD
DEDT=DET+DED*DDDT
DODP=DOD/DPD
DODT=DOT+DOD*DDDT
DUDP=DUD/DPD
DU DT=DUT+DUD*DDDT
DSDP=DSD/DPD
DSDT=DST+DSD*DDDT
CV=U/T*DUT
CP=CV+P/D/T*DPT*DPT/DPD
G0=CP/CV
G1=G0*DPD
G2=G0/(G0-(G0-1.0)/DPT)
G3=1.0+(G0-1.0)*DPD/DPT
DAD=(G2-1.0)/G2
DTPA=DAD
230 CONTINUE
FM=FM+DFM
FL=FL+DFL
DTPR=DTPC*O*P/T**4*(FL/FM)
R=R+DR
IF(KPRT)150,150,15
15 CONTINUE
IF(MOD(IE,KPMOD))150,151,150
151 CONTINUE
WRITE(6,161)IE,T,D,P,R,FM,FL,O,E,DTPR,DTPA
150 CONTINUE
NE=IE
ME=IE+NA
IF(ME-LE)116,216,216
216 CONTINUE
WRITE(6,*)' MODENV : ENVELOPE EXHAUSTS ARRAY '
GO TO 16
116 CONTINUE
IF(FM)16,17,17
17 CONTINUE
IF(FL)16,18,18
18 CONTINUE
IF(R)16,19,19
16 WRITE(6,161)IE,T,D,P,R,FM,FL,O,E,DTP,DTPA
R=0.0
FM=0.0
FL=0.0
FMC=FM
FLC=FL
RC=R
GRAVC=GCG*FMI/RI/RI
C DO NOT TRY TO ESTIMATE CENTRAL PRESSURE
C DPC=3.0/8.0/PIE*GRAVC/RI/RI*FMI
C P=PI+DPC
C PCL=LOG(P)
C DPCL=PCL-PLI
DPCL=DPL
PCL=PLI+DPCL
P=EXP(PCL)
DTCL=DPCL*DTP
TCL=LOG(TI)+DTCL
T=EXP(TCL)
TC=T
PC=P
DC=D
19 CONTINUE
NE=IE
ME=IE+NA
SVS(1,ME)=FM
SVS(2,ME)=R
SVS(3,ME)=FL
SVS(4,ME)=T
SVS(5,ME)=P
SVS(6,ME)=D
SVS(7,ME)=O

```

```

SVS(8,ME)=E
IF(R)20,20,11
11 CONTINUE
20 CONTINUE
C FMC=FM
C FLC=FL
C RC=R
C TC=T
C PC=P
WRITE(6,161)ME,TC,DC,PC,RC,FMC,FLC
WRITE(6,*)' MODENV : SUMMARY : '
N1=1
WRITE(6,161)N1,(SVS(J,N1),J=1,8)
NP=NA
WRITE(6,161)NP,(SVS(J,NP),J=1,8)
MC=NA+NE-1
WRITE(6,161)MC,(SVS(J,MC),J=1,8)
NC=NA+NE
WRITE(6,161)NC,(SVS(J,NC),J=1,8)
FRM=SVS(1,MC)/SVS(1,NA)
FRR=SVS(2,MC)/SVS(2,NA)
FRL=SVS(3,MC)/SVS(3,NA)
WRITE(6,*)' MODENV : MASS, RADIUS, LUMINOSITY FRACTIONS'
WRITE(6,161)MC,FRM,FRR,FRL
WRITE(6,161)NC,STARM,STARR,STARL,GRAVP,TP,DP,PP,TC,DC,PC,
1X,Y,Z
IF(KPRT)21,21,22
22 CONTINUE
DO 23 IS=1,NC,KPRT
WRITE(6,161)IS,(SVS(JS,IS),JS=1,8)
23 CONTINUE
21 CONTINUE
IF(NDS1)31,31,32
32 CONTINUE
WRITE(6,*)' MODENV : WRITING ENVELOPE TO UNIT NDS1'
WRITE(NDS1)NC,NA,STARM,STARR,STARL,GRAVP,TP,DP,PP,TC,DC,PC,
1X,Y,Z
DO 33 IS=1,NC
WRITE(NDS1)NC,IS,(SVS(JS,IS),JS=1,8)
33 CONTINUE
REWIND NDS1
31 CONTINUE
RETURN
161 FORMAT(I10,(T11,1P10E10.3))
END
SUBROUTINE MODEL(N,K,SMS,SRS,SV)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION VM(512),VR(512),VP(512),VL(512),VT(512),VD(512)
DIMENSION SV(512,20)
DATA PIE/3.1415926536/,GCC/6.670E-8/,CVL/2.997925E+10/
DATA AVN/6.02350E23/,BCK/1.38054E-16/,ARC/7.56495E-15/
DATA SUNM/1.989E+33/,SUNR/6.960E+10/,SUNL/3.900E+33/
WRITE(6,*)' CALCULATING SIMPLE MODEL.'
SIG=ARC*CVL/4
B=1.0
W=1.0
K=256
H=1.0/K
SM=SMS*SUNM
IF(SRS)1,1,2
1 SRS=SQRT(SMS)
2 SR=0.75*SRS*SUNR
DA=0.75*SM/SR**3/PIE
D0=(3*N+1)*DA
SLS=SMS**3
SL=0.75*SLS*SUNL
TE4=SL/SR/SR/SIG/PIE/4
PRE=ARC*TE4/3
TE=SQRT(SQRT(TE4))
PGE=AVN*BCK*DE
PE=PRE+PGE
PC=3*GCC/8/PIE*(SM/SR/SR)**2
PC=(3*N+1)*(4*N+1)*PC/(N+1)/(2*N+1)
TC=W*PC/AVN/BCK/D0
WRITE(6,6)K,SM,SR,SL,TC,PC,D0
SCD=1.0
SCP=1.0

```

```

SCT=1.0
DO 3 I=1,K
X=(I-1)*H
VR(I)=SR*X
DF=1.0
IF(N)10,11,12
12 DF=DF-X**(1.0/N)
11 VD(I)=D0*DF
FM=1.0
IF(N)10,21,22
22 FM=FM-3*N*X**(1.0/N)/(3*N+1)
21 VM(I)=(3*N+1)*SM*FM*X**3
PF=(N+1)*(2*N+1)*(3*N+1)
IF(N)10,31,32
32 PF=PF-2*N*(N+1)*(6*N+1)*X**(1.0/N)+3*N**2*(2*N+1)*X**(2.0/N)
31 VP(I)=PC*(1.0-PF*X**2/(4*N+1))
VT(I)=W*VP(I)/AVN/BCK/VD(I)
C PR=ARC/3*VT(I)**4
C VP(I)=VP(I)+PR
VL(I)=SL
C WRITE(6,6)I,VM(I),VR(I),VP(I),VL(I),VT(I),VD(I)
J=K+1-I
SV(J,1)=VM(I)
SV(J,2)=VR(I)
SV(J,3)=VL(I)
SV(J,4)=SCT*VT(I)
SV(J,5)=SCP*VP(I)
SV(J,6)=SCD*VD(I)
10 CONTINUE
3 CONTINUE
SV(K-1,3)=0.25*SV(K-1,3)
SV(K-2,3)=0.50*SV(K-2,3)
SV(K-3,3)=0.75*SV(K-3,3)
DO 4 I=1,K
SV(I,4)=SV(I,4)-SV(I,4)+TE
C PR=RCA*SV(I,4)**4/3
C SV(I,5)=SV(I,5)+PR
4 CONTINUE
RETURN
6 FORMAT(I5,1P6E10.3)
END

```

'trceost.f'

```

C PANDER - PHYSICS AND DERIVATIVES
C*****WARNING - TO USE THIS PACKAGE:
C NOTE CALL SEQUENCE - USE ONLY WITH MATCHING KWHSP
C TO SELECT EOS SUBROUTINE PLANT REQUIRED VALUE OF NEL IN CONEQS-
C NEL=0 OR 1 SELECTS ROUGH AND READY(FAST) ROUTINE: SAHA OF 1 ELEMENT (H)
C NEL=2 OR 3 SELECTS ESTATENN- FULL SAHA WITH NN PART. FNS. OF H,HE,O
C NEL=4 OR 5 SELECTS ESTATE - FULL SAHA WITH NN PART. FNS. OF H,HE,C,N,O
C INCLUDES FORMATION OF IONS H- C- O- AND MOLECULES H2, N2 AND CO
C IN LATTER CASE PLANT RELATIVE MASS FRACTIONS OF C,N,O - CCM,CNM,COM
C NEL>5 WILL INTERPOLATE P,U,S FROM TABLES - NOTE MODIFICATIONS OF TABLES
C IN LATTER CASE DATA SET NUMBERS NDP=NDU=NDS=0 MUST BE REMOVED/REPLACED-
C THE DEFAULT VALUES IN DATA STATEMENTS ARE 91,92,93 - STANDARD FORMATS
C OPACITY IS FROM CHRISTY TABLE FOR IO=KO=1
C OPACITY IS FROM IGLESIAS-ROGERS POP1 AND POP2 TABLES FOR IO=KO=2
C OPACITY IS FROM IGLESIAS-ROGERS 30 OPAL TABLES FOR IO=KO=3 (opal.t.d)
C OPACITY IS FROM CARSON ZEUS TABLES FOR Z=0 (TO BE IMPLEMENTED)
C TO INTERPOLATE OPACITY TABLE - SET NDO NON-ZERO: DEFAULT IS 94
C LOW T MOLECULAR OPACITIES FOR POPS 1,2,3,4 ARE SPLICED (TTR,TFW)
C ENERGY GENERATION COULD BE TABLED BUT FORMULA BEST FOR CHANGING X,Y,Z
C FOR FASTER RUNNING SET/CALL ND=1 IN PANDER(BOTH D/DT AND D/DD STILL GIVEN)
C SECOND DERIVATIVES (DVDTT.....DADAD) STILL TO BE FULLY IMPLEMENTED
C FINAL GOAL IS SAME PHYSICS PACKAGE FOR STRUCTURE, EVOLUTION AND PULSATION
C AS FROM TODAY 12 APRIL 1993 THIS IS NOW POSSIBLE
C THIS UPDATE OF OPAL TABLES AND SPLICED MOLECULAR TABLES DATED 29 JUNE 1993
C*****IF IN DOUBT SEE T.R.C. - END OF WARNING!
SUBROUTINE PANDER(LM,ND,KS,X,Y,Z,TV,DV,PV,UV,SV,OV,EV,
1 DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,
C 2 CV,CP,AG0,AG1,AG2,AG3,DAD)
2 DVDTT,DVDTD,SHCPT,SHCPD,DADAT,DADAD)

C VERSION II
C 2 DVDTT,DVDTD,SHCPT,SHCPD,DADAT,DADAD,ZAC)

```

```

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION T(3),D(3),P(3),U(3),S(3),O(3),E(3)
DIMENSION DVDT(3),SHCP(3),DADA(3)

C  VERSION II
C  DIMENSION DVDT(3),SHCP(3),DADA(3),ZAC(8)

DATA ONE/1.0/,WM/1.0/,EPS/1.0E-3/
DATA KO/1/,KE/1/
EXP(X)=DEXP(X)
ALOG(X)=DLOG(X)
LMV=LM
C  TO RESTRICT TO 1 EXTRA PASS FOR D/DT SET ND=1; D/DD STILL FROM ITERATION
IF(LMV)29,20,21
29 STOP 29
20 CONTINUE
T(1)=EXP(TV)
D(1)=EXP(DV)
P(1)=EXP(PV)
GO TO 22
21 CONTINUE
T(1)=TV
D(1)=DV
P(1)=PV
22 CONTINUE
DO 100 K=1,ND+1
IF(K-2)1,2,3
1  KT=KS
GO TO 9
2  T(2)=T(1)*(1.0+EPS)
D(2)=D(1)
KT=+1
GO TO 9
3  T(3)=T(1)
D(3)=D(1)*(1.0+EPS)
KT=+1
GO TO 9
9  CONTINUE
CALL CONEQS(KT,KO,KE,X,Y,Z,T(K),D(K),P(K),U(K),S(K),O(K),E(K),
C 1 AM,AN,EN,WM,ETA,DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,
C 1 AM,AN,EN,WM,ETA,GPT,DPD,GUT,DUD,GST,DSD,GOT,DOD,GET,DED,
1 AM,AN,EN,WM,ETA,GPT,GPD,GUT,GUD,GST,GSD,GOT,GOD,GET,GED,
C 1 AM,AN,EN,WM,ETA,DPT,DPD,DUT,DUD,DST,DSD,GOT,GOD,GET,GED,
2 DVDT(K),SHCP(K),DADA(K))

C  VERSION II
C  2 DVDT(K),SHCP(K),DADA(K),ZAC)

IF(K-2)11,12,13
11 IF(LMV)119,110,111
119 STOP 119
110 CONTINUE
DV=ALOG(D(1))
PV=ALOG(P(1))
OV=ALOG(O(1))
IF(E(1))1100,1100,1101
1100 EV=-99
GO TO 1102
1101 CONTINUE
EV=ALOG(E(1))
1102 CONTINUE
GO TO 112
111 CONTINUE
DV=D(1)
PV=P(1)
OV=O(1)
EV=E(1)
112 CONTINUE
C  BEWARE OF QUANTITIES WHICH MAY BE NEGATIVE!
UV=U(1)
SV=S(1)
DPD=GPD
DUD=GUD
DSD=GSD
DOD=GOD
DED=GED

```

```

GO TO 19
12 DPT=(P(K)/P(1)-1.0)/EPS
   DUT=(U(K)/U(1)-1.0)/EPS
   DST=(S(K)/S(1)-1.0)/EPS
   DOT=(O(K)/O(1)-1.0)/EPS
C   DET=(E(K)/E(1)-1.0)/EPS
   DVDTT=(DVDT(K)-DVDT(1))/EPS
   SHCPT=(SHCP(K)-SHCP(1))/EPS
   DADAT=(DADA(K)-DADA(1))/EPS
C   CV=U(1)*DUT/T(1)
C   CP=CV+P(1)/T(1)/D(1)*DPT*DPT/DPD
C   AG0=CP/CV
C   AG1=AG0*DPD
C   AG2=AG0/(AG0-(AG0-ONE)/DPT)
C   AG3=ONE+(AG0-ONE)*DPD/DPT
C   DAD=(AG2-ONE)/AG2
GO TO 19
13 DPD=(P(K)/P(1)-1.0)/EPS
   DUD=(U(K)/U(1)-1.0)/EPS
   DSD=(S(K)/S(1)-1.0)/EPS
   DOD=(O(K)/O(1)-1.0)/EPS
C   DED=(E(K)/E(1)-1.0)/EPS
   DVDTD=(DVDT(K)-DVDT(1))/EPS
   SHCPD=(SHCP(K)-SHCP(1))/EPS
   DADAD=(DADA(K)-DADA(1))/EPS
C   CV=U(1)*DUT/T(1)
C   CP=CV+P(1)/T(1)/D(1)*DPT*DPT/DPD
C   AG0=CP/CV
C   AG1=AG0*DPD
C   AG2=AG0/(AG0-(AG0-ONE)/DPT)
C   AG3=ONE+(AG0-ONE)*DPD/DPT
C   DAD=(AG2-ONE)/AG2
GO TO 19
19 CONTINUE
C   WRITE(6,16)T(K),D(K),P(K),U(K),S(K),O(K),E(K),AM,AN,EN,WM,ETA
C   WRITE(6,16)DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED
C   WRITE(6,16)GPT,GPD,GUT,GUD,GST,GSD,GOT,GOD,GET,GED
C   WRITE(6,16)DVDT(K),SHCP(K),DADA(K)
16 FORMAT(1P12E10.3)
100 CONTINUE
RETURN
END
C   PHYSICS - CONSTITUTIVE EQUATIONS
SUBROUTINE PVECTR(KS,KO,KE,XHY,YHE,ZMZ,V,Z)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION V(5),Z(5)
DATA WM/1.0/
CALL CONEQS(KS,KO,KE,XHY,YHE,ZMZ,V(4),Z(1),V(5),U,S,Z(3),Z(2),
1 AM,AN,EN,WM,ETA,DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,
2 DVDT,SHCP,DADA)
RETURN
END
SUBROUTINE CONEQS(KS,KO,KE,X,Y,Z,T,D,P,U,S,O,E,AM,AN,EN,WM,ETA,
1 DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,DVDT,SHCP,DAD)

C   VERSION II
C   1 DPT,DPD,DUT,DUD,DST,DSD,DOT,DOD,DET,DED,DVDT,SHCP,DAD,ZAC)

IMPLICIT REAL*8 (A-H,O-Z)

C   VERSION II
C   DIMENSION ZAC(8)
ABS(X)=DABS(X)
DATA RCA/7.56495E-15/,AVN/6.02252E23/,BCK/1.38054E-16/
DATA ONE/1.0/,TEN/10.0/,CEN/100.0/,EPD/1.0E-6/
TK=BCK*T
PR=RCA*T**4/3.0
IO=KO
IE=KE
DC=D
ITN=0
KDD=0
EDD=ONE
IF(KS)15,10,17
15 CONTINUE
D=WM*(P-PR)/AVN/TK
11 DC=D

```

```

ITN=ITN+1
IF(ITN-20)17,17,18
18 WRITE(6,16)ITN,T,D,DC,P,PC,WM,DPDD,DD,ETA,EDD
16 FORMAT(I10,1P10E10.3)
GO TO 14
17 CONTINUE
IF(DC)170,170,171
170 WRITE(6,16)ITN,T,D,DC,P,PC,WM,DPDD,DD,ETA,EDD
171 CONTINUE
NEL=0
NEL=1
NEL=3
NEL=5
CALL ESTATE (NEL,X,Y,Z,T,DC,PC,U,S,AM,AN,EN,WM,ETA,
1 DPT,DPD,DUT,DUD,DST,DSD)

C VERSION II
C 1 DPT,DPD,DUT,DUD,DST,DSD,ZAC)

C 1 DPT,GPD,DUT,GUD,DST,GSD)
IF(KS)172,172,180
172 CONTINUE
C IF(EDD-CEN*EPD)180,180,181
180 CONTINUE
IF(KO)200,200,201
201 CONTINUE
CALL OPACTY(IO,X,Y,Z,T,D,EN,O,DOT,DOD)
C CALL OPACTY(IO,X,Y,Z,T,D,EN,O,DOT,GOD)
200 CONTINUE
IF(KE)300,300,301
301 CONTINUE
C E AND DERIVATIVES ALSO NOW DIRECTLY AVAILABLE FROM ENUC
CALL ENERGY(IE,X,Y,Z,T,D,E,DET,DED)
C CALL ENERGY(IE,X,Y,Z,T,D,E,DET,GED)
300 CONTINUE
KDD=KDD+1
181 CONTINUE
IF(KS)12,10,13
12 IF(ITN-1)120,120,121
120 DPDD=AVN*TK/WM
GO TO 122
121 IF(DD)127,122,127
127 DPDD=(PC-PCP)/DD
DUDD=(U-UP)/DD
DSDD=(S-SP)/DD
DODD=(O-OP)/DD
C DEDD=(E-EP)/DD
122 DD=(P-PC)/DPDD
123 CONTINUE
D=DC+DD
IF(D)124,125,126
124 CONTINUE
125 DD=DD/10.0
GO TO 123
126 CONTINUE
PCP=PC
UP=U
SP=S
OP=O
EP=E
EDD=ABS(DD/DC)
IF(EDD-EPD)14,11,11
14 CONTINUE
IF(KDD-1)11,11,19
19 CONTINUE
C WRITE(6,16)ITN,T,D,DC,P,PC,WM,DPDD,DD
DPD=D/P*DPDD
DUD=D/U*DUDD
DSD=D/S*DSDD
DOD=D/O*DODD
C DED=D/E*DEDD
GO TO 10
13 P=PC
GO TO 10
10 IF(KO)20,20,21
21 CONTINUE
C NOW IMMEDIATELY AFTER ESTATE

```

```

C CALL OPACTY(IO,X,Y,Z,T,D,EN,O,DOT,DOD)
C CALL OPACTY(IO,X,Y,Z,T,D,EN,O,OT,OD)
20 IF(KE)30,30,31
31 CONTINUE
C NOW IMMEDIATELY AFTER ESTATE
C CALL ENERGY(IE,X,Y,Z,T,D,E,DET,DED)
30 CONTINUE
   DDDP=ONE/DPD
   DDDT=-DPT/DPD
   DVDP=-DDDP
   DVDT=-DDDT
   DUDD=DUD/DPD
   DUDT=DUT+DUD*DDDT
C   DSDP=DSD/DPD
C   DSDT=DST+DSD*DDDT
C   DODP=DOD/DPD
C   DODT=DOT+DOD*DDDT
C   DEDP=DED/DPD
C   DEDT=DET+DED*DDDT
   SHCV=U*DUT/T
   SHCP=SHCV+P/T/D*DPT*DPT/DPD
   AG0=SHCP/SHCV
   AG1=AG0*DPD
   AG2=AG0/(AG0-(AG0-ONE)/DPT)
   AG3=ONE+(AG0-ONE)*DPD/DPT
   DAD=(AG2-ONE)/AG2
C   WRITE(6,16)ITN,AM,AN,EN,WM,ETA,EDD
   RETURN
   END
C EQUATION OF STATE
SUBROUTINE ESTATE(NEL,X,Y,Z,T,D,P,U,S,AM,AN,EN,WM,ETA,
1 DPT,DPD,DUT,DUD,DST,DSD)

C VERSION II
C 1 DPT,DPD,DUT,DUD,DST,DSD,ZAC)

   IMPLICIT REAL*8 (A-H,O-Z)
   DIMENSION INZ(10),JMIN(10),JMAX(10)

C VERSION II
C DIMENSION INZ(10),JMIN(10),JMAX(10),ZAC(8)

   DIMENSION ANI(10),FF(10),ANF(10),CPF(10)
   DIMENSION VI(10,10),SL(10,10),SM(10),RL(10,10)
   DIMENSION PFL(10,10),EEV(10,10),SV(10,10)
   DIMENSION XAP(64),YAP(64),ZAP(64,64)
   DIMENSION XAU(64),YAU(64),ZAU(64,64)
   DIMENSION XAS(64),YAS(64),ZAS(64,64)
   DATA ILQ/2/,ITP/0/,NDP/91/,INP/0/,ENP/8.0/
   DATA ITU/0/,NDU/92/,INU/0/,ENU/8.0/
   DATA ITS/0/,NDS/93/,INS/0/,ENS/8.0/
   DATA AVN/6.02252E23/BCK/1.38054E-16/VLC/2.997925E10/
   DATA RCA/7.56495E-15/,EVE/1.60210E-12/,BHR/5.29167E-9/
   DATA PCH/6.6256E-27/,RYD/13.595/,GTL/15.68391/,GTD/20.27400/
   DATA PIE/3.1415926536/,ALOG2/0.30105/,THT/5039.935/
   DATA INZ/1,2,6,7,8,0,0,0,0,0/
   DATA NZC,NZN,NZO/6,7,8/,XZ,YZ,ZZ/1,0,2,0,0,0/
   DATA XM,YM,ZM/1,0,4,0,14,0/,ZMC,ZMN,ZMO/12,0,14,0,16,0/
   DATA CCM,CNM,COM/0,3,0,1,0,6/,CCN,CNN,CON/0,0,0,0,0,0/
C   DATA CCM,CNM,COM/0,0,0,0,1,0/,CCN,CNN,CON/0,0,0,0,0,0/
   DATA CPF/2,0,1,0,2,0,1,0,6,0,15,0,20,0,15,0,6,0,1,0/
   DATA VX/13.595/,VY/24.581/,VZ/13.614/,ID/2/
   DATA VHM/0.754/,VCM/1.270/,VOM/1.465/
   DATA VI/13.595,9*0.0,24.581,54.420,8*0.0,
311.264,24.376,47.864,64.476,392.077,489.981,4*0.0,
414.530,29.593,47.426,77.450,97.888,552.057,667.029,3*0.0,
513.614,35.108,54.886,77.394,113.896,138.116,739.315,871.387,
62*0.0,50*0.0/
   DATA REDH2/0.742/,WEDH2/4395.0/,DEDH2/ 4.477 /
   DATA REDN2/1.094/,WEDN2/2558.6/,DEDN2/ 9.756 /
   DATA REDCO/1.128/,WEDCO/2170.0/,DEDCO/11.090/
   DATA ONE/1.0/,TEN/10.0/,TCUT/1.0E+04/,EPSN/1.0E-06/
   ALOG(X)=DLOG(X)
   ALOG10(X)=DLOG10(X)
   SQRT(X)=DSQRT(X)
   ABS(X)=DABS(X)
   AMAX1(X1,X2)=DMAX1(X1,X2)

```

```

C  VERSION II
C  THIS ALTERATION TO ALLOW CCM,CNM,COM TO TAKE VALUES FROM
C  ARRAY ZAC, BROUGHT THROUGH FROM MAIN CODE. ALL ELEMENTS
C  ABOVE O16 ARE ASSUMED TO BE O16.
  IF(Z)1235,1235,1233
1233 CONTINUE
  CNM=0.1
  CCM=ZAC(1)/Z
  COM=0.0
  DO 1234 K=2,8
    COM=COM+ZAC(K)/Z
1234 CONTINUE
1235 CONTINUE
C  END OF ALTERATION

  IF(NEL-1)1000,1100,1200
1000 CONTINUE
1100 CONTINUE
  TK=BCK*T
  UR=RCA*T**4
  PR=UR/3.0
  AM=1.0/(X/XM+Y/YM+Z/ZM)
  VA=AM*(X/XM*VX+Y/YM*VY+Z/ZM*VZ)
  HEKL=GTL-ALOG2+1.5*ALOG10(T)-THT/T*VA
  AD=AVN*D
  AN=AD/AM
  C=TEN**HEKL/AN
  IF(C-0.01)1,1,2
1  EX=SQRT(C)*(1.0+C/2.0*(1.0-C/64.0))-C/2.0
  GO TO 5
2  IF(C-100.0)3,4,4
4  EX=1.0-(1.0-2.0/C)/C
  GO TO 5
3  EX=C/2.0*(SQRT(1.0+4.0/C)-1.0)
5  EN=EX*AD*(1.0+X)/2.0
  H=1.0
  GE=2.0*(TEN**GTL)*T*SQRT(T/PIE)
  F=EN/GE
  CALL DFIS(1,F,ETA)
  CALL FDIS(3,ETA,G)
  H=2.0/3.0*G/F
  TN=AN+EN
  PN=AN+H*EN
  PG=PN*TK
  P=PR+PG
  WM=AD*TK/PG
  UG=1.5*PG
  UI=AN*EX*VA*EVE
  U=UR+UG+UI
  DPRT=4.0*PR/T
  DPGT=PG/T
  DPGD=AN*TK/D
  DPT=DPRT+DPGT
  DPD=DPGD
C  NON=LOGARITHMIC DERIVATIVES - PER UNIT VOLUME
  DURT=4.0*UR/T
  DUGT=UG/T
  DUGD=1.5*DPGD
  DUID=UI/D
  DUT=DURT+DUGT
  DUD=DUGD+DUID
C  PER UNIT MASS
  S=(P+U)/T/D
  DST=(DPT+DUT)/T/D-S/T
  DSD=(DPD+DUD)/T/D-S/D
  DUT=DUT/D
  DUD=(DUD-U/D)/D
  U=U/D
C  LOGARITHMIC DERIVATIVES
  DPT=T/P*DPT
  DPD=D/P*DPD
  GO TO 2000
1200 CONTINUE
  IF(NEL-3)1300,1300,1400
1300 CONTINUE
  CALL ESTATENN(NEL,X,Y,Z,T,D,P,U,S,AM,AN,EN,WM,ETA,

```

```

1 DPT,DPD,DUT,DUD,DST,DSD)
C 1 DPT,GPD,DUT,GUD,DST,GSD)
GO TO 2000
1400 CONTINUE
IF(NEL-5)1500,1500,1600
1500 CONTINUE
ALNX=ALOG(TEN)
ALOGT=ALOG10(T)
TK=BCK*T
THTT=THT/T
GE=2.0*(TEN**GTL)*T*SQRT(T/PIE)
ZM=1.0/(CCM/ZMC+CNM/ZMN+COM/ZMO)
CCN=ZM*CCM/ZMC
CNN=ZM*CNM/ZMN
CON=ZM*COM/ZMO
AM=1/(X/XM+Y/YM+Z/ZM)
AD=AVN*D
AN=AD/AM
XN=AN*AM*X/XM
YN=AN*AM*Y/YM
ZN=AN*AM*Z/ZM
ANI(1)=XN
ANI(2)=YN
ANI(3)=ZN*CCN
ANI(4)=ZN*CNN
ANI(5)=ZN*CON
ZZ=CCN*NZC+CNN*NZN+CON*NZO
ENMIN=0.0
ENLIM=XZ*XN+YZ*YN+ZZ*ZN
ENMAX=ENLIM
EN=ENMAX*TEN**(-2.0E4/T)
RTL=GTL+1.5*ALOGT
EHM=THTT*VHM-RTL
ECM=THTT*VCM-RTL
EOM=THTT*VOM-RTL
DO 40 I=1,NEL
JMIN(I)=1
JMAX(I)=INZ(I)+1
40 FF(I)=1.0
RS=(3.0/4.0/PIE/AN)**(1.0/3.0)/BHR
DVC=3.0/AN/RS*RYD
C DVC=0.0
IF(T-1.0E4)41,41,42
41 CONTINUE
DVDH2=AMAX1(0.0D0,(1.0/RS-2.0E8*BHR/REDH2)*RYD)
DVDN2=AMAX1(0.0D0,(1.0/RS-2.0E8*BHR/REDN2)*RYD*NZN*NZN)
DVDCO=AMAX1(0.0D0,(1.0/RS-2.0E8*BHR/REDCO)*RYD*NZC*NZO)
RMDH2=XM/2.0
RMDN2=ZMN/2.0
RMDCO=ZMC*ZMO/(ZMC+ZMO)
BRDC=(1.0E16*PCH)*(PCH*AVN)/8.0/PIE/PIE
BRDH2=BRDC/RMDH2/REDH2**2
BRDN2=BRDC/RMDN2/REDN2**2
BRDCO=BRDC/RMDCO/REDCO**2
C WRITE(6,16)RMDH2,RMDCO,BRDC,BRDH2,BRDCO
TKHC=TK/PCH/VLC
XH2=WEDH2/TKHC
EXH2=EXP(XH2)
XN2=WEDN2/TKHC
EXN2=EXP(XN2)
XCO=WEDCO/TKHC
EXCO=EXP(XCO)
ZH2=TK/BRDH2/2*EXH2/(EXH2-1.0)
ZN2=TK/BRDN2/2*EXN2/(EXN2-1.0)
ZCO=TK/BRDCO*EXCO/(EXCO-1.0)
EQKH2=4.0*SQRT(RMDH2)**3/ZH2*TEN**(GTD+1.5*ALOGT-THTT*DEDH2)
EQKN2=400*SQRT(RMDN2)**3/ZN2*TEN**(GTD+1.5*ALOGT-THTT*DEDN2)
EQKCO=225*SQRT(RMDCO)**3/ZCO*TEN**(GTD+1.5*ALOGT-THTT*DEDCO)
C WRITE(6,16)XN,YN,ZN,ANI(3),ANI(4),ANI(5),EQKH2,EQKN2,EQKCO
42 CONTINUE
KMIN=0
KMAX=0
ITN=0
11 ITN=ITN+1
DV=DVC*EN
ENL=ALOG10(EN)
ETAD=ENL-RTL

```

```

ETAE=ALNX*ETAD
H=1.0
ETA=ETAE
IF(ETAE+15)31,31,32
32 CONTINUE
F=EN/GE
CALL DFIS(1,F,ETA)
CALL FDIS(3,ETA,G)
H=2.0/3.0*G/F
ETAD=ETA/ALNX
31 CONTINUE
ENLC=ENL-THTT*DV
IF(ITN-1)700,710,700
710 CONTINUE
IF(T-TCUT)720,720,700
720 CONTINUE
C H-
CALL PARTFN(1,2,VHM,T,RS,ZLN,ZMG,ER)
ZHML=ZLN/ALNX
C C-
CALL PARTFN(6,7,VCM,T,RS,ZLN,ZMG,ER)
ZCML=ZLN/ALNX
C O-
CALL PARTFN(8,9,VOM,T,RS,ZLN,ZMG,ER)
ZOML=ZLN/ALNX
700 CONTINUE
DO 71 I=1,NEL
IF(ANI(I))71,71,711
711 CONTINUE
NZ=INZ(I)
SM(I)=0.0
DO 72 J=1,NZ+1
NE=NZ+1-J
V=VI(J,I)
IF(ITN-1)713,712,713
712 CONTINUE
CALL PARTFN(NZ,NE,V,T,RS,ZLN,ZMG,ER)
PFL(J,I)=ZLN/ALNX
EEV(J,I)=ER*RYD
713 CONTINUE
IF(J-1)72,721,722
721 SL(J,I)=0.0
GO TO 723
722 SL(J,I)=SL(J-1,I)+RL(J-1,I)+PFL(J,I)
723 RL(J,I)=-ETAD-THTT*(V-DV)-PFL(J,I)
IF(SM(I)-SL(J,I))77,77,72
77 SM(I)=SL(J,I)
72 CONTINUE
71 CONTINUE
C FHM=TEN**(ZHML+EHM+ENLC-SM(1))
C FCM=TEN**(ZCML+ECM+ENLC-SM(3))
C FOM=TEN**(ZOML+EOM+ENLC-SM(5))
FHM=0.0
FCM=0.0
FOM=0.0
DO 73 I=1,NEL
IF(ANI(I))73,73,731
731 CONTINUE
SN=SM(I)
SM(I)=0.0
IF(T-TCUT)747,747,740
747 CONTINUE
IF(I-1)741,741,742
741 FHM=TEN**(ZHML+EHM+ENLC-SN)
SM(1)=SM(1)+FHM
GO TO 740
742 IF(I-3)740,743,744
743 FCM=TEN**(ZCML+ECM+ENLC-SN)
SM(3)=SM(3)+FCM
GO TO 740
744 IF(I-5)740,745,740
745 FOM=TEN**(ZOML+EOM+ENLC-SN)
SM(5)=SM(5)+FOM
740 CONTINUE
JM=1
SLM=0.0
DO 74 J=1,INZ(I)+1

```

```

      K=INZ(I)+1-J
      CPMK=1.0
      IF(K)70,70,90
C90  CPMK=CPM(K)
90  CPMK=1.0
70  SL(J,I)=CPMK*TEN**(SL(J,I)-SN)
      SM(I)=SM(I)+SL(J,I)
      IF(SLM-SL(J,I))87,87,74
87  SLM=SL(J,I)
      JM=J
74  CONTINUE
      IF(I-3)73,78,78
78  JMIN(I)=MAX(1,JM-ID)
      JMAX(I)=MIN(JM+ID,INZ(I)+1)
73  CONTINUE
      DO 75 I=1,NEL
      IF(ANI(I))75,75,751
751  CONTINUE
      IF(I-1)760,761,762
761  FHM=FHM/SM(1)
      GO TO 760
762  IF(I-3)760,763,764
763  FCM=FCM/SM(3)
      GO TO 760
764  IF(I-5)760,765,760
765  FOM=FOM/SM(5)
760  CONTINUE
      DO 76 J=JMIN(I),JMAX(I)
      SL(J,I)=SL(J,I)/SM(I)
76  CONTINUE
75  CONTINUE
      IF(T-TCUT)51,51,52
51  CONTINUE
      ANH1=ANI(1)*SL(1,1)
      ANC1=ANI(3)*SL(1,3)
      ANN1=ANI(4)*SL(1,4)
      ANO1=ANI(5)*SL(1,5)
      FF(1)=FREE(ANH1,ANH1,EQKH2/2)
      FF(3)=FREE(ANC1,ANO1,EQKCO)
      FF(4)=FREE(ANN1,ANN1,EQKN2/2)
      FF(5)=FREE(ANO1,ANC1,EQKCO)
52  ENN=0.0
      DO 81 I=1,NEL
      ANF(I)=ANI(I)*FF(I)
      IF(ANI(I))81,81,811
811  CONTINUE
      DO 82 J=JMIN(I),JMAX(I)
      ENN=ENN+(J-1)*ANF(I)*SL(J,I)
82  CONTINUE
81  CONTINUE
      ENN=ENN-ANF(1)*FHM-ANF(3)*FCM-ANF(5)*FOM
C   WRITE(6,15)NEL,ITN
C   WRITE(6,15)JMIN,JMAX
C   WRITE(6,16)T,D,EN,ENN,ETA,H,FHM,FCM,FOM
C   WRITE(6,16)FF,SM,SL
16  FORMAT(1P10E10.3)
      IF(ENN)91,91,92
91  KMAX=KMAX+1
      IF(KMAX-1)191,191,192
191  ENMAX=EN
      GO TO 193
192  ENMAX=MIN(ENMAX,EN)
193  IF(KMIN-1)194,195,195
194  ENP=EN/TEN/TEN
      GO TO 93
195  ENP=SQRT(ENMAX)*SQRT(ENMIN)
      GO TO 93
92  EPEN=ABS(ENN/EN-1.0)
      IF(EPEN-EPSN)10,150,150
150  IF(EN-ENN)160,10,170
160  KMIN=KMIN+1
      IF(KMIN-1)161,161,162
161  ENMIN=EN
      GO TO 163
162  ENMIN=MAX(ENMIN,EN)
163  KMAX=KMAX+1
      IF(KMAX-1)164,164,165

```

```

164 ENMAX=ENN
    GO TO 166
165 ENMAX=MIN(ENMAX,ENN)
166 ENP=SQRT(ENMIN)*SQRT(ENMAX)
    GO TO 93
170 KMAX=KMAX+1
    IF(KMAX-1)171,171,172
171 ENMAX=EN
    GO TO 173
172 ENMAX=MIN(ENMAX,EN)
173 KMIN=KMIN+1
    IF(KMIN-1)174,174,175
174 ENMIN=ENN
    GO TO 176
175 ENMIN=MAX(ENMIN,ENN)
176 ENP=SQRT(ENMAX)*SQRT(ENMIN)
    GO TO 93
93 EN=ENP
    IF(ITN-50)110,111,111
111 WRITE(6,16)T,D,AN,EN,ENL,ETA,ETAD,EPEN
    GO TO 10
110 CONTINUE
    GO TO 11
10 CONTINUE
    EV=0.0
    DO 83 I=1,NEL
    IF(ANI(I))83,83,831
831 CONTINUE
    SV(1,I)=0.0
    DO 84 J=2,JMAX(I)
    SV(J,I)=SV(J-1,I)+VI(J-1,I)
    EEV(J,I)=EEV(J,I)+SV(J,I)
    IF(J-JMIN(I))84,85,85
85 EV=EV+ANF(I)*SL(J,I)*EEV(J,I)
84 CONTINUE
83 CONTINUE
    PN=AN
    IF(T-TCUT)832,833,833
832 CONTINUE
    EV=EV-ANF(1)*FHM*VHM-ANF(3)*FCM*VCM-ANF(5)*FOM*VOM
    DH2=ANH1*(1.0-FF(1))/2.0
    DN2=ANN1*(1.0-FF(4))/2.0
    DCO=(ANC1*(1.0-FF(3))+ANO1*(1.0-FF(5)))/2.0
    EIH2=TK/EVE*(1.0+XH2/(EXH2-1.0))
    EIN2=TK/EVE*(1.0+XN2/(EXN2-1.0))
    EICO=TK/EVE*(1.0+XCO/(EXCO-1.0))
    EV=EV+DH2*(EIH2-DEDH2)+DN2*(EIN2-DEDN2)+DCO*(EICO-DEDCO)
    PN=PN-DH2-DN2-DCO
833 CONTINUE
    UR=RCA*T**4
    PR=UR/3.0
    PG=TK*(PN+H*EN)
    P=PR+PG
    WM=AD*TK/PG
    UG=3.0*PG/2.0
    UI=EVE*EV
    U=UR+UG+UI
    DPRT=4.0*PR/T
    DPGT=PG/T
    DPGD=PN*TK/D
    DPT=DPRT+DPGT
    DPD=DPGD
C   NON-LOGARITHMIC DERIVATIVES - PER UNIT VOLUME
    DURT=4.0*UR/T
    DUGT=1.5*DPGT
    DUGD=1.5*DPGD
    DUID=UI/D
    DUT=DURT+DUGT
    DUD=DUGD+DUID
C   PER UNIT MASS
    S=(P+U)/T/D
    DST=(DPT+DUT)/T/D-S/T
    DSD=(DPD+DUD)/T/D-S/D
    DUT=DUT/D
    DUD=(DUD-U/D)/D
    U=U/D
C   LOGARITHMIC DERIVATIVES

```

```

DPT=T/P*DPT
DPD=D/P*DPD
15 FORMAT(10I5)
IF(NEL)116,116,115
116 CONTINUE
WRITE(6,15)NEL,ITN
WRITE(6,15)JMIN,JMAX
WRITE(6,16)T,D,EN,ENLIM,ETA,H,FHM,FCM,FOM
WRITE(6,16)ANI,ZZ,ZM,CCN,CNN,CON
WRITE(6,16)FF,SM,SL
WRITE(6,16)ZH2,ZN2,ZCO,EQKH2,EQKN2,EQKCO,DH2,DN2,DCO
115 CONTINUE
GO TO 2000
1600 CONTINUE
NDP=0
IF(NDP)1700,1700,1701
1701 CONTINUE
TL=ALOG10(T)
DL=ALOG10(D)
C CALL TABLES(ILQ,ITP,NDP,INP,NXP,NYP,DXP,ENP,XAP,YAP,ZAP,
C *DL,TL,PL,CP,DPD,DPT)
CALL TABLES(ILQ,ITP,NDP,INP,NXP,NYP,DXP,CP,XT,YT,ZT,
* ENP,XAP,YAP,ZAP,DL,TL,PL,DPD,DPT)
PT=TEN**PL
PC=PT-CP
DPD=(PT/PC)*DPD
DPT=(PT/PC)*DPT
P=PC
NDU=0
CALL TABLES(ILQ,ITU,NDU,INU,NXU,NYU,DXU,CU,XT,YT,ZT,
* ENU,XAU,YAU,ZAU,DL,TL,UU,DUD,DUT)
UT=UU
UC=UT-CU
DUD=(UT/UC)*DUD
DUT=(UT/UC)*DUT
U=UC
NDS=0
CALL TABLES(ILQ,ITS,NDS,INS,NXS,NYS,DXS,CS,XT,YT,ZT,
* ENS,XAS,YAS,ZAS,DL,TL,SS,DSD,DST)
ST=SS
SC=ST-CS
DSD=(ST/SC)*DSD
DST=(ST/SC)*DST
S=SC
1700 CONTINUE
GO TO 2000
2000 CONTINUE
RETURN
END
C FREE FRACTION
FUNCTION FREE(AN,BN,CN)
IMPLICIT REAL*8 (A-H,O-Z)
AMAX1(X1,X2)=DMAX1(X1,X2)
B=BN-AN+CN
D=AMAX1(AMAX1(AN,ABS(B)),CN)
A=AN/D
B=B/D
C=-CN/D
E=B*B
F=-4.0*A*C
IF(B)1,1,2
1 FREE=(SQRT(E+F)-B)/A/2.0
GO TO 3
2 IF(E-F)1,1,4
4 G=F/E
IF(G-1.0E-02)5,5,1
5 FREE=-C/B*(1.0-G/4.0*(1.0-G/2.0))
3 RETURN
END
C SOLVE SAHA IONIZATION EQUATIONS FOR H,HE AND Z
SUBROUTINE ESTATENN(NEL,X,Y,Z,T,D,P,U,S,AM,AN,EN,WM,ETA,
1 DPT,DPD,DUT,DUD,DST,DSD)
IMPLICIT REAL*8 (A-H,O-Z)
DATA AVN/6.02252E23/,THT/5039.935/,GEL/15.68391/
DATA BHR/5.29167E-9/,BCK/1.38054E-16/,RCA/7.56495E-15/
DATA RYD/13.595/,EVE/1.60210E-12/,THTE/11604.879/,GELE/36.11354/
DATA PIE/3.1415926536/,HRP/0.886226926/,TFP/0.238732415/

```

```

DATA RPB3/0.59081795/,PIEL/1.144729886/
DATA ZX/1.0/,ZY/2.0/,ZZ/8.0/,XM/1.0/,YM/4.0/,ZM/16.0/
DATA GH00/1.0/,GH10/2.0/,GH20/1.0/,GHE10/1.0/GHE20/2.0/,GHE30/1.0/
DATA VX0/0.754/,VX1/13.596/,VY1/24.581/,VY2/54.420/
DATA VZ1/13.618/,VZ2/35.116/,VZ3/54.934/,VZ4/77.412/
DATA VZ5/113.896/,VZ6/138.116/,VZ7/739.315/,VZ8/871.387/
DATA GZ10/15.0/,GZ20/20.0/,GZ30/15.0/,GZ40/6.0/,GZ50/1.0/
DATA GZ60/2.0/,GZ70/1.0/,GZ80/2.0/,GZ90/1.0/
DATA BCON/1.0/,TEN/10.0/,EPS/1.0E-6/
COMPL(X1,X2)=MAX(X1,X2)+LOG(1.0+EXP(MIN(X1,X2)-MAX(X1,X2)))
ZEX0=SQRT(VX0/RXD)
ZEX1=1.0
ZEY1=SQRT(VY1/RXD)
ZEY2=2.0
TK=BCK*T
THETA=THE/T
RYBKT=THETA*RYD
TVX0=THETA*VX0
TVX1=THETA*VX1
TVY1=THETA*VY1
TVY2=THETA*VY2
TVZ1=THETA*VZ1
TVZ2=THETA*VZ2
TVZ3=THETA*VZ3
TVZ4=THETA*VZ4
TVZ5=THETA*VZ5
TVZ6=THETA*VZ6
TVZ7=THETA*VZ7
TVZ8=THETA*VZ8
ZEL=GELE+1.5*LOG(T)
ZE=EXP(ZEL)
AM=1.0/(X/XM+Y/YM+Z/ZM)
AZ=AM*(X*ZX/XM+Y*ZY/YM+Z*ZZ/ZM)
AD=AVN*D
AN=AD/AM
ANAM=AN*AM
ANX=ANAM*X/XM
ANY=ANAM*Y/YM
ANZ=ANAM*Z/ZM
RNN=(TFP/AN)**(1.0/3.0)/BHR
IF(X)10,10,11
11 CONTINUE
C H-
N0=1
G0=GH00
ZN0=ZEX0
GI=0.0
ZNI=0.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBH0)
ZHL0=ZLOG
EBH0=-VX0
ESUM0=0.0
C H0
N0=1
G0=GH10
ZN0=ZEX1
GI=1.0
ZNI=1.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBH1)
ZHL1=ZLOG
EBH1=ESUM0+EBH1*RYD
ESUM1=ESUM0+VX1
C H+
ZHL2=0.0
EBH2=0.0
EBH2=ESUM1+EBH2*RYD
10 CONTINUE
IF(Y)20,20,21
21 CONTINUE
ESUM=0.0
C HE0
N0=1
G0=GHE10
ZN0=ZEY1
GI=1.0
ZNI=1.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBHE1)

```

```

ZHEL1=ZLOG
EBHE1=ESUM0+EBHE1*RYD
ESUM1=ESUM0+VY1
C HE+1
N0=1
G0=GHE20
ZN0=ZEY2
GI=1.0
ZNI=2.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBHE2)
ZHEL2=ZLOG
EBHE2=ESUM1+EBHE2*RYD
ESUM2=ESUM1+VY2
C HE+2
ZHEL3=0.0
EBHE3=0.0
EBHE3=ESUM2+EBHE3*RYD
20 CONTINUE
C WRITE(6,60)T,D,AM,AZ,ZHL0,ZHL1,ZHL2,ZHEL1,ZHEL2,ZHEL3
C WRITE(6,60)T,D,AN,RNN,EBH0,EBH1,EBH2,EBHE1,EBHE2,EBHE3
C FOR SPEED ASSUME Z COMPLETELY IONIZED
C IF(Z)31,30,30
IF(Z)30,30,31
31 CONTINUE
ESUM0=0.0
C M0
N0=2
G0=GZ10
ZN0=N0*SQRT(VZ1/RYD)
GI=GZ20
ZNI=1.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM1)
ZML1=ZLOG
EBM1=ESUM0+EBM1*RYD
ESUM1=ESUM0+VZ1
C M+1
N0=2
G0=GZ20
ZN0=N0*SQRT(VZ2/RYD)
GI=GZ30
ZNI=2.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM2)
ZML2=ZLOG
EBM2=ESUM1+EBM2*RYD
ESUM2=ESUM1+VZ2
C M+2
N0=2
G0=GZ30
ZN0=N0*SQRT(VZ3/RYD)
GI=GZ40
ZNI=3.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM3)
ZML3=ZLOG
EBM3=ESUM2+EBM3*RYD
ESUM3=ESUM2+VZ3
C M+3
N0=2
G0=GZ40
ZN0=N0*SQRT(VZ4/RYD)
GI=GZ50
ZNI=4.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM4)
ZML4=ZLOG
EBM4=ESUM3+EBM4*RYD
ESUM4=ESUM3+VZ4
C M+4
N0=2
G0=GZ50
ZN0=N0*SQRT(VZ5/RYD)
GI=GZ60
ZNI=5.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM5)
ZML5=ZLOG
EBM5=ESUM4+EBM5*RYD
ESUM5=ESUM4+VZ5
C M+5
N0=2

```

```

G0=GZ60
ZN0=N0*SQRT(VZ6/RYD)
GI=GZ70
ZNI=6.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM6)
ZML6=ZLOG
EBM6=ESUM5+EBM6*RYD
ESUM6=ESUM5+VZ6
C M+6
N0=1
G0=GZ70
ZN0=N0*SQRT(VZ7/RYD)
GI=GZ80
ZNI=7.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM7)
ZML7=ZLOG
EBM7=ESUM6+EBM7*RYD
ESUM7=ESUM6+VZ7
C M+7
N0=1
G0=GZ80
ZN0=N0*SQRT(VZ8/RYD)
GI=GZ90
ZNI=8.0
CALL PARTFNN(N0,G0,ZN0,GI,ZNI,RYBKT,RNN,ZLOG,ZMAG,EBM8)
ZML8=ZLOG
EBM8=ESUM7+EBM8*RYD
ESUM8=ESUM7+VZ8
C M+8
ZM9=GZ90
ZML9=0.0
EBM9=0.0
EBM9=ESUM8+EBM9*RYD
C WRITE(6,60)Z,ZML1,ZML2,ZML3,ZML4,ZML5,ZML6,ZML7,ZML8,ZML9
C WRITE(6,60)Z,EBM1,EBM2,EBM3,EBM4,EBM5,EBM6,EBM7,EBM8,EBM9
30 CONTINUE
ENX=ANX*ZX
ENY=ANY*ZY
ENZ=ANZ*ZZ
ENZT=ENZ
EN=ENX+ENY+ENZ
EN=EN*10.0**(-2.0E4/T)
FH2=1.0
C THIS VALUE OF FH2 IS USED WHEN X=0
ITN=0
KMIN=0
KMAX=0
3 CONTINUE
ITN=ITN+1
ENL=LOG(EN)
ZENL=ZEL-ENL
ETA=-ZENL
IF(ETA+15.0)50,50,51
51 CONTINUE
FD1=HRP*EN/ZE
CALL DFIS(1,FD1,ETA)
ZENL=-ETA
50 CONTINUE
C REPLACES NON-DEGENERATE BY DEGENERATE FORMULA
IF(X)110,110,111
111 CONTINUE
SRHL0=0.0
RHL01=ZHL1-ZHL0+ZENL-TVX0
SRHL1=SRHL0+RHL01
RHL12=ZHL2-ZHL1+ZENL-TVX1
SRHL2=SRHL1+RHL12
C SRHLM=MAX(SRHL0,MAX(SRHL1,SRHL2))
SRHLM=MAX(SRHL0,SRHL1,SRHL2)
FHL0=SRHL0-SRHLM
FHL1=SRHL1-SRHLM
FHL2=SRHL2-SRHLM
C WRITE(6,60)SRHL0,RHL01,SRHL1,RHL12,SRHL2,FHL0,FHL1,FHL2
FH0=EXP(FHL0)
FH1=EXP(FHL1)
FH2=EXP(FHL2)
SFH=FH0+FH1+FH2
FH0=FH0/SFH

```

```

FH1=FM1/SFH
FH2=FM2/SFH
ENX=ANX*(FH2-FH0)
110 CONTINUE
IF(Y)120,120,121
121 CONTINUE
SRHEL1=0.0
RHEL12=ZHEL2-ZHEL1+ZENL-TVY1
SRHEL2=SRHEL1+RHEL12
RHEL23=ZHEL3-ZHEL2+ZENL-TVY2
SRHEL3=SRHEL2+RHEL23
C SRHELM=MAX(SRHEL1,MAX(SRHEL2,SRHEL3))
SRHELM=MAX(SRHEL1,SRHEL2,SRHEL3)
FHEL1=SRHEL1-SRHELM
FHEL2=SRHEL2-SRHELM
FHEL3=SRHEL3-SRHELM
C WRITE(6,60)SRHEL1,RHEL12,SRHEL2,RHEL23,SRHEL3,FHEL1,FHEL2,FHEL3
FHE1=EXP(FHEL1)
FHE2=EXP(FHEL2)
FHE3=EXP(FHEL3)
SFHE=FHE1+FHE2+FHE3
FHE1=FHE1/SFHE
FHE2=FHE2/SFHE
FHE3=FHE3/SFHE
ENY=ANY*(FHE2+2.0*FHE3)
120 CONTINUE
C FOR SPEED ASSUME Z COMPLETELY IONIZED
ENZ=FM2*ENZT
C IF(Z)131,130,130
IF(Z)130,130,131
131 CONTINUE
SRML1=0.0
RML12=ZML2-ZML1+ZENL-TVZ1
SRML2=SRML1+RML12
RML23=ZML3-ZML2+ZENL-TVZ2
SRML3=SRML2+RML23
RML34=ZML4-ZML3+ZENL-TVZ3
SRML4=SRML3+RML34
RML45=ZML5-ZML4+ZENL-TVZ4
SRML5=SRML4+RML45
RML56=ZML6-ZML5+ZENL-TVZ5
SRML6=SRML5+RML56
RML67=ZML7-ZML6+ZENL-TVZ6
SRML7=SRML6+RML67
RML78=ZML8-ZML7+ZENL-TVZ7
SRML8=SRML7+RML78
RML89=ZML9-ZML8+ZENL-TVZ8
SRML9=SRML8+RML89
C SRMLM=MAX(SRML1,MAX(SRML2,SRML3))
SRMLM=MAX(SRML1,SRML2,SRML3,SRML4,SRML5,SRML6,SRML7,SRML8,SRML9)
FML1=SRML1-SRMLM
FML2=SRML2-SRMLM
FML3=SRML3-SRMLM
FML4=SRML4-SRMLM
FML5=SRML5-SRMLM
FML6=SRML6-SRMLM
FML7=SRML7-SRMLM
FML8=SRML8-SRMLM
FML9=SRML9-SRMLM
C WRITE(6,60)SRML1,RML12,SRML2,RML23,SRML3,FML1,FML2,FML3
FM1=EXP(FML1)
FM2=EXP(FML2)
FM3=EXP(FML3)
FM4=EXP(FML4)
FM5=EXP(FML5)
FM6=EXP(FML6)
FM7=EXP(FML7)
FM8=EXP(FML8)
FM9=EXP(FML9)
SFM=FM1+FM2+FM3+FM4+FM5+FM6+FM7+FM8+FM9
FM1=FM1/SFM
FM2=FM2/SFM
FM3=FM3/SFM
FM4=FM4/SFM
FM5=FM5/SFM
FM6=FM6/SFM
FM7=FM7/SFM

```

```

FM8=FM8/SFM
FM9=FM9/SFM
ENZ=ANZ*(FM2+2*FM3+3*FM4+4*FM5+5*FM6+6*FM7+7*FM8+8*FM9)
130 CONTINUE
ENN=ENX+ENY+ENZ
C WRITE(*,*)ITN
C WRITE(6,60)T,D,FH0,FH1,FH2,FHE1,FHE2,FHE3,AN,EN
C WRITE(6,60)Z,FM1,FM2,FM3,FM4,FM5,FM6,FM7,FM8,FM9
IF(ITN-25)44,4,4
44 CONTINUE
IF(ENN)1,1,2
1 KMAX=KMAX+1
IF(KMAX-1)81,81,82
81 ENMAX=EN
GO TO 83
82 ENMAX=MIN(ENMAX,EN)
83 IF(KMIN-1)84,85,85
84 ENP=EN/TEN/TEN
GO TO 8
85 ENP=SQRT(ENMAX)*SQRT(ENMIN)
GO TO 8
2 IF(ABS(ENN/EN-1.0)-EPS)4,5,5
5 IF(EN-ENN)6,4,7
6 KMIN=KMIN+1
IF(KMIN-1)61,61,62
61 ENMIN=EN
GO TO 63
62 ENMIN=MAX(ENMIN,EN)
63 KMAX=KMAX+1
IF(KMAX-1)64,64,65
64 ENMAX=ENN
GO TO 66
65 ENMAX=MIN(ENMAX,ENN)
66 ENP=SQRT(ENMIN)*SQRT(ENMAX)
GO TO 8
7 KMAX=KMAX+1
IF(KMAX-1)71,71,72
71 ENMAX=EN
GO TO 73
72 ENMAX=MIN(ENMAX,EN)
73 KMIN=KMIN+1
IF(KMIN-1)74,74,75
74 ENMIN=ENN
GO TO 76
75 ENMIN=MAX(ENMIN,ENN)
76 ENP=SQRT(ENMAX)*SQRT(ENMIN)
GO TO 8
8 EN=ENP
GO TO 3
-4 CONTINUE
C WRITE(*,*)ITN
C WRITE(6,60)T,D,FH0,FH1,FH2,FHE1,FHE2,FHE3,AN,EN
C WRITE(6,60)Z,FM1,FM2,FM3,FM4,FM5,FM6,FM7,FM8,FM9
EBH=ANX*(FH0*EBH0+FH1*EBH1+FH2*EBH2)
EBHE=ANY*(FHE1*EBHE1+FHE2*EBHE2+FHE3*EBHE3)
EBM=ANZ*(FM1*EBM1+FM2*EBM2+FM3*EBM3+FM4*EBM4+FM5*EBM5+
1 FM6*EBM6+FM7*EBM7+FM8*EBM8+FM9*EBM9)
UI=(EBH+EBHE+EBM)*EVE
UR=RCA*T**4
PR=UR/3.0
PA=AN*TK
RFD=1.0
IF(ETA+15.0)40,40,41
41 CONTINUE
CALL FDIS(3,ETA,FD3)
RFD=2.0/3.0*FD3/FD1
-40 CONTINUE
PE=EN*TK*RFD
PG=PA+PE
P=PR+PG
WM=AD*TK/PG
UA=1.5*PA
UE=1.5*PE
UG=UA+UE+UI
UV=UR+UG
U=UV
SR=(PR+UR)/T

```

```

SG=(PG+UG)/T
SV=SR+SG
S=SV/D
DPRT=4.0*PR/T
DPGT=PG/T
DPGD=PA/D
DPT=DPRT+DPGT
DPD=DPGD
C  NON=LOGARITHMIC DERIVATIVES - PER UNIT VOLUME
DURT=4.0*UR/T
DUGT=(UA+UE)/T
DUGD=UG/D
DUT=DURT+DUGT
DUD=DUGD
C  PER UNIT MASS
DST=(DPT+DUT)/T/D-S/T
DSD=(DPD+DUD)/T/D-S/D
DUT=DUT/D
DUD=(DUD-U/D)/D
U=U/D
C  LOGARITHMIC DERIVATIVES
DPT=T/P*DPT
DPD=D/P*DPD
DUT=T/U*DUT
DUD=D/U*DUD
DST=T/S*DST
DSD=D/S*DSD
C  WRITE(6,60)ZENL,ETA,RFD,PE,PA,PR,PG,PT,U,S
RETURN
60  FORMAT(1P10E10.3)
END
C  PARTITION FUNCTION USING NEAREST NEIGHBOUR PROBABILITY
SUBROUTINE PARTFN(NZ,NE,VI,T,R,ZL,ZM,EBR)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION NA(10),NG(10)
DATA NA/1,1,2,2,2,2,2,2,2,2/
DATA NG/2,1,2,1,6,15,20,15,6,1/
DATA BCK/1.38054E-16/,RYD/13.595/
DATA EVE/1.60210E-12/
IF(NE)1,1,2
1  ZM=1.0
   ZL=0.0
   EBR=0.0
   GO TO 9
2  TK=BCK*T
   RYT=EVE/BCK*RYD/T
   N0=NA(NE)
   G0=NG(NE)
   Z0=N0*SQRT(VI/RYD)
   IF(NE-1)9,3,4
3  GI=1.0
   GO TO 5
4  GI=NG(NE-1)
5  Z1=NZ-NE+1
   CALL PARTFNN(N0,G0,Z0,GI,Z1,RYT,R,ZL,ZM,EBR)
9  RETURN
END
C  PARTITION FUNCTION FROM NEAREST NEIGHBOUR PROBABILITY
SUBROUTINE PARTFNN(N0,G0,Z0,GI,Z1,RYBKT,RNN,ZL,ZM,EBR)
IMPLICIT REAL*8 (A-H,O-Z)
DATA CON/1.0/,CR3/1.44224957/,RPB3/0.59081795/
COMPL(X1,X2)=MAX(X1,X2)+LOG(1.0+EXP(MIN(X1,X2)-MAX(X1,X2)))
R0=N0*N0/Z0
PL0=(R0/RNN/CON)**3
ZL=-PL0+LOG(G0)
IF(ZI)1,1,2
2  CZR=CON*ZI*RNN
   RBI=1.0/CZR/SQRT(CZR)
   SNM=CZR/CR3
   XL=RBI*(N0+0.5)**3
   CALL ERFCL(XL,E,C,EL,CL)
   SNB=SNM+N0*N0
   ZB=ZI+(N0*N0)*(Z0-ZI)/SNB
   EBR=(Z0/N0)**2-ZB*ZB/SNB
   EBKT=EBR*RYBKT
   SL=-EBKT+CL+LOG(GI*RPB3/RBI)
   ZL=COMPL(ZL,SL)

```

```

EBR=EBR*EXP(SL-ZL)
1 ZM=EXP(ZL)
RETURN
END
C ERROR FUNCTION AND COMPLIMENT - ABRAMOWITZ AND STEGUN P. 297
C RANGE 0 TO 1: ERFC(X) = 1 - ERF(X) ... BOTH WITH NATURAL LOGS
SUBROUTINE ERFCL(X,E,C,EL,CL)
IMPLICIT REAL*8 (A-H,O-Z)
DATA P/0.3275911/
DATA A1/0.254829592/,A2/-0.284496736/,A3/1.421413741/
DATA A4/-1.453152027/,A5/1.061405430/
T=1.0/(1.0+P*X)
S=T*(A1+T*(A2+T*(A3+T*(A4+T*A5))))
X2=X*X
C=S*EXP(-X2)
E=1.0-C
IF(X)1,1,2
1 E=0.0
EL=-99.0
C=1.0
CL=0.0
GO TO 3
2 EL=LOG(E)
CL=LOG(S)-X2
3 RETURN
END
C OPACITY
C NB - OPACITY CHOSEN DEPENDS ON VALUE OF IO AND/OR NDO
SUBROUTINE OPACTY(IO,X,Y,Z,T,D,EN,RK,DKT,DKD)
C NB - RK IS MAGNITUDE OF OPACITY (NOT LOG) - DKT,DKD ARE LOG DERIVATIVES
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XAO(64),YAO(64),ZAO(64,64)
DATA ILQ/2/,ITO/0/,NDO/94/,INO/0/,ENO/8.0/
DATA BCK/1.38054E-16/,TTR/8.0E3/,TFW/4.0E3/
C NB - MOLECULAR OPACITIES USED WHEN T<TTR
C SPLICING OF OPACITIES OVER INTERVAL TFW (TLO,THI) CENTRED ON TTR
SQRT(X)=DSQRT(X)
ALOG10(X)=DLOG10(X)
TLO=TTR-TFW/2.0
THI=TTR+TFW/2.0
IF(T-THI)21,22,22
C INVOKE MOLECULAR OPACITY ROUTINES
21 CALL RMKMPZ(X,Y,Z,T,D,RKL,DKT,DKD)
RK=10.0**RKL
RKL=RKL
DKTL=DKT
DKDL=DKD
IF(T-TLO)23,23,24
23 FLO=1.0
FHI=0.0
GO TO 30
24 FLO=(THI-T)/TFW
FHI=(T-TLO)/TFW
GO TO 25
22 CONTINUE
FLO=0.0
FHI=1.0
25 CONTINUE
C INVOKE OTHER OPACITY PROCEDURES
C CHECK IF TABLES ARE REQUIRED: NDO>0
NDO=0
IF(NDO)10,10,20
10 CONTINUE
C CHOOSE PROCEDURE ACCORDING TO VALUE OF IO
IO=3
IF(IO-2)11,12,13
11 CONTINUE
C USE CHRISTY FORMULA (NB - THIS NEEDS ELECTRON DENSITY EN OR USE KC=0)
C DERIVATIVES COMPUTED ACCORDING TO VALUE OF ND
ND=2
KC=0
CALL CHRISTY (ND,KC,X,Y,Z,T,D,EN,RK,DKT,DKD)
RKLH=ALOG10(RK)
DKTH=DKT
DKDH=DKD
GO TO 30
12 CONTINUE

```

```

C USE IGLESIAS-ROGERS TABLES
CALL RKIRPZ(X,Y,Z,T,D,RKL,DKT,DKD)
RK=10.0**RKL
RKLH=RKL
DKTH=DKT
DKDH=DKD
GO TO 30
13 CONTINUE
C USE AN ALTERNATIVE
C FOR Z=0 USE ZEROZ TABLES
IF(Z)14,14,15
14 CONTINUE
C CALL ZEROZK(X,Y,Z,T,D,RKL,RKLT,RKLD)
C GO TO 19
15 CONTINUE
C INVOKE OPAL TABLES
CALL OPALS(X,Y,Z,T,D,RKL,RKLT,RKLD,RKLT,TKLDD,RKLT,DKLDT)
19 CONTINUE
RK=10.0**RKL
DKT=RKLT
DKD=RKLD
RKLH=RKL
DKTH=DKT
DKDH=DKD
GO TO 30
20 CONTINUE
C USE ROUTINE TABLES READING FROM A DATA FILE
TL=ALOG10(T)
DL=ALOG10(D)
CALL TABLES(ILQ,ITP,NDO,INO,NXO,NYO,DXO,CO,XT,YT,ZT,
* ENO,XAO,YAO,ZAO,DL,TL,OL,DOD,DOT)
RK=10.0**OL
C CORRECTION IN SCATTERING LIMIT IF X<>XT
RK=RK*(1.0+X)/(1.0+XT)
DKT=DOT
DKD=DOD
RKLH=ALOG10(RK)
DKTH=DKT
DKDH=DKD
30 CONTINUE
RKL=FLO*RKLL+FHI*RKLH
DKT=FLO*DKTL+FHI*DKTH
DKD=FLO*DKDL+FHI*DKDH
RK=10.0**RKL
RETURN
END
C CHRISTY OPACITY FIT
C WITH OR WITHOUT FREE ELECTRON DENSITY EN
SUBROUTINE CHRISTY (ND,KC,X,Y,Z,T,D,EN,RK,DKT,DKD)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RKI(3)
DATA EPS/0.001/
DO 10 ID=1,ND+1
GO TO (11,12,13),ID
11 TI=T
DI=D
GO TO 15
12 TI=T*(1.0+EPS)
DI=D
GO TO 15
13 TI=T
DI=D*(1.0+EPS)
GO TO 15
15 CONTINUE
CALL CHRISTYS (KC,X,Y,Z,TI,DI,EN,RK,DOT,DOD)
RKI(ID)=RK
GO TO (21,22,23),ID
21 CONTINUE
DKT=DOT
DKD=DOD
GO TO 10
22 DKT=(RKI(2)/RKI(1)-1.0)/EPS
GO TO 10
23 DKD=(RKI(3)/RKI(1)-1.0)/EPS
GO TO 10
10 CONTINUE
RK=RKI(1)

```

```

RETURN
END
C  CHRISTY FIT SUBROUTINE
SUBROUTINE CHRISTYS (KC,X,Y,Z,T,D,EN,RK,DKT,DKD)
IMPLICIT REAL*8 (A-H,O-Z)
DATA BCK/1.38054E-16/,AVN/6.022169E+23/,THT/5040.0/,GEL/15.68391/
DATA XZ/1.0/YZ/2.0/ZZ/6.0/XM/1.0/YM/4.0/ZM/12.0/
DATA XI/13.598/,YI/24.587/,ZI/11.260/
SQRT(X)=DSQRT(X)
IF(KC)1,1,2
1  EN=AVN*(1+X)*D/2.0
C  EN=EN*10.0**(-5040/T)
AM=1.0/(X/XM+Y/YM+Z/ZM)
AN=AVN*D/AM
XN=AN*AM*X/XM
YN=AN*AM*Y/YM
ZN=AN*AM*Z/ZM
PC=10.0**GEL*T**1.5
EC=THT/T
CX=PC*10.0**(-EC*XI)
FX=1.0
IF(XN)10,10,11
11  CX=CX/XN
FX=FION(CX)
10  CONTINUE
CY=PC*10.0**(-EC*YI)
FY=1.0
IF(YN)20,20,21
21  CY=CY/YN
FY=FION(CY)
20  CONTINUE
CZ=PC*10.0**(-EC*ZI)
FZ=1.0
IF(ZN)30,30,31
31  CZ=CZ/ZN
FZ=FION(CZ)
30  CONTINUE
FF=XZ*X/XM*FX+YZ*Y/YM*FY+ZZ*Z/ZM*FZ
EN=2.0/(1.0+X)*EN*FF
2  CONTINUE
T4=1.0E-4*T
T44=T4**4
T46=T44*T4*T4
T4H=SQRT(T4)
SK=0.485E-12/(T4*D*(1+2.2E-5*T4))
XK1=2.1*T46+2.0E6/T44
XK2=T4*(4.0E-3/T44+2.0E-4/SQRT(SQRT(D)))
XK=X*(T4H/XK1+1/(4.5*T46+1/XK2))
YK=Y*15/(T46+1.0E7)
ZK=Z*T4H/(20*T4+35*T44+18*((1+X)*D)**0.33*T4**4.71)
RK=BCK*EN*T*(SK+XK+YK+ZK)
C  NB - NO INLINE DERIVATIVES
DKT=0.0
DKD=0.0
RETURN
END
C  SOLVE QUADRATIC FOR IONIZED FRACTION
FUNCTION FION(R)
IMPLICIT REAL*8 (A-H,O-Z)
SQRT(X)=DSQRT(X)
IF(R-0.01)1,1,2
1  FION=SQRT(R)*(1.0+R/8.0-R*R/128.0+R*R*R/1024.0)-R/2.0
GO TO 5
2  IF(R-100.0)3,3,4
3  FION=(SQRT(R*(R+4.0))-R)/2.0
GO TO 5
4  FION=1.0-1.0/R+2.0/R/R-5.0/R/R/R
5  RETURN
END
C  TEST RMKMPZ
SUBROUTINE TRMKMPZ
C  INTERPOLATE FOR GIVEN X,Y,Z IN MOLECULAR OPACITY TABLES
IMPLICIT REAL*8 (A-H,O-Z)
X=0.776
Y=0.214
Z=0.010
DO 1 IT=1,5

```

```

TL=3+0.1*(2*IT+1)
T=10.0**TL
T6L=TL-6
DO 2 ID=1,5
RL=ID-6
DL=RL+3.0*T6L
DL=2*RL
D=10.0**DL
CALL RMKMPZ(X,Y,Z,T,D,RKL,RKLT,RKLD)
C WRITE(6,6)T,D,RKL,RKLT,RKLD
2 CONTINUE
1 CONTINUE
STOP
6 FORMAT(1P5E15.5)
END
C FOR GIVEN X,Y,Z INTERPOLATE IN MOLECULAR OPACITY TABLES
SUBROUTINE RMKMPZ(X,Y,Z,T,D,RKL,DKT,DKD)
IMPLICIT REAL*8 (A-H,O-Z)
DATA XT1/0.770/,YT1/0.212/,ZT1/0.018/
DATA XT2/0.783/,YT2/0.216/,ZT2/0.001/
DATA XT3/0.750/,YT3/0.250/,ZT3/0.000/
DATA XT4/1.000/,YT4/0.000/,ZT4/0.000/
DATA KON/0/
IF(KON)100,100,101
100 WRITE(6,*)' RMKMPZ: ACCESSING MOLECULAR OPACITIES'
KON=1
101 CONTINUE
IF(Z)1,1,2
1 DY=Y-YT4
FY=DY/(YT3-YT4)
CALL RMKMP3(D,T,RKL3,DKD3,DKT3)
CALL RMKMP4(D,T,RKL4,DKD4,DKT4)
RKL=FY*RKL3+(1.0-FY)*RKL4
DKT=FY*DKT3+(1.0-FY)*DKT4
DKD=FY*DKD3+(1.0-FY)*DKD4
C WRITE(6,6)RKL,DKT,DKD
GO TO 10
2 IF(Z-ZT2)3,4,4
3 DY=Y-YT4
FY=DY/(YT3-YT4)
CALL RMKMP3(D,T,RKL3,DKD3,DKT3)
CALL RMKMP4(D,T,RKL4,DKD4,DKT4)
RKLY=FY*RKL3+(1.0-FY)*RKL4
DKTY=FY*DKT3+(1.0-FY)*DKT4
DKDY=FY*DKD3+(1.0-FY)*DKD4
C WRITE(6,6)RKLY,DKTY,DKDY
ZY=ZT4
DZ=Z-ZY
FZ=DZ/(ZT2-ZY)
CALL RMKMP2(D,T,RKL2,DKD2,DKT2)
RKL=FZ*RKL2+(1.0-FZ)*RKLY
DKT=FZ*DKT2+(1.0-FZ)*DKTY
DKD=FZ*DKD2+(1.0-FZ)*DKDY
C WRITE(6,6)RKL,DKT,DKD
GO TO 10
4 DZ=Z-ZT2
FZ=DZ/(ZT1-ZT2)
CALL RMKMP1(D,T,RKL1,DKD1,DKT1)
CALL RMKMP2(D,T,RKL2,DKD2,DKT2)
RKL=FZ*RKL1+(1.0-FZ)*RKL2
DKT=FZ*DKT1+(1.0-FZ)*DKT2
DKD=FZ*DKD1+(1.0-FZ)*DKD2
C WRITE(6,6)RKL,DKT,DKD
GO TO 10
10 RK=10.0**RKL
RETURN
6 FORMAT(3F10.6)
END
C ROSSELAND MEAN OPACITY FOR POP1 (X=0.770,Y=0.212,Z=0.018)
C WITH MOLECULES ()
C CONTAINS REVISIONS OF AUGUST 1989
SUBROUTINE RMKMP1(X,Y,Z,ZX,ZY)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RMK(12,8)
ALOG10(X)=DLOG10(X)
DATA DX/1.0/,DY/0.1/
DATA OX/-13.0/,OY/3.3/

```

```

DATA RMK/
1 -4.8519,-3.8143,-3.4351,-3.3359,-3.2879,-3.1696,
2 -3.1488,-3.0539,-2.8333,-1.8675,-1.2470,-0.8055,
3 -4.7305,-4.8717,-4.7283,-3.8230,-2.7508,-2.3034,
4 -2.0431,-2.0198,-1.8235,-1.0334,-0.4012,-0.0834,
5 -4.1706,-4.1404,-4.1145,-3.9506,-3.5047,-2.6620,
6 -1.6976,-1.1834,-0.8911,-0.2212, 0.4315, 0.6858,
7 -3.6982,-3.7504,-3.7048,-3.3392,-2.7168,-2.0579,
8 -1.4265,-0.8063,-0.2242, 0.5125, 1.1701, 1.3762,
9 -2.1740,-2.5399,-2.7415,-2.6190,-2.2030,-1.6110,
A -0.9037,-0.1955, 0.4806, 1.2823, 1.8860, 1.9960,
B -0.8175,-1.1522,-1.4525,-1.4995,-1.2223,-0.8056,
C -0.3269, 0.2207, 0.8586, 1.6029, 2.4178, 2.5891,
D -0.4856,-0.3697,-0.1974,-0.0796, 0.0484, 0.2573,
E 0.5532, 0.9194, 1.3816, 1.9729, 2.8130, 3.2851,
F -0.4972,-0.4179,-0.0851, 0.5643, 1.1301, 1.4066,
G 1.5833, 1.7978, 2.1133, 2.5740, 3.2357, 3.9228/
XL=ALOG10(X)
YL=ALOG10(Y)
IX=(XL-OX)/DX+1
IX=MIN(11,(MAX(1,IX)))
IY=(YL-OY)/DY+1
IY=MIN( 7,(MAX(1,IY)))
XIX=OX+DX*(IX-1)
YIY=OY+DY*(IY-1)
FX=(XL-XIX)/DX
FY=(YL-YIY)/DY
Z=(1.0-FX)*((1.0-FY)*RMK(IX ,IY)+FY*RMK(IX ,IY+1))+
* FX *((1.0-FY)*RMK(IX+1,IY)+FY*RMK(IX+1,IY+1))
Z00=RMK(IX,IY)-RMK(IX+1,IY)-RMK(IX,IY+1)+RMK(IX+1,IY+1)
ZX=(RMK(IX+1,IY)-RMK(IX,IY)+FY*Z00)/DX
ZY=(RMK(IX,IY+1)-RMK(IX,IY)+FX*Z00)/DY
RETURN
END
C  ROSSELAND MEAN OPACITY FOR POP2 (X=0.783,Y=0.216,Z=0.001)
C  WITH MOLECULES ()
C  CONTAINS REVISIONS OF AUGUST 1989
SUBROUTINE RMKMP2(X,Y,Z,ZX,ZY)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RMK(12,8)
ALOG10(X)=DLOG10(X)
DATA DX/1.0/,DY/0.1/
DATA OX/-13.0/,OY/3.3/
DATA RMK/
1 -5.5111,-4.5023,-3.9222,-3.8427,-3.7936,-3.8692,
2 -3.8305,-3.7394,-3.5337,-2.7096,-1.9004,-1.4013,
3 -5.2816,-5.3820,-5.3196,-4.7145,-3.6791,-3.2699,
4 -3.1433,-2.9554,-2.7050,-1.8271,-1.1178,-0.7485,
5 -4.9629,-4.9350,-4.8687,-4.6893,-4.2920,-3.6263,
6 -2.7646,-2.2360,-1.8644,-1.1789,-0.4234,-0.0717,
7 -3.7915,-4.1155,-4.3029,-4.2514,-3.7933,-3.0785,
8 -2.3026,-1.6377,-1.0908,-0.3087, 0.3784, 0.5751,
9 -2.1744,-2.6105,-2.9310,-2.9167,-2.5884,-2.1295,
A -1.6083,-1.0187,-0.3693, 0.3434, 0.9853, 1.2658,
B -0.8129,-1.1481,-1.4535,-1.5372,-1.3075,-0.9270,
C -0.4958,-0.0207, 0.5081, 1.0807, 1.7285, 2.3117,
D -0.4784,-0.3616,-0.1893,-0.0718, 0.0560, 0.2624,
E 0.5533, 0.9122, 1.3543, 1.9000, 2.5611, 3.2320,
F -0.4903,-0.4098,-0.0747, 0.5763, 1.1401, 1.4151,
G 1.5914, 1.8054, 2.1196, 2.5758, 3.2184, 3.9209/
XL=ALOG10(X)
YL=ALOG10(Y)
IX=(XL-OX)/DX+1
IX=MIN(11,(MAX(1,IX)))
IY=(YL-OY)/DY+1
IY=MIN( 7,(MAX(1,IY)))
XIX=OX+DX*(IX-1)
YIY=OY+DY*(IY-1)
FX=(XL-XIX)/DX
FY=(YL-YIY)/DY
Z=(1.0-FX)*((1.0-FY)*RMK(IX ,IY)+FY*RMK(IX ,IY+1))+
* FX *((1.0-FY)*RMK(IX+1,IY)+FY*RMK(IX+1,IY+1))
Z00=RMK(IX,IY)-RMK(IX+1,IY)-RMK(IX,IY+1)+RMK(IX+1,IY+1)
ZX=(RMK(IX+1,IY)-RMK(IX,IY)+FY*Z00)/DX
ZY=(RMK(IX,IY+1)-RMK(IX,IY)+FX*Z00)/DY
RETURN
END

```

```

C  ROSSELAND MEAN OPACITY FOR POP3 (X=0.750,Y=0.250,Z=0.000)
C  WITH MOLECULES (H2,H2+)
SUBROUTINE RMKMP3(X,Y,Z,ZX,ZY)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RMK(12,8)
ALOG10(X)=DLOG10(X)
DATA DX/1.0/,DY/0.1/
DATA OX/-13.0/,OY/3.3/
DATA RMK/
1 -7.7090,-7.6466,-7.2489,-6.6594,-6.1653,-5.6111,
2 -4.9947,-4.5160,-4.1111,-3.7207,-3.3277,-2.9284,
3 -6.5621,-6.6278,-6.5880,-6.5048,-6.2525,-5.5635,
4 -4.9204,-4.5000,-4.1453,-3.7977,-3.4457,-3.0851,
5 -5.3023,-5.4977,-5.6024,-5.5624,-5.4448,-5.3019,
6 -4.9639,-4.3927,-4.0679,-3.7629,-3.3957,-2.5760,
7 -3.8097,-4.1414,-4.3837,-4.4274,-4.1954,-3.8028,
8 -3.3513,-2.9289,-2.5906,-2.2834,-1.7784,-0.7330,
9 -2.1848,-2.6222,-2.9470,-2.9563,-2.6459,-2.2094,
A -1.7315,-1.2336,-0.7466,-0.3648, 0.1048, 0.9458,
B -0.8260,-1.1603,-1.4678,-1.5575,-1.3329,-0.9544,
C -0.5254,-0.0559, 0.4604, 1.0071, 1.5732, 2.2712,
D -0.4981,-0.3835,-0.2109,-0.0926, 0.0342, 0.2390,
E 0.5286, 0.8860, 1.3263, 1.8700, 2.5285, 3.2119,
F -0.5089,-0.4316,-0.1030, 0.5444, 1.1138, 1.3926,
G 1.5693, 1.7821, 2.0943, 2.5489, 3.1914, 3.9003/
XL=ALOG10(X)
YL=ALOG10(Y)
IX=(XL-OX)/DX+1
IX=MIN(11,(MAX(1,IX)))
IY=(YL-OY)/DY+1
IY=MIN( 7,(MAX(1,IY)))
XIX=OX+DX*(IX-1)
YIY=OY+DY*(IY-1)
FX=(XL-XIX)/DX
FY=(YL-YIY)/DY
Z=(1.0-FX)*(1.0-FY)*RMK(IX ,IY)+FY*RMK(IX ,IY+1))+
* FX *(1.0-FY)*RMK(IX+1,IY)+FY*RMK(IX+1,IY+1)
Z00=RMK(IX,IY)-RMK(IX+1,IY)-RMK(IX,IY+1)+RMK(IX+1,IY+1)
ZX=(RMK(IX+1,IY)-RMK(IX,IY)+FY*Z00)/DX
ZY=(RMK(IX,IY+1)-RMK(IX,IY)+FX*Z00)/DY
RETURN
END
C  ROSSELAND MEAN OPACITY FOR POP4 (X=1.000,Y=0.000,Z=0.000)
C  WITH MOLECULES (H2,H2+)
SUBROUTINE RMKMP4(X,Y,Z,ZX,ZY)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RMK(12,8)
ALOG10(X)=DLOG10(X)
DATA DX/1.0/,DY/0.1/
DATA OX/-13.0/,OY/3.3/
DATA RMK/
1 -7.5836,-7.5246,-7.0907,-6.4821,-5.9943,-5.4276,
2 -4.8226,-4.3561,-3.9558,-3.5677,-3.1766,-2.7797,
3 -6.4538,-6.5066,-6.4583,-6.3754,-6.0691,-5.3741,
4 -4.7546,-4.3460,-3.9941,-3.6478,-3.2968,-2.9376,
5 -5.2062,-5.3952,-5.4837,-5.4302,-5.3083,-5.1673,
6 -4.7988,-4.2372,-3.9232,-3.6186,-3.2546,-2.4526,
7 -3.7305,-4.0543,-4.2803,-4.2915,-4.0328,-3.6292,
8 -3.1776,-2.7622,-2.4499,-2.1632,-1.6650,-0.6135,
9 -2.1170,-2.5476,-2.8455,-2.8080,-2.4731,-2.0294,
A -1.5498,-1.0439,-0.5819,-0.2357, 0.2155, 1.0495,
B -0.7394,-1.0785,-1.3708,-1.4204,-1.1665,-0.7794,
C -0.3470, 0.1331, 0.6420, 1.1697, 1.6989, 2.3764,
D -0.3656,-0.2367,-0.0683, 0.0456, 0.1793, 0.3952,
E 0.6928, 1.0622, 1.5051, 2.0476, 2.6791, 3.3280,
F -0.3830,-0.2840, 0.0888, 0.7557, 1.2866, 1.5411,
G 1.7165, 1.9400, 2.2622, 2.7257, 3.3598, 4.0322/
XL=ALOG10(X)
YL=ALOG10(Y)
IX=(XL-OX)/DX+1
IX=MIN(11,(MAX(1,IX)))
IY=(YL-OY)/DY+1
IY=MIN( 7,(MAX(1,IY)))
XIX=OX+DX*(IX-1)
YIY=OY+DY*(IY-1)
FX=(XL-XIX)/DX
FY=(YL-YIY)/DY

```

```

Z=(1.0-FX)*((1.0-FY)*RMK(IX ,IY)+FY*RMK(IX ,IY+1))+
* FX *((1.0-FY)*RMK(IX+1,IY)+FY*RMK(IX+1,IY+1))
Z00=RMK(IX,IY)-RMK(IX+1,IY)-RMK(IX,IY+1)+RMK(IX+1,IY+1)
ZX=(RMK(IX+1,IY)-RMK(IX,IY)+FY*Z00)/DX
ZY=(RMK(IX,IY+1)-RMK(IX,IY)+FX*Z00)/DY
RETURN
END
C TEST ZEROZ - USE OF ZERO METALS OPACITY TABLES
SUBROUTINE TZEROZ
IMPLICIT REAL*8 (A-H,O-Z)
Z=0.0
DO 1 IX=1,11
X=0.1*(IX-1)
Y=1.0-X
T=1.0E+05
D=1.0E-07
CALL ZEROZK(X,Y,Z,T,D,RK,RKLT,RKLD)
WRITE(6,66)X,Y,Z,T,D,RK,RKLT,RKLD
1 CONTINUE
STOP
66 FORMAT(1P8E10.3)
END
C INTERPOLATION IN N TABLES
C ACCESS N TABLES FROM A SINGLE FILE
SUBROUTINE ZEROZK(X,Y,Z,T,D,RK,RKLT,RKLD)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XA(64,8),YA(64,8),ZA(16,64,8)
DIMENSION ITA(8),INC(8),NXA(8),NYA(8),DXA(8),ZCA(8),END(8),NAME(8)
DIMENSION XTA(8),YTA(8),ZTA(8)
DIMENSION RKLTA(2),RKLDA(2),RKLTA(2)
DATA ITA /8*0/,N/5/,KON/0/
DATA EDY /1.0E-6/
IF(KON-N)100,101,101
100 CONTINUE
OPEN(UNIT=1,FILE="zeroz.d")
C OPEN(UNIT=1,NAME='ZEROZ.DAT',TYPE='OLD')
C WRITE(6,*) ZEROZK: ACCESSING ZEROZ.DAT FOR ZERO METALS OPACITY'
WRITE(6,*) ZEROZK: ACCESSING zeroz.d FOR ZERO METALS OPACITY'
NDS=1
DO 1 I=1,N
IF(ITA(I))10,10,11
10 WRITE(6,50)NDS
50 FORMAT(1X,'FILE ',I4)
READ (NDS,51)NAME(I),NYA(I),NXA(I),DXA(I),ZCA(I),
1XTA(I),YTA(I),ZTA(I)
WRITE( 6,51)NAME(I),NYA(I),NXA(I),DXA(I),ZCA(I),
1XTA(I),YTA(I),ZTA(I)
51 FORMAT(6X,A4,2I5,F10.5,E10.3,3F10.5)
J=0
20 J=J+1
READ (NDS,52)YA(J,I),XA(J,I),(ZA(K,J,I),K=1,NXA(I))
C WRITE( 6,52)YA(J,I),XA(J,I),(ZA(K,J,I),K=1,NXA(I))
52 FORMAT(F4.1,F6.1,10F10.5,(10X,10F10.5))
IF(J-NYA(I))22,23,22
22 IF(YA(J,I)-END(I))20,21,20
21 NYA(I)=J+INC(I)
23 ITA(I)=1
11 CONTINUE
1 CONTINUE
KON=N
101 CONTINUE
LQ=2
TL=LOG10(T)
DL=LOG10(D)
DO 2 IT=1,N
KT=IT
DY=Y-YTA(IT)
IF(ABS(DY)-EDY)3,3,4
4 CONTINUE
IF(DY)5,3,2
2 CONTINUE
5 KT=KT-1
GO TO 6
3 CALL NINTERP(KT,LQ,NXA(KT),NYA(KT),DXA(KT),XA,YA,ZA,
1DL,TL,RKL,RKLD,RKLT)
WRITE(6,66)KT,XTA(KT),YTA(KT),ZTA(KT),RKL,RKLD,RKLT
GO TO 9

```

```

6 DO 7 K=1,2
  IT=KT+K-1
  CALL NINTERP(IT,LQ,NXA(IT),NYA(IT),DXA(IT),XA,YA,ZA,
  1DL,TL,RKLA(K),RKLDA(K),RKLTA(K))
  WRITE(6,66)IT,XTA(IT),YTA(IT),ZTA(IT),RKLA(K),RKLDA(K),RKLTA(K)
7 CONTINUE
  DYT=YTA(KT+1)-YTA(KT)
  F1=(YTA(KT+1)-Y)/DYT
  F2=(Y-YTA(KT))/DYT
  RKL=F1*RKLA(1)+F2*RKLA(2)
  RKLD=F1*RKLD(1)+F2*RKLD(2)
  RKLTA=F1*RKLTA(1)+F2*RKLTA(2)
9 CONTINUE
  RK=10.0**RKL
  RETURN
66 FORMAT(I5,1P6E10.3)
END
SUBROUTINE NINTERP(IT,LQ,NX,NY,DX,XA,YA,ZA,X,Y,Z,DZX,DZY)
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION XA(64,8),YA(64,8),ZA(16,64,8),XY(64)
  DIMENSION XX(3),YY(3),ZX(3),ZY(3),DZ(3)
  CF(X1,X2,X3,Z1,Z2,Z3)=((Z1-Z2)/(X1-X2)-(Z2-Z3)/(X2-X3))/(X1-X3)
  BF(X1,X2,Z1,Z2)=(Z1-Z2)/(X1-X2)-C*(X1+X2)
  AF(X1,Z1)=Z1-(B+C*X1)*X1
  NLQ=LQ+1
  IY=1
3 IF(Y-YA(IY,IT))1,20,20
20 IF(IY-NY+1)2,2,1
2 IY=IY+1
  GO TO 3
1 IF(2*Y-YA(IY,IT)-YA(IY+1,IT))4,5,5
5 IY=IY+1
4 IY=MAX0(IY,2)
  IY=MIN0(IY,NY+1-LQ)
  DO 6 JY=1,NLQ
  KY=IY+JY-2
  YY(JY)=YA(KY,IT)
  KK=0
  IF(DX)14,15,14
15 KK=1
14 DO 7 IX=1,NX
7 XY(IX)=XA(KY+KK*(IX-1),IT)+(IX-1)*DX
  IX=1
10 IF(X-XY(IX))8,90,90
90 IF(IX-NX+1)9,8,8
9 IX=IX+1
  GO TO 10
8 IF(2*X-XY(IX)-XY(IX+1))11,12,12
12 IX=IX+1
11 IX=MAX0(IX,2)
  IX=MIN0(IX,NX+1-LQ)
  DO 13 JX=1,NLQ
  KX=IX+JX-2
  XX(JX)=XY(KX)
13 ZX(JX)=ZA(KX,KY,IT)
  C=0.0
  GO TO (21,22),LQ
22 C=CF(XX(1),XX(2),XX(3),ZX(1),ZX(2),ZX(3))
21 B=BF(XX(1),XX(2),ZX(1),ZX(2))
  A=AF(XX(1),ZX(1))
  DZX=B+2*C*X
D WRITE(6,61)XX,ZX,A,B,C,DZX
  DZ(JY)=DZX
6 ZY(JY)=A+(B+C*X)*X
  C=0.0
  GO TO (31,32),LQ
32 C=CF(YY(1),YY(2),YY(3),ZY(1),ZY(2),ZY(3))
31 B=BF(YY(1),YY(2),ZY(1),ZY(2))
  A=AF(YY(1),ZY(1))
D WRITE(6,61)YY,ZY,A,B,C
  Z=A+(B+C*Y)*Y
  DZY=B+2*C*Y
  C=0.0
  GO TO (41,42),LQ
42 C=CF(YY(1),YY(2),YY(3),DZ(1),DZ(2),DZ(3))
41 B=BF(YY(1),YY(2),DZ(1),DZ(2))
  A=AF(YY(1),DZ(1))

```

```

DZX=A+(B+C*Y)*Y
RETURN
D61 FORMAT(10F10.5)
END
C TEST RKIRPZ
SUBROUTINE TRKIRPZ
IMPLICIT REAL*8 (A-H,O-Z)
X=0.0
Y=0.0
Z=0.01
DO 1 IT=1,5
  TL=3+IT
  T=10.0**TL
  T6L=TL-6
  DO 2 ID=1,5
    RL=ID-6
    DL=RL+3.0*T6L
    D=10.0**DL
    CALL RKIRPZ(X,Y,Z,T,D,RKL,RKLT,RKLD)
  WRITE(6,6)T,D,RKL,RKLT,RKLD
2 CONTINUE
1 CONTINUE
STOP
6 FORMAT(1P5E15.5)
END
C INTERPOLATE IN IGLESIAS-ROGERS TABLES FOR GIVEN Z
SUBROUTINE RKIRPZ(X,Y,Z,T,D,RKL,RKLT,RKLD)
IMPLICIT REAL*8 (A-H,O-Z)
DATA ZT1/0.020/,ZT2/0.001/,DZT/0.019/
DATA KON/0/
IF(KON)100,100,101
100 WRITE(6,*)' RKIRPZ: ACCESSING IGLESIAS-ROGERS OPACITIES'
    KON=1
101 CONTINUE
    DZ=Z-ZT1
    FZ=DZ/DZT
    CALL RKIRP1(T,D,RKL1,RKLT1,RKLD1)
    CALL RKIRP2(T,D,RKL2,RKLT2,RKLD2)
    RKL=RKL1-FZ*(RKL2-RKL1)
    RKLT=RKLT1-FZ*(RKLT2-RKLT1)
    RKLD=RKLD1-FZ*(RKLD2-RKLD1)
    RETURN
    END
C TEST RKIRP1
SUBROUTINE TRKIRP1
IMPLICIT REAL*8 (A-H,O-Z)
DO 1 IT=1,5
  TL=IT+3
  T=10.0**TL
  T6L=TL-6
  DO 2 ID=1,5
    RL=ID-6
    DL=RL+3.0*T6L
    D=10.0**DL
    CALL RKIRP1(T,D,RKL,RKLT,RKLD)
  WRITE(6,6)T,D,RKL,RKLT,RKLD
2 CONTINUE
1 CONTINUE
STOP
6 FORMAT(1P5E15.5)
END
SUBROUTINE RKIRP1(T,D,RKL,RKLT,RKLD)
C OPACITY FOR X=0.700,Y=0.280,Z=0.020
C IGLESIAS AND ROGERS: AP. J. (LETTERS),371,L73-L75,20 APRIL 1991
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RLA(9),T6A(49),T6LA(49),RKLA(9,49)
ALOG10(X)=DLOG10(X)
DATA T6A/
1 0.006, 0.007, 0.008, 0.009, 0.010, 0.012, 0.014, 0.016, 0.018,
2 0.020, 0.025, 0.030, 0.035, 0.040, 0.045, 0.050, 0.055, 0.060,
3 0.070, 0.080, 0.090, 0.100, 0.120, 0.150, 0.200, 0.250, 0.300,
4 0.400, 0.500, 0.600, 0.800, 1.000, 1.200, 1.500, 2.000, 2.500,
5 3.000, 4.000, 5.000, 6.000, 8.000, 10.00, 15.00, 20.00, 30.00,
6 40.00, 60.00, 80.00, 100.0/
DATA RLA/
1-5.000,-4.500,-4.000,-3.500,-3.000,-2.500,-2.000,-1.500,-1.000/
DATA RKLA/

```

1-1.572,-1.650,-1.801,-1.818,-1.782,-1.597,-1.437,-1.224,-1.000,
 2-0.709,-0.749,-0.814,-0.808,-0.748,-0.646,-0.517,-0.346,-0.158,
 3-0.207,-0.127,-0.019, 0.053, 0.118, 0.211, 0.312, 0.435, 0.579,
 4-0.068, 0.194, 0.451, 0.649, 0.805, 0.916, 1.014, 1.117, 1.223,
 5-0.069, 0.272, 0.614, 0.960, 1.226, 1.421, 1.558, 1.660, 1.755,
 6-0.130, 0.201, 0.584, 1.026, 1.473, 1.855, 2.149, 2.333, 2.472,
 7-0.139, 0.130, 0.482, 0.915, 1.393, 1.872, 2.282, 2.624, 2.869,
 8-0.143, 0.113, 0.442, 0.845, 1.304, 1.808, 2.275, 2.717, 3.076,
 9-0.150, 0.088, 0.409, 0.814, 1.262, 1.758, 2.234, 2.727, 3.168,
 X-0.159, 0.066, 0.380, 0.784, 1.232, 1.713, 2.220, 2.725, 3.211,
 1-0.144, 0.060, 0.364, 0.771, 1.248, 1.751, 2.280, 2.799, 3.314,
 2-0.068, 0.125, 0.423, 0.833, 1.322, 1.833, 2.374, 2.916, 3.462,
 3 0.018, 0.231, 0.536, 0.937, 1.419, 1.940, 2.477, 3.016, 3.581,
 4 0.046, 0.317, 0.657, 1.066, 1.533, 2.041, 2.569, 3.108, 3.657,
 5 0.044, 0.346, 0.724, 1.179, 1.647, 2.135, 2.644, 3.161, 3.698,
 6 0.020, 0.309, 0.693, 1.169, 1.675, 2.180, 2.679, 3.184, 3.699,
 7-0.009, 0.279, 0.654, 1.115, 1.633, 2.161, 2.676, 3.177, 3.671,
 8-0.039, 0.235, 0.595, 1.041, 1.553, 2.092, 2.627, 3.141, 3.630,
 9-0.072, 0.177, 0.517, 0.947, 1.445, 1.967, 2.508, 3.047, 3.545,
 X-0.094, 0.140, 0.463, 0.876, 1.365, 1.885, 2.424, 2.969, 3.482,
 1-0.091, 0.136, 0.456, 0.868, 1.343, 1.852, 2.390, 2.928, 3.444,
 2-0.085, 0.157, 0.477, 0.874, 1.350, 1.852, 2.380, 2.913, 3.427,
 3-0.026, 0.198, 0.513, 0.922, 1.392, 1.875, 2.382, 2.907, 3.406,
 4 0.100, 0.311, 0.611, 1.004, 1.465, 1.944, 2.427, 2.903, 3.339,
 5 0.148, 0.380, 0.682, 1.050, 1.476, 1.931, 2.387, 2.801, 3.134,
 6 0.016, 0.285, 0.612, 0.995, 1.412, 1.838, 2.268, 2.640, 2.919,
 7-0.110, 0.154, 0.492, 0.904, 1.333, 1.750, 2.146, 2.484, 2.738,
 8-0.293,-0.077, 0.264, 0.728, 1.174, 1.591, 1.943, 2.239, 2.475,
 9-0.363,-0.224, 0.058, 0.507, 1.003, 1.448, 1.797, 2.085, 2.316,
 X-0.372,-0.271,-0.056, 0.325, 0.829, 1.319, 1.692, 1.984, 2.220,
 1-0.394,-0.304,-0.132, 0.163, 0.603, 1.099, 1.516, 1.835, 2.089,
 2-0.403,-0.327,-0.177, 0.093, 0.498, 0.979, 1.416, 1.754, 2.017,
 3-0.392,-0.326,-0.193, 0.049, 0.432, 0.900, 1.356, 1.712, 1.980,
 4-0.374,-0.310,-0.187, 0.028, 0.379, 0.843, 1.304, 1.671, 1.935,
 5-0.382,-0.298,-0.155, 0.066, 0.382, 0.817, 1.253, 1.593, 1.817,
 6-0.413,-0.330,-0.179, 0.064, 0.402, 0.804, 1.181, 1.462, 1.648,
 7-0.429,-0.362,-0.230, 0.011, 0.356, 0.735, 1.058, 1.297, 1.469,
 8-0.448,-0.407,-0.315,-0.127, 0.180, 0.504, 0.775, 0.990, 1.174,
 9-0.456,-0.428,-0.364,-0.225, 0.017, 0.297, 0.548, 0.764, 0.962,
 X-0.462,-0.440,-0.391,-0.279,-0.086, 0.155, 0.388, 0.602, 0.810,
 1-0.468,-0.453,-0.421,-0.343,-0.198,-0.007, 0.194, 0.402, 0.622,
 2-0.469,-0.457,-0.431,-0.374,-0.257,-0.097, 0.092, 0.292, 0.514,
 3-0.475,-0.463,-0.437,-0.385,-0.299,-0.181,-0.026, 0.152, 0.335,
 4-0.485,-0.475,-0.454,-0.407,-0.324,-0.221,-0.098, 0.038, 0.198,
 5-0.497,-0.493,-0.482,-0.455,-0.397,-0.320,-0.229,-0.122, 0.034,
 6-0.507,-0.505,-0.499,-0.482,-0.443,-0.386,-0.313,-0.216,-0.069,
 7-0.526,-0.524,-0.520,-0.512,-0.491,-0.455,-0.405,-0.331,-0.205,
 8-0.544,-0.545,-0.543,-0.534,-0.520,-0.497,-0.465,-0.411,-0.298,
 9-0.563,-0.562,-0.560,-0.557,-0.546,-0.532,-0.514,-0.477,-0.370/
 DATA KON/0/
 IF(KON)1,1,2
 1 DO 3 I=1,49
 3 T6LA(I)=ALOG10(T6A(I))
 KON=1
 2 T6=1.0E-06*T
 T6L=ALOG10(T6)
 R=D/T6**3
 RL=ALOG10(R)
 IR1=1
 DO 4 I=2,8
 IF(RL-RLA(I))4,5,5
 5 IR1=1
 4 CONTINUE
 IR2=IR1+1
 IT1=1
 DO 6 I=2,48
 IF(T6L-T6LA(I))6,7,7
 7 IT1=1
 6 CONTINUE
 IT2=IT1+1
 DRL=RLA(IR2)-RLA(IR1)
 DT6L=T6LA(IT2)-T6LA(IT1)
 URL=(RL-RLA(IR1))/DRL
 UTL=(T6L-T6LA(IT1))/DT6L
 RKL1=RKLA(IR1,IT1)
 RKL2=RKLA(IR2,IT1)
 RKL12=RKLA(IR1,IT2)

```

RKL22=RKLA(IR2,IT2)
URLC=1.0-URL
UTLC=1.0-UTL
RKL=URLC*UTLC*RKL11+URL*UTLC*RKL21+URLC*UTL*RKL12+URL*UTL*RKL22
VZ=RKL11-RKL21-RKL12+RKL22
RKL21=(RKL21-RKL11+UTL*VZ)/DRL
RKL21=(RKL12-RKL11+URL*VZ)/DT6L
RKLD=RKLR
RKL21=RKL21-3.0*RKLR
C WRITE(6,16)IT1,IT2,IR1,IR2,RKL11,RKL21,RKL12,RKL22,URL,UTL
C WRITE(6,16)IT1,IT2,IR1,IR2,T,D,RKL,RKLT,RKLD
RETURN
16 FORMAT(4I5,1P6E10.3)
END
C TEST RKIRP2
SUBROUTINE TRKIRP2
IMPLICIT REAL*8 (A-H,O-Z)
DO 1 IT=1,5
  TL=IT+3
  T=10.0**TL
  T6L=TL-6
  DO 2 ID=1,5
    RL=ID-6
    DL=RL+3.0*T6L
    D=10.0**DL
    CALL RKIRP2(T,D,RKL,RKLT,RKLD)
  WRITE(6,6)T,D,RKL,RKLT,RKLD
2 CONTINUE
1 CONTINUE
STOP
6 FORMAT(1P5E15.5)
END
SUBROUTINE RKIRP2(T,D,RKL,RKLT,RKLD)
C OPACITY FOR X=0.700,Y=0.299,Z=0.001
C IGLESIAS AND ROGERS: AP. J. (LETTERS),371,L73-L75,20 APRIL 1991
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION RLA(9),T6A(49),T6LA(49),RKLA(9,49)
ALOG10(X)=DLOG10(X)
DATA T6A /
1 0.006, 0.007, 0.008, 0.009, 0.010, 0.012, 0.014, 0.016, 0.018,
2 0.020, 0.025, 0.030, 0.035, 0.040, 0.045, 0.050, 0.055, 0.060,
3 0.070, 0.080, 0.090, 0.100, 0.120, 0.150, 0.200, 0.250, 0.300,
4 0.400, 0.500, 0.600, 0.800, 1.000, 1.200, 1.500, 2.000, 2.500,
5 3.000, 4.000, 5.000, 6.000, 8.000, 10.00, 15.00, 20.00, 30.00,
6 40.00, 60.00, 80.00, 100.0 /
DATA RLA /
1-5.000,-4.500,-4.000,-3.500,-3.000,-2.500,-2.000,-1.500,-1.000 /
DATA RKLA /
1-1.585,-1.665,-1.822,-1.842,-1.810,-1.624,-1.465,-1.256,-1.043,
2-0.720,-0.760,-0.827,-0.823,-0.765,-0.665,-0.540,-0.374,-0.195,
3-0.220,-0.139,-0.030, 0.041, 0.102, 0.194, 0.291, 0.409, 0.546,
4-0.089, 0.178, 0.438, 0.636, 0.790, 0.898, 0.992, 1.090, 1.188,
5-0.097, 0.245, 0.588, 0.937, 1.204, 1.399, 1.533, 1.630, 1.716,
6-0.168, 0.155, 0.534, 0.976, 1.429, 1.819, 2.114, 2.287, 2.420,
7-0.188, 0.068, 0.413, 0.845, 1.331, 1.819, 2.227, 2.571, 2.817,
8-0.206, 0.042, 0.365, 0.763, 1.224, 1.742, 2.204, 2.654, 3.016,
9-0.223, 0.007, 0.325, 0.732, 1.187, 1.694, 2.165, 2.662, 3.106,
X-0.232,-0.020, 0.291, 0.702, 1.161, 1.656, 2.161, 2.665, 3.152,
1-0.208,-0.017, 0.280, 0.690, 1.183, 1.695, 2.224, 2.747, 3.268,
2-0.143, 0.043, 0.342, 0.761, 1.259, 1.773, 2.313, 2.855, 3.414,
3-0.053, 0.153, 0.456, 0.862, 1.354, 1.878, 2.419, 2.968, 3.535,
4-0.036, 0.244, 0.590, 1.001, 1.474, 1.985, 2.515, 3.062, 3.621,
5-0.065, 0.245, 0.635, 1.106, 1.580, 2.072, 2.585, 3.116, 3.658,
6-0.098, 0.187, 0.580, 1.070, 1.592, 2.105, 2.611, 3.128, 3.649,
7-0.131, 0.144, 0.517, 0.985, 1.512, 2.055, 2.585, 3.099, 3.606,
8-0.158, 0.104, 0.458, 0.901, 1.415, 1.957, 2.512, 3.044, 3.550,
9-0.191, 0.037, 0.363, 0.788, 1.288, 1.802, 2.358, 2.912, 3.439,
X-0.209,-0.003, 0.303, 0.713, 1.207, 1.718, 2.264, 2.817, 3.362,
1-0.204,-0.010, 0.287, 0.693, 1.178, 1.686, 2.226, 2.774, 3.323,
2-0.205,-0.004, 0.292, 0.686, 1.174, 1.687, 2.222, 2.765, 3.311,
3-0.193, 0.014, 0.315, 0.715, 1.196, 1.706, 2.231, 2.766, 3.296,
4-0.193, 0.021, 0.329, 0.733, 1.208, 1.699, 2.189, 2.679, 3.156,
5-0.220,-0.035, 0.241, 0.614, 1.047, 1.489, 1.918, 2.335, 2.749,
6-0.278,-0.117, 0.128, 0.463, 0.851, 1.251, 1.641, 2.024, 2.412,
7-0.329,-0.188, 0.032, 0.341, 0.695, 1.063, 1.426, 1.791, 2.169,
8-0.400,-0.311,-0.144, 0.135, 0.465, 0.801, 1.137, 1.486, 1.853,
9-0.424,-0.364,-0.246,-0.030, 0.282, 0.626, 0.963, 1.309, 1.659,

```

```

X-0.432,-0.386,-0.294,-0.119, 0.165, 0.507, 0.852, 1.192, 1.532,
1-0.442,-0.408,-0.338,-0.201, 0.038, 0.359, 0.701, 1.035, 1.369,
2-0.448,-0.420,-0.361,-0.245,-0.030, 0.280, 0.624, 0.958, 1.280,
3-0.445,-0.421,-0.370,-0.268,-0.071, 0.229, 0.578, 0.909, 1.209,
4-0.441,-0.418,-0.371,-0.277,-0.103, 0.183, 0.522, 0.832, 1.097,
5-0.449,-0.425,-0.375,-0.278,-0.111, 0.140, 0.427, 0.686, 0.918,
6-0.458,-0.438,-0.396,-0.310,-0.156, 0.068, 0.309, 0.535, 0.771,
7-0.463,-0.448,-0.416,-0.348,-0.215,-0.024, 0.188, 0.403, 0.655,
8-0.468,-0.459,-0.438,-0.391,-0.296,-0.152, 0.028, 0.239, 0.520,
9-0.471,-0.463,-0.447,-0.411,-0.339,-0.221,-0.061, 0.149, 0.429,
X-0.473,-0.467,-0.454,-0.423,-0.361,-0.260,-0.115, 0.093, 0.366,
1-0.475,-0.471,-0.461,-0.436,-0.384,-0.301,-0.173, 0.024, 0.285,
2-0.477,-0.473,-0.463,-0.443,-0.398,-0.327,-0.199,-0.021, 0.231,
3-0.483,-0.479,-0.472,-0.453,-0.416,-0.359,-0.251,-0.092, 0.129,
4-0.488,-0.485,-0.478,-0.465,-0.431,-0.378,-0.287,-0.146, 0.056,
5-0.498,-0.497,-0.493,-0.483,-0.455,-0.412,-0.338,-0.223,-0.050,
6-0.508,-0.507,-0.504,-0.496,-0.474,-0.438,-0.376,-0.280,-0.129,
7-0.526,-0.525,-0.522,-0.518,-0.503,-0.477,-0.436,-0.369,-0.246,
8-0.544,-0.545,-0.545,-0.538,-0.527,-0.511,-0.486,-0.440,-0.333,
9-0.563,-0.562,-0.561,-0.560,-0.551,-0.542,-0.531,-0.501,-0.401/
DATA KON/0/
IF(KON)1,1,2
1 DO 3 I=1,49
3 T6LA(I)=ALOG10(T6A(I))
KON=1
2 T6=1.0E-06*T
T6L=ALOG10(T6)
R=D/T6**3
RL=ALOG10(R)
IR1=1
DO 4 I=2,8
IF(RL-RLA(I))4,5,5
5 IR1=I
4 CONTINUE
IR2=IR1+1
IT1=1
DO 6 I=2,48
IF(T6L-T6LA(I))6,7,7
7 IT1=I
6 CONTINUE
IT2=IT1+1
DRL=RLA(IR2)-RLA(IR1)
DT6L=T6LA(IT2)-T6LA(IT1)
URL=(RL-RLA(IR1))/DRL
UTL=(T6L-T6LA(IT1))/DT6L
RKL11=RKLA(IR1,IT1)
RKL21=RKLA(IR2,IT1)
RKL12=RKLA(IR1,IT2)
RKL22=RKLA(IR2,IT2)
URLC=1.0-URL
UTLC=1.0-UTL
RKL=URLC*UTLC*RKL11+URL*UTLC*RKL21+URLC*UTL*RKL12+URL*UTL*RKL22
VZ=RKL11-RKL21-RKL12+RKL22
RKLr=(RKL21-RKL11+UTL*VZ)/DRL
RKLt6=(RKL12-RKL11+URL*VZ)/DT6L
RKLD=RKLr
RKLt=RKLt6-3.0*RKLr
C WRITE(6,16)IT1,IT2,IR1,IR2,RKL11,RKL21,RKL12,RKL22,URL,UTL
C WRITE(6,16)IT1,IT2,IR1,IR2,T,D,RKL,RKLt,RKLD
RETURN
16 FORMAT(4H5,1P6E10.3)
END
C PROGRAM TOPALS - TEST OPALS
SUBROUTINE TOPALS
IMPLICIT REAL*8 (A-H,O-Z)
DATA T/1.0E5/,D/1.0E-7/
DO 1 I=1,3
X=0.2*I
DO 2 J=1,3
Z=2.5E-4*10**(J-1)
Y=1.0-X-Z
CALL OPALS(X,Y,Z,T,D,RKL,RKLt,RKLD,RKLtT,RKLDD,RKLtD,RKLDT)
WRITE(6,6)X,Y,Z,T,D
WRITE(6,6)RKL,RKLt,RKLD,RKLtT,RKLDD,RKLtD,RKLDT
2 CONTINUE
1 CONTINUE
STOP

```

```

6  FORMAT(1P8E10.3)
   END
C  ACCESSING ROGERS-IGLESIAS OPAL TABLES
C  NB - SECOND DERIVATIVES TEMPORARILY COMMENTED OUT
SUBROUTINE OPALS(X,Y,Z,T,D,RKL,RKLT,RKLD,RKLT,TKLDD,RKLT,DKLDT)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION OPALK(17,50,10,4)
DIMENSION LINE(20),RLA(17),T6A(50),XA(4),ZA(10)
DIMENSION OP(4),OPR(4),OPT(4),OPRR(4),OPTT(4),OPRT(4),OPTR(4)
DATA NR/17/,NT/50/,NZ/10/,NX/3/
DATA EBTL/0.43429448/,TBEL/2.3025850/
DATA XC/-1.0/,YC/-1.0/,ZC/-1.0/
DATA FRACX/1.0E-06/,FRACZ/1.0E-06/
DATA KON/0/
IF(KON)1,1,2
1  CONTINUE
C  WRITE(6,*) OPALS: ACCESSING ROGERS-IGLESIAS TABLES IN OPALT.DAT'
WRITE(6,*) OPALS: ACCESSING ROGERS-IGLESIAS TABLES IN opalt.d'
OPEN (UNIT=1,FILE="opalt.d")
C  OPEN (UNIT=1,NAME='OPALT.DAT',TYPE='OLD')
DO 11 IX=1,NX
  IY=4-IX
  DO 12 IZ=1,NZ
    READ (1,(20A4))LINE
  C  WRITE(6,(20A4))LINE
  READ (1,*)XA(IY),ZA(IZ)
  C  WRITE(6,*)XA(IY),ZA(IZ)
  READ (1,(9X,17F7.1))(RLA(J),J=1,NR)
  C  WRITE(6,(9X,17F7.1))(RLA(J),J=1,NR)
  RLA(14)=-0.5
  RLA(15)= 0.0
  RLA(16)= 0.5
  RLA(17)= 1.5
  DO 13 IT=1,NT
    READ (1,(F9.3,17F7.3))T6A(IT),(OPALK(IR,IT,IZ,IY),IR=1,NR)
13  CONTINUE
12  CONTINUE
11  CONTINUE
    CLOSE(UNIT=1)
    KON=1
2  CONTINUE
  T6=1.0E-6*T
  TL=LOG10(T)
  DL=LOG10(D)
  RL=DL-3*(TL-6)
  NT6=50
  NRL=13
  DO 14 IT=1,NT6
    IT6=IT
  C  IF(T6-T6A(IT))15,15,14
  IF(T6-T6A(IT))15,14,14
14  CONTINUE
15  CONTINUE
    IT6=IT6-1
    IT6=MIN(MAX(2,IT6),NT6-1)
    DO 16 IR=1,NRL
      IRL=IR
      IF(RL-RLA(IR))17,16,16
16  CONTINUE
17  CONTINUE
      IRL=IRL-1
      IRL=MIN(MAX(2,IRL),NRL-1)
      IF(X-XC)124,125,124
124 CONTINUE
      MIX=2
      DO 24 IX=1,NX
        KX=IX
  C  IF(X-XA(IX))25,24,24
      DELX=X-XA(IX)
      IF(ABS(DELX)-FRACX)224,224,225
225 IF(DELX)25,224,24
224 MIX=1
      GO TO 125
24  CONTINUE
25  KX=KX-1
    KX=MIN(MAX(1,KX),NX-1)
    XC=X

```

```

C WRITE(6,*)XC,YC,ZC
125 CONTINUE
IF(Z-ZC)126,127,126
126 CONTINUE
MIZ=2
DO 26 IZ=1,NZ
KZ=IZ
C IF(Z-ZA(IZ))27,26,26
DELZ=Z-ZA(IZ)
IF(ABS(DELZ)-FRACZ)226,226,227
227 IF(DELZ)27,226,26
226 MIZ=1
GO TO 127
26 CONTINUE
27 KZ=KZ-1
KZ=MIN(MAX(1,KZ),NZ-1)
ZC=Z
C WRITE(6,*)XC,YC,ZC
127 CONTINUE
K=0
DO 31 IZ=1,2
ICZ=KZ+IZ-1
DO 32 IX=1,2
ICX=KX+IX-1
K=K+1
IF(IX-MIX)33,33,34
33 IF(IZ-MIZ)35,35,34
34 CONTINUE
OP(K)=0.0
OPR(K)=0.0
OPT(K)=0.0
GO TO 32
35 CONTINUE
CALL FLAG(RLA,T6A,OPALK,IRL,IT6,ICZ,ICX,RL,T6,
*F0,FX,FY,FXX,FYY,FXY,FYX)
C WRITE(6,6)IRL,IT6,T,D,T6,RL
C WRITE(6,6)ICX,ICZ,F0,FX,FY,FXX,FYY,FXY,FYX
6 FORMAT(2I5,1P7E10.3)
OP(K)=F0
OPR(K)=FX
OPT(K)=FY
C OPRR(K)=FXX
C OPTT(K)=FYY
C OPRT(K)=FXY
C OPTR(K)=FYX
32 CONTINUE
31 CONTINUE
DX=XA(KX+1)-XA(KX)
DZ=ZA(KZ+1)-ZA(KZ)
XF=(X-XA(KX))/DX
ZF=(Z-ZA(KZ))/DZ
XFC=1.0-XF
ZFC=1.0-ZF
F0=ZFC*(XFC*OP(1)+XF*OP(2))+ZF*(XFC*OP(3)+XF*OP(4))
FX=ZFC*(XFC*OPR(1)+XF*OPR(2))+ZF*(XFC*OPR(3)+XF*OPR(4))
FY=ZFC*(XFC*OPT(1)+XF*OPT(2))+ZF*(XFC*OPT(3)+XF*OPT(4))
C FXX=ZFC*(XFC*OPRR(1)+XF*OPRR(2))+ZF*(XFC*OPRR(3)+XF*OPRR(4))
C FYY=ZFC*(XFC*OPTT(1)+XF*OPTT(2))+ZF*(XFC*OPTT(3)+XF*OPTT(4))
C FXY=ZFC*(XFC*OPRT(1)+XF*OPRT(2))+ZF*(XFC*OPRT(3)+XF*OPRT(4))
C FYX=ZFC*(XFC*OPTR(1)+XF*OPTR(2))+ZF*(XFC*OPTR(3)+XF*OPTR(4))
RKL=F0
RK=10.0**RKL
RKLDD=FX
RKLTD=EBTL*T6*FY
RKLTT=RK*TL6-3.0*RKLDD
C RKLDD=EBTL*FXX
C RKLDT=T6*FXY-3.0*RKLDD
C RKLTD=T6*FYX-3.0*RKLDD
C RKLTT=T6*(TBEL*T6*FYY-6.0*FXY)+9.0*EBTL*FXX
RETURN
END
C TWO-DIMENSIONAL LAGRANGIAN INTERPOLATION AND DIFFERENTIATION
SUBROUTINE FLAG(XT,YT,FT,IX,IY,ICZ,ICX,X,Y,
*F0,FX,FY,FXX,FYY,FXY,FYX)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XT(17),YT(50),FT(17,50,10,4)
DIMENSION CX(3),CY(3),C0X(3),C1X(3),C0Y(3),C1Y(3)

```

```

DIMENSION F0X(3),F1X(3),F2X(3),F0Y(3),F1Y(3),F2Y(3)
DATA XC,0.0/,YC,0.0/
F(IX,IY)=FT(IX,IY,ICZ,ICX)
IF(X-XC)11,12,11
12 IF(Y-YC)11,10,11
11 CONTINUE
DO 1 I=1,3
CX(I)=1.0
CY(I)=1.0
C0X(I)=1.0
C0Y(I)=1.0
C1X(I)=2.0*X
C1Y(I)=2.0*Y
DO 2 J=1,3
IF(J-I)3,2,3
3 CONTINUE
CX(I)=CX(I)*(XT(IX+I-2)-XT(IX+J-2))
CY(I)=CY(I)*(YT(IY+I-2)-YT(IY+J-2))
C0X(I)=C0X(I)*(X-XT(IX+J-2))
C1X(I)=C1X(I)-XT(IX+J-2)
C0Y(I)=C0Y(I)*(Y-YT(IY+J-2))
C1Y(I)=C1Y(I)-YT(IY+J-2)
2 CONTINUE
C0X(I)=C0X(I)/CX(I)
C1X(I)=C1X(I)/CX(I)
C0Y(I)=C0Y(I)/CY(I)
C1Y(I)=C1Y(I)/CY(I)
1 CONTINUE
XC=X
YC=Y
C WRITE(6,*)XC,YC
10 CONTINUE
DO 4 I=1,3
F0X(I)=0.0
F1X(I)=0.0
F2X(I)=0.0
F0Y(I)=0.0
F1Y(I)=0.0
F2Y(I)=0.0
DO 5 J=1,3
F0X(I)=F0X(I)+C0Y(J)*F(IX+I-2,IY+J-2)
F1X(I)=F1X(I)+C1Y(J)*F(IX+I-2,IY+J-2)
C F2X(I)=F2X(I)+2/CY(J)*F(IX+I-2,IY+J-2)
F0Y(I)=F0Y(I)+C0X(J)*F(IX+J-2,IY+I-2)
F1Y(I)=F1Y(I)+C1X(J)*F(IX+J-2,IY+I-2)
C F2Y(I)=F2Y(I)+2/CX(J)*F(IX+J-2,IY+I-2)
5 CONTINUE
4 CONTINUE
FXY0=0.0
FXX0=0.0
FX=0.0
FY=0.0
FXX=0.0
FYY=0.0
FXY=0.0
FXX=0.0
DO 6 J=1,3
FXY0=FXY0+C0Y(I)*F0Y(I)
FXX0=FXX0+C0X(I)*F0X(I)
FX=FX+C0Y(I)*F1Y(I)
FY=FY+C0X(I)*F1X(I)
C FXY=FXY+C1X(I)*F1X(I)
C FXX=FXX+C1Y(I)*F1Y(I)
C FXX=FXX+C0Y(I)*F2Y(I)
C FYY=FYY+C0X(I)*F2X(I)
6 CONTINUE
F0=(FXY0+FXX0)/2.0
C WRITE(6,*)FXY,FXX0,F0
C WRITE(6,*)FX,FY
C WRITE(6,*)FXX,FYY
C WRITE(6,*)FXY,FXX
RETURN
END
C ENERGY
SUBROUTINE ENERGY(IE,X,Y,Z,T,D,E,DET,DED)
IMPLICIT REAL*8 (A-H,O-Z)
EXP(X)=DEXP(X)

```

```

ALOG(X)=DLOG(X)
SQRT(X)=DSQRT(X)
AMIN1(X1,X2)=DMIN1(X1,X2)
AMAX1(X1,X2)=DMAX1(X1,X2)
SUML(X1,X2)=AMAX1(X1,X2)+ALOG(1+EXP(AMIN1(X1,X2)-AMAX1(X1,X2)))
DATA RCA/7.56495E-15/,AVN/6.02252E23/,BCK/1.38054E-16/
DATA EVE6/1.60210E-06/,AMU/931.4812/,VLC/2.997925E10/
DATA QPP/13.811/,QNPO/24.970/,QHEC/7.275/,QHEO/14.436/
DATA W1/1.0/,W4/4.0/,W14/14.0/
X1=X
X4=Y
IF(IE-2)1,2,2
1 CONTINUE
T6=1.0E-6*T
T9=1.0E-3*T6
T913=T9**(1.0/3.0)
T923=T913**2
C RPP=4.210E-15/T923*EXP(-3.380/T913)
RPPL=ALOG(4.210E-15/T923)-3.380/T913
FPP=1+0.123*T913+1.090*T923+0.938*T9
SPP=1+0.250*SQRT(D/T6)
EPP=AVN*(X1/W1)**2*D*RPP*FPP*SPP/2*QPP*EVE6
IF(EPP)11,11,12
11 EPPL=-99.0
GO TO 13
C12 EPPL=ALOG(AVN*(X1/W1)**2*D*FPP*SPP/2*QPP*EVE6)+RPPL
12 EPPL=ALOG(EPP)+RPPL
13 CONTINUE
C RNPO=5.080E7/T923*EXP(-15.228/T913-(T9/3.090)**2)
RNPOL=ALOG(5.080E7/T923)-15.228/T913-(T9/3.090)**2
FNPO=1+0.027*T913-0.778*T923-0.149*T9+T923*(0.261*T923+0.127*T9)
RNPOL1=RNPOL+ALOG(FNPO)
C +2.280E3/T9/SQRT(T9)*EXP(-3.011/T9)+1.650E4*T913*EXP(-12.007/T9)
RNPOL2=ALOG(2.280E3/T9/SQRT(T9))-3.011/T9
RNPOL3=ALOG(1.654E4*T913)-12.007/T9
C ELMIN=AMIN1(RNPOL2,RNPOL3)
C ELMAX=AMAX1(RNPOL2,RNPOL3)
C RNPOL2=ELMAX+ALOG(1+EXP(ELMIN-ELMAX))
C ELMIN=AMIN1(RNPOL1,RNPOL2)
C ELMAX=AMAX1(RNPOL1,RNPOL2)
C RNPOL=ELMAX+ALOG(1+EXP(ELMIN-ELMAX))
RNPOL=SUML(RNPOL1,SUML(RNPOL2,RNPOL3))
SNPO=1+1.750*SQRT(D/T6)
X14=Z/2
C ENPO=AVN*(X1*X14/W1/W14)*D*RNPO*FNPO*SNPO*QNPO*EVE6
ENPO=AVN*(X1*X14/W1/W14)*D *SNPO*QNPO*EVE6
IF(ENPO)21,21,22
21 ENPOL=-99.0
GO TO 23
C22 ENPOL=ALOG(AVN*(X1*X14/W1/W14)*D *SNPO*QNPO*EVE6)+RNPOL
22 ENPOL=ALOG(ENPO)+RNPOL
23 CONTINUE
E=EPP+ENPO
C ELMIN=AMIN1(EPPL,ENPOL)
C ELMAX=AMAX1(EPPL,ENPOL)
C E=ELMAX+ALOG(1+EXP(ELMIN-ELMAX))
EL=SUML(EPPL,ENPOL)
E=EXP(EL)
GO TO 5
2 CONTINUE
C NEEDS FURTHER DEFINITION
E=0.0
GO TO 5
5 RETURN
END
C COMPUTE NUCLEAR ENERGY RATES
C THERMONUCLEAR REACTION RATES FROM FOWLER, CAUGHLAN, ZIMMERMAN & HARRIS
C FCZ I : 1967, ANN. REV. ASTRON. AP. 5:525
C FCZ II : 1975, ANN. REV. ASTRON. AP. 13:69
C HFCZ III : 1983, ANN. REV. ASTRON. AP. 21:165
C THIS REVISION ON 31 JANUARY 1995
SUBROUTINE ENUC(TV,DV,X,Y,Z,EPPC,ECNO,EHEC,EV,DET,DED)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ZAC(8),EAC(8)
DATA AN/6.0222E+23/,ANL/54.754909/,EV6/1.6022E-06/
DATA AMH/1.007825/,AMHE/4.002603/,AMZ/14.0/
DATA AMC/12.0/,AMO/16.0/,AMNE/20.0/,AMMG/24.0/,AMSI/28.0/

```

```

DATA QPPC/13.811/,QCNO/24.970/,QHEC/14.436/
DATA QCAO/7.162/,QOANE/4.730/,QNEMG/9.315/,QMGS1/9.986/
DATA KON/0/
EXP(X)=DEXP(X)
ALOG(X)=DLOG(X)
IF(KON)100,100,101
100 KON=1
WRITE(6,*) ENUC : NO HEAVY ELEMENT BURNING'
101 CONTINUE
T=EXP(TV)
D=EXP(DV)
T9=T*1.0E-09
T913=EXP(ALOG(T9)/3)
T923=T913**2
T943=T9*T913
T953=T9*T923
T912=SQRT(T9)
T932=T912**3
T976=T912*T923
C H1(P,E+NU)H2
F11=3.820E-15/T923*EXP(ANL-3.380/T913)
F12=(1.000+0.123*T913+1.090*T923+0.938*T9)
F1=F11*F12
D11=(3.380/T913-2.0)/3.0/T9
D12=(0.123/T923+2*1.090/T913)/3+0.938
D1=F11*D12+F1*D11
CPPC=D*(X/AMH)**2/2.0*QPPC*EV6
EPPC=CPPC*F1
C N14(P,G)O15 (II)
F21=5.080E+07/T923*EXP(ANL-15.228/T913-(T9/3.090)**2)
F22=(1.000+0.027*T913-0.778*T923-0.149*T9+0.261*T943+0.127*T953)
F23=2.280E+03/T932*EXP(ANL-3.011/T9)
F24=1.650E+04*T913*EXP(ANL-12.007/T9)
F2=F21*F22+F23+F24
D21=F21*((15.228/T913-2.0)/3.0-2.0*(T9/3.090)**2)/T9
D22=(0.027/T923-1.556/T913-0.447+1.044*T913+0.635*T923)/3.0
D23=F23*(3.011/T9-1.5)/T9
D24=F24*(12.007/T9+1.0/3.0)/T9
D2=F21*D22+F22*D21+D23+D24
CCNO=D*(X/AMH)*(Z/AMZ)*QCNO*EV6
ECNO=CCNO*F2
C HE4(2A,G)C12 (II,III)
C CHECK Q-VALUE
F31=2.790E-08/T9**3*EXP(ANL-4.4027/T9)
F32=1.350E-07/T932*EXP(ANL-24.811/T9)
F3=F31+F32
D3=F31*(4.4027/T9-3.0)/T9+F32*(24.811/T9-1.5)/T9
CHEC=D*D*(Y/AMHE)**3/6.0*QHEC*EV6
EHEC=HEC*F3
C C12(A,G)O16 (II,III)
F410=1.580E+08/T9**2*EXP(ANL-32.120/T913-(T9/5.863)**2)
F411=(1.000+0.621*T923)
F412=(1.000+0.047/T923)
F413=(F411/F412)**2
F41=F410*F413
F42=2.740E+07/T923*EXP(ANL-32.120/T913)
F43=1.250E+03/T923*EXP(ANL-27.499/T9)
F44=1.430E-02*T9**5*EXP(ANL-15.541/T9)
F4=F41+F42+F43+F44
D410=F410*(32.120/T913/3.0-2.0-2.0*(T9/5.863)**2)/T9
D411=4.0/3.0*0.621/T913*F411
D412=-4.0/3.0*0.047/T953*F412
D413=(F412**2*D411-F411**2*D412)/F412**4
D41=F410*D413+D410*F413
D42=F42*(32.120/T913-2.0)/3.0/T9
D43=F43*(27.499/T9-2.0/3.0)/T9
D44=F44*(15.541/T9+5.0)/T9
D4=D41+D42+D43+D44
CCAO=D*(Y/AMHE)*(ZC/AMC)*QCAO*EV6
ECAO=CCAO*F4
EAC(1)=ECAO
C O16(A,G)NE20 (II)
F51=5.490E+09/T923*EXP(ANL-39.756/T913)
F52=4.090E+01/T932*EXP(ANL-10.359/T9)
F53=3.920E+02/T932*EXP(ANL-12.243/T9)
F54=8.050E+00*T9**2*EXP(ANL-20.093/T9)
F5=F51+F52+F53+F54

```

```

D51=F51*(39.756/T913-2.0)/T9/3.0
D52=F52*(10.359/T9-1.5)/T9
D53=F53*(12.243/T9-1.5)/T9
D54=F54*(20.093/T9+2.0)/T9
D5=D51+D52+D53+D54
COANE=D*(Y/AMHE)*(ZCO/AMO)
COANE=D*(Y/AMHE)*(ZCO/AMO)*QOANE*EV6
EOANE=COANE*F5
EAC(2)=EOANE
C  NE20(A,G)MG24 (II)
  THETA6=0.1
  F61=4.110E+11/T923*EXP(ANL-46.766/T913-(T9/2.219)**2)
  F62=(1.000+0.009*T913+0.882*T923+0.055*T9+0.749*T943+0.119*T953)
  F63=5.270E+03/T932*EXP(ANL-15.869/T9)
  F64=6.510E+03*T912*EXP(ANL-16.223/T9)
  F65=4.210E+01/T932*EXP(ANL-9.115/T9)
  F66=3.200E+01/T923*EXP(ANL-9.383/T9)
  F60=1.000+5.000*EXP(-18.960/T9)
  F6=(F61*F62+F63+F64+THETA6*(F65+F66))/F60
  D61=F61*(46.766/T913-2.0)/3.0-2.0*(T9/2.219)**2/T9
  D62=(0.009/T923+1.764/T913+0.165+2.996*T913+0.595*T923)/3.0
  D63=F63*(15.869/T9-1.5)/T9
  D64=F64*(16.223/T9+0.5)/T9
  D65=F65*(9.115/T9-1.5)/T9
  D66=F66*(9.383/T9-2.0/3.0)/T9
C  D60=18.960/T9**2*(F60-1.0)
  D60=18.960/T9**2*5.000*EXP(-18.960/T9)
  D6=(F62*D61+F61*D62+D63+D64+THETA6*(D65+D66))/F60-F6*D60/F60
  CNEMG=D*(Y/AMHE)*(ZCNE/AMNE)*QNEMG*EV6
  ENEMG=CNEMG*F6
  EAC(3)=ENEMG
C  MG24(A,G)SI28 (II,III)
  THETA7=0.1
  F71=4.780E+01/T932*EXP(ANL-13.506/T9)
  F72=2.380E+03/T932*EXP(ANL-15.218/T9)
  F73=2.470E+02*T932*EXP(ANL-15.147/T9)
  F74=1.720E-09/T932*EXP(ANL-5.028/T9)
  F75=1.250E-03/T932*EXP(ANL-7.929/T9)
  F76=2.430E+01/T9 *EXP(ANL-11.523/T9)
  F70=1.000+5.000*EXP(-15.882/T9)
  F7=(F71+F72+F73+THETA7*(F74+F75+F76))/F70
  D71=F71*(13.506/T9-1.5)/T9
  D72=F72*(15.218/T9-1.5)/T9
  D73=F73*(15.147/T9+1.5)/T9
  D74=F74*(5.028/T9-1.5)/T9
  D75=F75*(7.929/T9-1.5)/T9
  D76=F76*(11.523/T9-1.0)/T9
C  D70=15.882/T9**2*(F70-1.0)
  D70=15.882/T9**2*5.000*EXP(-15.882/T9)
  D7=(D71+D72+D73+THETA7*(D74+D75+D76))/F70-F7*D70/F70
  CMGSI=D*(Y/AMHE)*(ZCMG/AMMG)*QMGSI*EV6
  EMGSI=CMGSI*F7
  EAC(4)=EMGSI
  EAC(5)=0.0
  EAC(6)=0.0
  EAC(7)=0.0
  EAC(8)=0.0
  SEN=EPPC+ECNO+EHEC+ECAO+EOANE+ENEMG+EMGSI
  EN=SEN
  DPPCT=0.0
  IF(F1)1,1,10
10 DPPCT=T9*D1/F1
1  CONTINUE
  DCNOT=0.0
  IF(F2)2,2,20
20 DCNOT=T9*D2/F2
2  CONTINUE
  DHECT=0.0
  IF(F3)3,3,30
30 DHECT=T9*D3/F3
3  CONTINUE
  IF(EN)80,80,85
80 EV=-99
  DET=0.0
  DED=1.0
  GO TO 90
85 CONTINUE

```

```

DET=CPPC*D1+CCNO*D2+CHEC*D3+CCAO*D4+COANE*D5+CNEMG*D6+CMGSI*D7
DET=T9*DET/EN
DED=(EPPC+ECNO+2.0*EHEC+ECAO+EOANE+ENEMG+EMGSI)/EN
EV=ALOG(EN)
90 CONTINUE
RETURN
END
C COMPUTE NUCLEAR ENERGY RATES
C INCLUDING HEAVY ELEMENT C,O,NE,MG ... ALPHA REACTIONS
C THERMONUCLEAR REACTION RATES FROM FOWLER, CAUGHLAN, ZIMMERMAN & HARRIS
C FCZ I : 1967, ANN. REV. ASTRON. AP. 5:525
C FCZ II : 1975, ANN. REV. ASTRON. AP. 13:69
C HFCZ III : 1983, ANN. REV. ASTRON. AP. 21:165
C THIS REVISION ON 31 JANUARY 1995
C SUBROUTINE ENUC(TV,DV,X,Y,Z,EPPC,ECNO,EHEC,EV,DET,DED)
SUBROUTINE ENUCZ(TV,DV,X,Y,Z,ZAC,EPPC,ECNO,EHEC,EAC,EV,DET,DED)
C SUBROUTINE ENUCZT(TV,DV,X,Y,Z,ZAC,EPPC,ECNO,EHEC,EAC,EV,DET,DED),
C 1DPPCT,DCNOT,DHECT,DETA)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ZAC(8),EAC(8),DETA(8)
DATA AN/6.0222E+23/,ANL/54.754909/,EV6/1.6022E-06/
DATA AMH/1.007825/,AMHE/4.002603/,AMZ/14.0/
DATA AMC/12.0/,AMO/16.0/,AMNE/20.0/,AMMG/24.0/,AMSI/28.0/
DATA AMS/32.0/,AMAR/36.0/,AMCA/40.0/
DATA QPPC/13.811/,QCNO/24.970/,QHEC/7.274/
DATA QCAO/7.162/,QOANE/4.730/,QNEMG/9.315/,QMGSI/9.986/
DATA QSIAS/0.0/,QSAAR/0.0/,QARCA/0.0/
DATA QCCMG/13.931/,QCOSI/16.754/,QOOS/16.541/
DATA KON/0/
EXP(X)=DEXP(X)
ALOG(X)=DLOG(X)
IF(KON)100,100,101
100 KON=1
WRITE(6,*)' ENUCZ : HEAVY ELEMENT BURNING'
101 CONTINUE
ZCC=ZAC(1)
ZCO=ZAC(2)
ZCNE=ZAC(3)
ZCMG=ZAC(4)
ZCSI=ZAC(5)
ZCS=ZAC(6)
ZCAR=ZAC(7)
ZCCA=ZAC(8)
T=EXP(TV)
D=EXP(DV)
T9=T*1.0E-09
T913=EXP(ALOG(T9)/3)
T923=T913**2
T943=T9*T913
T953=T9*T923
T912=SQRT(T9)
T932=T912**3
T976=T912*T923
C H1(P,E+NU)H2
F11=3.820E-15/T923*EXP(ANL-3.380/T913)
F12=(1.000+0.123*T913+1.090*T923+0.938*T9)
F1=F11*F12
D11=(3.380/T913-2.0)/3.0/T9
D12=(0.123/T923+2*1.090/T913)/3+0.938
D1=F11*D12+F1*D11
CPPC=D*(X/AMH)**2/2.0*QPPC*EV6
EPPC=CPPC*F1
C N14(P,C)015 (II)
F21=5.080E+07/T923*EXP(ANL-15.228/T913-(T9/3.090)**2)
F22=(1.000+0.027*T913-0.778*T923-0.149*T9+0.261*T943+0.127*T953)
F23=2.280E+03/T932*EXP(ANL-3.011/T9)
F24=1.650E+04*T913*EXP(ANL-12.007/T9)
F2=F21*F22+F23+F24
D21=F21*((15.228/T913-2.0)/3.0-2.0*(T9/3.090)**2)/T9
D22=(0.027/T923-1.556/T913-0.447+1.044*T913+0.635*T923)/3.0
D23=F23*(3.011/T9-1.5)/T9
D24=F24*(12.007/T9+1.0/3.0)/T9
D2=F21*D22+F22*D21+D23+D24
ZAMZ=ZCC/AMC+ZCO/AMO
CCNO=D*(X/AMH)*ZAMZ*QCNO*EV6
C CCNO=D*(X/AMH)*(Z/AMZ)*QCNO*EV6
ECNO=CCNO*F2

```

- C HE4(2A,G)C12 (II,III)
 C CHECK Q-VALUE
 $F31=2.790E-08/T9^{**3}*EXP(ANL-4.4027/T9)$
 $F32=1.350E-07/T932*EXP(ANL-24.811/T9)$
 $F3=F31+F32$
 $D3=F31*(4.4027/T9-3.0)/T9+F32*(24.811/T9-1.5)/T9$
 $CHEC=D*D*(Y/AMHE)^{**3}/6.0*QHEC*EV6$
 $EHEC=CHEC*F3$
- C C12(A,G)O16 (II,III)
 $F410=1.580E+08/T9^{**2}*EXP(ANL-32.120/T913-(T9/5.863)^{**2})$
 $F411=(1.000+0.621*T923)$
 $F412=(1.000+0.047/T923)$
 $F413=(F411/F412)^{**2}$
 $F41=F410*F413$
 $F42=2.740E+07/T923*EXP(ANL-32.120/T913)$
 $F43=1.250E+03/T923*EXP(ANL-27.499/T9)$
 $F44=1.430E-02*T9^{**5}*EXP(ANL-15.541/T9)$
 $F4=F41+F42+F43+F44$
 $D410=F410*(32.120/T913/3.0-2.0-2.0*(T9/5.863)^{**2})/T9$
 $D411=4.0/3.0*0.621/T913*F411$
 $D412=-4.0/3.0*0.047/T953*F412$
 $D413=(F412^{**2}*D411-F411^{**2}*D412)/F412^{**4}$
 $D41=F410*D413+D410*F413$
 $D42=F42*(32.120/T913-2.0)/3.0/T9$
 $D43=F43*(27.499/T9-2.0/3.0)/T9$
 $D44=F44*(15.541/T9+5.0)/T9$
 $D4=D41+D42+D43+D44$
 $CCAO=D*(Y/AMHE)*(ZCC/AMC)*QCAO*EV6$
 $ECAO=CCAO*F4$
 $EAC(1)=ECAO$
- C O16(A,G)NE20 (II)
 $F51=5.490E+09/T923*EXP(ANL-39.756/T913)$
 $F52=4.090E+01/T932*EXP(ANL-10.359/T9)$
 $F53=3.920E+02/T932*EXP(ANL-12.243/T9)$
 $F54=8.050E+00*T9^{**2}*EXP(ANL-20.093/T9)$
 $F5=F51+F52+F53+F54$
 $D51=F51*(39.756/T913-2.0)/T9/3.0$
 $D52=F52*(10.359/T9-1.5)/T9$
 $D53=F53*(12.243/T9-1.5)/T9$
 $D54=F54*(20.093/T9+2.0)/T9$
 $D5=D51+D52+D53+D54$
 $COANE=D*(Y/AMHE)*(ZCO/AMO)*QOANE*EV6$
 $EOANE=COANE*F5$
 $EAC(2)=EOANE$
- C NE20(A,G)MG24 (II)
 $THETA6=0.1$
 $F61=4.110E+11/T923*EXP(ANL-46.766/T913-(T9/2.219)^{**2})$
 $F62=(1.000+0.009*T913+0.882*T923+0.055*T9+0.749*T943+0.119*T953)$
 $F63=5.270E+03/T932*EXP(ANL-15.869/T9)$
 $F64=6.510E+03*T912*EXP(ANL-16.223/T9)$
 $F65=4.210E+01/T932*EXP(ANL-9.115/T9)$
 $F66=3.200E+01/T923*EXP(ANL-9.383/T9)$
 $F60=1.000+5.000*EXP(-18.960/T9)$
 $F6=(F61*F62+F63+F64+THETA6*(F65+F66))/F60$
 $D61=F61*((46.766/T913-2.0)/3.0-2.0*(T9/2.219)^{**2})/T9$
 $D62=(0.009/T923+1.764/T913+0.165+2.996*T913+0.595*T923)/3.0$
 $D63=F63*(15.869/T9-1.5)/T9$
 $D64=F64*(16.223/T9+0.5)/T9$
 $D65=F65*(9.115/T9-1.5)/T9$
 $D66=F66*(9.383/T9-2.0/3.0)/T9$
- C D60=18.960/T9^{**2}*(F60-1.0)
 $D60=18.960/T9^{**2}*5.000*EXP(-18.960/T9)$
 $D6=(F62*D61+F61*D62+D63+D64+THETA6*(D65+D66))/F60-F6*D60/F60$
 $CNEMG=D*(Y/AMHE)*(ZCNE/AMNE)*QNEMG*EV6$
 $ENEMG=CNEMG*F6$
 $EAC(3)=ENEMG$
- C MG24(A,G)SI28 (II,III)
 $THETA7=0.1$
 $F71=4.780E+01/T932*EXP(ANL-13.506/T9)$
 $F72=2.380E+03/T932*EXP(ANL-15.218/T9)$
 $F73=2.470E+02*T932*EXP(ANL-15.147/T9)$
 $F74=1.720E-09/T932*EXP(ANL-5.028/T9)$
 $F75=1.250E-03/T932*EXP(ANL-7.929/T9)$
 $F76=2.430E+01/T9*EXP(ANL-11.523/T9)$
 $F70=1.000+5.000*EXP(-15.882/T9)$
 $F7=(F71+F72+F73+THETA7*(F74+F75+F76))/F70$
 $D71=F71*(13.506/T9-1.5)/T9$

$D72 = F72 * (15.218 / T9 - 1.5) / T9$
 $D73 = F73 * (15.147 / T9 + 1.5) / T9$
 $D74 = F74 * (5.028 / T9 - 1.5) / T9$
 $D75 = F75 * (7.929 / T9 - 1.5) / T9$
 $D76 = F76 * (11.523 / T9 - 1.0) / T9$
 C $D70 = 15.882 / T9 ** 2 * (F70 - 1.0)$
 $D70 = 15.882 / T9 ** 2 * 5.000 * EXP(-15.882 / T9)$
 $D7 = (D71 + D72 + D73 + THETA7 * (D74 + D75 + D76)) / F70 - F7 * D70 / F70$
 $CMGSI = D * (Y / AMHE) * (ZCMG / AMMG) * QMGSI * EV6$
 $EMGSI = CMGSI * F7$
 $EAC(4) = EMGSI$
 $F8 = 0.0$
 $D8 = 0.0$
 $CSIAS = D * (Y / AMHE) * (ZCSI / AMSI) * QSIAS * EV6$
 $ESIAS = CSIAS * F8$
 $EAC(5) = ESIAS$
 $ESAR = 0.0$
 $EARCA = 0.0$
 C $C + C, C + O, O + O$ RATES FROM FCZ II pp85,103
 C $C12 + C12 = MG24$
 $T9A = T9 / (1.000 + 0.067 * T9)$
 $T9A13 = T9A ** (1.0 / 3.0)$
 $T9A23 = T9A13 ** 2$
 $T9A56 = T9A13 * SQRT(T9A)$
 $DT9A = (T9A / T9) ** 2$
 $FCC1 = 1.260E + 27 * T9A56 / T932 * EXP(ANL - 84.165 / T9A13)$
 $FCC2 = EXP(-0.010 * T9A ** 4) + 5.560E - 03 * EXP(1.685 * T9A23)$
 $FCC = FCC1 / FCC2$
 $DCC1 = FCC1 * ((84.165 / T9A13 + 2.5) / T9A * DT9A / 3.0 - 1.5 / T9)$
 $DCC2 = (-0.010 * 4.0 * T9A ** 3 * EXP(-0.010 * T9A ** 4) + 5.560E - 03 * 1.685 * 2.0 / 3.0 / T9A13 * EXP(1.685 * T9A23)) * DT9A$
 $DCC = (FCC2 * DCC1 - FCC1 * DCC2) / FCC2 ** 2$
 $CCCMG = D / 2.0 * (ZCC / AMC) ** 2 * QCCMG * EV6$
 $ECCMG = CCCMG * FCC$
 $EAC(6) = ECCMG$
 C $C + O = SI$
 $T9A = T9 / (1.000 + 0.055 * T9)$
 $T9A13 = T9A ** (1.0 / 3.0)$
 $T9A23 = T9A13 ** 2$
 $T9A56 = T9A13 * SQRT(T9A)$
 $DT9A = (T9A / T9) ** 2$
 $FCO1 = 1.720E + 31 * T9A56 / T932 * EXP(ANL - 106.594 / T9A13)$
 $FCO2 = EXP(-0.180 * T9A ** 2) + 1.060E - 03 * EXP(2.562 * T9A23)$
 $DCO1 = FCO1 * ((106.594 / T9A13 + 2.5) / T9A * DT9A / 3.0 - 1.5 / T9)$
 $DCO2 = (-0.180 * 2.0 * T9A * EXP(-0.180 * T9A ** 2) + 1.060E - 03 * 2.562 * 2.0 / 3.0 / T9A13 * EXP(2.562 * T9A23)) * DT9A$
 $FCO = FCO1 / FCO2$
 $DCO = (FCO2 * DCO1 - FCO1 * DCO2) / FCO2 ** 2$
 $CCOSI = D * (ZCC / AMC) * (ZCO / AMO) * QCOSI * EV6$
 $ECOSI = CCOSI * FCO$
 $EAC(7) = ECOSI$
 C $O + O = S$
 $T9A = T9 / (1.000 + 0.067 * T9)$
 $T9A13 = T9A ** (1.0 / 3.0)$
 $T9A23 = T9A13 ** 2$
 $T9A56 = T9A13 * SQRT(T9A)$
 $DT9A = (T9A / T9) ** 2$
 $FOO1 = 3.610E + 37 * T9A56 / T932 * EXP(ANL - 135.930 / T9A13)$
 $FOO2 = EXP(-0.032 * T9A ** 2) + 3.890E - 04 * EXP(2.659 * T9A23)$
 $DOO1 = FOO1 * ((135.930 / T9A13 + 2.5) / T9A * DT9A / 3.0 - 1.5 / T9)$
 $DOO2 = (-0.032 * 2.0 * T9A * EXP(-0.032 * T9A ** 2) + 3.890E - 04 * 2.659 * 2.0 / 3.0 / T9A13 * EXP(2.659 * T9A23)) * DT9A$
 $FOO = FOO1 / FOO2$
 $DOO = (FOO2 * DOO1 - FOO1 * DOO2) / FOO2 ** 2$
 $COOS = D / 2.0 * (ZCO / AMO) ** 2 * QOOS * EV6$
 $EOOS = COOS * FOO$
 $EAC(8) = EOOS$
 C $SEN = EPPC + ECNO + EHEC + ECAO + EOANE + ENEMG + EMGSI$
 C $SEN = CPPC * F1 + CCNO * F2 + CHEC * F3 + CCAO * F4 + COANE * F5 + CNEMG * F6 + CMGSI * F7$
 $SEN = EPPC + ECNO + EHEC + ECAO + EOANE + ENEMG + EMGSI + ESIAS + ESAAR + EARCA$
 $SEN = SEN + ECCMG + ECOSI + EOOS$
 $SET = CPPC * D1 + CCNO * D2 + CHEC * D3 + CCAO * D4 + COANE * D5 + CNEMG * D6 + CMGSI * D7$
 $SET = SET + CSIAS * D8 + CCCMG * DCC + CCOSI * DCO + COOS * DOO$
 C $SED = CPPC * F1 + CCNO * F2 + 2.0 * CHEC * F3 + CCAO * F4 + COANE * F5 + CNEMG * F6 + CMGSI * F7$
 C $SED = SED + CSIAS * F8 + CCCMG * FCC + CCOSI * FCO + COOS * FOO$
 $SED = EPPC + ECNO + 2 * EHEC + ECAO + EOANE + ENEMG + EMGSI + ESIAS + ESAAR + EARCA$
 $SED = SED + ECCMG + ECOSI + EOOS$

```

SED=SED/D
TEN=0.0
DEN=0.0
SENL=-100.0
IF(SEN)200,200,201
201 CONTINUE
TEN=T9*SET/SEN
DEN=D*SED/SEN
SENL=ALOG(SEN)
200 CONTINUE
EN=SEN
EV=SENL
DET=TEN
DED=DEN
DPPCT=0.0
IF(F1)1,1,10
10 DPPCT=T9*D1/F1
1 CONTINUE
DCNOT=0.0
IF(F2)2,2,20
20 DCNOT=T9*D2/F2
2 CONTINUE
DHECT=0.0
IF(F3)3,3,30
30 DHECT=T9*D3/F3
3 CONTINUE
DCAOT=0.0
IF(F4)4,4,40
40 DCAOT=T9*D4/F4
4 CONTINUE
DETA(1)=DCAOT
DOANET=0.0
IF(F5)5,5,50
50 DOANET=T9*D5/F5
5 CONTINUE
DETA(2)=DOANET
DNEMGT=0.0
IF(F6)6,6,60
60 DNEMGT=T9*D6/F6
6 CONTINUE
DETA(3)=DNEMGT
DMGSIT=0.0
IF(F7)7,7,70
70 DMGSIT=T9*D7/F7
7 CONTINUE
DETA(4)=DMGSIT
DSIAST=0.0
IF(F8)8,8,80
80 DSIAST=T9*D8/F8
8 CONTINUE
DETA(5)=DSIAST
DCCMGT=0.0
IF(FCC)190,190,191
191 DCCMGT=T9*DCC/FCC
190 CONTINUE
DETA(6)=DCCMGT
DCOSIT=0.0
IF(FCO)290,290,291
291 DCOSIT=T9*DCO/FCO
290 CONTINUE
DETA(7)=DCOSIT
DOOST=0.0
IF(FOO)390,390,391
391 DOOST=T9*DOO/FOO
390 CONTINUE
DETA(8)=DOOST
RETURN
END
C COMPUTE NUCLEAR ENERGY RATES
C INCLUDING HEAVY ELEMENT C,O,NE,MG ... ALPHA REACTIONS
C AND ALSO C+C=MG, C+O=SI, O+O=S REACTIONS
C THERMONUCLEAR REACTION RATES FROM FOWLER, CAUGHLAN, ZIMMERMAN & HARRIS
C FCZ I : 1967, ANN. REV. ASTRON. AP. 5:525
C FCZ II : 1975, ANN. REV. ASTRON. AP. 13:69
C HFCZ III : 1983, ANN. REV. ASTRON. AP. 21:165
C THIS REVISION ON 31 JANUARY 1995
C SUBROUTINE ENUC(TV,DV,X,Y,Z,EPPC,ECNO,EHEC,EV,DET,DED)

```

```

C  SUBROUTINE ENUCZ(TV,DV,X,Y,Z,ZAC,EPPC,ECNO,EHEC,EAC,EV,DET,DED)
SUBROUTINE ENUCZT(TV,DV,X,Y,Z,ZAC,EPPC,ECNO,EHEC,EAC,EV,DET,DED)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ZAC(8),EAC(8),DETA(8)
DATA AN/6.0222E+23/,ANL/54.754909/,EV6/1.6022E-06/
DATA AMH/1.007825/,AMHE/4.002603/,AMZ/14.0/
DATA AMC/12.0/,AMO/16.0/,AMNE/20.0/,AMMG/24.0/,AMSI/28.0/
DATA AMS/32.0/,AMAR/36.0/,AMCA/40.0/
DATA QPPC/13.811/,QCNO/24.970/,QHEC/7.274/
DATA QCAO/7.162/,QOANE/4.730/,QNEMG/9.315/,QMGS1/9.986/
DATA QSIAS/0.0/,QSAAR/0.0/,QARCA/0.0/
DATA QCCMG/13.931/,QCOSI/16.754/,QOOS/16.541/
DATA KON/0/
EXP(X)=DEXP(X)
ALOG(X)=DLOG(X)
IF(KON)100,100,101
100 KON=1
WRITE(6,*) ENUCZT : HEAVY ELEMENT BURNING'
101 CONTINUE
ZCC=ZAC(1)
ZCO=ZAC(2)
ZCNE=ZAC(3)
ZCMG=ZAC(4)
ZCSI=ZAC(5)
ZCS=ZAC(6)
ZCAR=ZAC(7)
ZCCA=ZAC(8)

C  VERSION II
C  LINES TO ENSURE THAT N14=0.1Z
C  ZCN=Z
C  DO 2234 K=1,8
C  ZCN=ZCN-ZAC(K)
C2234 CONTINUE

T=EXP(TV)
D=EXP(DV)
T9=T*1.0E-09
T913=EXP(ALOG(T9)/3)
T923=T913**2
T943=T9*T913
T953=T9*T923
T912=SQRT(T9)
T932=T912**3
T976=T912*T923
C  H1(P,E+NU)H2
F11=3.820E-15/T923*EXP(ANL-3.380/T913)
F12=(1.000+0.123*T913+1.090*T923+0.938*T9)
F1=F11*F12
D11=(3.380/T913-2.0)/3.0/T9
D12=(0.123/T923+2*1.090/T913)/3+0.938
D1=F11*D12+F1*D11
CPPC=D*(X/AMH)**2/2.0*QPPC*EV6
EPPC=CPPC*F1
C  N14(P,G)015 (II)
F21=5.080E+07/T923*EXP(ANL-15.228/T913-(T9/3.090)**2)
F22=(1.000+0.027*T913-0.778*T923-0.149*T9+0.261*T943+0.127*T953)
F23=2.280E+03/T932*EXP(ANL-3.011/T9)
F24=1.650E+04*T913*EXP(ANL-12.007/T9)
F2=F21*F22+F23+F24
D21=F21*((15.228/T913-2.0)/3.0-2.0*(T9/3.090)**2)/T9
D22=(0.027/T923-1.556/T913-0.447+1.044*T913+0.635*T923)/3.0
D23=F23*(3.011/T9-1.5)/T9
D24=F24*(12.007/T9+1.0/3.0)/T9
D2=F21*D22+F22*D21+D23+D24
ZCNO=Z
ZAMZ=ZCNO/AMZ
ZAMZ=ZCC/AMC+ZCO/AMO

C  VERSION II
C  ZAMZ=ZCC/AMC+ZCN/AMZ+ZCO/AMO

CCNO=D*(X/AMH)*ZAMZ*QCNO*EV6
ECNO=CCNO*F2

C  VERSION II
C  MAKE CNO RATE ZERO IF NO CARBON

```

```

C IF(ZCC,EQ,0)THEN
C ECNO=0
C END IF

C HE4(2A,G)C12 (II,III)
C CHECK Q-VALUE
F31=2.790E-08/T9**3*EXP(ANL-4.4027/T9)
F32=1.350E-07/T932*EXP(ANL-24.811/T9)
F3=F31+F32
D3=F31*(4.4027/T9-3.0)/T9+F32*(24.811/T9-1.5)/T9
CHEC=D*D*(Y/AMHE)**3/6.0*QHEC*EV6
EHEC=CHEC*F3
C C12(A,G)O16 (II,III)
F410=1.580E+08/T9**2*EXP(ANL-32.120/T913-(T9/5.863)**2)
F411=(1.000+0.621*T923)
F412=(1.000+0.047/T923)
F413=(F411/F412)**2
F41=F410*F413
F42=2.740E+07/T923*EXP(ANL-32.120/T913)
F43=1.250E+03/T923*EXP(ANL-27.499/T9)
F44=1.430E-02*T9**5*EXP(ANL-15.541/T9)
F4=F41+F42+F43+F44
D41=0.0
D410=F410*(32.120/T913/3.0-2.0-2.0*(T9/5.863)**2)/T9
D411=4.0/3.0*0.621/T913*F411
D412=-4.0/3.0*0.047/T953*F412
D413=(F412**2*D411-F411**2*D412)/F412**4
D41=F410*D413+D410*F413
D42=F42*(32.120/T913-2.0)/3.0/T9
D43=F43*(27.499/T9-2.0/3.0)/T9
D44=F44*(15.541/T9+5.0)/T9
D4=D41+D42+D43+D44
CCAO=D*(Y/AMHE)*(ZCC/AMC)*QCAO*EV6
ECAO=CCAO*F4
EAC(1)=ECAO
C O16(A,G)NE20 (II)
F51=5.490E+09/T923*EXP(ANL-39.756/T913)
F52=4.090E+01/T932*EXP(ANL-10.359/T9)
F53=3.920E+02/T932*EXP(ANL-12.243/T9)
F54=8.050E+00*T9**2*EXP(ANL-20.093/T9)
F5=F51+F52+F53+F54
D51=F51*(39.756/T913-2.0)/T9/3.0
D52=F52*(10.359/T9-1.5)/T9
D53=F53*(12.243/T9-1.5)/T9
D54=F54*(20.093/T9+2.0)/T9
D5=D51+D52+D53+D54
COANE=D*(Y/AMHE)*(ZCO/AMO)*QOANE*EV6
EOANE=COANE*F5
EAC(2)=EOANE
C NE20(A,G)MG24 (II)
THETA6=0.1
F61=4.110E+11/T923*EXP(ANL-46.766/T913-(T9/2.219)**2)
F62=(1.000+0.009*T913+0.882*T923+0.055*T9+0.749*T943+0.119*T953)
F63=5.270E+03/T932*EXP(ANL-15.869/T9)
F64=6.510E+03*T912*EXP(ANL-16.223/T9)
F65=4.210E+01/T932*EXP(ANL-9.115/T9)
F66=3.200E+01/T923*EXP(ANL-9.383/T9)
F60=1.000+5.000*EXP(-18.960/T9)
F6=(F61*F62+F63+F64+THETA6*(F65+F66))/F60
D61=F61*((46.766/T913-2.0)/3.0-2.0*(T9/2.219)**2)/T9
D62=(0.009/T923+1.764/T913+0.165+2.996*T913+0.595*T923)/3.0
D63=F63*(15.869/T9-1.5)/T9
D64=F64*(16.223/T9+0.5)/T9
D65=F65*(9.115/T9-1.5)/T9
D66=F66*(9.383/T9-2.0/3.0)/T9
C D60=18.960/T9**2*(F60-1.0)
D60=18.960/T9**2*5.000*EXP(-18.960/T9)
D6=(F62*D61+F61*D62+D63+D64+THETA6*(D65+D66))/F60-F6*D60/F60
CNEMG=D*(Y/AMHE)*(ZCNE/AMNE)*QNEMG*EV6
ENEMG=CNEMG*F6
EAC(3)=ENEMG
C MG24(A,G)SI28 (II,III)
THETA7=0.1
F71=4.780E+01/T932*EXP(ANL-13.506/T9)
F72=2.380E+03/T932*EXP(ANL-15.218/T9)
F73=2.470E+02*T932*EXP(ANL-15.147/T9)
F74=1.720E-09/T932*EXP(ANL-5.028/T9)

```

$F75=1.250E-03/T932*EXP(ANL-7.929/T9)$
 $F76=2.430E+01/T9 *EXP(ANL-11.523/T9)$
 $F70=1.000+5.000*EXP(-15.882/T9)$
 $F7=(F71+F72+F73+THETA7*(F74+F75+F76))/F70$
 $D71=F71*(13.506/T9-1.5)/T9$
 $D72=F72*(15.218/T9-1.5)/T9$
 $D73=F73*(15.147/T9+1.5)/T9$
 $D74=F74*(5.028/T9-1.5)/T9$
 $D75=F75*(7.929/T9-1.5)/T9$
 $D76=F76*(11.523/T9-1.0)/T9$
C $D70=15.882/T9**2*(F70-1.0)$
 $D70=15.882/T9**2*5.000*EXP(-15.882/T9)$
 $D7=(D71+D72+D73+THETA7*(D74+D75+D76))/F70-F7*D70/F70$
 $CMGSI=D*(Y/AMHE)*(ZCMG/AMMG)*QMGSI*EV6$
 $EMGSI=CMGSI*F7$
 $EAC(4)=EMGSI$
 $F8=0.0$
 $D8=0.0$
 $CSIAS=D*(Y/AMHE)*(ZCSI/AMSI)*QSIAS*EV6$
 $ESIAS=CSIAS*F8$
 $EAC(5)=ESIAS$
 $ESAAR=0.0$
 $EARCA=0.0$
C $C+C, C+O, O+O$ RATES FROM FCZ II pp85,103
C $C12+C12=MG24$
 $T9A=T9/(1.000+0.067*T9)$
 $T9A13=T9A**(1.0/3.0)$
 $T9A23=T9A13**2$
 $T9A56=T9A13*SQRT(T9A)$
 $DT9A=(T9A/T9)**2$
 $FCC1=1.260E+27*T9A56/T932*EXP(ANL-84.165/T9A13)$
 $FCC2=EXP(-0.010*T9A**4)+5.560E-03*EXP(1.685*T9A23)$
 $FCC=FCC1/FCC2$
 $DCC1=FCC1*((84.165/T9A13+2.5)/T9A*DT9A/3.0-1.5/T9)$
 $DCC2=(-0.010*4.0*T9A**3*EXP(-0.010*T9A**4)$
 $1 +5.560E-03*1.685*2.0/3.0/T9A13*EXP(1.685*T9A23))*DT9A$
 $DCC=(FCC2*DCC1-FCC1*DCC2)/FCC2**2$
 $CCCMG=D/2.0*(ZCC/AMC)**2*QCCMG*EV6$
 $ECCMG=CCCMG*FCC$
 $EAC(6)=ECCMG$
C $C+O=SI$
 $T9A=T9/(1.000+0.055*T9)$
 $T9A13=T9A**(1.0/3.0)$
 $T9A23=T9A13**2$
 $T9A56=T9A13*SQRT(T9A)$
 $DT9A=(T9A/T9)**2$
 $FCO1=1.720E+31*T9A56/T932*EXP(ANL-106.594/T9A13)$
 $FCO2=EXP(-0.180*T9A**2)+1.060E-03*EXP(2.562*T9A23)$
 $DCO1=FCO1*((106.594/T9A13+2.5)/T9A*DT9A/3.0-1.5/T9)$
 $DCO2=(-0.180*2.0*T9A*EXP(-0.180*T9A**2)$
 $1 +1.060E-03*2.562*2.0/3.0/T9A13*EXP(2.562*T9A23))*DT9A$
 $FCO=FCO1/FCO2$
 $DCO=(FCO2*DCO1-FCO1*DCO2)/FCO2**2$
 $CCOSI=D*(ZCC/AMC)*(ZCO/AMO)*QCOSI*EV6$
 $ECOSI=CCOSI*FCO$
 $EAC(7)=ECOSI$
C $O+O=S$
 $T9A=T9/(1.000+0.067*T9)$
 $T9A13=T9A**(1.0/3.0)$
 $T9A23=T9A13**2$
 $T9A56=T9A13*SQRT(T9A)$
 $DT9A=(T9A/T9)**2$
 $FOO1=3.610E+37*T9A56/T932*EXP(ANL-135.930/T9A13)$
 $FOO2=EXP(-0.032*T9A**2)+3.890E-04*EXP(2.659*T9A23)$
 $DOO1=FOO1*((135.930/T9A13+2.5)/T9A*DT9A/3.0-1.5/T9)$
 $DOO2=(-0.032*2.0*T9A*EXP(-0.032*T9A**2)$
 $1 +3.980E-04*2.659*2.0/3.0/T9A13*EXP(2.659*T9A23))*DT9A$
 $FOO=FOO1/FOO2$
 $DOO=(FOO2*DOO1-FOO1*DOO2)/FOO2**2$
 $COOS=D/2.0*(ZCO/AMO)**2*QOOS*EV6$
 $EOOS=COOS*FOO$
 $EAC(8)=EOOS$
C $SEN=EPPC+ECNO+EHEC+ECAO+EOANE+ENEMG+EMGSI$
C $SEN=CPPC*F1+CCNO*F2+CHEC*F3+CCAO*F4+COANE*F5+CNEMG*F6+CMGSI*F7$
 $SEN=EPPC+ECNO+EHEC+ECAO+EOANE+ENEMG+EMGSI+ESIAS+ESAAR+EARCA$
 $SEN=SEN+ECCMG+ECOSI+EOOS$
 $SET=CPPC*D1+CCNO*D2+CHEC*D3+CCAO*D4+COANE*D5+CNEMG*D6+CMGSI*D7$

```

SET=SET+CSIAS*D8+CCCMG*DCC+CCOSI*DCO+COOS*DOO
C SED=CPPC*F1+CCNO*F2+2.0*CHEC*F3+CCAO*F4+COANE*F5+CNEMG*F6+CMGSI*F7
C SED=SED+CSIAS*F8+CCCMG*FCC+CCOSI*FCO+COOS*FOO
SED=EPPC+ECNO+2*EHEC+ECAO+EOANE+ENEMG+EMGSI+ESIAS+ECCMG+ECOSI+EOOS
SED=SED/D
TEN=0.0
DEN=0.0
SENL=-100.0
IF(SEN)200,200,201
201 CONTINUE
TEN=T9*SET/SEN
DEN=D*SED/SEN
SENL=ALOG(SEN)
200 CONTINUE
EN=SEN
EV=SENL
DET=TEN
DED=DEN
DPPCT=0.0
IF(F1)1,1,10
10 DPPCT=T9*D1/F1
1 CONTINUE
DCNOT=0.0
IF(F2)2,2,20
20 DCNOT=T9*D2/F2
2 CONTINUE
DHECT=0.0
IF(F3)3,3,30
30 DHECT=T9*D3/F3
3 CONTINUE
DCAOT=0.0
IF(F4)4,4,40
40 DCAOT=T9*D4/F4
4 CONTINUE
DETA(1)=DCAOT
DOANET=0.0
IF(F5)5,5,50
50 DOANET=T9*D5/F5
5 CONTINUE
DETA(2)=DOANET
DNEMGT=0.0
IF(F6)6,6,60
60 DNEMGT=T9*D6/F6
6 CONTINUE
DETA(3)=DNEMGT
DMGSIT=0.0
IF(F7)7,7,70
70 DMGSIT=T9*D7/F7
7 CONTINUE
DETA(4)=DMGSIT
DSIAST=0.0
IF(F8)8,8,80
80 DSIAST=T9*D8/F8
8 CONTINUE
DETA(5)=DSIAST
DCCMGT=0.0
IF(FCC)190,190,191
191 DCCMGT=T9*DCC/FCC
190 CONTINUE
DETA(6)=DCCMGT
DCOSIT=0.0
IF(FCO)290,290,291
291 DCOSIT=T9*DCO/FCO
290 CONTINUE
DETA(7)=DCOSIT
DOOST=0.0
IF(FOO)390,390,391
391 DOOST=T9*DOO/FOO
390 CONTINUE
DETA(8)=DOOST
RETURN
END
C INVERT FDIS
SUBROUTINE DFIS(L,F,X)
IMPLICIT REAL*8 (A-F,I,O-Z)
ABS(X)=DABS(X)
ALOG(X)=DLOG(X)

```

```

DATA C/1.0/,E/1.0E-07/
ITN=0
P=2.0/(L+2)
IF(F-C)1,1,2
1 X=ALOG(F)
GO TO 3
2 X=(F/P)**P
3 ITN=ITN+1
CALL FDIS(L,X,FN)
M=L-2
CALL FDIS(M,X,FP)
FP=L*FP/2.0
DX=-(FN-F)/FP
X=X+DX
IF(ITN-20)7,8,8
8 WRITE(6,16)ITN,X,DX
GO TO 4
7 CONTINUE
IF(ABS(X)-C)5,5,6
5 IF(ABS(DX)-E)4,3,3
6 IF(ABS(DX/X)-E)4,3,3
4 RETURN
16 FORMAT(I10,1P2E10.3)
END
SUBROUTINE FDIS(L,X,Y)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION GAM(3)
DIMENSION CNL(5,3),CNM(5,3),CNH(5,3),CDL(5,3),CDM(5,3),CDH(5,3)
DIMENSION CNL1(5),CDL1(5),CNM1(5),CDM1(5),CNH1(5),CDH1(5)
DIMENSION CNL2(5),CDL2(5),CNM2(5),CDM2(5),CNH2(5),CDH2(5)
DIMENSION CNL3(5),CDL3(5),CNM3(5),CDM3(5),CNH3(5),CDH3(5)
EQUIVALENCE (CNL(1,1),CNL1(1)),(CNL(1,2),CNL2(1)),
1(CNL(1,3),CNL3(1)),(CDL(1,1),CDL1(1)),(CDL(1,2),CDL2(1)),
2(CDL(1,3),CDL3(1))
EQUIVALENCE (CNM(1,1),CNM1(1)),(CNM(1,2),CNM2(1)),
1(CNM(1,3),CNM3(1)),(CDM(1,1),CDM1(1)),(CDM(1,2),CDM2(1)),
2(CDM(1,3),CDM3(1))
EQUIVALENCE (CNH(1,1),CNH1(1)),(CNH(1,2),CNH2(1)),
1(CNH(1,3),CNH3(1)),(CDH(1,1),CDH1(1)),(CDH(1,2),CDH2(1)),
2(CDH(1,3),CDH3(1))
SQRT(X)=DSQRT(X)
EXP(X)=DEXP(X)
DATA GAM/1.00,0.50,0.75/
DATA PIE/3.141592653589793/
DATA CNL1/
1-1.253314128820E 00,-1.723663557701E 00,-6.559045729258E-01,
2-6.342283197682E-02,-1.488383106116E-05/
DATA CDL1/
1 1.000000000000E 00, 2.191780925980E 00, 1.605812955406E 00,
2 4.443669527481E-01, 3.624232288112E-02/
DATA CNM1/
1 1.073812769400E 00, 5.600330366000E 00, 3.688221127000E 00,
2 1.174339281600E 00, 2.364193552700E-01/
DATA CDM1/
1 1.000000000000E 00, 4.603184066700E 00, 4.307591067400E-01,
2 4.215113214500E-01, 1.183260160100E-02/
DATA CNH1/
1-8.222559330000E-01,-3.620369345000E+01,-3.015385410000E+03,
2-7.049871579000E+04,-5.698145924000E+04/
DATA CDH1/
1 1.000000000000E 00, 3.935689841000E+01, 3.568756266000E+03,
2 4.181893625000E+04, 3.385138907000E+05/
DATA CNL2/
1-3.133285305570E-01,-4.161873852293E-01,-1.502208400588E-01,
2-1.339579375173E-02,-1.513350700138E-05/
DATA CDL2/
1 1.000000000000E 00, 1.872608675902E 00,1.145204446578E 00,
2 2.570225587573E-01, 1.639902543568E-02/
DATA CNM2/
1 6.781766266600E-01, 6.331240179100E-01, 2.944796517720E-01,
2 8.013207114190E-02, 1.339182129400E-02/
DATA CDM2/
1 1.000000000000E 00, 1.437404003970E-01, 7.086621484500E-02,
2 2.345794947350E-03,-1.294499288350E-05/
DATA CNH2/
1 8.224499762600E-01, 2.004630339300E+01, 1.826809344600E+03,
2 1.222653037400E+04, 1.404075009200E+05/

```

```

DATA CDH2/
1 1.000000000000E 00, 2.348620765900E+01, 2.201348374300E+03,
2 1.144267359600E+04, 1.658471590000E+05/
DATA CNL3/
1-2.349963985406E-01,-2.927373637547E-01,-9.883097588738E-02,
2-8.251386379551E-03,-1.874384153223E-05/
DATA CDL3/
1 1.000000000000E 00, 1.608597109146E 00, 8.275289530880E-01,
2 1.522322382850E-01, 7.695120475064E-03/
DATA CNM3/
1 1.153021340200E 00, 1.059155897200E 00, 4.689880309500E-01,
2 1.188290878400E-01, 1.943875578700E-02/
DATA CDM3/
1 1.000000000000E 00, 3.734895384100E-02, 2.324845813700E-02,
2-1.376677087400E-03, 4.646639278100E-05/
DATA CNH3/
1 2.467400236840E 00, 2.191675823680E+02, 1.238293790750E+04,
2 2.206677249680E+05, 8.494429200340E+05/
DATA CDH3/
1 1.000000000000E 00, 8.911251406190E+01, 5.045756696670E+03,
2 9.090759463040E+04, 3.899609156410E+05/
FN=0.0
FD=0.0
N=(L+3)/2
IF(X-1.0)1,4,4
4 IF(X-4.0)2,2,3
1 CONTINUE
EX=EXP(X)
DO 10 M=1,5
K=6-M
FN=EX*FN+CNL(K,N)
FD=EX*FD+CDL(K,N)
10 CONTINUE
DD=FN/FD
Y=EX*(GAM(N)* SQRT(PIE)+EX*DD)
GO TO 5
2 CONTINUE
DO 20 M=1,5
K=6-M
FN=X*FN+CNM(K,N)
FD=X*FD+CDM(K,N)
20 CONTINUE
DD=FN/FD
Y=DD
GO TO 5
3 CONTINUE
C=2*N-1
PX=1.0/X/X
SX=SQRT(X)
DO 30 M=1,5
K=6-M
FN=PX*FN+CNH(K,N)
FD=PX*FD+CDH(K,N)
30 CONTINUE
DD=FN/FD
Y=SX**(2*N-1)*(2.0/C+PX*DD)
GO TO 5
5 RETURN
END
C INTERPOLATION IN TABLES
C SUBROUTINE TABLES(LQ,ITA,NDS,INC,NX,NY,DX,END,XA,YA,ZA,
C 1 X,Y,Z,ZC,DZX,DZY)
SUBROUTINE TABLES(LQ,ITA,NDS,INC,NX,NY,DX,ZC,XHY,YHE,ZME,
1 END,XA,YA,ZA,X,Y,Z,DZX,DZY)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XA(64),YA(64),ZA(64,64)
IF(ITA)10,10,11
10 WRITE( 6,50)NDS
50 FORMAT(1X,'TABLE',14)
READ(NDS,51)NAME,NY,NX,DX,ZC,XHY,YHE,ZME
WRITE( 6,51)NAME,NY,NX,DX,ZC,XHY,YHE,ZME
51 FORMAT(6X,A4,2I5,F10.5,E10.3,3F10.5)
I=0
20 I=I+1
READ(NDS,52)YA(I),XA(I),(ZA(J,I),J=1,NX)
C WRITE( 6,52)YA(I),XA(I),(ZA(J,I),J=1,NX)
52 FORMAT(F4.1,F6.1,10F10.5,(10X,10F10.5))

```

```

      IF(I-NY)22,23,22
22  IF(YA(I)-END)20,21,20
21  NY=I+INC
23  ITA=1
11  CALL INTERP(LQ,NX,NY,DX,XA,YA,ZA,X,Y,Z,DZX,DZY)
      RETURN
      END
      SUBROUTINE INTERP(LQ,NX,NY,DX,XA,YA,ZA,X,Y,Z,DZX,DZY)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XA(64),YA(64),ZA(64,64),XY(64),XX(3),YY(3),ZX(3),ZY(3)
      DIMENSION DZ(3)
      CF(X1,X2,X3,Z1,Z2,Z3)=((Z1-Z2)/(X1-X2)-(Z2-Z3)/(X2-X3))/(X1-X3)
      BF(X1,X2,Z1,Z2)=(Z1-Z2)/(X1-X2)-C*(X1+X2)
      AF(X1,Z1)=Z1-(B+C*X1)*X1
      NLQ=LQ+1
      IY=1
3  IF(Y-YA(IY))1,20,20
20  IF(IY-NY+1)2,2,1
2  IY=IY+1
   GO TO 3
1  IF(2*Y-YA(IY)-YA(IY+1))4,5,5
5  IY=IY+1
4  IY=MAX0(IY,2)
   IY=MIN0(IY,NY+1-LQ)
   DO 6 JY=1,NLQ
     KY=IY+JY-2
     YY(JY)=YA(KY)
     KK=0
     IF(DX)14,15,14
15  KK=1
14  DO 7 IX=1,NX
7  XY(IX)=XA(KY+KK*(IX-1))+(IX-1)*DX
   IX=1
10  IF(X-XY(IX))8,90,90
90  IF(IX-NX+1)9,8,8
9  IX=IX+1
   GO TO 10
8  IF(2*X-XY(IX)-XY(IX+1))11,12,12
12  IX=IX+1
11  IX=MAX0(IX,2)
     IX=MIN0(IX,NX+1-LQ)
     DO 13 JX=1,NLQ
       KX=IX+JX-2
       XX(JX)=XY(KX)
13  ZX(JX)=ZA(KX,KY)
     C=0.0
     GO TO (21,22),LQ
22  C=CF(XX(1),XX(2),XX(3),ZX(1),ZX(2),ZX(3))
21  B=BF(XX(1),XX(2),ZX(1),ZX(2))
     A=AF(XX(1),ZX(1))
     DZX=B+2*C*X
D  WRITE(6,61)XX,ZX,A,B,C,DZX
   DZ(JY)=DZX
6  ZY(JY)=A+(B+C*X)*X
   C=0.0
   GO TO (31,32),LQ
32  C=CF(YY(1),YY(2),YY(3),ZY(1),ZY(2),ZY(3))
31  B=BF(YY(1),YY(2),ZY(1),ZY(2))
     A=AF(YY(1),ZY(1))
D  WRITE(6,61)YY,ZY,A,B,C
   Z=A+(B+C*Y)*Y
   DZY=B+2*C*Y
   C=0.0
   GO TO (41,42),LQ
42  C=CF(YY(1),YY(2),YY(3),DZ(1),DZ(2),DZ(3))
41  B=BF(YY(1),YY(2),DZ(1),DZ(2))
     A=AF(YY(1),DZ(1))
     DZX=A+(B+C*Y)*Y
     RETURN
D61  FORMAT(10F10.5)
      END

```

D.2 Other Codes

In addition to the TRC code, three other small codes were used while the data for this work was being produced. The first, 'dread.f', was used to read the model contained in the file named 'fort.102'. This proved necessary, as models could otherwise easily be lost or wrongly labelled. The second, 'dchecker.f', was used to read data from each zone of a model in the file 'fort.102'. For each zone it will print mass fractions, radius, mass contained within that zone, and zone number. The third and final code, 'dzcomp.f', was used to calculate the fractions and masses of metals contained within a model as a whole, and within the hydrogen-depleted core of a model.

'dread.f'

```

PROGRAM READER
C CODE BY PDT TO INTERPRET DATA FROM TRC MODELS
C TAKEN FROM TRC CODE WITH MINOR MODIFICATIONS
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION ZAC(8)
READ ( 102)MB,ITS,DYT,TTY,SMASSI,SMASS,STARRS,STARLS,
1 TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
2 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,
3 (ZAC(J),J=1,6)
WRITE(6,163)
163 FORMAT(' STEP', ' GRID', ' TIME STEP', ' TOTL TIME',
1 5X, 'IMASS',5X, 'CMASS',4X, 'RADIUS', ' LUMNOSITY',
2 /,T11,8X, 'TP',8X, 'PP',8X, 'DP',8X, 'XP',8X, 'YP',8X, 'ZP',
3 /,T11,8X, 'TC',8X, 'PC',8X, 'DC',8X, 'XC',8X, 'YC',8X, 'ZC',
4 /,T11,6X, 'ZC12',6X, 'ZO16',5X, 'ZNE20',5X, 'ZMG24',5X, 'ZS128',
56X, 'ZS32')
WRITE(6,161)ITS,ICN,DTY,TTY,SMASSI,SMASS,STARRS,STARLS,
1 TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
2 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,
3 (ZAC(J),J=1,6)
WRITE(6,162) (SMASSI-SMASS),LOG10(STARLS),LOG10(TPHOT)
161 FORMAT(2I5,(T11,1P6E10.3))
162 FORMAT(1P3E10.3)
END

```

'dchecker.f'

```

PROGRAM TESTER
C CODE BY PDT TO CHECK ON MASS DISTRIBUTION IN TRC MODELS
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION VMC(512),VRC(512),VLC(512),VPC(512),VTC(512)
DIMENSION VDC(512),VUC(512),VSC(512),VKC(512),VEC(512)
DIMENSION VGC(512),VVC(512),VXC(512),VYC(512),VZC(512)
DIMENSION VXD(512),VYD(512),VZD(512),AZC(512,8)
DIMENSION EC1(512),EC2(512),EC3(512),ZMASS(512)
DIMENSION RADIUS(512)
NMZ=8
SUNM=1.989E33
SUNR=6.96E10
READ(102)MB,ITS,DTY,TTY,SMASSI,SMASS,STARRS,STARLS,
1 TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
2 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,
3 (AZC(MB,JB),JB=1,NMZ)

DO 270 IB=1,MB
READ( 102) VMC(IB),VRC(IB),VLC(IB),VPC(IB),VTC(IB),
1 VDC(IB),VUC(IB),VSC(IB),VKC(IB),VEC(IB),VGC(IB),VVC(IB),
2 VXC(IB),VYC(IB),VZC(IB),(AZC(IB,JB),JB=1,NMZ)
ZMASS(IB)=(1-EXP(VMC(IB)))*SMASS*(1.001)
RADIUS(IB)=EXP(VRC(IB))/SUNR
HB=IB
WRITE(6,161)VXC(IB),VYC(IB),VZC(IB),RADIUS(IB),
1(ZMASS(IB)/SMASS),HB

```

```

270 CONTINUE
161 FORMAT(1P6E10.3)
END

```

'dzcomp.f'

```

PROGRAM TESTER
C CODE BY PDT TO CHECK ON METALS IN TRC MODELS
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION VMC(512),VRC(512),VLC(512),VPC(512),VTC(512)
DIMENSION VDC(512),VUC(512),VSC(512),VKC(512),VEC(512)
DIMENSION VGC(512),VVC(512),VXC(512),VYC(512),VZC(512)
DIMENSION AZC(512,8)
DIMENSION VZM(8),VZT(8),VZTC(8)
DIMENSION VZTF(8),VZTCF(8)
NMZ=8
SUNM=1.989E33
SUNR=6.96E10
READ(102)MB,ITS,DTY,TTY,SMASSI,SMASS,STARRS,STARLS,
1 TPHOT,PPHOT,DPHOT,XPHOT,YPHOT,ZPHOT,
2 TCEN,PCEN,DCEN,XCEN,YCEN,ZCEN,
3 (AZC(MB,JB),JB=1,NMZ)
DO 270 IB=1,MB
READ( 102) VMC(IB),VRC(IB),VLC(IB),VPC(IB),VTC(IB),
1 VDC(IB),VUC(IB),VSC(IB),VKC(IB),VEC(IB),VGC(IB),VVC(IB),
2 VXC(IB),VYC(IB),VZC(IB),(AZC(IB,JB),JB=1,NMZ)
270 CONTINUE
DO 290 IB=MB,1,-1
RMASS=(1-EXP(VMC(IB)))*SMASS*(1.001)
DMASS=RMASS-PMASS
DO 260 JB=1,NMZ
VZM(JB)=DMASS*AZC(IB,JB)
VZT(JB)=VZT(JB)+VZM(JB)
ZTOT=ZTOT+VZM(JB)
IF(VXC(IB))295,295,300
295 CONTINUE
VZTC(JB)=VZTC(JB)+VZM(JB)
ZTOTC=ZTOTC+VZM(JB)
CMASS=RMASS
300 CONTINUE
260 CONTINUE
PMASS=RMASS
290 CONTINUE
DO 280 JB=1,6
VZTF(JB)=VZT(JB)/SMASS
VZTCF(JB)=VZTC(JB)/CMASS
280 CONTINUE
ZTOTF=ZTOT/SMASS
ZTOTCF=ZTOTC/CMASS
WRITE(6,161)(VZT(JB),JB=1,6),ZTOT
WRITE(6,161)(VZTF(JB),JB=1,6),ZTOTF
WRITE(6,161)(VZTC(JB),JB=1,6),ZTOTC
WRITE(6,161)(VZTCF(JB),JB=1,6),ZTOTCF
WRITE(6,162)SMASS,CMASS
161 FORMAT(1P7E10.3)
162 FORMAT(1P2E10.3)
END

```

References

- Anders, E., Grevesse, N., 1989, *Geochim. Cosmochim. Acta.*, **53**, 197
- Andriessse, C. D., Donn, B. D., Viotti, R., 1978, *M. N. R. A. S.*, **185**, 771
- Applegate, J. H., Hogan, C. R., Scherrer, J., 1988, *Ap. J.*, **329**, 572
- Arnett, W. D., 1978, *Ap. J.*, **219**, 1008
- Bertotti, B., Carr, B. J., Rees, M. J., 1983, *M. N. R. A. S.*, **203**, 945
- Böhm-Vitense, E., 1992, 'Introduction to Stellar Astrophysics, volume 3', Cambridge University Press
- Bond, H. E., 1981, *Ap. J.*, **248**, 606
- Carney, B. W., 1979, *Ap. J.*, **233**, 211
- Carney, B. W., Peterson, R. C., 1981, *Ap. J.*, **231**, 238
- Carr, B. J., Bond, J. R., Arnett, W. D., 1984, *Ap. J.*, **277**, 445
- Carson, T. R., 1992, *Rev. Mexicana. Astron. Astrof.*, **23**, 151
- Carson, T.R., 1995, Private Communications
- Carson, T. R., Sharp, C. M., 1991, in preparation
- Carson, T. R., Luo, G., Sharp, C. M. 1992, *Rev. Mexicana. Astron. Astrof.*, **23**, 217
- Cassinelli, J. P., Lamers, H. J. G. L. M., 1987, 'Exploring the Universe with the IUE Satellite', Ed., Kondo, Y., et al, D. Reidel Publishing Company
- Cassinelli, J. P., Mathis, J. S., Savage, B. D., 1981, *Science*, **212**, 1497
- Cassisi, S., Castellani, V., 1993, *Ap. J. Suppl.*, **88**, 509
- Castellani, V., Chieffi, A., Tornambé, A., 1983, *Ap. J.* **272**, 249
- Castor, J., Abbott, D. C., Klein, R., 1975, *Astrophys. J.*, **195**, 157
- Chieffi, A., Tornambé, A., 1984, *Ap. J.*, **287**, 745
- Chieffi, A., Tornambé, A., 1986, *M. N. R. A. S.*, **220**, 529
- Chiosi, C., 1980, 'I.A.U. Colloquium no. 59', Ed. Chiosi, C., Stalio., R., D. Reidel Publishing Company

- Christy, R. F., 1966, *Ap. J.*, **144**, 108
- Cody, W. J., Thatcher, B. M., 1967, *Math. Comp.*, **21**, 30
- D'Antona, F., 1982, *Astron. Astrophys.*, **115**, L1
- Dorman, B., Rood, R. T., O'Connell, R. W., 1993, *Ap. J.*, **419**, 596
- Fowler, W. A., Caughlan, G. R., Zimmerman, B. A., 1967, *Ann. Rev. Astron. Ap.*, **5**, 525
- Fowler, W. A., Caughlan, G. R., Zimmerman, B. A., 1975, *Ann. Rev. Astron. Ap.*, **13**, 69
- Fujimoto, M. Y., Iben, I., Chieffi, A., Tornambé, A., 1984, *Ap. J.*, **287**, 749
- Guenther, D. B., Demarque, P., 1983, *Astron. Astrophys.*, **118**, 262
- Harris, W. E., Canterna, R., 1979, *Ap. J.*, **231**, L19
- Harris, M. J., Fowler, W. A., Caughlan, G. R., Zimmerman, B. A., 1983, *Ann. Rev. Astron. Ap.*, **21**, 165
- Henyey, L. G., Vardya, M. S., Bodenheimer, P. L., 1965, *Ap. J.*, **142**, 841
- Hogan, C. J., 1983, *Ap. J.*, **274**, 7
- Jura, M., 1986, *Ap. J.*, **301**, 624
- Karimabadi, H., Blitz, L., 1984, *Ap. J.*, **283**, 169
- Kippenhahn, R., Weigert, A., 1990, 'Stellar Structure and Evolution', Springer-Verlag
- Kippenhahn, R., Weigert, A., Hofmeister, E., 1967, *Meth. Comp. Phys.*, **7**, 129
- Kirshner, R. P., 1981, 'Supernovae: A Survey of Current Research', Ed. Rees, M. J., Stoneham, R. J., D. Reidel Publishing Company
- Klapp, J., 1983, *Astrophys. Spac. Sci.*, **93**, 313
- Kolesnik, I. G., 1978, *Sov. Astron.*, **22**(5)
- Kudritzki, R. P., Pauldrach A., Puls, J., 1986, *Astron. Astrophys.*, **164**, 86
- Kudritzki, R. P., Pauldrach A., Puls, J., 1987, *Astron. Astrophys.*, **173**, 293

- Kudritzki, R. P., Pauldrach A., Puls, J., Abbott, D.C., 1989, *Astron. Astrophys.*, **219**, 205
- Lacey, C. G., Field, G. B., 1988, *Ap. J.*, **330**, L1
- Lahav, O., 1986, *Mon. Not. R. Astr. Soc.*, **220**, 259
- Lamers, H. J. G. K. M., 1892, 'I.A.U. Colloquium no. 59', Ed., Chiosi, C., Stalio, R., D. Reidel Publishing Company
- Limongi, M., Tornambé, A., 1991, *Ap. J.*, **371**, 317
- Matsumoto, T., et al, 1984, *Ap. J.*, **329**, 567
- Milligan, H., Carson, T. R., 1992, in preparation.
- Nieuwenhuijzen, H., de Jager, C., 1990, *Astr. Astrophys.*, **231**, 134
- Ober, W. W., El Eid, M. F., Fricke, K. J., 1983, *Astr. Astrophys.*, **119**, 61
- Paczynski, B., 1971, *Act. Ast.*, **21**, 1
- Palla, F., Stahler, S., Salpeter, E., 1983, *Ap. J.*, **271**, 632
- Rees, M. J., 1978, *Nature*, **275**, 35
- Rogers, F. J., Iglesias, C. A., 1991, *Ap. J.*, **371**, L73
- Rogers, F. J., Iglesias, C. A., 1992, *Ap. J. Suppl.*, **79**, 507
- Rosseland, S., 1924, *Mon. Not. R. Astr. Soc.*, **84**, 525
- Salpeter, E. E., 1955, *Ap. J.*, **121**, 161
- Silk, J., 1977, *Ap. J.*, **211**, 638
- Stahler, S. W., Palla, F., Salpeter, E. E., 1986, *Ap. J.* **302**, 590
- Steigman, G., 1985, 'Nucleosynthesis ; Challenges and New Developments', Ed., Arnett, W. D., Truran, J. W., University of Chicago Press
- Thacker, P., 1993, undergraduate dissertation
- Vainer, B. V., 1990, *Sov. Astron.*, **34**(2)
- Vainer, B.V., Chuvenkov, V.V., Shchekinov, Y.A., 1986, *Astrofizika* **25**(3), 559

Wagner, R., 1974, *Ap. J.*, **191**, 173

Woosley, S. E., Weaver, T. A., 1982, 'Essays in Nuclear Astrophysics', Ed.,
Barnes, C. A., Clayton, D. O., Schraum, D. N., Cambridge University Press

Zinn, R., 1980, *Ap. J. Suppl.*, **42**, 19