Draft version March 16, 2026
Typeset using LaTeX **twocolumn** style in AASTeX7

The YREC Stellar Evolution Code: Public Data Release

Marc H. Pinsonneault [1,2] Jennifer L. van Saders [3] Lyra Cao [4] Jamie Tayar [5] Franck Delahaye [6]
Leslie M. Morales [5] Rachel A. Patton [7] Matthew C. Rendina [1] Joel C. Zinn [8]
Zachary R. Claytor [9] Amanda L. Ash [1] Susan Byrom [5] Kaili Cao [2,10] and Vincent A. Smedile [1,2]

[1] *Department of Astronomy, The Ohio State University, Columbus, OH 43210, USA*
[2] *Center for Cosmology and AstroParticle Physics (CCAPP), The Ohio State University, 191 West Woodruff Ave, Columbus, OH 43210, USA*
[3] *Institute for Astronomy, University of Hawai'i, 2680 Woodlawn Drive, Honolulu, HI 96822, USA*
[4] *Department of Physics and Astronomy, Vanderbilt University, Nashville, TN 37212, USA*
[5] *Department of Astronomy, University of Florida, Gainesville, FL 32611, USA*
[6] *LUX, Observatoire de Paris, Université PSL, Sorbonne Université, CNRS, 75005 Paris, France*
[7] *Department of Physics and Astronomy and Pittsburgh Particle Physics, Astrophysics and Cosmology Center (PITT PACC), University of Pittsburgh, 3941 O'Hara Street, Pittsburgh, PA 15260, USA*
[8] *Department of Physics and Astronomy, California State University, Long Beach, Long Beach, CA 90840, USA*
[9] *Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA*
[10] *Department of Physics, The Ohio State University, 191 West Woodruff Ave, Columbus, OH 43210, USA*

ABSTRACT

In this paper we present the public release of the Yale Rotating Evolution Code (YREC). YREC is a stellar evolution code that covers brown dwarfs and stars across a wide range of masses, and evolutionary states from the pre-MS through helium burning. We summarize the key ingredients of the code, document the code performance, and discuss its strengths and limitations. We present libraries of input files, documentation, sample use cases, and scripts. In addition to usage as a research tool, we highlight the utility of the code for educational purposes.

Keywords: Stellar astronomy (1583) — Solar physics (1476)

1. INTRODUCTION

The theory of stellar structure and evolution is deep and beautiful. Its bedrock lies in fundamental physical principles. The equations of stellar structure are expressions of the conservation of mass, energy, and momentum, complemented by the second law of thermodynamics. The remarkably short free-fall timescale ensures that hydrostatic balance is a superb approximation for almost all stars in almost all circumstances, greatly simplifying the general problem. The transport and generation of energy are also closely matched in stars, so stars are near-equilibrium structures. As a consequence, stellar structure, or the solution of the equations at an instant in time, is a tractable problem. It is only over oceans of time that the graceful arc of stellar evolution proceeds, driven by the slow but irreversible transformation of light species into heavy ones by nuclear reactions.

Email: pinsonneault.1@osu.edu

What might seem at first to be an impossible problem can then be cast in a straightforward form. It simply requires tools from the full arsenal of modern physics—quantum mechanics, thermodynamics, statistical mechanics, electricity and magnetism, nuclear physics, and fluid mechanics. The transport of radiation through matter can be captured by an effective mean free path of a photon, expressed as an opacity. Nuclear reaction rates and the equation of state can similarly be defined as functions of the local temperature, density and composition. Even difficult physical problems, such as the transport of energy by turbulent convective motions, can be collapsed into a local prescription. Solving these equations, of course, is not the same as setting them up.

Analytic work has a distinguished history in stellar theory, but the limitations of these tools became very clear at an early stage. The microphysics are clearly posed questions, but not ones with answers in convenient analytical forms. The correct solution to the equations need not be easy to compute. The dawn of the computer age in the 1950s and 1960s led directly to the first

generation of numerical stellar structure codes. Astrophysicists were able to use compact codes and kilobytes of memory, deployed with punch cards, to decode the life cycle of stars. A number of independent codes, still in use today, were developed. In this paper we present the public release of the Yale Rotating Stellar Evolution Code (YREC), which had its origin in this era.

Despite the variety of stellar codes that were developed, only a few major codes are public: the MESA code (B. Paxton et al. 2011, 2013, 2015, 2018, 2019; A. S. Jermyn et al. 2023), CESAM (P. Morel 1997; P. Morel & Y. Lebreton 2008; L. Manchon et al. 2025), the Eggleton STARS code (P. P. Eggleton 1972; P. P. Eggleton et al. 2011), and (recently) the ESTER code (R. White et al. 2025). MESA has been particularly high in impact and is widely used. This is not because other authors were unwilling to share and compare results: a good example of detailed code comparisons within the community can be found in J. N. Bahcall & M. H. Pinsonneault (1992) for solar neutrinos. Instead, the scarcity of public tools reflects the mismatch between the era in which those codes were developed and the time frame when the open source movement blossomed. Proprietary codes are also typically not built with the needs of a large and diverse user base in mind. Some are carefully curated gems. But walled gardens can also be filled with weeds. Releasing such codes therefore requires careful boundaries, which we lay out below.

Users are free to modify the YREC source code, but the code as written was not designed with modular user inputs in mind. In future releases, we will prioritize making the code easier to alter. YREC will also do what you tell it to do, without second-guessing. This means that the code can and will crash if you assign, for example, a super-critical rotation rate. The documentation does discuss known failure modes and how to diagnose them. Many of the inputs for YREC were designed for specific science cases, and the code is not guaranteed to perform well outside of that domain, where the assumptions break down. Finally, YREC is written in glorious Fortran, F77 to be specific, but it is F90 compatible. With these cautions in mind, the modern open source movement has proven to be fruitful and powerful, and we believe that YREC will be a useful tool for research and education.

1.1. *Why YREC?*

YREC is a stellar evolution code optimized for the study of hydrostatic stars from the pre-main sequence through to the helium burning phase of evolution. YREC can also be used to study giant planets and white dwarfs. It is fast, precise, and accurate in the domains where it functions. YREC has been extensively used for both stellar population and stellar physics studies, including high-precision solar models (C. Basinger et al. 2024).

YREC uses modern input microphysics (equation of state, energy transport, energy generation, surface boundary conditions) with a variety of user-configurable options. YREC is a relaxation code, and as written, it requires a starting model to run. These models can then be relaxed to a different set of initial conditions. It has the option to include important physical processes not considered in classical stellar models. For example, YREC includes the structural effects of rotation using a pseudo-2D approximation. The treatment of angular momentum evolution is flexible, including star-disk coupling, rotationally induced mixing, internal angular momentum transport, angular momentum loss from magnetized winds, and the ability to model different rotation profiles in convective regions. It also includes gravitational settling and the structural effects of starspots and their associated magnetic fields.

1.2. *Outline of the Paper*

We describe the input physics and numerical approach of the YREC code in Section 2. We also note similarities and differences with the public MESA code.

Stellar evolution codes adopt a set of input physics and run a single initial condition (mass, composition, rotation rate, and age) to a prescribed end state. A test suite of runs highlights the various capabilities of the code. The test suite is also an essential tool for code development, as it allows users to check that code changes have not caused unanticipated problems. In Section 3, we present our test suite, sample applications, and comparisons with other codes in the literature. Scripts and helper codes are discussed in Section 4. In the discussion and summary, we lay out the current domains where the code is recommended for usage and future plans for upgrades. Details of how to access the code repository, input files, documentation, and other resources are given in the appendices.

2. PHYSICS AND NUMERICS OF THE YREC CODE

YREC has been in use as a stellar structure and evolution code for more than 60 years. The code was first described in R. B. Larson & P. R. Demarque (1964), and subsequently updated and expanded as described by M. J. Prather (1976). The most recent summaries of the standard model physics are P. Demarque et al. (2008), which included the addition of output for computing pulsations; and J. L. van Saders & M. H. Pinsonneault (2012), which includes a series of updates to

the input physics. Microscopic diffusion was added by J. N. Bahcall & M. H. Pinsonneault (1992); J. N. Bahcall et al. (1995), along with a complete overhaul of the treatment of nuclear reaction rates. We also note that the DSEP code, used for a popular set of isochrones (A. Dotter et al. 2008), was based on YREC and shares much of its structure.

M. H. Pinsonneault et al. (1989) added rotation and rotationally induced mixing. There were then a series of updates to the angular momentum evolution model. This included updates to the angular momentum loss model (A. Krishnamurthi et al. 1997; J. L. van Saders & M. H. Pinsonneault 2013), updates to the rotationally induced mixing model (M. H. Pinsonneault et al. 2002; G. Somers & M. H. Pinsonneault 2016), and the inclusion of the structural effects of starspots (G. Somers & M. H. Pinsonneault 2015; G. Somers et al. 2020).

2.1. *Overall Code Structure and Repositories*

YREC is a relaxation code that can be run in 2 modes. An evolving run takes an input model and runs to a specified end state. A rescaling model uses a prior solution that is close to the desired conditions and perturbs it. Rescaling can be done in the fully convective pre-main sequence (pre-MS), the zero-age main sequence (ZAMS), or the zero-age core He burning state (ZAHB). For the pre-MS, there is no time independent solution of the structure equations. A pre-MS model must therefore evolve in time when rescaling. On the ZAMS and the ZAHB, the code rescales with a zero time step. We note that many useful stop conditions are not enabled for rescaling runs of any form. If rescaling is desired, it is therefore recommended to do runs in two stages: an initial rescaling run followed by an evolving run with a stop condition.

Running YREC requires the user to download the source code, and then to use a fortran compiler to compile the subroutines and link them. The code, documentation, and instructions on how to run the code can be found at https://github.com/yreclab/yrec. [11] YREC also requires libraries of input tables and starting models, which are available on the repository under the `input` and `startmodels` directories, respectively. These data are also available in the YREC Zenodo repository (https://zenodo.org/communities/yrecmodels.)

The code uses two namelists that include global run instructions, file input and output paths, numerical controls, and an extensive list of user-controllable options for the input physics. Running the code also requires a converged starting model and tabulated input data for the microphysics. Suites of models can be run using scripts, and the output can be adopted to investigate a range of astrophysical problems.

The github repository contains a `testsuite` directory that includes the namelists and output for the test suite, which can also be found on the Zenodo repository. The Zenodo repository also has sections for scripts, classroom exercises, a base model grid covering a range of masses and metallicities, and sample cases used to generate all of the figures and tables in this paper.

2.2. *Numerical Methods*

YREC solves the equations of stellar structure using the Henyey technique. In this method, the four coupled equations of stellar structure are converted into finite difference equations. A valid solution of the structure equations at a given time is then perturbed to account for composition changes and gravitational potential energy release. The start-of-timestep structure determines the locations of convective regions. It is also used to compute nuclear reaction rates, which are then used to infer composition changes in an implicit scheme. Composition changes drive changes in the structure variables, which are inferred, along with their derivatives, in microphysics routines. As an example, the conversion of hydrogen to helium lowers the number of free particles per gram, reducing pressure at fixed temperature and density. The model will therefore require compensating changes in temperature and density to restore hydrostatic equilibrium, which in turn cause feedback for other state variables. The code then solves iteratively for the required changes in the structure variables until it reaches a user-specified set of numerical precision criteria.

Stellar evolution is a two point boundary value problem. At the center, YREC uses a Taylor series expansion to map the inner shell to the central values. This approach is widely used in stellar codes, as it avoids the divergence of logarithmic structure variables at zero mass and radius. Close to the surface the assumption of a short photon mean free path becomes invalid. All stellar evolution codes therefore impose a solution of the radiative transfer equations as an atmospheric boundary condition. This can either be a direct integration, such as a gray atmosphere, or a table look-up from a grid of full model atmosphere calculations.

An important difference between YREC and MESA is the treatment of the outer boundary condition below the atmosphere. For a two point boundary value problem, the governing equations must be integrated both inwards from the surface and outwards from the center.

[11] The release paper version of the code can be found in the https://github.com/yreclab/yrec/tree/yrec_2026_release branch.

The two solutions and their derivatives are matched at a fitting point, and the location of the fitting point is a numerical choice that reflects tradeoffs. The base of the atmosphere is a possible location for the fitting point, but this is not required. A deeper fitting point is usually more stable, but the inwards integration usually makes simplifying assumptions. For example, in YREC the luminosity is constant in the envelope integration. MESA models typically adopt a fitting point at the base of the atmosphere.[12]

YREC adopts an outer fitting point well below the photosphere. At this fitting point, the structure variables P, T, R and L from the interior solution and their derivatives must be matched to the corresponding values from a downwards envelope integration. The code also needs information on how these envelope integration properties change as a function of L and $T_{\rm eff}$. YREC therefore integrates three solutions close to the model L and $T_{\rm eff}$, and interpolates between them in this envelope triangle as the model solution evolves. If the solution moves outside of the triangle the code will flip it to re-center the triangle around the current model location.

Although very little mass is usually in this envelope, typically one part in 10^4 or less, this is a regime where the pressure and density are dropping very rapidly in stars. As a result, not retaining the outer layer structure yields a significant increase in computing speed. The structure equations are also more stable with a deeper fitting point. The cost is that information on the detailed outer layer physics is not stored in the interiors model. YREC therefore has options to post-process an envelope solution, for both informational purposes and for practical applications such as computing oscillation frequencies.

YREC solves the structure equations in up to four iterative loops, two of which are mandatory. In the first loop, the L and $T_{\rm eff}$ at which the surface boundary conditions (SBCs) are evaluated is held fixed. In the second, main loop, the location of the "envelope triangle" is allowed to change dynamically. The third, optional, level of iteration allows the code to use the end-of-timestep structure to compute the composition change, rather than the start-of-timestep structure. Microscopic diffusion and mixing from rotation are computed after the model structure has converged in a separate step, so the converged structure in the first through third level of iteration does not account for all of the changes in the composition profile or for changes in the structural effects of rotation during a timestep. The fourth level of iteration allows the code to iterate between the rotation, diffusion, and structure solutions. The overall program flow is illustrated in Figure 1.

[12] For some tabulated model atmospheres, there is a choice in MESA to adopt a deeper fitting point, up to $\tau = 100$, but this is still relatively close to the surface.

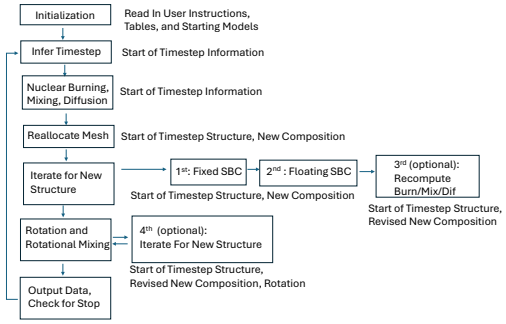

Figure 1. A schematic illustration of the steps in the YREC code.

For each model, the code begins by reallocating the spatial mesh points. Mesh points are allocated based on the requirement that the state variables not change from one point to the next by more than a critical value. In the deep core, mass is usually the limiting factor. Luminosity governs the spacings in nuclear burning regions, while pressure is the governing ingredient for envelopes. Points with composition or rotation rate discontinuities can be flagged to prevent rezoning across them.

At the end of the timestep, the code computes a series of limits for the next timestep; the most stringent relevant criterion is used. The code uses the instantaneous rate of nuclear reactions to set a limit on the timestep for the next model. It also computes the rate of change of the structure variables, from start to end of timestep, to set another limit on the timestep; this is adopted if shorter than the nuclear burning timestep. This second criterion is essential for pre-MS models, for which the energy source is gravitational potential energy. For the main sequence and beyond, this alternative timestep criterion is disabled; otherwise it induces very small time steps in the shell hydrogen burning stage.

The code then writes data to output files as requested, and checks whether a stop condition has been achieved. If the run continues, the code then returns to the start of the loop described above.

2.3. *Classical Stellar Model Physics*

Stellar evolution codes are designed for exploration. It is therefore natural that YREC has many options for how to treat important physical processes. The YREC code has evolved over time, and it has retained backwards compatibility with historical choices for the input physics. In this paper we do not dwell on these charm-

ing relics, turning our attention instead to the options recommended for use today.

2.3.1. *The Equation of State*

The equation of state relates pressure to temperature, density, and composition, and it also encodes extensive thermodynamic information. As such, it is used for a wide range of purposes in stellar evolution codes. The current version of YREC has three different choices for an equation of state.

The recommended equation of state is the 2005 update of the OPAL EoS (F. J. Rogers & A. Nayfonov 2002). We note that this EoS has a fixed heavy element content. Users can generate and use input tables with different metal contents, but the metallicity is fixed for a given run of the code. In the high-density and cool domain, the OPAL table does not cover some of the phase space of interest. The D. Saumon et al. (1995), or SCV, equation of state is a good alternative here.

The default equation of state, which we refer to as the Yale EoS (M. J. Prather 1976), is available across the full phase space. It includes radiation pressure and degeneracy pressure. The latter is solved for using tabulated solutions to the Fermi integrals, and it includes relativistic corrections. For the gas pressure, the code solves the Saha equations for hydrogen, helium, and metals at low temperature. Above a user-specified threshold, it assumes a fully ionized plasma. The Debye-Huckle correction is also included, to account for electrostatic effects. This EoS is extremely fast, with smooth analytic derivatives, and we recommend its usage for educational applications. However, the code assumptions break down for cool dense plasmas, especially for lower main sequence stars. Most published applications of YREC therefore use tabulated EoS data that includes a more sophisticated treatment, particularly for collective plasma effects.

YREC has a tie-breaker algorithm. It will use the OPAL EoS when possible if it is selected as an option. Outside the table domain, it then checks if the SCV EoS is available (if specified); the Yale EoS is the backup if neither is available. In all cases the code ramps between solutions close to table edges. We caution that thermodynamic consistency is not in general preserved in these cases, which can cause numerical issues in model domains close to table edges. The implementation is different in detail than that in MESA, but both codes use a broadly similar approach.

2.3.2. *Radiative and Conductive Energy Transport*

The opacity of matter to light depends, in complicated ways, on the composition, density, and temperature.

The calculation of opacities requires several steps. It is necessary i) to determine the ionic state of the absorbing medium as well as the population of the atomic levels of its constituents, ii) to calculate the absorbing coefficients for different processes (photo-excitation, photoionization), and iii) to include plasma effects (the ions cannot be considered as isolated). Accuracy versus completeness is still at play nowadays as it was 30 years ago. The atomic systems considered today may be much larger, thanks to the computer power available, but the calculations still require approximations and compromises since they cannot be infinite.

In stellar interiors, local thermal equilibrium (LTE) is a very good approximation since the mean free path of a photons is very small. In this case, we can use a diffusion approximation for energy transport, with a diffusion coefficient that is the product of c/3 and an effective photon mean free path. The latter can be well approximated by a weighted average of the monochromatic opacity of the mixture of elements present in the stellar plasma. This Rosseland mean opacity is defined, per gram, by

$$\frac{1}{\kappa} = \int \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu \qquad (1)$$

with B_ν being the Planck function characterizing the energy distribution of blackbody radiation and κ_ν being the monochromatic opacity of the mixture.

$$\kappa_\nu = \sum_{s=species} f_\nu(s)(\kappa^s_{BB}(s) + \kappa^s_{BF}(s)) + \kappa_{FF} \qquad (2)$$

with f_s being the fractions of species s, and where BB stands for bound-bound transitions (photo excitation processes), BF stands for bound-free transitions (photo ionization processes) and FF stands for free-free transitions (inverse Bremsstrahlung processes). For the atomic opacities, the main providers are the OPAL project (C. A. Iglesias & F. J. Rogers 1996) and the Opacity Project (N. R. Badnell et al. 2005), hereafter OP. For these, each of the processes (BB and BF) have to be calculated for each ion of each species and combined with a detailed equation of state. If we consider the opacities the OP distributes over its website (https://cdsweb.u-strasbg.fr/topbase/testop/home.html) this corresponds to 241 ions for the 17 species (H to Ni). Since the Rosseland mean opacity is a harmonic mean, the details of the lines and resonances as well as the plasma effects (mainly their broadening) are crucial and can lead to large differences depending on the resolution of the photo-ionization cross sections, line transitions, and the broadening theory.

Molecular opacities require a molecular EoS, which is far more complex than the atomic case. The same holds

true for the molecular energy levels and transitions relative to the atomic ones. However, for astrophysical conditions molecules are rare above temperatures of order 6000 K, and hydrogen ionization is the dominant opacity source in any case at temperatures of order 10^4 K. YREC, like other stellar evolution codes, therefore uses dedicated molecular opacity tables computed only for low temperatures.

YREC uses three families of opacity tables: conductive, atomic, and molecular. Our conductive opacities are taken from S. Cassisi et al. (2007). We use molecular opacities from J. W. Ferguson et al. (2005) below log T = 4.0, atomic opacities above log T = 4.1, and a ramp between the two for intermediate temperatures. There are three families of atomic opacities that the code can use—OPAL tables for carbon and oxygen-rich mixtures (C. A. Iglesias & F. J. Rogers 1993), OPAL tables for H, He, and metal mixtures (C. A. Iglesias & F. J. Rogers 1996) and OP tables for H, He, and metal mixtures (N. R. Badnell et al. 2005). The first set of opacities are used in helium-burning regions, where the heavy element mixture changes significantly. For both OP and OPAL, the heavy element mixture is held fixed, and the opacity is interpolated as a function of X, Z, ρ and T. These tables, with these assumptions, are widely used in stellar models, and there are web tools available to build tables for user-specified heavy element mixtures (for the Opacity Project, https://opacity-cs.obspm.fr/opacity/index.html). YREC uses tables constructed for a variety of heavy element mixtures, typically tied to different assumed solar abundance measurements. Our base heavy element mixture for this paper is N. Grevesse & A. J. Sauval (1998), which produces solar models in good agreement with helioseismology. We also include tables based on the published solar mixtures of E. Magg et al. (2022). M. Asplund et al. (2021), and K. Lodders et al. (2025).

2.3.3. *Convection and Convective Boundary Layer Physics*

YREC uses the mixing length theory (MLT) of convection (E. Böhm-Vitense 1958) to infer the actual temperature gradient in regimes that are unstable against convection. This requires knowledge of the radiative and adiabatic temperature gradients. The code considers the Schwarzschild criterion for instability, not the Ledoux criterion. See P. Demarque et al. (2008) for a fuller discussion of this point.

The effective convective flux can be modified in models that include starspots (section 2.5 below). In convective cores the specific heat is quite high and there is an option to skip MLT calculations; in this case the adopted temperature gradient is the adiabatic one.

YREC adopts a step overshoot model, expressed as a number of pressure scale heights at the boundary. Overshoot can be enabled separately for cores, envelopes, and intermediate regimes.

The code has a second, recommended control, which limits the overshoot domain to a fraction of the physical size of the core (J.-H. Woo & P. Demarque 2001). If adopted, the code selects the smaller of these two measures. Checking against the size of the core is important to properly model small convective cores, because the pressure scale height formally diverges at zero radius and becomes very large for small cores.

Overshoot as treated here is more properly overmixing, as the radiative temperature gradient is retained. There is a code option to enforce adiabatic overshoot.

YREC also considers semi-convection, using the V. Castellani et al. (1971) approach. We are in the process of updating this to a newer approach, but do not yet have a publicly supported alternative.

2.3.4. *Nuclear Reaction Rates*

Nuclear reaction rates are important for both energy generation and composition change. YREC is designed for the hydrogen and helium burning stages of evolution. It also includes light element burning (^{6}Li, ^{7}Li, ^{9}Be) in the trace element approximation, where the contribution to energy generation is neglected. This is an excellent approximation for these rare species. The code includes full non-equilibrium burning for ^{1}H, ^{3}He, ^{4}He, ^{12}C, ^{13}C, ^{14}N, ^{16}O, and ^{18}O.

Deuterium burning is not explicitly traced in the pp chain; however, for young stars non-equilibrium burning of ^{2}H into ^{3}He is important. The code therefore does track the burning, and associated energy generation, for pre-MS deuterium. It does not, however, treat the pp reaction as a source, so deuterium in the models simply goes to zero, typically early in the pre-MS phase.

For dense nuclear reaction grids, most stellar evolution codes use either analytic fits to reaction rates or a table look-up. The NACRE (C. Angulo et al. 1999) and NACRE-2 (Y. Xu et al. 2013) compilations of rates are commonly used. By default, MESA nuclear reaction rates are from JINA REACLIB (R. H. Cyburt et al. 2010) and NACRE (C. Angulo et al. 1999). The YREC code uses this approach for He burning, adopting the energy yields and rates of the G. R. Caughlan & W. A. Fowler (1988) compilation for triple α, $^{12}\mathrm{C}(\alpha,\gamma)^{16}\mathrm{O}$, $^{13}\mathrm{C}(\alpha,n)^{16}\mathrm{O}$, and $^{14}\mathrm{N}(\alpha,\gamma)^{18}\mathrm{O}$.

From the 1990s onwards, YREC has adopted the J. N. Bahcall & M. H. Pinsonneault (1992) approach for hydrogen burning, appropriate for reactions where resonances do not play a strong role. In this approach, we

use the charges, masses, the cross section extrapolated to zero energy, S_0, and its first and second derivatives, to directly compute reaction rates. YREC includes electron screening of reactions, which includes the intermediate and strong screening cases. However, for general use the default code settings recommend the usage of weak screening (E. E. Salpeter 1954). The adopted cross sections are taken from E. G. Adelberger et al. (2011), sometimes referred to as Solar Fusion II. The recently published Solar Fusion III rates (B. Acharya et al. 2025) are an alternative, and a case using these rates is included in our test suite.

Solar neutrino fluxes are computed for all pp and CNO neutrino sources, adopting equilibrium abundances for all isotopes not explicitly tracked. Neutrino losses are subtracted from the effective energy yield of nuclear reactions. YREC uses the N. Itoh et al. (1996) rates for continuum neutrino cooling. These are important processes for evolved stars, and they can lead to off-center He ignition.

2.3.5. *Mixing and Composition Change*

YREC solves for non-equilibrium abundance changes due to nuclear reactions using an implicit scheme. For convective regions, instantaneous mixing is assumed, and the reaction rates are therefore mass-averaged across the convection zone. The nuclear reaction rates are computed using start-of-timestep values for the structure variables. There are options to refine these estimates using end-of-timestep values instead. Microscopic diffusion and rotationally induced mixing are treated in separate, sequential steps.

We note that there is a diversity of approaches in the literature in how to treat the interaction between composition change and the evolution of the stellar structure. At one limit, the P. P. Eggleton (1972) method simultaneously solves for composition change and the change in the structural variables. The prime author of the MESA code built his first evolution code using Eggleton as a basis (B. Paxton 2004), and the MESA code is built using the same principle. The YREC sequential approach is more common, and faster, but it is by no means universal. The same comment applies to the choice of where to evaluate the nuclear reaction rates in the timestep, or how to treat a mixed convection zone whose depth varies during the timestep.

In practice, the numerical controls are typically adjusted such that these different approximations yield consistent results within the domains where they are valid. As an example of a case where these differences matter, consider the case of instantaneous mixing in convection zones. This is usually an excellent approximation, but it breaks down in interesting domains, such as AGB shell flashes.

2.3.6. *Boundary Conditions*

The boundary conditions are evaluated at three pairs of (L, $T_{\rm eff}$) close to the model values (see section 2.2). The structure variables (P, T, R, L) of the interior solution and their derivatives are matched to the envelope solution, and the three different solutions in turn provides derivatives as a function of (L, $T_{\rm eff}$).

For the atmosphere, the code assumes constant M, R, and L, and integrates P($T(\tau)$) as a function of optical depth τ to a fitting point. For the default option, a gray atmosphere, this is $\tau = 2/3$. Other fitting points can be chosen. The integration starts at an arbitrary low density and requires a $T(\tau)$ relationship. Users can also use a table look-up from libraries of model atmospheres. For the existing tables, an optical depth of 2/3 is also adopted. The available options are R. Kurucz (1993), F. Castelli & R. L. Kurucz (2003), and F. Allard & P. H. Hauschildt (1995). The Allard tables are available for solar metallicity only, while the other two are available for a range of metallicities. If the models go outside of the table grid on the hot edge, the table look-up is disabled and the code defaults to an Eddington gray atmosphere. The run will terminate if the code falls outside of the table at high L or low $T_{\rm eff}$.

The atmosphere is then used as a starting point for an envelope integration, which assumes a constant L. The remaining three structure equations are integrated down to a fitting point using a Bulirsch-Stoer integration scheme with an adaptive step size. The physics used in the envelope integration are the same as those used in the interiors model. Only the lower boundary condition of the envelope integration is retained, and these results are output in the last and stored model headers.

2.4. *Microscopic Diffusion*

In a traditional stellar model, radiative regions are strictly stable. In the real world, this is not true. Heavy species sink relative to light ones due to gravitational settling and thermal diffusion. Large atoms can be driven up relative to nuclei and small atoms by radiation pressure, sometimes referred to as radiative levitation. Collectively, we refer to these mechanisms as microscopic diffusion. The timescale for diffusion is shortest in the outer layers of stars, where the density is low, and it becomes shorter in stars with thin surface convection zones. Mass loss can counteract diffusion, and diffusion therefore becomes less important on the upper main sequence with the onset of strong radiation pressure-driven winds. Microscopic diffusion is therefore particularly important in intermediate mass stars,

between 1.3 and 9 $M_\odot$ at solar metallicity. Even in the Sun, however, it impacts surface abundances relative to birth values at the 10% level.

YREC models can include gravitational settling and thermal diffusion. We follow the approach of J. M. Burgers (1969) as implemented by A. A. Thoul et al. (1994). The code includes microscopic diffusion for metals as a group, helium, and the light species ^{6}Li, ^{7}Li and ^{9}Be. Metals are assumed to all settle at the same rate as fully ionized iron. This is consistent with the approximations used in the opacity tables, which assume a fixed heavy element mixture.

In the limit of very efficient microscopic diffusion, stars will develop layer cake structures in their outer layers; pure H at the top, with a layer of pure He below, followed by zones with progressively heavier species. This causes difficulties for codes in thin surface convection zones due to interpolation issues in opacity and equation of state tables. The opacity for zero H, for example, is quite different from that for even a small trace contribution—see E. Farag et al. (2024) for a discussion of this point. YREC disables microscopic diffusion calculations when the central hydrogen drops below a user-specified threshold (default 0.001), or if the outer point in the model is radiative. In practice, this means that one can choose an outer fitting point deep enough to avoid pathological solutions on the main sequence, which manifest as the code failing due to interpolation errors.

2.5. *The Structural Effects of Star Spots*

Magnetic fields are remarkably hard to model, and thus usually neglected in stellar models. At some level, this omission is reasonable; it is unlikely, for example, that there are core magnetic fields strong enough to compete with gas or radiation pressure, at least in sensible stars. However, there are contexts where magnetic fields cannot be ignored, despite the fondest wishes of theorists. Angular momentum loss by magnetized winds, and magnetic instabilities for rotating stars, can play important roles in models of rotating stars. We discuss magnetic fields in these contexts in Section 2.6, where we address rotation.

Another regime, included in YREC, is the direct impact of star spots on the structure and evolution of cool stars. Stars with modest rotation rates and deep surface convection zones have dynamo-generated magnetic fields. The dynamo produces cool star spots with equipartition-strength magnetic fields. As a consequence, the ambient temperature in un-spotted regions is not the same at the surface averaged effective temperature. Convection is also observed to be inhibited in sunspots, and the same should apply to star spots. As a consequence, star spots can be consequential for stellar structure, as originally pointed out in H. C. Spruit (1982).

Although the solar sunspot filling factor is small, of order 0.1 percent, there are stars where spots are far more important. In active main sequence stars, the filling factor of spots can reach levels of order 25 percent; much higher filling factors, 70 percent or more, are seen in pre-main sequence stars (L. Cao & K. G. Stassun 2025).

YREC accounts for both flux blocking and a modified surface boundary condition (G. Somers & M. H. Pinsonneault 2015). Star spots can induce radius inflation and impact pre-MS lithium depletion. In fully convective stars the luminosity can also be substantively altered. The star spot filling factor and temperature contrast is specified at the start of the run and held fixed, which allows users to test for the structural effects of star spots.

2.6. *Rotation*

Rotation is a fundamental stellar property. Stars are born with a wide range of rotation rates. As a consequence stars of the same birth mass, composition, and age can have very different properties. Rotation changes stellar structure, altering the surface appearance and changing fundamental properties such as lifetimes. It induces mixing in otherwise stable regions, which can be consequential for chemical evolution. Rotation can also be an age indicator, making it a key diagnostic tool for stellar populations.

Rotation is also challenging to model, as it is intertwined with frontier challenges in stellar theory. The interplay of rotation and magneto-convection generates stellar dynamos. Magnetized winds drive angular momentum loss in some regimes, but not others. In radiative regions magnetic fields, waves, and hydrodynamic mechanisms can transport angular momentum, and induce mixing of chemical species, to varying degrees. In convective regions the overturn timescale is short compared to the evolutionary timescale, so a rotation profile will be enforced by the interplay of rotation and turbulence. The natural timescales of both rotation and turbulence vary by orders of magnitude, however, so there is no reason to expect all stars to have some universal profile. It is thus unsurprising that we lack a consensus model for stellar rotation. YREC includes a variety of options, which includes the freedom to explore quite different underlying physical models for the angular momentum evolution of stars.

There is a global correspondence between the MESA and YREC codes in terms of the treatment of rotation. The MESA method is based on the approach of A. Heger

et al. (2000), which in turn used M. H. Pinsonneault et al. (1989) as a starting point. Both papers, in turn, broadly follow the approach of A. S. Endal & S. Sofia (1976). In detail, however, the different codes have distinctive approaches. Due to the contributions of many people, over the years these models have evolved and mutated. We document this natural process of rejuvenation, particularly in the treatment of angular momentum transport, loss, and mixing, here.

2.6.1. *The Initial Conditions for Rotation*

The natural initial condition for rotation is in the star formation phase, which in turn divides into two phases. Stellar modeling typically begins at the end of the hydrodynamic collapse phase. At this phase the star is assumed to have reached its final mass and hydrostatic equilibrium is a good approximation. In YREC, this would correspond to the deuterium burning birthline. Spherical models can be converted into uniformly rotating ones at a user-specified rate for any model, but the birthline is a physically motivated state, at least for low mass stars.

Starting models much further up the Hayashi track exist, but these do not correspond to real stars. This is an important point for initializing rotation. If a user initializes rotation in an artificially large starting model, angular momentum conservation will transform a reasonable sounding guess into an unreasonable outcome.

During the hydrostatic pre-MS, star-disk coupling can occur (A. Koenigl 1991; F. Shu et al. 1994). It is also necessary to explain the evolution of rotation in young stars. As a result, YREC also includes the option to enforce star-disk coupling for a user-specified timescale at a user-specified rate.

For massive stars, it is less clear that the deuterium burning birthline is a true representation of the starting state (F. Palla & S. W. Stahler 1991). Above a critical mass, of order 5 to 8 solar masses, accreting models develop radiative cores and are born close to the main sequence. For higher mass models, an initial rotation state on the main sequence may therefore be closer to being physically reasonable.

2.6.2. *The Structural Effects of Rotation*

Rotation induces a departure from spherical symmetry which can be significant. The net effect is a reduction in the effective gravitational force, which makes rotating stars behave for most purposes as if they had a lower effective mass. However, in detail the impact depends on the angular momentum distribution. The general case would require a two or three dimensional model. Fortunately, there are reasonable families of models that can be captured using a pseudo-2D angular velocity distribution. In this case, the structural effects of rotation can be computed by measuring the shape of equipotential surfaces as a function of mass. The impact of rotation on the structure can then be included with correction terms applied to the spherically symmetric equations of stellar structure.

YREC uses a modified version of the R. Kippenhahn & H. C. Thomas (1970) technique. See M. H. Pinsonneault et al. (1989) for a discussion of the differences. This method assumes uniform rotation on equipotential surfaces. The code then solves for the shape of these surfaces, including contributions from the quadrupole term in the potential, and solves a modified set of stellar structure equations.

2.6.3. *Rotation in Convective Regions*

YREC assumes that a rotation law is enforced in convective regions on a short timescale. The code permits rotation profiles of the form $\omega \sim R^{\alpha}, -2 \leq \alpha \leq 0$. There are also options to couple the rotation profile to adjacent radiative regions - for example, to match the rate at the base of a convective region to the rate in the radiative zone below it. These options have a significant impact for the rotational evolution of post-main sequence stars, where observational data suggests that core rotation rates can be tens to hundreds of times faster than the envelope rotation rates (B. Mosser et al. 2012; C. Gehan et al. 2018, e.g.), with important implications for internal angular momentum transport (J. Tayar & M. H. Pinsonneault 2013; J. Fuller et al. 2019). Both the surface and internal rotation of core He burning stars are important tests of angular momentum loss (J. Tayar & M. H. Pinsonneault 2018). Rotation and magnetism may also affect mass loss on the red giant branch (Y. Li 2025).

YREC does not include differential rotation with latitude. For sunlike stars, the average rotation with depth is nearly constant, even if there is latitudinal differential rotation. We therefore recommend that users impose rigid rotation for pre-MS and MS cases. However, they should keep in mind that imposing a uniform rotation rate in convection zones may be a poor approximation for luminous giants. It is difficult to explain the survival of rotation in blue core He-burning stars without strong differential rotation in luminous red giants (M. H. Pinsonneault et al. 1991; A. Sills & M. H. Pinsonneault 2000). This differential rotation can also impact the interpretation of red giant core rotation rates (Y. Kissin & C. Thompson 2015) and core He burning stars (J. Tayar & M. H. Pinsonneault 2018).

2.6.4. *Angular Momentum Loss From Magnetized Winds*

Users may choose between braking laws that describe the angular momentum loss via magnetized winds. Magnetic braking is the consequence of the interaction of mass loss and magnetic fields. As both are frontier fields in stellar physics, the description of their interaction is unsurprisingly burdened with many assumptions.

Broadly, the rate of angular momentum (AM) loss is set by a combination of the mass loss rate ($\dot{M}$), the rotation rate (Ω), and the effective magnetic lever arm, parameterized as the Alfvén radius (R_A). The Alfvén radius itself generally depends on $\dot{M}$, stellar magnetic field (B), and wind driving physics. The overall field configuration (dipole, quadropole, etc.) affects how strongly the torque scales with the magnetic lever arm. YREC has three built-in options for the braking law that differ in their assumptions about each of these ingredients.

The first is a modified version of the S. D. Kawaler (1988) prescription proposed by A. Krishnamurthi et al. (1997). The Kawaler braking law assumes that a dipole magnetic field scales as $BR^2 \sim \Omega$. The torque for slow rotators therefore scales as Ω^3. A. Krishnamurthi et al. (1997) added a Rossby-scaled magnetic saturation threshold to the standard form of the Kawaler law, beyond which the AM loss scales more weakly with rotation rate. This saturation is a ubiquitous feature of magnetic proxies in stars: they take maximal but constant values at sufficiently rapid rotation and low Rossby number Ro (the rotation period divided by the convective overturn timescale).

The second form is the loss law from J. L. van Saders & M. H. Pinsonneault (2013), which uses a parametrized model of the steady-state MHD wind solutions from S. Matt & R. E. Pudritz (2008) and S. P. Matt et al. (2012). The magnetic field is scaled assuming a linear (Rossby-scaled) dynamo and pressure equipartition at the photosphere: $B \sim Ro^2 P_{phot}^{1/2}$. Mass loss is scaled as $\dot{M} \sim L_X \sim (\omega\tau_{cz})^2$ following the observed correlation of $\dot{M}$ with X-ray luminosity (B. E. Wood et al. 2005) and scaling of the X-ray luminosity with Rossby number (N. Pizzolato et al. 2003). The liberal use of Rossby scalings causes the torque to be significantly weaker in stars with thin convective envelopes than that in a Kawaler prescription, which better matches the actual rotational behavior of stars near the Kraft break.

Finally, users can enable a "custom" mode where they can manually set the exponents that describe the dynamo scalings, Rossby scalings, and effective magnetic index.

Although the braking utilizes an inferred $\dot{M}$, this $\dot{M}$ is not self-consistently incorporated into the structural evolution. All prescriptions available in YREC allow the user to specify a disk-locking rotation period and duration. Magnetic saturation of the torque is accounted for, with the saturation threshold set by the user. Because the field strengths, mass loss rates, and overall torque magnitudes are not known from first principles, practical use always involves the adjustment of a scale factor to calibrate the magnitude of the torque against benchmark observations. Sample cases of such a calibration are discussed in Section 3.4.4.

2.6.5. *Angular Momentum Transport and Rotationally Induced Mixing in Radiative Regions*

The original YREC code included angular momentum transport and the associated mixing via hydrodynamic mechanisms. YREC solves diffusion equations for the transport of angular momentum and the associated rotationally induced mixing. Instantaneous mixing is assumed in convective regions. Angular momentum loss is induced in the surface convection zone as a surface boundary condition.

The secular mechanisms included are meridional circulation, the secular shear instability, and the Goldreich-Schubert-Fricke instability. See G. Somers & M. H. Pinsonneault (2016) for a recent summary of the assumed input physics. This differs from the M. H. Pinsonneault et al. (1989) approach in that the characteristic length scale for the instabilities is the radius, rather than the velocity scale height. The meridional circulation treatment is also modeled on the J. P. Zahn (1992) approach, neglecting terms of order the second or higher derivative of the angular velocity but including the quadrupole component of the potential. The hyper-diffusion terms in the J. P. Zahn (1992) method require a very different numerical treatment than the one used here. This expression is broadly similar to others in the literature. We include the reduction in the effective meridional circulation velocity induced by mean molecular weight (μ) gradients following the approach of A. Maeder & J.-P. Zahn (1998). We also adopt the A. Maeder (1997) approach for the secular shear instability. The code permits different treatments of the inhibiting effect of mean molecular weight gradients on angular momentum transport and the associated mixing.

A central result from M. H. Pinsonneault et al. (1989) was that the efficiency of angular momentum transport must be much higher than that of the associated mixing. This was placed on a physical basis by B. Chaboyer & J. P. Zahn (1992), who argued that the turbulence induced by rotation is much more effective in the horizontal direction than in the vertical direction. The former can exchange angular momentum, while mixing is only induced in the vertical direction. YREC treats this as

a free parameter, f_c, which can be used to adjust the efficiency of rotationally induced mixing. This differs from the J. P. Zahn (1992) approach, where this is a fixed parameter of the model. In our view, the assumptions embedded in the approach make a free parameter an appropriate choice.

Wave-driven transport is not supported in the public release, but we intend to add in in later releases. Magnetic transport is modeled under two different treatments. Strong magnetic coupling, as that predicted from the Taylor-Spruit mechanism, is effectively equivalent to uniform rotation on the lower main sequence. We therefore include an option to simply assume uniform rotation at all times, which mimics the effects of strong coupling. We also include an option for a diffusive magnetic transport term (G. Somers & M. H. Pinsonneault 2016) that does not induce mixing; this can be adjusted to reproduce, for example, the core-envelope coupling timescale inferred from the spin down of low mass stars, or the core to surface rotation contrast in the Sun. This approach is now used in other codes to mimic magnetic transport (P. Eggenberger et al. 2019). The code also allows users to couple core rotation profiles to differentially rotating convective envelopes.

3. SAMPLE APPLICATIONS AND LITERATURE COMPARISONS

The life cycles of stars are wondrously complex. YREC is a flexible tool, which can simulate many scenarios. It is important to understand the precision, accuracy, and domain of validity for the code. We therefore present different physical scenarios as well as different mass, composition, and evolutionary state domains for the code. We begin by laying out the properties of our "base case" models, and follow by discussing models with different input physics or initial conditions. We discuss scripts to run sets of models, and tools to extract information from the code, in Section 4.

3.1. *The Base Case and Numerical Precision Settings*

Our base case adopts the physical assumptions listed in Table 1, and—coupled with the test suite cases—represents a reasonable starting point for users developing their own input namelists. We use a solar calibration for the convective mixing length α and the birth solar helium $Y_{\odot,init}$. For models with non-solar metal content, we use the difference between the solar and Big Bang nucleosynthetic Y abundance (Planck Collaboration et al. 2020) to define the helium to metal enrichment ratio $\Delta Y/\Delta Z$. We note that other codes and isochrones make different assumptions about this ratio, which also depends on the choice of input physics. Our adopted birth hydrogen and metal abundances (X and Z) as a function of [Fe/H] are also given in Table 1, assuming a GS98 abundance pattern.

Table 1. Summary of Input Physics for the Base Model Grid

Parameter	YREC
Atmosphere	Gray
Convective Core Overshoot	Lesser of Step: $0.2H_p$, 0.15 core size
Diffusion	None
Equation of State	OPAL[a,b] + SCV[c]
High-Temperature Opacities	OP[d]
Low-Temperature Opacities	J. W. Ferguson et al. (2005)
Mixing Length	1.707873
Nuclear Reaction Rates	E. G. Adelberger et al. (2011)
Rotation	None
Weak Screening	E. E. Salpeter (1954)
Mixture and Solar Z/X	N. Grevesse & A. J. Sauval (1998)
Solar X	0.719600
Solar Y	0.263908
Solar Z	0.016492
$\Delta Y/\Delta Z$	0.951249
Surface $(Z/X)_\odot$	0.02292

Metallicity Map

[Fe/H]	X	Z
−1.0	0.750950	0.001721
−0.5	0.743176	0.005386
+0.5	0.654060	0.047401

[a] F. J. Rogers et al. (1996); [b] F. J. Rogers & A. Nayfonov (2002); [c] D. Saumon et al. (1995); [d] N. R. Badnell et al. (2005)

The numerical choices in our models are customized to the requirements for different use cases. In some cases, the tolerances for numerical precision had to be relaxed to obtain a solution. We note the domains where the recommended numerical controls differ significantly from the general case in the discussion below.

Our spatial resolution, temporal resolution, and convergence criteria for all cases are defined such that the lifetime of each phase is precise at the 1 percent level. Solar models require $L = L_\odot$, 3.827×10^{33} erg s^{-1}, and $R = R_\odot$, 6.958×10^{10} cm, at the solar age, 4.568 Gyr, within one part in 10^{-4}. The surface $\frac{Z}{X}$ is required to be within 1 percent of the current solar surface $\frac{Z_\odot}{X_\odot}$, 0.02292 for the N. Grevesse & A. J. Sauval (1998) mixture. We also require that neutrino fluxes were precise at the 1 percent level. Solar models including rotation also reproduce the solar equatorial rotation rate (25.4 days) and the difference between the meteoritic and the cur-

rent solar surface ^{7}Li abundance (2.3 dex) within 1 percent. These are achieved by varying the scaling constant in the magnetized wind and the efficiency of mixing relative to angular momentum transport, respectively.

The inner fitting point is set at $10^{-4}M_{\rm tot}$. For typical main sequence conditions, this places the inner fitting point close to $0.01R_{tot}$. Solar models, and the models showing diffusion and mixing to the terminal-age main sequence (TAMS), have the outer fitting point set at $10^{-7}M_{\odot}$ below the surface. For solar models, this places the outer fitting point close to $0.99R_{\rm tot}$. General purpose stellar evolution runs have the outer fitting point set at $10^{-4}M_{\rm tot}$ below the surface, which on the main sequence corresponds to roughly $0.95R_{\rm tot}$. In some cases the fitting point location is adjusted differently to achieve proper numerical results. We also evolved some models in stages, with somewhat different numerical tolerances in different regimes.

We use Hayashi track seed models to generate models at the beginning of the deuterium burning birthline, defined as the location where 1 percent of the birth deuterium has been burned. These models are the starting point for the large majority of the cases that we present here. We provide libraries of starting models with a range of masses and metallicities. In addition to the GS98 mix, we also provide models with the M. Asplund et al. (2021), E. Magg et al. (2022), and K. Lodders et al. (2025) heavy element mixtures. The default YREC models use the K. Lodders (2021) isotopic ratios for CNO and the light element abundances D, ^{3}He, ^{6}Li, ^{7}Li and ^{9}Be.

3.1.1. *Run-speed comparison with MESA*

We perform rough speed comparisons between MESA and YREC for low mass stars. For YREC, we run a solar mass, solar metallicity model up to the tip of the red giant branch using the same numerical and physics settings as the base model grid in Section 3.3. For the MESA comparison model, we use MESA version r24.08.1 with most inlist values left at their default settings except those required to match the input physics choices of the YREC model; an example inlist is provided in the sample cases. We adjusted the global mesh variable `mesh_delta_coeff` in MESA for each phase of evolution to roughly match the number of mesh points in the YREC models. The average number of shells across each phase of evolution we consider is within 3% for both codes. Models were run on a single thread to mimic YREC execution and to make the comparison more hardware agnostic, although in practice users of MESA would utilize all available cores on their machines. From pre-main sequence initialization to the ZAMS, YREC is roughly 8 times faster than MESA at similar mesh resolution; from the ZAMS to TAMS it is $\sim 2\times$ faster; and from the TAMS to TRGB, YREC is $\sim 6\times$ faster. "Default" YREC runs with fewer shells than default MESA: the mesh was made sparser in each of these evolutionary phases to match YREC. In practice, the speed gains may therefore be even greater when models are tuned to achieve the numerical precision required for the science question at hand. Users can use the simplified Yale EoS for another factor of ~ 9 speed gain over the base case pre-MS to TRGB run, which allows for rapid testing (54 to 72 times faster execution time than MESA) with only modest sacrifices in physical accuracy.

3.2. *The Test Suite*

A test suite is provided to demonstrate YREC's capabilties and provide for a baseline reference to aid in regression testing during development. It is not yet comprehensive, but is being added to regularly and our aim is to provide full coverage of use cases and model types that YREC supports. Information on setting up and running all or a portion of the pre-defined test suite or adding one's own test cases may be found in the 'examples' subdirectory of the project file tree.

Our test suite is designed to test physical scenarios in current use, and to include a wide range of physical conditions. We recommend that users who modify the code validate their changes against the standard library of test suite results. Our physical assumptions, the masses, birth compositions, and mixing lengths are listed in Table 2. We will present the results from the test suite and other models in subsequent sections. The namelists also provide examples of how users can customize the code for different use cases.

3.3. *The Base Case Model Grid*

We adopted the test suite templates to run our base grid of models, using the same physical assumptions described in Section 3.1. The grid spans masses from 0.3 to 8.0 $M_{\odot}$ in increments of 0.1 $M_{\odot}$, and metallicities of [Fe/H] $= -1.0$, -0.5, 0.0, and $+0.5$. Models with initial masses below 2.1 $M_{\odot}$ were evolved to the tip of the red giant branch, while the remainder were evolved to the onset of core helium burning. We list in Table 1 the physical assumptions made to generate the grid. We use this grid to define the regime where YREC is expected to produce reliable results for stellar evolution and to demonstrate its predictions.

3.3.1. *From the Deuterium Birthline to Core He ignition.*

Snapshots of the structure of zero-age main sequence models is a logical starting point. In Figure 2, we show

Table 2. Cases in the YREC test suite. Column 1 contains the name, or names, of the cases; Column 2 is the type, and Column 3 is a brief description. Solar type runs are calibrated solar models, designed to showcase common use cases. Zero-age main sequence (ZAMS) runs, defined as where core H drops 0.005 below the birth value, are for the lower main sequence, and highlight the impact of different choices of surface boundary conditions and equations of state. Brown dwarf runs, taken to 15 Gyr, test code performance below the H burning limit. Terminal-age main sequence (TAMS) runs are taken to core H exhaustion, defined as when the central H drops to 0.0001. TAMS runs show the effects of diffusion, rotation, and rotationally induced mixing below and just above the Kraft break. Evolution runs are longer sets chained together. The 0.3 $M_\odot$ case runs first to the ZAMS and next to the TAMS. The 1.0 $M_\odot$ case runs first to the TAMS, then to He ignition in a degenerate flash. It then restarts at the zero-age horizontal branch (ZAHB), and ends at core He exhausion (TAHB). The 3.0 and 9.0 $M_\odot$ cases are evolved to the TAMS, the ZAHB, and the TAHB in three runs.

Case Name	Type	Descriptions
Test solarGS98 base	Solar	Base case.
Test solarGS98 dif	Solar	Diffusion included.
Test solarAAG21 dif	Solar	AAG21 mix, diffusion included.
Test solarM22M dif	Solar	Magg22 mix, diffusion included.
Test solarGS98 dif rot	Solar	Diffusion, rotation, and rotational mixing included.
Test solarGS98 dif rot fast	Solar	As the dif rot case, but rapid birth rotation.
Test solarGS98 dif rot solid	Solar	As the dif rot case, but solid body rotation.
Test solarGS98 kurucz	Solar	Kurucz atmosphere SBC.
Test solarGS98 SF3	Solar	Solar fusion III nuclear reaction rates.
Test solarGS98 yaleeos	Solar	Yale EoS.
m0p03feh+0.0GS98 allard 15 Gyr	Brown Dwarf	$0.03M_\odot$, Allard SBC, to 15 Gyr.
m0p05feh+0.0GS98 allard 15 Gyr	Brown Dwarf	$0.05M_\odot$, Allard SBC, to 15 Gyr.
m0p08feh+0.0GS98 allard 15 Gyr	Brown Dwarf	$0.08M_\odot$, Allard SBC, to 15 Gyr.
m0p1feh+0.0GS98 allard ZAMS	ZAMS	$0.1M_\odot$, Allard SBC.
m0p1feh+0.0GS98 base ZAMS	ZAMS	$0.1M_\odot$, base case.
m0p1feh+0.0GS98 spot25 ZAMS	ZAMS	$0.1M_\odot$, fspot 0.25 case.
m0p3feh+0.0GS98 allard ZAMS	ZAMS	$0.3M_\odot$, Allard SBC.
m0p3feh+0.0GS98 spot25 ZAMS	ZAMS	$0.3M_\odot$, fspot 0.25 case.
m0p3feh+0.0GS98 scveos ZAMS	ZAMS	$0.3M_\odot$, SCV EoS.
m0p3feh+0.0GS98 yaleeos ZAMS	ZAMS	$0.3M_\odot$, Yale EoS.
m1p0feh+0.0GS98 spot25 ZAMS	ZAMS	$1.0M_\odot$, fspot 0.25 case.
m1p0feh+0.0GS98 dif TAMS	TAMS	$1.0M_\odot$, diffusion included, evolved to TAMS.
m1p4feh+0.0GS98 dif TAMS	TAMS	$1.4M_\odot$, diffusion included, evolved to TAMS.
m0p3feh+0.0GS98 base ZAMS, TAMS	Evolution	$0.3M_\odot$, base case evolved to ZAMS, TAMS.
m1p0feh+0.0GS98 base TAMS, HeIgnite, TAHB	Evolution	$1.0M_\odot$, base cases, evolved to the TAMS, He ignition, and from core He burning to core He exhaustion.
m3p0feh+0.0GS98 base TAMS, ZAHB, TAHB	Evolution	$3.0M_\odot$, base cases, evolved to the TAMS, ZAHB and TAHB.
m9p0feh+0.0GS98 base TAMS, ZAHB, TAHB	Evolution	$9.0M_\odot$, base cases, evolved to the TAMS, ZAHB and TAHB.
m9p0feh+0.0GS98 yaleeos TAMS, ZAHB, TAHB	Evolution	$9.0M_\odot$, base cases with Yale EoS, evolved to the TAMS, ZAHB and TAHB.
Solar_m1p0feh+0p0_GN93_TAHB	TAHB	$1.0M_\odot$ Z=0.019 ZAHB evolved to the TAHB, with GN93 opacity tables.
Solar_m0p7feh-0p75_GN93_TAHB	TAHB	$0.7M_\odot$ Z=0.003 ZAHB evolved to the TAHB, with GN93 opacity tables.
Solar_m0p95feh+0p0_GN93_TAHB	TAHB	$0.95M_\odot$ Z=0.019 ZAHB evolved to the TAHB via ZAHB seed envelope mass re-scaling, with GN93 opacity tables.

the internal structure of representative ZAMS models in the $\rho - T$ plane, an important indicator of the predominant pressure sources from the equation of state. We also show the boundaries where radiation pressure (upper left) and degeneracy pressure (lower right) dominate.

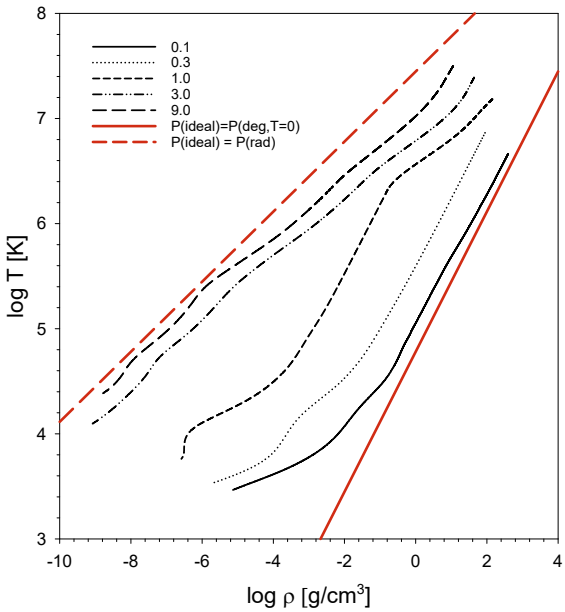

Figure 2. The locus in the $\rho - T$ plane of 0.1(lower right), 0.3, 1, 3 and 9 $M_\odot$ models (upper left) at the zero-age main sequence, defined here as the point where the central hydrogen mass fraction is 0.005 lower than the birth value. These are solar metallicity models using the base model physics. The domains where radiation pressure equals gas pressure, and where zero-temperature degeneracy pressure equals gas pressure, are indicated by the respective red lines.

The central temperature and density for models as a function of mass is also an interesting diagnostic of stellar structure. We present our models at the start (ZAMS) and end (TAMS) of the main sequence, compared with PARSEC and MIST, in Figure 3. The upper MS has a shallow slope in central T as a function of mass due to the strong temperature dependence of the CNO cycle; in lower mass stars the pp chain is more important and the slope is steeper there. The steep change in central density close to solar mass reflects the transition from radiative to convective envelopes. Different codes are very close above 0.5 $M_\odot$ on the ZAMS, with the differences at lower mass reflecting the importance of the equation of state and surface boundary conditions for these stars. Although both ρ and T increase on the MS, the increases are much larger for ρ. Physical conditions can change rapidly after the MS turnoff, which accounts for some of the differences seen between 1 and 2 $M_\odot$.

The depth of core and surface convection zones are another interesting diagnostic of stellar evolution, frequently displayed as a function of time in Kippenhahn diagrams. We show the histories of 1 and 3 $M_\odot$ models in Figure 4. The 1 $M_\odot$ model has a shallow (in mass) convective envelope on the MS, develops a deep surface convection zone during an extended red giant branch phase, and has a convective core during the core He-burning phase. The 3 $M_\odot$ model has a convective core that gradually shrinks during the MS phase, followed by a brief shell H-burning red giant phase. The core He-burning phase is a larger fraction of the lifetime in this model than in the lower mass one.

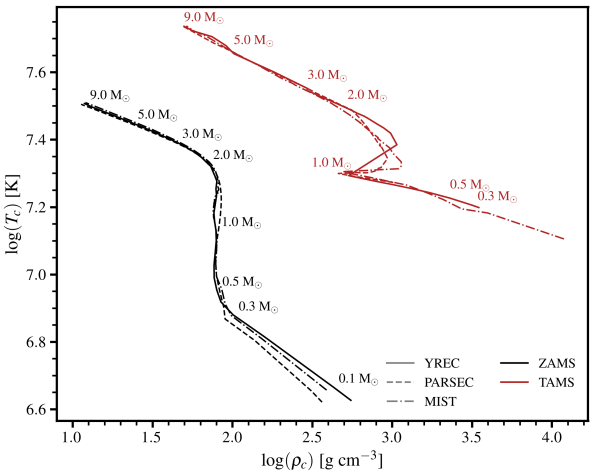

Figure 3. Central temperature as a function of central density at ZAMS (black) and TAMS (red) locations. Comparisons are shown between YREC, MIST, and PARSEC evolution tracks for solar metallicity (YREC, $Z = 0.016492$; MIST, $Z = 0.0142$) and $Z = 0.017$ (PARSEC).

Figure 5 shows the stellar radius as a function of mass, with the black and red lines corresponding to the ZAMS and TAMS locations. The lower envelope is in excellent agreement with fundamental data from eclipsing binary stars (G. Torres et al. 2010). The apparent agreement is deceptive for the lower MS, where the TAMS greatly exceeds the age of the Universe. We discuss the "radius inflation" problem in Section 3.4.5.

In Figure 6, we show the evolutionary tracks for solar metallicity models at 0.3, 1.0, 3.0, and 9.0 $M_\odot$. The main sequence band is superposed for reference. The lower branch for each mass track is the pre-MS phase, defined at deuterium ignition. This definition breaks down for the more massive stars, where a physical model would predict a starting point closer to the MS (F. Palla & S. W. Stahler 1991; H. W. Yorke & C. Sonnhalter

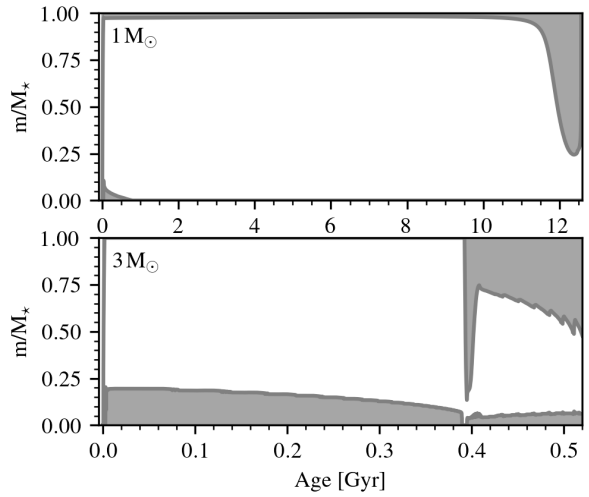

Figure 4. Kippenhahn diagram showing the evolution of convective boundaries for the $1M_{\odot}$ (top panel) and $3M_{\odot}$ (bottom panel) models. Shaded gray regions indicate convective zones shown as a function of age and fractional mass coordinate.

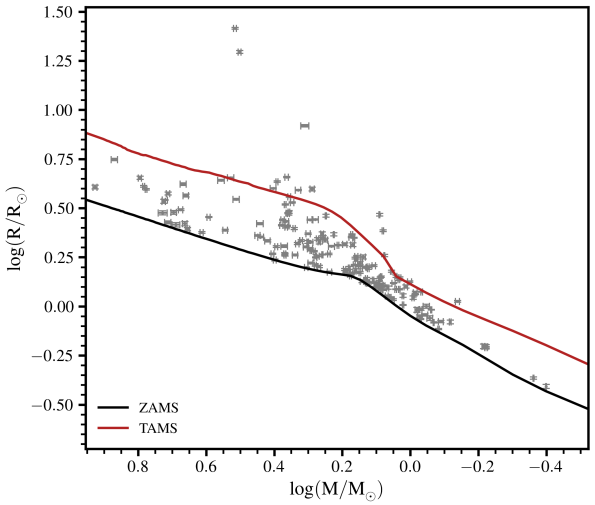

Figure 5. Radius as a function of mass for tracks 0.3 - 9.0 $M_{\odot}$ at ZAMS and TAMS. G. Torres et al. (2010) masses and radii of eclipsing binaries plotted for reference.

2002). A transient loop above the MS, due to non-equilibrium CN burning, is clearly seen in the higher mass tracks, as is the sudden adjustment at the end of the MS phase from H depletion in a large convective core. The 1 and 3 $M_{\odot}$ cases are similar in L during the core He-burning phase, while the 9 $M_{\odot}$ case is significantly brighter due to a larger H-exhausted core on the MS.

The non-rotating models from the MIST (J. Choi et al. 2016) and PARSEC (v1.2 and v2.0, A. Bressan et al. 2012; Y. Chen et al. 2014; Y. Chen et al. 2015; X. Fu et al. 2018 C. T. Nguyen et al. 2022a; G. Costa et al. 2025) libraries are overlaid for comparison with YREC [13]. The tracks for different codes are broadly similar at solar mass and above. Tracks for the lowest mass model ($0.3\ M_{\odot}$) are quite different, reflecting the importance of the equation of state and surface boundary conditions. See Section 3.4.2 for a further discussion of this point.

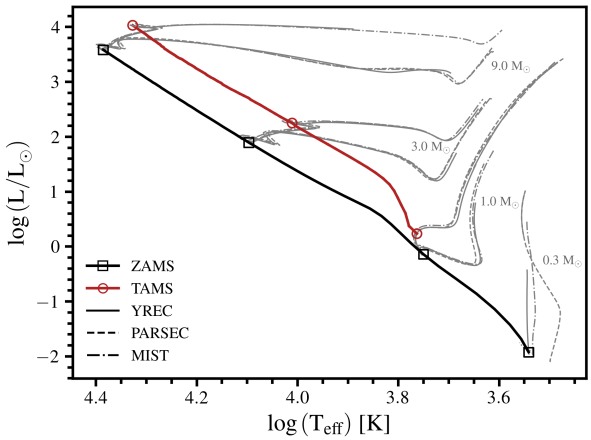

Figure 6. Luminosity as a function of temperature for the solar metallicity models. The solid black and red lines show the location of ZAMS and TAMS, respectively. We show a comparison of evolutionary tracks from YREC (solid), PARSEC (dashed), and MIST (dash-dot) model grids for solar metallicity and Z = 0.017 for PARSEC. Tracks are shown for masses of 0.3, 1.0, 3.0, and 9.0 $M_{\odot}$, as available in each model library. 1.0, 3.0, and 9.0 $M_{\odot}$ tracks are ended at He core ignition.

Table 3 provides a summary of stellar parameters at TAMS for the $1M_{\odot}$ tracks, including age, luminosity, effective temperature, radius, central density, and central temperature. The TAMS location is defined as the point where the central H abundance drops to 0.0001. Both MIST and PARSEC models show close agreement with our base grid in central temperature, effective temperature, and radius, each agreeing within 3%. The MIST models exhibit slightly larger offsets in luminosity and age (6 and 10%, respectively), while the PARSEC models differ primarily in central density and age (6% and 7%). These discrepancies can be partially attributed to differences in the precise definition of the TAMS point in each grid and variations in the adopted input physics. For instance, the MIST models use the R. Kurucz (1993)

[13] The PARSEC library provides only pre-MS to ZAMS for 0.3 $M_{\odot}$ and does not include tracks at its defined solar metallicity, where $Z_{\odot} = 0.01524$ and $Z_{\rm initial} = 0.01774$ (A. Bressan et al. 2012; Y. Chen et al. 2014). For this reason, we use the Z = 0.017 tracks for comparison.

Table 3. Comparison of stellar properties between YREC, MIST, and PARSEC for 1$M_\odot$ at TAMS.

Parameter	YREC	MIST	PARSEC
Age [Gyr]	9.807	8.781	9.081
$\log(L/L_\odot)$	0.235	0.260	0.223
$\log(T_{eff})$ [K]	3.763	3.766	3.765
$\log(R/R_\odot)$	0.114	0.120	0.106
$\log(\rho_c)$ [g cm^{-3}]	2.812	2.798	2.784
$\log(T_c)$ [K]	7.285	7.294	7.290

surface boundary conditions, whereas this grid adopts the Eddington-grey approximation.

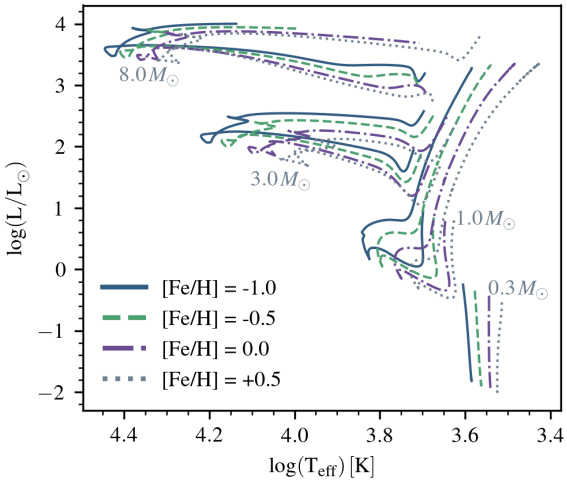

Figure 7. H-R diagram showing evolutionary tracks for masses 0.3, 1.0, 3.0, and 8.0 $M_\odot$ from the base grid. We show how each track evolves across metallicities [Fe/H] = -1.0 (solid blue), -0.5 (dashed green), 0.0 (dash-dotted purple), and +0.5 (dotted grey). The 0.3 $M_\odot$ tracks extend to ZAMS, the 1.0 $M_\odot$ tracks to the tip of the RGB, and the 3.0 and 8.0 $M_\odot$ tracks to ZAHB, defined by $Y_{cen} = 1 - Z - 0.06$.

Our final stop before the core He-burning phase is the red giant phase, where we now have rich and precise data sets due to asteroseismology. Stellar models make strong predictions about the metallicity dependence of stellar properties, as illustrated in Figure 7. We are now in a position to test these predictions. Modern time domain surveys have revealed a rich landscape of stellar variability. One of the highest impact results was the discovery of solar-like oscillations in cool and evolved giant stars. The global oscillation properties can be combined with spectra to infer mass, radius and age for large numbers of stars. In Figure 8, we compare our models with data from the APOKASC-3 catalog (M. H. Pinsonneault et al. 2025). There is a clear offset, which reflects the limitations of the solar calibration for stars very different from the Sun. See J. Tayar et al. (2017) for a discussion.

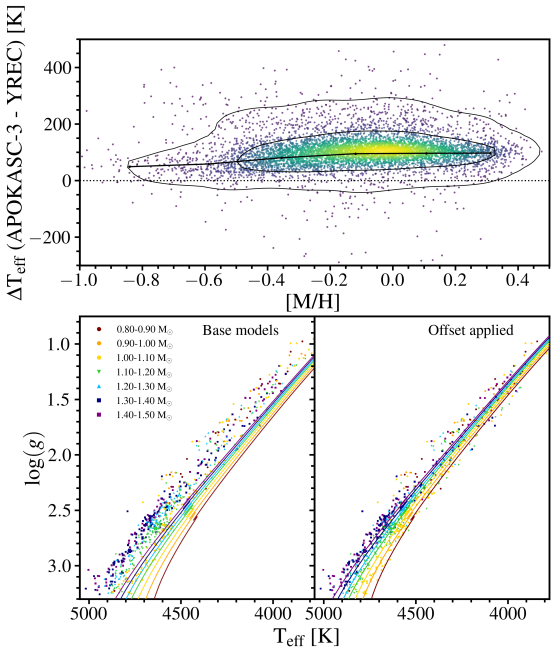

Figure 8. *Top:* Difference between APOKASC-3 and YREC effective temperatures for the RGB sample in the APOKASC-3 catalog. Contours represent 68% and 95% of stars in the sample. *Bottom:* Kiel diagrams of APOKASC-3 RGB stars with metallicities in the range -0.05 < [M/H] < +0.05. The right panel shows the sample overlaid with the base grid solar metallicity tracks. The track colors correspond to the lower boundary of the mass bins. The left panel shows the same tracks after applying the temperature offset shown in the top panel.

3.3.2. *Core He-burning models*

The core He-burning phase is complex, and there can be significant differences between evolutionary codes. These complications arise in earnest even at helium ignition. The onset of He burning is a smooth process in high mass stars, though it does not proceed smoothly for low-mass stars. For stars below $\approx 2M_\odot$, He-burning ignites in a highly degenerate medium. Evolving a model through this helium flash is numerically challenging because the usual stabilizing effects of the ideal gas equation of state are short-circuited. The assumptions of the code that work well for most conditions - in particular, instantaneous mixing of convective regions and exact hydrostatic equilibrium - become poor ones during the helium flash. Great care with numerics is also required;

the standard mesh allocation procedure in YREC works well for smooth and stable evolution, but less so for domains with rapid structural change. It is a testament to the strengths of the MESA code that it can routinely be evolved through the He flash (L. Bildsten et al. 2012). It is worth noting that YREC *can* solve these issues, and in fact it was one of the first codes to carry evolution through the helium flash (P. W. Cole et al. 1985). However, this capability greatly complicated the code, and was removed from modern versions over many years of revisions.

Instead, YREC uses a two-step procedure for low-mass horizontal branch stars that experience the helium flash. The equations of stellar structure can be solved in a time-independent fashion for stars with nuclear energy sources. Given a composition profile, it is therefore possible for these stars to solve for a core He-burning model, similar to the approach used for the core H-burning main sequence. The most substantive difference is that the helium core mass, in addition to the total mass and metallicity, needs to be specified. For a self-consistent run, one evolves to the RGB tip, infers the core mass as a function of total mass and metallicity, and then rescales a seed core He burning model to the desired specifications. See Section 4.1 for a discussion of the procedure that we use. The current suite of seed models for core He-burning were built with the N. Grevesse & A. Noels (1993) mixture, and some namelists in the test suites also use the corresponding atomic opacity tables. It is not entirely trivial to adjust the non-uniform ZAHB abundances to a different mix; we intend to expand the suite to different mixtures in later releases. The default, however, is to use atomic opacity tables calculated with the GS98 mixture. Neutrino cooling rates also differ from the default, taken from J. N. Bahcall (1989) in order to be consistent with the seed models. The helium flash models described in this section ($< 2M_{\odot}$) use this two-step procedure.

The precise physics appropriate for modeling the regions above the convective cores of the HB are not settled, which we discuss briefly here. The fundamental complication arises from carbon- and oxygen-rich material in the convective core 'overshooting' the nominal convective boundary due to inertia from convective motions at the boundary. Since this material is more opaque than the material it mixes into, this region becomes unstable to convection according to the Schwarzschild criterion. Nevertheless, the region is stable against convection when accounting for the composition gradient in the Ledoux stability criterion.

It was therefore first proposed by P. Ledoux (1947) that such a 'semi-convection' region would experience strong enough convection to mix abundances, but not strong enough to significantly transport energy; i.e., the region would transport energy via radiation. Discussed in detail in the context of very massive stars (M. Schwarzschild & R. Härm 1958) and later in core helium core burning stars (e.g., B. Paczyński 1970), the typical assumption is that the semi-convection region settles into a composition at every point such that $\nabla_{\rm rad} = \nabla_{\rm ad}$.

An alternate formulation to the mixing dilemma above a core helium-burning model is overshooting, in which the convective boundary is only extended to a certain extent that depends on the motion of the convective elements as they pass the nominal convective boundary. Both prescriptions predict that the lifetime of the horizontal branch will be extended as the core grows, since it has access to more helium than it otherwise would. This results in a different ratio in the number of horizontal branch stars to asymptotic giant branch stars in clusters compared to a case when new helium is not introduced into the core (e.g. F. Caputo et al. 1989; T. Constantino et al. 2016). In detail, the evolution of the helium abundance proceeds differently under these two scenarios, which produces different predictions for asteroseismic gravity mode frequencies. Unfortunately, neither standard overshooting nor semi-convection reproduce satisfactorily, though non-standard models can (T. Constantino et al. 2015; D. Bossini et al. 2015, 2017).

YREC adopts an overshooting formalism to model the physics of mixing around the core helium-burning core. We compare our horizontal branch models against PARSEC v2.0, since it, too, adopts an overshoot formalism (A. G. Bressan et al. 1981) that is different in detail, but similar in extent.

We illustrate the evolution of our solar metallicity horizontal branch models from the ZAHB (defined to be when $Y_{\rm cen} < 1 - Z - 0.06$) in Figure 9, with tracks also shown from rotating PARSEC v2.0 (G. Costa et al. 2019b,a; C. T. Nguyen et al. 2022b, 2025a) models, which cover the low-mass ZAHB via a similar core mass rescaling as YREC uses. The stepwise evolution in the core of low-mass stars is evident in YREC models, which reflects episodes where fresh fuel is ingested into the He-burning core. These are less evident in the PARSEC v2.0 models at the scale of the figure, athough the qualitative behavior is present, as we show below when looking at the central helium evolution. Modest differences in luminosity ($\approx 10\%$) and in temperature ($\approx 3\%$) between low-mass YREC and PARSEC v2.0 ZAHB locations in luminosity–effective temperature space are consistent with previous findings of agreement among publicly-available solar metallicity ZAHB models (D. An et al. 2019).

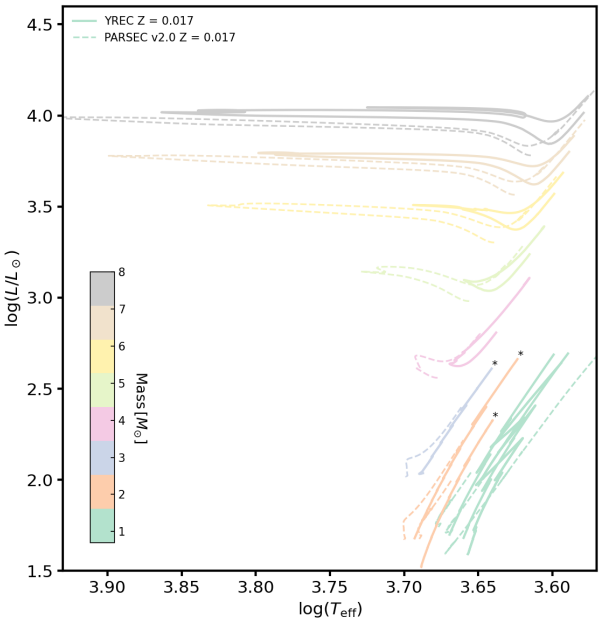

Figure 9. Comparison of the low-mass core helium burning phase from YREC (solid) and PARSEC v2.0 (dashed) from beginning of central core helium burning to $Y_{\rm cen} = 10^{-3}$ except where noted with *. The tracks include masses between and including $0.8M_\odot$ and $2.4M_\odot$ in steps of $0.4M_\odot$ and $3M_\odot$ and $8M_\odot$ in steps of $1M_\odot$. The low-mass PARSEC v2.0 models, which have suppressed helium ingestion episodes, generally reach core helium exhaustion at lower luminosities than YREC models, which experience helium ingestion episodes that allow for continued growth of the core and extension to larger luminosities.

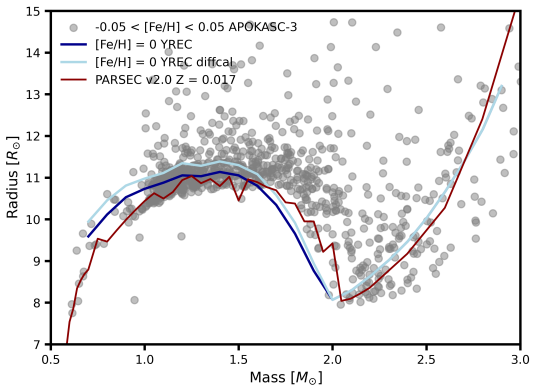

Figure 10. Comparison of the core helium burning phase from YREC and PARSEC v2.0 against APOKASC-3 masses and radii for solar-metallicity stars. Cases with and without solar diffusion-calibrated $Y = .273$ and $Z = 0.0188$ are shown. YREC models with $M \geq 2M_\odot$ use a two-step procedure. See text for details. Although there is qualitative agreement, asteroseismic constraints suggest opportunities for understanding the physics of the horizontal branch better in both YREC and PARSEC v2.0 models.

YREC models proceed to much higher luminosities and lower temperatures than PARSEC. This can be traced to a higher ingestion rate of helium into the core during overshoot in YREC, which lengthens the lifetime of the horizontal branch. Importantly, PARSEC does not instantaneously mix abundances in convective regions, instead following a diffusion approach (C. T. Nguyen et al. 2022b). This latter difference would reduce the amount of helium that enters into the core at any given time, and correspondingly reduce the severity of mixing zone mergers compared to those in YREC, which assumes instantaneous mixing. (We discuss these mixing episodes below, which are also responsible for the sharp decreases in luminosity in Figure 9.)

Asteroseismic data is a powerful test of stellar models, and we compare ZAHB models with data in Figure 10. We note that we define the ZAHB as when $Y_{\rm cen} = 1 - Z - 0.06$ for YREC, which corrects for the energy required to lift degeneracy in the core. A sharp boundary in radius as a function of mass is observed in the data, which corresponds to the ZAHB. The handful of stars with radii below the edge are the products of binary star interactions (Y. Li et al. 2022). The predicted mass range for the onset of degenerate helium ignition agrees well with the local minimum in radius at $M \approx 2.0M_\odot$. In detail, none of the models trace the lower boundary precisely. The core mass in the YREC case is constant to within 1% ($M_{\rm core} \approx 0.459M_\odot$) for $0.7M_\odot \leq M \leq 1.2M_\odot$. PARSEC v2.0 models have slightly higher core masses, sitting within 1% of $0.475M_\odot$ until $M \approx 1.35M_\odot$. Above $M \approx 1.4M_\odot$ ($M \approx 1.35M_\odot$), YREC (PARSEC v2.0) the core mass begins decreasing as neutrino cooling becomes less efficient, which is reflected in the turnover of the mass-radius relation. Neither YREC nor PARSEC v2.0 completely agrees with the ZAHB location in mass-radius space. YREC models at higher masses may be brought into better agreement by adopting an initial helium mass fraction consistent with the PARSEC choice (smaller than YREC's by 0.015), while adopting a higher helium abundance consistent with solar-calibrated diffusion models worsens the agreement (light blue in Fig. 10). Further work is clearly needed to reconcile models with the powerful asteroseismic constraints.

Turning to a closer inspection of the growth of the core with time, we show in Figure 11 the PARSEC v2.0 and YREC central abundance evolution of a solar horizontal branch model in the presence of overshooting. Initially helium is converted into carbon; at higher carbon abundance, $C^{12} + \alpha$ becomes more efficient than triple α, leading to higher oxygen production.

The small stepwise increases in helium visible in Figure 11 and in the corresponding sharp decreases in luminosity in low-mass tracks shown in Figure 9 are due to the subtleties of the physics exterior to the convective core mentioned above. The emergence of a semi-convection region will split the convective core into two if using the Schwarzschild stability criterion and adopting the radiative temperature gradient in this region, as PARSEC v2.0 and YREC both do. The outer convection zone shrinks in size even as the convective core grows with the opacity increase from nuclear burning. Overshooting helps bridge the gap at which point the core reaches the convective shell discontinuity, ingesting helium, shrinking due to the temporary decrease in opacity this causes, and splitting again. This split does not occur if the temperature gradient is assumed to be adiabatic and not radiative as is commonly assumed in semi-convection prescriptions, and these episodes do not occur. The lifetime of the horizontal branch is still increased in this case, though, as the core can mix with the semi-convection region. Classical breathing pulses as described in A. V. Sweigart & P. Demarque 1973 and occurring at $Y_{\rm cen} < 0.1$ are also present in YREC models for both the overshooting and no overshooting cases. Such pulses are related to the smaller mixing zone merger episodes, though in this case the increased temperature of the core at this phase in the evolution can produce luminosities in the core large enough to overcome decreases in opacity due to helium ingestion; the result is a large increase in core size with helium ingestion followed by quiescent increase in core size due to nuclear burning (e.g., V. Castellani et al. 1971).

We compare against MESA r24.03.1 to further demonstrate differences in models, even in the absence of the details of overshooting. For this purpose, we modify the 1MS_pre_ms_to_wd inlists from the MESA test suite, turning off mass loss as well as semi-convection and overshooting, and adopting GN93 abundances with Z=0.02, Y=0.28. We also enforce the same mixing length formalism as YREC (E. Böhm-Vitense 1958) and do not use Type 2 opacity tables in order to be consistent with YREC interior opacity tables. For this comparison, our YREC models use GN93 OPAL atomic opacities to be as consistent as possible with the YREC ZAHB seed models, which were initially generated using these abundances.

As seen in Fig. 11, whereas for YREC we see small increases in central helium content as the convective core grows in size and ingests new helium, the same does not occur in MESA. The large jump at ≈ 80 Myr would correspond to the classical breathing pulses that occur at low helium content (Y< 0.1). There is also a small difference in the luminosity and the corresponding duration of the core He-burning phase from central ΔY$= 0.06$ until reaching $Y_{\rm cen} = 10^{-3}$.

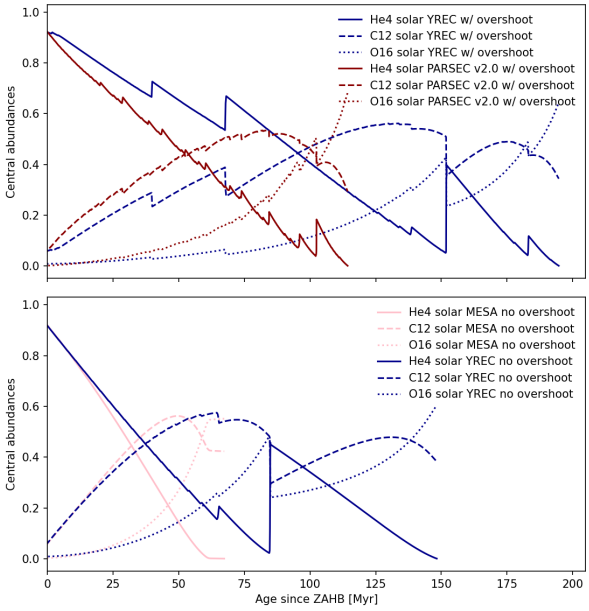

Figure 11. Comparison of the core helium burning phase for a solar model from YREC (dark blue) with and without overshooting, matched to the core masses of PARSEC v2.0 (red; top) and MESA (pink; bottom). Note that MESA and PARSEC predict slightly different core masses: $0.443 M_\odot$ and $0.469 M_\odot$, respectively, though most of the lifetime differences in the top and bottom panels is due to the difference in helium ingestion due to overshooting. A significant core breathing pulse is observed at ≈ 80 Myr in the no overshooting case. The evolution of the region outside the convective core in horizontal branch models even with similar treatments can differ.

3.4. *Sample Cases*

It can be useful for readers to see examples of use cases for YREC, and we present some examples here. For solar models, and ones on the lower and upper MS, we discuss the relevant models from the test suite. We also include models with rotation and starspots, which include grids not given in the testsuite. The namelists for these cases can be found in the sample cases directory and are listed in Table 4.

3.4.1. *Solar Models*

YREC has a long, and high-impact, history of usage for solar models starting in the 1960s (P. R. Demarque & J. R. Percy 1964). The first generation of solar models including rotation (M. H. Pinsonneault et al. 1989) was also achieved with YREC. The "standard" solar model for neutrinos in the 1990s was produced with YREC

Table 4. Families of sample cases discussed in the paper. The case prefix is presented in the first column. The location of the namelists (test suite or sample cases) is in the second, and a description is in the third. "Both" for the location is for situations where a subset of the cases are in the test suite.

Case Name	Location	Descriptions
Test solar	Test Suite	Solar models calibrated with a range of input physics and birth mixtures
Sample base ZAMS	Sample Cases/LowerMS	Models of $0.09 - 1M_{\odot}$ evolved from the DBl to the ZAMS.
Sample allard ZAMS	Sample Cases/LowerMS	Models of $0.03-1M_{\odot}$, evolved from the Dbl to 15 Gyr($0.03 - 0.08M_{\odot}$) or the ZAMS ($0.09 - 1M_{\odot}$) with Allard atmosphere SBCs
Sample kurucz ZAMS	Sample Cases/LowerMS	Models of $0.15 - 1M_{\odot}$, evolved from the Dbl to the ZAMS with Kurucz atmosphere SBCs
Sample yaleeos ZAMS	Sample Cases/LowerMS	Models of $0.3-1M_{\odot}$, evolved from the Dbl to the ZAMS with the Yale EoS
Sample m9p0feh0+0GS98 Cases	Both	$9M_{\odot}$ models with different EoS, degress of overshoot, with and without semi-convection
Sample Gyrochronology	Sample Cases	Models of $1.4M_{\odot}$ and below, including rotation, evolved from the Dbl to an age of 4 Gyr with different assumptions about angular momentum loss and internal angular momentum transport.
Sample Starspots	Sample Cases	Models of $1.4M_{\odot}$ and below, including star spots, evolved from the Dbl to the ZAMS with different star spot filling factors.
Sample ZAHB	Sample Cases/TAHB	Models of $1.8M_{\odot}$ and below, evolved from the ZAHB with core sizes consistent with the base case model grid of §3.3, with overshooting.
ZAHB diffcal	Sample Cases/TAHB	Same as ZAHB, but with diffusion-calibrated Y and Z.
Solar PARSEC	Sample Cases/TAHB	Solar model horizontal branch evolution matched to PARSEC v2.0, with overshooting.
Solar GN93 MESA	Sample Cases/TAHB	Solar model horizontal branch evolution matched to MESA, with GN93 opacity tables.

(J. N. Bahcall & M. H. Pinsonneault 1992; J. N. Bahcall et al. 1995), and these papers added gravitational settling as a key ingredient for solar models. YREC models also established the tension between a low solar metallicity and helioseismic data J. N. Bahcall et al. (2005); F. Delahaye & M. H. Pinsonneault (2006); F. L. Villante et al. (2014) YREC solar model work continues to the present day C. Basinger et al. (2024), including work on the faint young Sun problem. Our test suite includes a number of different solar models, calibrated under commonly used physical scenarios. Our sample cases account for differences in the mixture of heavy elements, the equation of state, nuclear reaction rates, surface boundary conditions, microscopic diffusion, the treatment of rotation, the initial angular momentum, and the associated mixing.

These are precise models, which require tight numerical controls that are not required for general purpose models. For example, the convergence criteria for these models can be difficult to achieve in the general case. Small timesteps in the pre-MS are required for precise measurements of lithium depletion, but not for a precise pre-MS lifetime or HR diagram evolution. For our solar models, this is achieved by lowering the maximum change in the structure variables across a timestep relative to the general case. The inner fitting point is also moved inwards, for more precise solar neutrino calculations. This change has a negligible impact on broader classes of stellar models.

We show the sound speed profiles of our calibrated solar models in Figure 12. These results, as well as model solar neutrino fluxes, are in excellent agreement with literature values. See C. Basinger et al. (2024) for a detailed discussion. The solar models without microscopic diffusion show large deviations from the true solar profile, a well-known literature result (J. N. Bahcall & M. H. Pinsonneault 1992; J. Christensen-Dalsgaard et al. 1993). The choice of nuclear reaction rates and SBCs has a small impact, but the older Yale EoS is a worse fit to the data in the envelope. The birth mixture has a significant impact on the sound speed profile, although it is less than that for diffusion. The most metal poor mixture, AAG21, has the largest sound speed deviations. Rotationally induced mixing has a measurable effect, but broadly follows the same patterns as solar models with diffusion alone. Different initial angular momenta and models of internal angular momentum transport yield similar results.

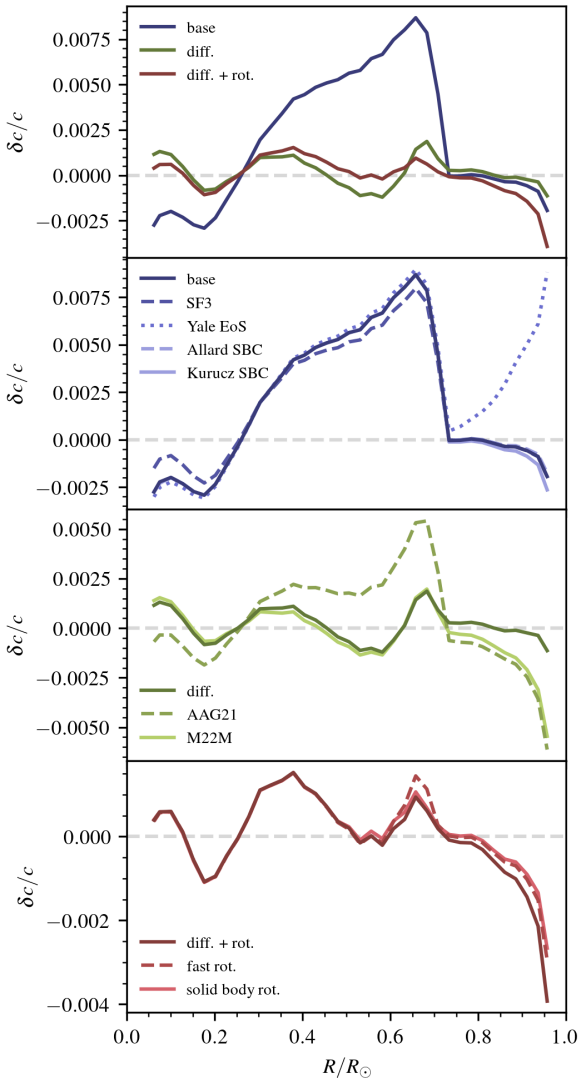

Figure 12. Fractional differences in sound speed between our test suite solar models and solar data from S. Basu et al. (2009). The top panel shows differences between models with the same mixture that include different physical processes. Differences in nuclear reaction rates and surface boundary conditions are compared in the second panel. The third panel shows the impact of changes in the heavy element mixture; all of these models included diffusion but not rotation. The impact of different initial rotation rates and the transport model are compared in the bottom panel.

3.4.2. *The Lower MS*

The choices for the surface boundary condition and the inclusion of star spots are quite important for lower MS stars, and we show the impact on the ZAMS for representative choices in Figure 13. The Kurucz and Allard table edges define the cool end of the runs. The general trend is that the more simplified models (Yale EoS and gray atmospheres) depart significantly from those with more sophisticated SBCs (Kurucz and Allard) and equations of state (OPAL and SCV) below about 4000 K. We only show the Yale EoS down to 0.3 solar masses, and the Kurucz atmospheres down to 0.15. For 0.5 $M_\odot$ and above, the Yale EoS and gray atmosphere case performs quite well on the main sequence. Because this combination is much faster to compute than the full case, this is a good choice for educational purposes. We also show brown dwarf tracks here, along with the onset of the brown dwarf cooling curves.

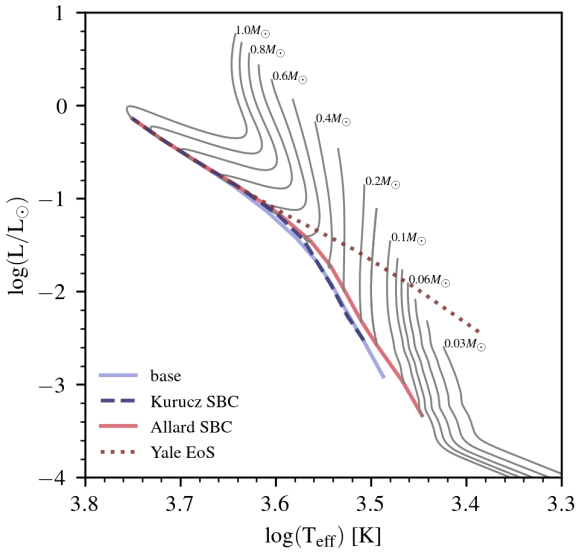

Figure 13. The impact of the choice of surface boundary conditions and equation of state on lower MS solar metallicity models. ZAMS curves are shown for the base models, Allard model atmospheres, Kurucz model atmospheres, and the Yale EoS. Model tracks to 15 Gyr (solid black lines) are shown for the Allard case.

3.4.3. *The Upper MS*

Example high-mass models are spread across the test suite and sample cases. Test suite models include two 9 $M_\odot$ stars evolved to the TAHB, one with the base set of input physics (`base`) and the other with the base set of input physics but the Yale EoS (`base_yale`). Sample case models, selected to showcase different types of extra convective mixing, include three 9 $M_\odot$ stars evolved with base case physics and either 1) no overshooting and no semi-convection (`base_none`), 2) overshooting with 0.4 H_p (`base_0.4os`), and 3) semi-convection and the Yale EoS (`base_yale_sc`). All models are run with all four levels of iterations. High-mass models evolving beyond the main sequence are very sensitive to adopted physics and numerical choices. Presently, the central fitting point must be moved in during the initial phase

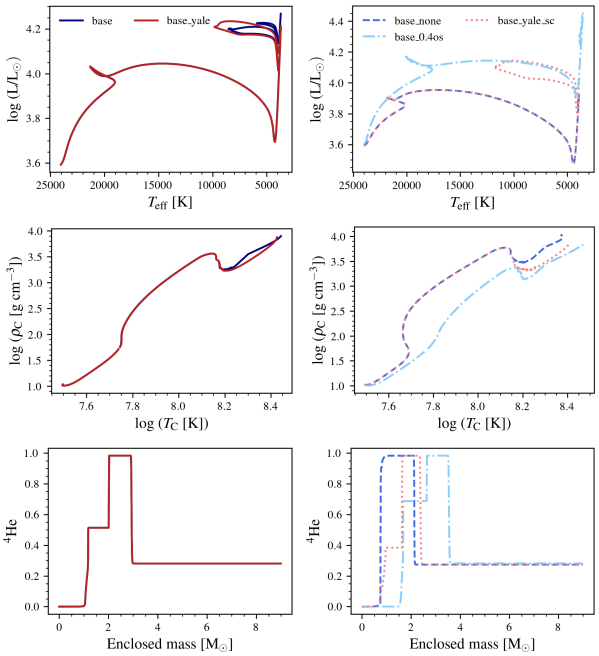

Figure 14. The evolution and final helium profile of 9 $M_\odot$ models run with different assumptions about convection and two different EoSs. The left-hand panels compare the two test suite models, which largely overlap, and the right-hand panels compare the sample case models. Blue models are for the default OPAL EoS and red models are for the default Yale EoS.

of evolution for the later stages of evolution to run properly. Models with semi-convection are particularly sensitive. Presently, a 9 $M_\odot$ model with semi-convection and the OPAL EoS does not reach the TAHB.

Figure 14 shows the evolution in an HR diagram, the evolution of central density and temperature, and the final helium profile of the models. The test suite models, where only the EoS is changed between them, shown on the left-hand side of the figure, have nearly identical evolution, though the `base_yale` model lives ∼1 Myr longer. Central density and temperature diverge during core helium burning, corresponding to differences in the blue loops the models exhibit. Despite differences in core helium burning, the models end up with the same helium profile at the TAHB.

Sample case models largely behave as expected. The `base_none` model evolves the fastest, does not exhibit a blue loop, and has the lowest luminosity and helium-core mass of the set. The `base_yale_sc` model evolves nearly identically to the `base_none` the model, deviating during core helium-burning, when semi-convection becomes important in the region surrounding the core. The `base_0.4os` model has the highest luminosity and core mass, the longest lifetime, and the hottest and least dense core.

Note the two-tiered helium abundance in the cores of most models. This behavior, though peculiar, is linked to extra convective mixing, as it is seen in all models with semi-convection or overshooting and not in the model without them. There ends up being a partially helium-depleted layer around the core where helium-rich material from the base of the helium shell was mixed into the core and burned. This partial He-depletion in models with overshooting or semi-convection only shows up at intermediate masses (4-5 $M_\odot$, depending on adopted physics) and above.

3.4.4. *Main Sequence Gyrochronology*

The use of open cluster rotation sequences to calibrate both theoretical and empirical angular momentum loss models has a long history. Empirical fits to cluster sequences generally provide physics-agnostic interpolating functions for period as a function of age and mass (e.g. E. E. Mamajek & L. A. Hillenbrand 2008; S. A. Barnes 2010; A. C. Lanzafame & F. Spada 2015; L. G. Bouma et al. 2023). Theoretical approaches leverage stellar structure models coupled to a physically motivated magnetic braking law, but remain fundamentally anchored to cluster (and other) calibrators (e.g. A. Krishnamurthi et al. 1997; J. Bouvier et al. 1997; A. Sills & M. H. Pinsonneault 2000; P. A. Denissenkov et al. 2010a; F. Gallet & J. Bouvier 2015; J. L. van Saders & M. H. Pinsonneault 2013; G. Somers & M. H. Pinsonneault 2016; L. Amard & S. P. Matt 2020; J. Roquette et al. 2021). We present an example of this calibration exercise with YREC rotating models in Figure 15. Two sets of physical assumptions are shown: one in which models include microscopic diffusion, internal angular momentum transport, and magnetized braking from the surface. A second suite shows the behavior of strongly magnetically coupled models, approximated by enforcing solid body rotation at all times. Both sets of models are independently calibrated by varying a normalization parameter f_k to match the rotational behavior at 0.9 solar masses in < 3 Gyr old clusters (J. L. Curtis et al. 2020).

Models are run with mass-dependent initial rotation periods following G. Somers et al. (2017) and subject to saturated spin-down below Rossby numbers of $0.1\mathrm{Ro}_\odot$. For those models allowed to differentially rotate, AM transport is performed using both hydrodynamical instabilities and an extra AM diffusion term following G. Somers & M. H. Pinsonneault (2016) that better reproduces both the chemical and rotational histories of cluster stars. This diffusion term is constant in time, but a strongly varying function of stellar mass; we adopt the mass-dependence determined in G. Somers & M. H.

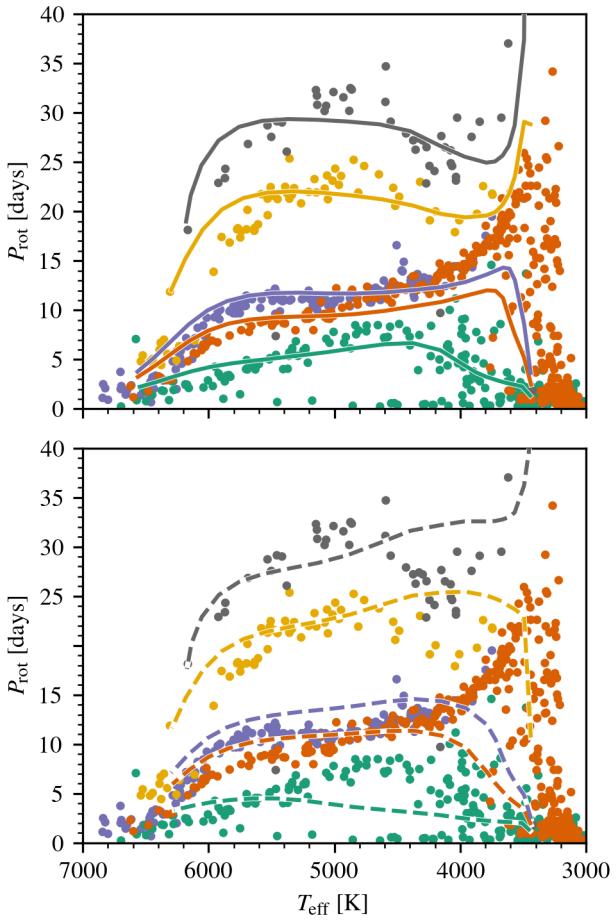

Figure 15. Rotating models compared against a select sample of open cluster rotation sequences from J. L. Curtis et al. (2020) and D. Gruner et al. (2023). Solid curves (top panel) denote models that include the full treatment of microscopic diffusion, internal AM transport, and wind braking. Dashed curves (bottom panel) represent strongly magnetically coupled models, where solid body rotation is enforced at all times.

Pinsonneault (2016) for display here. The braking is performed with the J. L. van Saders & M. H. Pinsonneault (2013) braking law.

Departures from the cluster sequences shown in Figure 15 are generically areas of active research; the gyrochrones we show here represent a reasonable but not exhaustive exploration of the possible parameter choices and their optimization.

3.4.5. *Young Stars and Star Spots*

Starspots are frequently observed on the surfaces of cool, active stars. The presence of spots has long been associated with anomalies in color (B. Campbell 1984; J. R. Stauffer et al. 2003), abundances (D. R. Soderblom et al. 1993), and spectra (M. A. Gully-Santiago et al. 2017). As a magnetic activity proxy, it shows a clear relationship with the stellar Rossby number, similar to other activity proxies thought to have an origin in the stellar dynamo (L. Cao & M. H. Pinsonneault 2022). Spots have also associated with surface temperature systematics (L. Cao & M. H. Pinsonneault 2022; F. Pérez Paolino et al. 2025), lithium depletion anomalies in young stars (e.g. R. D. Jeffries et al. 2021, 2023; L. Cao et al. 2026), age systematics (T. J. David et al. 2019; L. Cao et al. 2022), and radii excesses due to magnetic radius inflation (R. J. Jackson et al. 2018; L. Cao & K. G. Stassun 2025). In YREC, starspots were initially implemented in G. Somers & M. H. Pinsonneault (2015) using the formalism from H. C. Spruit & A. Weiss (1986), and presented as a set of starspot isochrones in G. Somers et al. (2020). Here, we explore the present capabilities of YREC to study cool stars with starspots.

In Figure 16, we show the A. W. Mann et al. (2015) sample of nearby field stars with empirically-calibrated masses and radii, with a set of solar-metallicity starspot isochrones overplotted at several representative ages (100 Myr, 300 Myr, and at ZAMS) and spot filling factors (0, 25%, and 50%). These models have gray atmospheres, are non-rotating, and have diffusion turned off, similar to the test suite case. The starspot isochrones trace the A. W. Mann et al. (2015) sample very well; though there is some metallicity variation which is not captured in this plot, many stars in the A. W. Mann et al. (2015) sample appear to depart from the solid lines, and their locations on this diagram are qualitatively better fit with some degree of magnetic inflation from starspots.

We subsequently run a series of runs observing the same microphysics conventions from the G. Somers et al. (2020) paper to produce Figure 17: a set of lithium depletion patterns at 100 Myr, using the convention that $A(\mathrm{Li}) = \log_{10}(\texttt{Li7\_sur}) + 12$. In this plot, we show non-rotating non-diffusive YREC models with a linear ramp from the Allard models at 0.4 $M_\odot$ to the Castelli–Kurucz models at 0.6 $M_\odot$. In the top panel, we show the variation of the lithium depletion curve with respect to metallicity, with an unspotted and a spot-saturated model ($f_{\rm spot} = 25\%$ plotted). We see that the effect of spots is similar in some respects to a decrease in metallicity, though the difference in the shapes of the distributions is more complicated in detail. The bottom plot shows lithium depletion patterns for a solar metallicity case with different starspot filling factors, showing that the effect of spots is also roughly to mimic a depletion pattern at cooler temperature. This is expected, because spotted model stars have cooler convective zone bases as a consequence of magnetic inflation (L. Cao et al. 2026).

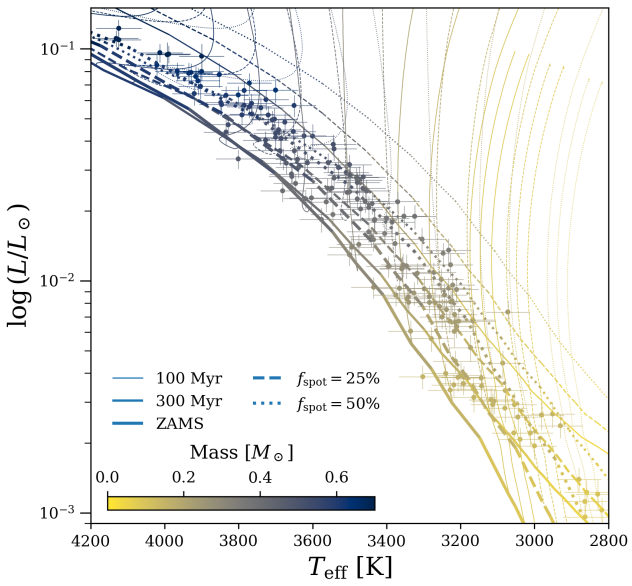

Figure 16. Starspot isochrones at solar composition evaluated at 100 Myr, 300 Myr, and the ZAMS (defined where $X_{cen}/X_{cen,0} = 0.993$); and 0 (solid), 25% (dashed) and 50% (dotted) spot filling factor. Mass tracks are shown for constituent tracks evolved to the ZAMS. The A. W. Mann et al. (2015) sample is overplotted with reported errors. Colors are assigned by the stellar mass, as shown in the colorbar.

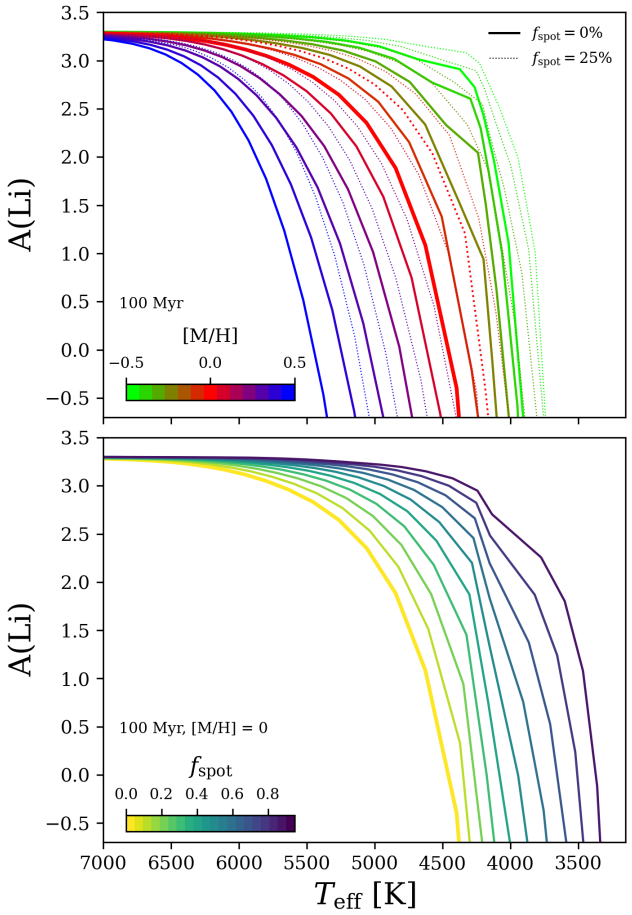

Figure 17. Starspot lithium isochrones evaluated at 100 Myr for various metallicities and starspot filling factors. Top plot shows the variation of the lithium abundance pattern with metallicity (between -0.5 and +0.5, stepping by 0.1 dex), for unspotted (solid) and 25% spotted (dotted) cases. Bottom plot shows the variation of a 100 Myr solar metallicity lithium abundance pattern with starspot filling factor from f_{spot} = 0.0 to 0.9, stepping by 0.1.

3.4.6. *Rotation Profiles in Evolved Stars*

While many main sequence stars appear to be rotating as close to solid bodies by the end of the main sequence, it is not clear that red giant branch stars with large envelopes and shorter lifetimes should do the same. In fact, the available asteroseismic results suggest that the cores of these stars are rotating significantly faster than their envelopes (J. Tayar et al. 2019; G. Li et al. 2024). It is therefore useful to be able to parameterize the radial rotation profiles in these stars in order to explore various limiting cases. YREC offers the option to force the entire star to rotate as a solid body or conversely, to conserve angular momentum within each zone (J. Tayar & M. H. Pinsonneault 2013). There are also options to enforce an angular momentum transport timescale, or allow only hydrodynamic mechanisms to act (G. Somers & M. H. Pinsonneault 2016). Finally, YREC also has the option to allow differential rotation in convection zones, where the differential rotation profile has the rotation frequency ω scaling with the radial coordinate r to some power between -2.0 (constant specific angular momentum) and 0.0 (solid body rotation). When setting up the differentially rotating convection zone, the code also has the option to enforce solid body rotation below the surface convection zone, as well as the option to force the entire interior to rotate at the rate of the base of the convection zone. We show examples of these various cases for a 3 $M_\odot$ star halfway through core-helium burning in Figure 18.

4. SCRIPTS

The YREC code was designed to run single initial value problems to a specified endpoint. However, there are a number of cases where scripts are essential or convenient. We provide tools and scripts in the YREC Wrappers folder[14] of the YREC GitHub repository. Within this repository are the `modelgrid_tools/main_tools` repository, where the

[14] https://urldefense.com/v3/__https://github.com/yreclab/yrec/tree/main/wrappers/modelgrid_tools

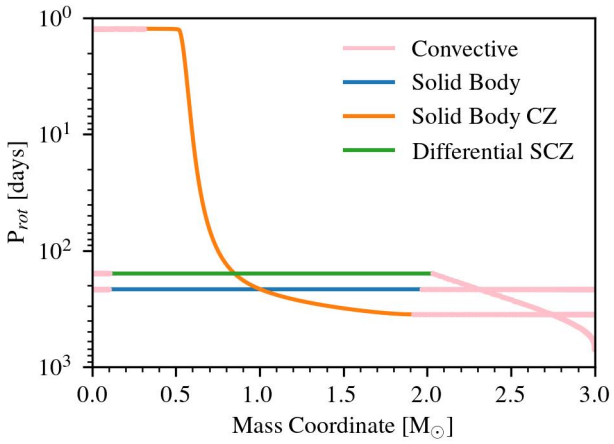

Figure 18. Radial rotation profiles for a 3 $M_\odot$ star halfway through helium burning. Convective regions are pink in all cases. We enforce various rotation profiles including solid body for the whole star (blue), a solid body interior and a differentially rotation surface convection zone (green), and solid body convection zones but a differentially rotating radiative region (orange).

base tools for new users are found. In addition, we feature the `modelgrid_tools/alternate_tools` repository, which houses our more advanced tools for experienced users and large workflows. We recommend starting with the `main_tools` repository and using the `alternate_tools` for large workflows, parallel processing, and backups in case the file readers in the `main_tools` fail. All scripts and wrappers are written in Python unless otherwise noted. We provide links to and brief descriptions of these tools in Appendix A. The specific tools released with this paper include:

- A means of building the executable code for different compilers and common compile options.
- A namelist converter, to customize the output and input directories for the code.
- A YREC output file reader.
- Scripts for running grids of models with different initial mass, composition, or rotation rate.
- A script to calibrate solar models, using the results of prior runs to estimate best parameters for future ones.
- Checks on the internal consistency of the input physics tables, used prior to running the code.
- Checks on the numerical precision of the calculations.
- Scripts for running the code in parallel, on multiple processors.

There are three other families of tools that we discuss in more detail here. The first is our procedure for restarting models after the He flash, which is more involved than typical YREC use cases. The second is the linkage of YREC with the `kiauhoku` tool, in the context of building isochrones from user-supplied grids of stellar models. We also discuss the `rotevol` code, which is a fast and efficient tracer code for generating angular momentum evolution scenarios from grids of standard stellar models.

Our final sub-section contains some sample exercises, illustrating the power of YREC as a teaching tool.

4.1. *RGB Tip to ZAHB*

Users may generate custom zero-age horizontal branch models by re-scaling the envelope mass and the core mass of the YREC seed ZAHB models to match those of a user-generated tip-of-the-RGB model. In this way, a user may run a physically consistent calculation from the pre-MS to the ZAHB, bypassing the helium flash. An example of the envelope mass rescaling procedure is provided in the test suite (see Table 2). An example is also provided for the core mass rescaling procedure in the context of the solar mass evolution run listed in Table 2. This mass rescaling approach has been used in the latest PARSEC v2.0 models (C. T. Nguyen et al. 2025b), and may be important, depending on the use case, to preserve consistency of the ZAHB models with those from previous stages of evolution (e.g., for isochrone generation purposes).

4.2. *Tracer Angular Momentum Evolution Grids*

For low mass stars there are a number of degrees of freedom in the model for angular momentum loss and in the initial conditions for rotation. The parameters in the model are usually constrained empirically, by matching the observed time evolution in star clusters against suites of theoretical models with multiple degrees of freedom.

For example, angular momentum loss is a steeply increasing function of rotation rate with a saturation threshold. Slow rotators are insensitive to this saturation threshold for the angular momentum loss law, but rapid rotator histories are strongly influenced by it. Core-envelope decoupling, by contrast, is a mass-dependent process whose primary observational signature is the spin down of slower rotators. At late stages, the asymptotic slope depends on the assumed dynamo scalings, while the zero point of the loss model sets the absolute value of the asymptotic trend. It can therefore

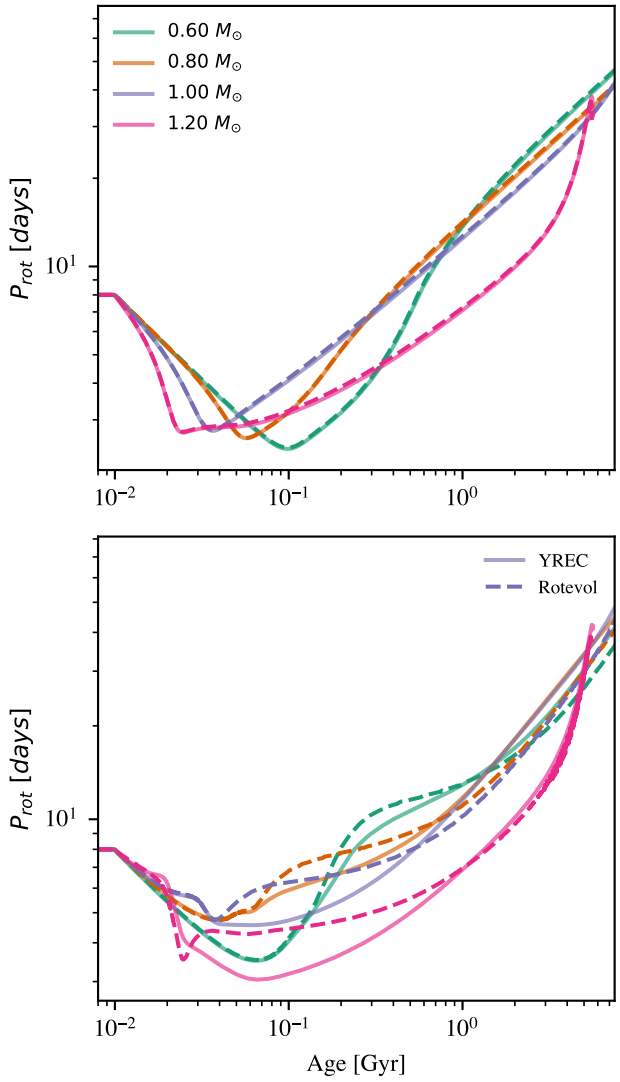

Figure 19. A comparison between full YREC rotation models and `rotevol` forward models for tracks with masses 0.6, 0.8, 1.0, and 1.2 $M_\odot$. The upper panel shows the comparison between the predicted rotation periods under the solid body approximation. The lower panel shows the difference in predicted rotation periods for non-solid body rotation with core-envelope coupling timescales given by P. A. Denissenkov et al. (2010b).

be useful to quickly generate large suites of models, with the intention of obtaining a best fit solution in a higher dimensional space. For slowly rotating stars the structural effects of rotation can be usefully treated as modest perturbations to the structure. We have therefore developed a program, `rotevol`, which reads in model tracks generated without rotation, and generates what-if scenarios for surface rotation as a function of time. `rotevol`, in turn, can be called by scripts that perturb the assumptions in the physical model and search for a best-fit solution relative to a user-defined figure of merit. `rotevol` has already seen extensive use in the literature (J. L. van Saders & M. H. Pinsonneault 2013; J. L. van Saders et al. 2016, 2019; G. Somers et al. 2017; T. S. Metcalfe et al. 2020; N. Saunders et al. 2024; F. Chiti et al. 2024; Y. Li et al. 2025; L. Cao et al. 2023; L. Cao & K. G. Stassun 2025); this represents the first public release of these scripts.

In addition to the test suites, we have included scripts which allow the user to run `rotevol` on pre-computed model grids. These scripts include functionalities for generating rotation tracks for different physical parameters and running a grid of models through `rotevol`. As `rotevol` forward models the rotational evolution of a pre-computed stellar track, one of its main advantages is speed. Compared to a full YREC run with rotation, `rotevol` reduces the computation time by 2 orders of magnitude. `rotevol` takes on order a tenth of a second to compute the rotation evolution of a mass track.

Using `rotevol`, we have computed comparison cases between rotating YREC models and `rotevol` forward models for stellar masses between $0.2M_\odot$ and $1.2M_\odot$. Figure 19 compares the predicted rotation periods from YREC and `rotevol` for the two rotation scenarios discussed in Section 3.4.4: solid body rotation and core-envelope decoupling. The figure shows `rotevol` runs with identical braking law parameters on tracks computed with identical physics to those in Section 3.4.4, but with rotation off. The solid body case produces excellent agreement (within$\approx$ 5%) in the predicted rotation periods compared to fully rotating YREC models. The `rotevol` models with core-envelope decoupling show disagreements up to $\approx 20\%$ for stellar masses $\geq 1.1\ M_\odot$—discrepancies that are not unexpected, given that the underlying physical treatment of the internal angular momentum transport is different (see also P. A. Denissenkov et al. 2010a; G. Somers & M. H. Pinsonneault 2016; F. Spada & A. C. Lanzafame 2020).

Users can find the `rotevol` code in the examples folder of the YREC release code. This folder includes scripts for running `rotevol` in non-rotating YREC tracks found in the sample output folder.

4.3. *ISTARY - The YREC Stellar Isochrone Set*

Isochrones are elegant tools for studying stellar populations. Here we present a sample set derived from our grid, a guide for isochrone creation in the form of a Jupyter .ipynb file, and a visualization of the results. The **I**sochrones and **S**tellar **T**racks for **A**strophysics **R**esearch by the **Y**REC Collaboration (ISTARY)[15]

[15] https://github.com/avincesmedile/yrec_isochrones

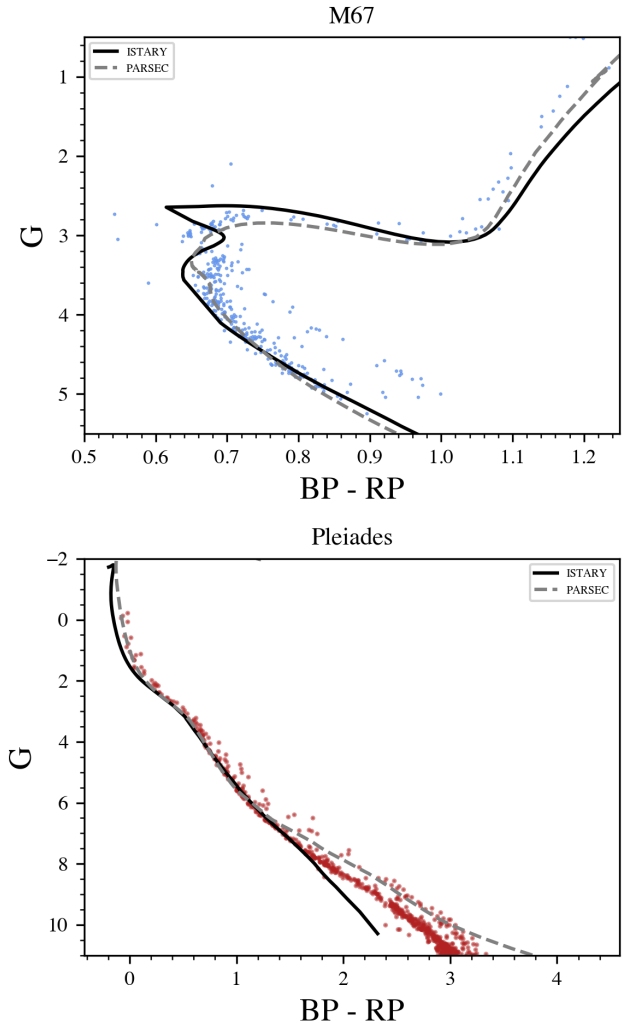

Figure 20. A CMD example of the isochrone set generated from the YREC sample grid. We plot the absolute Gaia DR3 G mag and Gaia DR3 BP-RP. Isochrones (black) were computed using the `kiauhoku`-generated EEPs derived from the YREC main grid. We adopted solar metallicity for M67 and an age of 3.95 Gyr (C. Reyes et al. 2024), and Fe/H = 0.03 with an age of 0.125 Myr (S. Gossage et al. 2018) for the Pleiades. Data for M67 and the Pleiades were taken from C. Reyes et al. (2024); M. G. J. van Groeningen et al. (2023), respectively. Isochrones are compared with Pleiades data (bottom panel) and M67 data (top panel).

isochrone set was produced using our custom isochrone functions in conjunction with the `kiauhoku` stellar interpolator code from Z. R. Claytor et al. (2020a). `kiauhoku` takes a grid of YREC .track files and converts it into a multi-index pandas dataframe by common initial mass, mixing length, model number, and metallicity. Then, it converts this dataframe into equivalent evolutionary phases (EEPs) using the method by A. Dotter (2016). We then create isochrones using a custom wrapper, `make_iso()`. ISTARY and our isochrone creation guide can be found in the YREC Github repository.

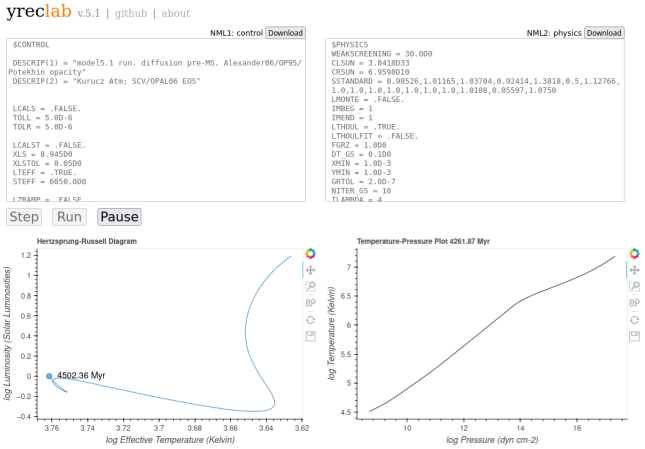

Figure 21. The `yreclab` web interface, showing a run in progress, with toggles to "step" (evaluate one timestep), "run" (evaluate until paused or execution ends), and "pause" the execution.

Our sample isochrones are for [Fe/H] = 0.0 and cover $0.250 \leq \text{Gyr} \leq 13$, although others can be generated with the same machinery. We compare with data in the Pleiades and M67 open clusters, which we show in Figure 20. The isochrones here were converted into bolometric magnitudes using the YBC Code Y. Chen et al. (2019). For the Pleiades, we employ individual distances for members derived from Gaia DR3 parallaxes, and utilize an extinction value of $A_v = 0.1$. We find good agreement between ISTARY, PARSEC, and the data up to the lower MS, where known discrepancies between the Pleiades and isochrones occur Y. Chen et al. (2019). For M67, we used the distances (an average distance modulus of 9.614) and extinctions from C. Reyes et al. (2024). We find good agreement between ISTARY, PARSEC, and the M67 data.

4.4. *YREC as a Tool for Stellar Structure and Evolution Classes*

Stellar structure and evolution is a core topic for undergraduate and graduate education in astronomy. The two topics—structure and evolution—demand different approaches. Stellar structure is deeply tied to the physics of stars, and most instructors adopt a semi-analytic approach close in nature to the tactics used in core physics courses. Stellar evolution, however, is more typically treated as a distillation of results. YREC is an excellent tool for both structure and evolution. Students can compare the idealized results from analytic structure calculations to those from full models, and they can generate their own evolutionary histories. We have provided some exercises in the examples folder. The first exercise

compares pressure as a function of density, temperature and composition to that predicted by analytic expressions used in classes, such as those for radiation pressure or the ideal gas law. The second exercise illustrates the effects of significant variations in entire families of rates (CNO and pp) as opposed to selective changes in single rates (pp and $N^{14}+p$.). The third has students compare analytic estimates for the equilibrium abundances of elements that participate in CNO burning to their model abundances.

Another unique teaching application of YREC is the web laboratory, also known as `yreclab`. Due to the efficient design of the code and its lightweight nature, it has been possible to compile YREC to WebAssembly with the code `emscripten` (A. Zakai 2011), which allows it to be run interactively client-side as a modern single-page web application hosted on Github Pages[16] (L. Cao & M. H. Pinsonneault 2026). An example run is shown in Figure 21. `yreclab` fully supports the current YREC implementation and its namelists can be modified just like the desktop version of YREC. Modifications to the code[17] allow it to be run in an interpreted state, and the stellar evolution code can be directly controlled via the Javascript scripting language—allowing it to be paused and restarted to examine the models of a run in progress. During any point of the evolutionary run, the virtual file system allows all files created during the run to be downloaded.

These architectural choices make `yreclab` uniquely suited for teaching & experimentation in stellar astrophysics (L. Cao & M. H. Pinsonneault 2026). Existing efforts such as `MESA-Web` (C. E. Fields et al. 2023) have been successful at providing a platform for teaching stellar physics, but require calculations to be submitted for later download by email, and have a limited subset of toggles available on their calculation submission page. With `yreclab`, the scriptable WebAssembly platform allows calculations to be supervised and interacted with at real-time, at near native desktop speeds (within a factor of two), with full support for the same namelists available in the desktop version of YREC.

5. SUMMARY

Stellar astrophysics is a rich topic, and it has deep connections across astronomy writ large. Stellar evolution codes are natural tools for understanding the life cycles of stars, which is why a number of were developed over the years. However, the list of public codes is short, and different codes are optimized for different kinds of stars.

In this paper we present a public release of the YREC stellar evolution code. YREC has been primarily used for the evolution of low and intermediate mass stars, and in that context it is a state-of-the-art code with sophisticated options. It can be run for a wide range of conditions, but caution should be taken when using any tools outside of the domains where development has been focused. We begin by summarizing the key domains for use, followed by the road map for improvements and updates. For the discussion below, we distinguish between brown dwarfs ($M <\sim 85 M_J$); low mass stars with deeper convective envelopes ($\sim 1.3 M_\odot > M >\sim 85 M_J$), intermediate mass stars ($\sim 8 M_\odot > M >\sim 1.3 M_\odot$), and high mass stars ($M >\sim 8 M_\odot$), with these ranges defined at solar metallicity.

YREC has been extensively used for low mass stars, starting from the pre-MS through core He burning. It can be used for hotter brown dwarfs, but more realistic models are limited by the surface boundary condition and equation of state tables to either younger ages or higher mass. In its current form we recommend caution below $30 M_J$.

The code requires a starting model, but fortunately rescaling allows users to generate starting models with mass and composition different from those in the library. There is also no rule against exporting models from other codes to YREC. Our initial conditions, on the upper Hayashi track, begin in a fictional place. Fortunately, it's also a useful one, as these models can be rescaled and evolved to a more physically reasonable deuterium birthline. Even in this case, real models would accrete up the birthline, so the real very early evolution could be quite different than that in the models. This has limited impact on classical model properties, but it's important for evolving models with rotation. Artificially large starting models will have extravagant angular momenta, so care needs to be taken to choose a proper point for initializing such models.

The code allows a range of angular momentum evolution models, including the structural effects of star spots, which will be especially useful for gyrochronology and rotationally induced mixing. Strong magnetic coupling, such as that inferred from the Taylor-Spruit instability, can be effectively modeled as solid body rotation for these purposes. Wave-driven angular momentum transport is not currently supported in the code. YREC makes state of the art solar models.

The code also treats microscopic diffusion well in the low mass star domain. The primary limitation is that metals are diffused at the same rate. We note, however, that this is of order a 1% level effect for solar models. The code crashes at the He flash, but our scripting tools

[16] https://yreclab.github.io/
[17] https://github.com/yreclab/yreclab

can be used to restart evolution in the core He-burning phase. The current code does not include support for the thermally pulsing AGB phase, nor does it include thermohaline mixing. The code does include mass loss, but it is being re-worked and the current version is not supported.

For intermediate mass stars, numerical issues arise with microscopic diffusion in stars with thin surface convection zones. These can be avoided by setting the outer fitting point deep enough, at the cost of shutting the microscopic diffusion calculations off. The detailed behavior of the code in the outer layers will depend on the location of the outer fitting point, which should be chosen with care. Note that this has very little impact on ages, or standard isoschrones, as radiative envelope stars are insensitive to the treatment of the outer boundary condition. YREC has limited options for the treatment of convective boundaries, which can become important in this mass domain. Users interested in rotation should ensure that the initial angular momenta of the models are reasonable for the stars under consideration.

The upper mass range of YREC is set by the onset of radiation-pressure driven winds. These induce negative pressure in the atmosphere solutions above 16 solar masses - a physical, but regrettable, feature in a system with logarithmic co-ordinates. YREC at present does not include radiation pressure driven winds, nor does it include nuclear burning stages beyond helium burning. It does capture the broad properties of models in the domain where it runs; above 12 solar masses, the code should be used with caution.

YREC will run with a wide range of birth mixtures, although the input physics tables in general do not support changes in the heavy element mixture. The maximum Z for the tables is 0.1, and users should be prepared to face challenges in the limit where the metal content is taken to zero. The isotopes and species that the code can track are hard-coded at present, although work is underway to generalize this.

The code continues to be actively developed. We intend to overhaul and bring the code up to date for the treatment of convective boundaries and mass loss. More generally, we are looking at adding in well-posed treatments of important physical processes. We are also in the process of evaluating and updating the basic input physics. This is an effort where we welcome external effort. Over a longer term, we intend to make the code easier to modify, and to generalize some of the hard-coded components that could make such changes more challenging. It is our hope that this code proves to be useful, and we welcome feedback.

ACKNOWLEDGMENTS

We thank an anonymous referee and Earl Bellinger for helpful comments on the manuscript. We thank Yaguang Li for example MESA inlists.

The YREC stellar evolution code has been the product of decades of effort and improvements from a variety of contributors. Many, but not all, of these contributions are preserved in comments in the code. Major contributors in addition to the authors on this paper include Pierre Demarque, Richard Larson, Michael Prather, Don van den Berg, David Guenther, Grant Newsham, Brian Chaboyer, and John Bahcall. Lawrence Pinsonneault also made major contributions to the organization and documentation of the code.

MHP acknowledges support from NASA grants 80NSSC24K0622 and 80NSSC24K0637. JT and LM acknowledge support from 80NSSC22K0812. JvS acknowledges support from NSF grants AST-2205888 and AST-2507789. ZRC acknowledges support from NASA ROSES 80NSSC24K0081. LC acknowledges support from the Vanderbilt Initiative in Data-intensive Astrophysics (VIDA) and NASA Grant 80NSSC24K0622.

AUTHOR CONTRIBUTIONS

MP is responsible for a large fraction of the YREC stellar evolution code. He also took primary responsibility for writing the text and developing the test suite, as well as documenting the code. JvS contributed to code testing and development, gyrochronology example cases, and documentation. LC contributed to code development, documentation, starspots example cases, test suite validation, and the `yreclab` web interface. JT contributed to the development of the radial differential rotation modules, the available atmosphere boundary condition options, assisted with grid tool development, and participated in writing and documentation. FD contributed opacity tables, some writing of the code, and he participated in the writing of the manuscript. LM contributed to the grid tool development, generation of the base stellar model grid, and writing of the relevant sections of the manuscript. RAP tested the structural and evolutionary behavior of high-mass models with various modifications to convection and contributed to associated portions of the manuscript. VAS wrote the scripts for installing the base model grid into the `kiauhoku` stellar model interpolator, the guide to creating isochrones with either a custom or the base YREC grid, and the ISTARY code/library. In addition, VAS organized the user tools repository, contributed to grid tool development, and wrote relevant sections of the manuscript. SB contributed to the grid tool development. AA contributed to the scripts for the angular momentum tracer

code and the relevant manuscript sections. KC participated in the maintenance and modernization of the YREC code and related discussions. MR developed the build system, test suite executor, hunted down bugs in the YREC code, and contributed to relevant sections of the manuscript. ZRC authored the `kiauhoku` package, advised on its usage, and provided feedback on the manuscript. JCZ contributed to horizontal branch model testing and associated manuscript writing.

Software: astropy (Astropy Collaboration et al. 2013) `kiauhoku` (Z. R. Claytor et al. 2020a,b),

APPENDIX

A. SCRIPTS AND TOOLS

A.1. *Building YREC*

Users can download YREC from https://github.com/yreclab/yrec. YREC is implemented entirely in Fortran 77 and is straightforward to build. The code is Fortran 90 compliant. A simple build system is provided to create binaries of YREC for several different purposes. Two major compiler families are currently supported, the GNU Fortran compiler (gfortran) and the Intel Fortran compiler (ifort, ifx). Building the software is supported on Linux, Mac OS, and Windows. The default build configuration is set to generate a program with the best performance for large scale model runs. Development and debug build configurations are also provided which may be useful to those who wish to modify or further develop YREC.

A.2. *Namelist Converters*

The YREC control namelists (nml) contain file paths to input files containing physical parameters and input/output controls. Users can run examples using the relative paths provided, or edit the locations of input and output files manually or by utilizing the `change_nml()` or `update_nml` codes. We provide a simpler namelist tool for new users, `update_nml`[18], which allows users to modify namelists one variable at a time using command-line prompts and inputs. In addition, we provide the more advanced `change_nml`[19] tool, which can intelligently modify multiple namelist files at once to specify input and output data locations for YREC. After modification, users can run their YREC namelists either from the command line or via the Python wrappers introduced here for coordinated parallel runs, with the outputs deposited in an organized fashion.

[18] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/main_tools/update_nml.py

[19] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/alternate_tools/change_nml.py

A.2.1. *YREC Output Reader*

YREC returns a set of output files, such as the .track files, which detail the evolution of the star from the designated starting point. The ability to read batches of such files can be useful. We provide functionality to locate and read large batches of files into Python programs for manipulation with pandas or astropy. We note that some minor modifications of the source code, discussed in the package, are required to use this option.

The wrappers themselves—`load_yrec_tracks()`[20] and `tracker()`[21]—are provided to users to read in .track files. `The tracker()` function can be used together with `load_yrec_tracks()` or on its own. Users provide a filepath to a .track file and it is read into Python as a Pandas DataFrame. The load_yrec_tracks() function takes a filepath to a folder containing all the outputted files. The code either uses the immediate directory or searches all subdirectories for .track files (to be used if the nml were modified using change_nml() function). Optional functionality is given to manipulate the stored data to create EEPs and Isochrones and return them alongside the main dataset.

A.3. *Model Grid Generators*

We provide wrappers that can be used to load the input physics and namelists and execute YREC to generate various grids of models. We provide the means to recreate our sample model grid (`make_modelgrid.py`)[22], as well as generate a grid of models with the same input physics but a range of masses and metallicities via Python (`batchrunner.py`)[23] and Slurm (`run_yrec_grid.slurm`)[24] scripts to run a grid. In ad-

[20] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/main_tools/load_yrec_tracks.py

[21] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/main_tools/Tracker.py

[22] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/main_tools/make_modelgrid.py

[23] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/main_tools/batchrunner.py

[24] https://github.com/avincesmedile/YREC-Wrappers/tree/main/modelgrid_tools/slurm_tools

dition to this, we also provide the `yrec_parallel.py`[25] script for running large batches of models on different CPUs of the users' native machine. This tool supports systematic variation of input values to explore a given physics parameter space across many independent YREC runs. For all batch scripts, users provide the path information for their YREC installation and namelists, then specify the masses and Fe/H they wish the grid to cover. The codes run the YREC executable with the corresponding namelists. Forthcoming additions include an option to generate ranges of birth rotation rates, which requires custom nml2 files as well as custom nml1 files.

A.4. *Solar Model Calibration*

Our solar model calibrations illustrate the power of scripting YREC execution. We provide the `solar_rot_calibrated.py` script[26] to this end. The code performs a template solar model run, including rotation, with user-specified initial X, Z, mixing length α, birth rotation rate, angular momentum loss scale factor f_k, and mixing efficiency parameter f_c. The user also specifies the solar M, Z/X, L, R, age, surface rotation period, and the target solar lithium depletion. In the classical or diffusion model case, the script evaluates L and R, and iterates X, Z, and α until they match the solar values within a user-specified tolerance. This is done with partial derivatives of L and R with respect to X and α taken from prior solar model runs; Z is adjusted to the user-specified Z/X. Users can also specify their own partial derivatives if desired. In the case where diffusion is included, the script adds a check on the surface Z/X to ensure that it matches the user-specified current value. In a classical model, the birth value is the same as the current value, so no such check is performed. Finally, the model with rotation adds the requirement that the model surface rotation rate and logarithmic lithium depletion match the solar values. This is done by adjusting the zero point of the angular momentum loss rate f_k and the efficiency of mixing with respect to angular momentum transport f_c, respectively.

A.5. *Tolerance Calibration*

These scripts perturb the spatial and temporal resolution of a specified run and infer the change in a figure of merit. This can be used to define the numerical controls required to meet a goal. In Figure 22 we show an example for the models used to construct the base grid; in this case, our figure of merit was the age at TAMS. We recommend using the `update_nml()` or `change_nml()` functions to make a model run and change the spatial and temporal tolerances as we do in Figure 22, and choose a metric of merit (such as Age in our case) to compare the end-of-run values to.

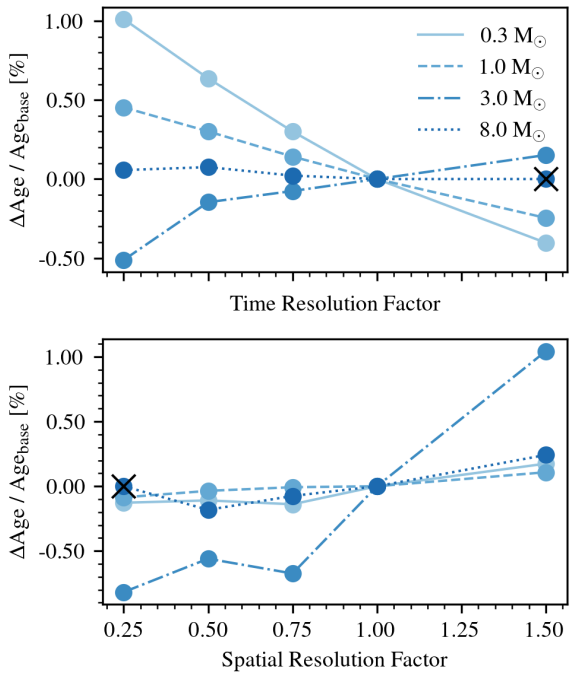

Figure 22. Spatial and temporal resolution tests for solar metallicity models at 0.3, 1.0, 3.0, and 8.0 $M_\odot$. The tolerances from the grid namelists were multiplied by the indicated factors, and the resulting percent change in the TAMS age is show relative to the base-resolution model. X's denote failure to converge.

B. TROUBLESHOOTING YREC

The YREC code will fail to execute in some cases, and it can be helpful to understand the origin of some of the failure modes. The .short output file contains very useful diagnostic data in case of a failure, and it should be consulted. Here are some common problems and solutions:

B.1. *Failure to converge*

The code will apply corrections to the structure variables iteratively. If it has not converged within the tolerances in a user-specified number of attempts, it will stop. Sometimes this can be solved by simply increasing the maximum iterations; this may be fruitful if the corrections are decreasing slowly in amplitude. More often, they reach a flat amplitude and change sign. The

[25] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/alternate_tools/yrec_parallel.py

[26] https://github.com/avincesmedile/YREC-Wrappers/blob/main/modelgrid_tools/main_tools/solar_rot_calibrated.py

most common cause is that the spatial or temporal tolerances are too loose, such that the solution cannot reach the desired precision. In some cases, changing the size of the envelope triangle can also help, as this influences the accuracy of the surface boundary conditon interpolator. Users may also relax the convergence criteria; be cautioned, however, that tolerances looser than the 10^{-4} level can produce erratic behavior in global properties.

Sometimes the code will cascade down to an extremely small timestep, which always results in a convergence failure. In this case, the solution is counter-intuitive: the timestep criteria need to be relaxed. In the limit of very small timesteps, the corrections become very sensitive to numerical noise.

The third and fourth levels of iteration can produce convergence failures in evolved red giant branch models; we recommend that users set these to 0 unless they are explicitly needed to study rotational mixing.

B.2. *Hard failure modes.*

Other failures are tied to the assumptions in the code, and they are not numerical in nature. The current code fails if both overshoot and semi-convection are used at the same time. Above $\sim$ 16 solar masses on the main sequence, gray atmospheres models will have a net negative gas pressure due to the onset of radiation pressure driven winds, which YREC does not currently account for. The code will also fail if rotation is so rapid that material at the equator is unbound; users should check that the internal rotation profile does not fail this criterion. Frequently this condition is triggered by a poor choice of starting conditions (see below). Finally, the code cannot evolve through a degenerate helium flash. This failure mode will manifest as runaway corrections, and it cannot be solved without changes to the code itself.

B.3. *Challenging domains*

Stellar physics is the most uncertain at very high and very low masses. We are actively working on the code, and anticipate progress on the phase space where it can be used. Below we document the current domains where YREC is either difficult to run or not well suited given its current assumptions.

For high mass stars, a combination of missing physics and numerical challenges makes it difficult to run models above 9 solar masses through core He exhaustion.

Models with rotation can be run across the full mass range. However, rotationally induced mixing is too efficient for stars above the Kraft break, and we do not recommend running models with mixing in this domain.

Models with microscopic diffusion encounter numerical difficulties when the hydrogen or helium mass fractions approach zero in the envelope. We therefore also recommend that the code not be used with diffusion enabled above ~ 1.4 solar masses at solar metallicity.

Brown dwarfs and stars close to the hydrogen-burning limit are numerically challenging, both due to the uncertain EoS and the importance of the surface boundary conditions for the solution. We do not recommend using the Yale EoS below $0.5M_{\odot}$, or using a gray atmosphere below $0.3M_{\odot}$. The F. Allard & P. H. Hauschildt (1995) grid is the currently recommended choice for low mass stars.

We also note that models close to $0.1M_{\odot}$ can encounter challenges because their physical conditions skirt close to the edge of published equation of state tables, making their structure sensitive to the ramping between different EoS sources.

Core helium burning stars may not achieve $Y_{\rm cen} = 10^{-4}$, reaching temperatures that may be outside the bounds of interior opacity tables before reaching low central helium abundance. The equilibrium solution for the seed ZAHB models used for the low-mass two-step rescaling procedure is also sensitive to the choice of neutrino cooling physics, and users are cautioned that their model may not run if attempting to use N. Itoh et al. (1996) rates, which are otherwise the standard for YREC. All low-mass core helium burning model namelists in the test suite set `LNULOS1` to false, which results in defaulting to rates from J. N. Bahcall (1989).

B.4. *Choices that can lead to sorrow.*

Our digital constructs lack imagination and foresight, merely doing what they are asked to do. It is therefore in the nature of software to fail. Some of these failures reflect poor design. There are times, however, where the fault lies not in the stars, but in ourselves (Shakespeare 1599). Here we list, from personal experience, lessons that users of the code should take to heart.

There may be a time when computers can read our minds, doing the things that we wish to be done. We do not yet live in such a time. If you do not tell YREC where to read from, or where to write to, it will fail. Check your pointers.

The code features a lovely variety of options, but human imagination is far greater in scope. If you include variables that the code does not recognize in the namelists, it will quit in disgust. Ensure that your comments are clearly labelled as such, and that the version of the code that you are using supports the choices that you are making.

It is tempting to use a single template to clone a series of runs, changing the output pointers to produce large grids of models. It is important to check that you are

also changing the inputs, such that the files that you produce are what you think they are. In a better world, you would always get what you desire. In this one, you get what you deserve.

The code does, in fact, provide diagnostic outputs. When it fails, or misbehaves, reading the messages on the screen, or those stored in the short file, can lead to enlightenment. Or not, of course.

If you assign the wrong input to the wrong name, the code will become confused at this offense to the natural order of things and quit. However, if you assign a file with the right format, but the wrong physics, to the right place—say, for example, an atmosphere table with a metal content different from that of the star—it will assume that you have wisdom, and it will allow you to proceed without comment. Ask yourself whether this trust is warranted.

Conservation laws are the foundation of modern physics, and YREC respects these iron rules. If you start with a very large star, and give it a modest spin, it will quickly become unruly as it shrinks and spins up. At some point it may become unbound at the equator—a situation that YREC finds intolerable, as it cannot conceive of imaginary numbers. Check your starting point for models with rotation. It is easy to give stars far more angular momentum than they can find a use for.

While we are on the subject of imaginary numbers, recall that the code uses logarithmic variables. Zero is a special number. Before you set something in the code to zero, some introspection is required. Do you really need that number to be zero, or will a very small number suffice instead? Do you feel lucky? Be prepared to experiment and learn. In some cases, the code will anticipate folly and quit. In other, more devious cases, it will not, and it may break in unusual ways.

Finally, a successful run of the code is the beginning, not the end, of the journey. You should plot your results and see if they appear sensible. You should change your numerics and see whether they change your answer to any interesting degree. If they do, ponder your life choices, and adjust. Ensure that the tables that you use are the ones you actually want to use. Ensure that the physics that you think that you are using are the same as the physics that you are, in fact, using.

REFERENCES

Acharya, B., Aliotta, M., Balantekin, A. B., et al. 2025, Reviews of Modern Physics, 97, 035002, doi: 10.1103/8lm7-gs18

Adelberger, E. G., García, A., Robertson, R. G. H., et al. 2011, Reviews of Modern Physics, 83, 195, doi: 10.1103/RevModPhys.83.195

Allard, F., & Hauschildt, P. H. 1995, ApJ, 445, 433, doi: 10.1086/175708

Amard, L., & Matt, S. P. 2020, ApJ, 889, 108, doi: 10.3847/1538-4357/ab6173

An, D., Pinsonneault, M. H., Terndrup, D. M., & Chung, C. 2019, ApJ, 879, 81

Angulo, C., Arnould, M., Rayet, M., et al. 1999, NuPhA, 656, 3, doi: 10.1016/S0375-9474(99)00030-5

Asplund, M., Amarsi, A. M., & Grevesse, N. 2021, A&A, 653, A141, doi: 10.1051/0004-6361/202140445

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33, doi: 10.1051/0004-6361/201322068

Badnell, N. R., Bautista, M. A., Butler, K., et al. 2005, MNRAS, 360, 458, doi: 10.1111/j.1365-2966.2005.08991.x

Bahcall, J. N. 1989, Neutrino Astrophysics (Cambridge University Press)

Bahcall, J. N., Basu, S., Pinsonneault, M., & Serenelli, A. M. 2005, ApJ, 618, 1049, doi: 10.1086/426070

Bahcall, J. N., & Pinsonneault, M. H. 1992, Reviews of Modern Physics, 64, 885, doi: 10.1103/RevModPhys.64.885

Bahcall, J. N., Pinsonneault, M. H., & Wasserburg, G. J. 1995, Reviews of Modern Physics, 67, 781, doi: 10.1103/RevModPhys.67.781

Barnes, S. A. 2010, ApJ, 722, 222, doi: 10.1088/0004-637X/722/1/222

Basinger, C., Pinsonneault, M., Bastelberger, S. T., Gaudi, B. S., & Domagal-Goldman, S. D. 2024, MNRAS, 534, 2968, doi: 10.1093/mnras/stae2280

Basu, S., Chaplin, W. J., Elsworth, Y., New, R., & Serenelli, A. M. 2009, ApJ, 699, 1403, doi: 10.1088/0004-637X/699/2/1403

Bildsten, L., Paxton, B., Moore, K., & Macias, P. J. 2012, ApJL, 744, L6, doi: 10.1088/2041-8205/744/1/L6

Böhm-Vitense, E. 1958, ZA, 46, 108

Bossini, D., Miglio, A., Salaris, M., et al. 2015, MNRAS, 453, 2290, doi: 10.1093/mnras/stv1738

Bossini, D., Miglio, A., Salaris, M., et al. 2017, MNRAS, 469, 4718, doi: 10.1093/mnras/stx1135

Bouma, L. G., Palumbo, E. K., & Hillenbrand, L. A. 2023, ApJL, 947, L3, doi: 10.3847/2041-8213/acc589

Bouvier, J., Forestini, M., & Allain, S. 1997, A&A, 326, 1023

Bressan, A., Marigo, P., Girardi, L., et al. 2012, MNRAS, 427, 127, doi: 10.1111/j.1365-2966.2012.21948.x

Bressan, A. G., Chiosi, C., & Bertelli, G. 1981, A&A, 102, 25

Burgers, J. M. 1969, Flow Equations for Composite Gases (Academic Press)

Campbell, B. 1984, ApJ, 283, 209, doi: 10.1086/162295

Cao, L., & Pinsonneault, M. H. 2022, MNRAS, 517, 2165, doi: 10.1093/mnras/stac2706

Cao, L., & Pinsonneault, M. H. 2026, submitted ApJ

Cao, L., Pinsonneault, M. H., Hillenbrand, L. A., & Kuhn, M. A. 2022, ApJ, 924, 84, doi: 10.3847/1538-4357/ac307f

Cao, L., Pinsonneault, M. H., & Sharifi, K. 2026, submitted ApJ

Cao, L., Pinsonneault, M. H., & van Saders, J. L. 2023, ApJL, 951, L49, doi: 10.3847/2041-8213/acd780

Cao, L., & Stassun, K. G. 2025, ApJL, 988, L1, doi: 10.3847/2041-8213/ade875

Caputo, F., Castellani, V., Chieffi, A., Pulone, L., & Tornambe, Jr., A. 1989, ApJ, 340, 241, doi: 10.1086/167387

Cassisi, S., Potekhin, A. Y., Pietrinferni, A., Catelan, M., & Salaris, M. 2007, ApJ, 661, 1094, doi: 10.1086/516819

Castellani, V., Giannone, P., & Renzini, A. 1971, Ap&SS, 10, 355, doi: 10.1007/BF00649680

Castelli, F., & Kurucz, R. L. 2003, in IAU Symposium, Vol. 210, Modelling of Stellar Atmospheres, ed. N. Piskunov, W. W. Weiss, & D. F. Gray, A20, doi: 10.48550/arXiv.astro-ph/0405087

Caughlan, G. R., & Fowler, W. A. 1988, Atomic Data and Nuclear Data Tables, 40, 283, doi: 10.1016/0092-640X(88)90009-5

Chaboyer, B., & Zahn, J. P. 1992, A&A, 253, 173

Chen, Y., Bressan, A., Girardi, L., et al. 2015, MNRAS, 452, 1068, doi: 10.1093/mnras/stv1281

Chen, Y., Girardi, L., Bressan, A., et al. 2014, MNRAS, 444, 2525, doi: 10.1093/mnras/stu1605

Chen, Y., Girardi, L., Fu, X., et al. 2019, A&A, 632, A105, doi: 10.1051/0004-6361/201936612

Chiti, F., van Saders, J. L., Heintz, T. M., et al. 2024, ApJ, 977, 15, doi: 10.3847/1538-4357/ad856c

Choi, J., Dotter, A., Conroy, C., et al. 2016, ApJ, 823, 102, doi: 10.3847/0004-637X/823/2/102

Christensen-Dalsgaard, J., Proffitt, C. R., & Thompson, M. J. 1993, ApJL, 403, L75, doi: 10.1086/186725

Claytor, Z. R., van Saders, J. L., Santos, Â. R. G., et al. 2020a, ApJ, 888, 43, doi: 10.3847/1538-4357/ab5c24

Claytor, Z. R., van Saders, J. L., Santos, Â. R. G., et al. 2020b, kiauhoku: Stellar model grid interpolation,, Astrophysics Source Code Library, record ascl:2011.027 http://ascl.net/2011.027

Cole, P. W., Demarque, P., & Deupree, R. G. 1985, ApJ, 291, 291, doi: 10.1086/163067

Constantino, T., Campbell, S. W., Christensen-Dalsgaard, J., Lattanzio, J. C., & Stello, D. 2015, MNRAS, 452, 123

Constantino, T., Campbell, S. W., Lattanzio, J. C., & van Duijneveldt, A. 2016, MNRAS, 456, 3866

Costa, G., Girardi, L., Bressan, A., et al. 2019a, A&A, 631, A128

Costa, G., Girardi, L., Bressan, A., et al. 2019b, MNRAS, 485, 4641

Costa, G., Shepherd, K. G., Bressan, A., et al. 2025, A&A, 694, A193, doi: 10.1051/0004-6361/202452573

Curtis, J. L., Agüeros, M. A., Matt, S. P., et al. 2020, ApJ, 904, 140, doi: 10.3847/1538-4357/abbf58

Cyburt, R. H., Amthor, A. M., Ferguson, R., et al. 2010, ApJS, 189, 240, doi: 10.1088/0067-0049/189/1/240

David, T. J., Hillenbrand, L. A., Gillen, E., et al. 2019, ApJ, 872, 161, doi: 10.3847/1538-4357/aafe09

Delahaye, F., & Pinsonneault, M. H. 2006, ApJ, 649, 529, doi: 10.1086/505260

Demarque, P., Guenther, D. B., Li, L. H., Mazumdar, A., & Straka, C. W. 2008, Ap&SS, 316, 31, doi: 10.1007/s10509-007-9698-y

Demarque, P. R., & Percy, J. R. 1964, ApJ, 140, 541, doi: 10.1086/147947

Denissenkov, P. A., Pinsonneault, M., Terndrup, D. M., & Newsham, G. 2010a, ApJ, 716, 1269, doi: 10.1088/0004-637X/716/2/1269

Denissenkov, P. A., Pinsonneault, M., Terndrup, D. M., & Newsham, G. 2010b, ApJ, 716, 1269, doi: 10.1088/0004-637X/716/2/1269

Dotter, A. 2016, ApJS, 222, 8, doi: 10.3847/0067-0049/222/1/8

Dotter, A., Chaboyer, B., Jevremović, D., et al. 2008, ApJS, 178, 89, doi: 10.1086/589654

Eggenberger, P., Deheuvels, S., Miglio, A., et al. 2019, A&A, 621, A66, doi: 10.1051/0004-6361/201833447

Eggleton, P. P. 1972, MNRAS, 156, 361, doi: 10.1093/mnras/156.3.361

Eggleton, P. P., Tout, C., Pols, O., et al. 2011, STARS: A Stellar Evolution Code,, Astrophysics Source Code Library, record ascl:1107.008 http://ascl.net/1107.008

Endal, A. S., & Sofia, S. 1976, ApJ, 210, 184, doi: 10.1086/154817

Farag, E., Fontes, C. J., Timmes, F. X., et al. 2024, ApJ, 968, 56, doi: 10.3847/1538-4357/ad4355

Ferguson, J. W., Alexander, D. R., Allard, F., et al. 2005, ApJ, 623, 585, doi: 10.1086/428642

Fields, C. E., Townsend, R. H. D., Dotter, A. L., Zingale, M., & Timmes, F. X. 2023, arXiv e-prints, arXiv:2309.15930, doi: 10.48550/arXiv.2309.15930

Fu, X., Bressan, A., Marigo, P., et al. 2018, MNRAS, 476, 496, doi: 10.1093/mnras/sty235

Fuller, J., Piro, A. L., & Jermyn, A. S. 2019, MNRAS, 485, 3661, doi: 10.1093/mnras/stz514

Gallet, F., & Bouvier, J. 2015, A&A, 577, A98, doi: 10.1051/0004-6361/201525660

Gehan, C., Mosser, B., Michel, E., Samadi, R., & Kallinger, T. 2018, A&A, 616, A24, doi: 10.1051/0004-6361/201832822

Gossage, S., Conroy, C., Dotter, A., et al. 2018, ApJ, 863, 67, doi: 10.3847/1538-4357/aad0a0

Grevesse, N., & Noels, A. 1993, in Perfectionnement de l'Association Vaudoise des Chercheurs en Physique, ed. B. Hauck, S. Paltani, & D. Raboud, 205–257

Grevesse, N., & Sauval, A. J. 1998, SSRv, 85, 161

Gruner, D., Barnes, S. A., & Weingrill, J. 2023, A&A, 672, A159, doi: 10.1051/0004-6361/202345942

Gully-Santiago, M. A., Herczeg, G. J., Czekala, I., et al. 2017, ApJ, 836, 200, doi: 10.3847/1538-4357/836/2/200

Heger, A., Langer, N., & Woosley, S. E. 2000, ApJ, 528, 368, doi: 10.1086/308158

Iglesias, C. A., & Rogers, F. J. 1993, ApJ, 412, 752, doi: 10.1086/172958

Iglesias, C. A., & Rogers, F. J. 1996, ApJ, 464, 943

Itoh, N., Hayashi, H., Nishikawa, A., & Kohyama, Y. 1996, ApJS, 102, 411, doi: 10.1086/192264

Jackson, R. J., Deliyannis, C. P., & Jeffries, R. D. 2018, MNRAS, 476, 3245, doi: 10.1093/mnras/sty374

Jeffries, R. D., Jackson, R. J., Sun, Q., & Deliyannis, C. P. 2021, MNRAS, 500, 1158, doi: 10.1093/mnras/staa3141

Jeffries, R. D., Jackson, R. J., Wright, N. J., et al. 2023, MNRAS, 523, 802, doi: 10.1093/mnras/stad1293

Jermyn, A. S., Bauer, E. B., Schwab, J., et al. 2023, ApJS, 265, 15, doi: 10.3847/1538-4365/acae8d

Kawaler, S. D. 1988, ApJ, 333, 236, doi: 10.1086/166740

Kippenhahn, R., & Thomas, H. C. 1970, in IAU Colloquium 4: Stellar Rotation, ed. A. Slettebak, 20

Kissin, Y., & Thompson, C. 2015, ApJ, 808, 35, doi: 10.1088/0004-637X/808/1/35

Koenigl, A. 1991, ApJL, 370, L39, doi: 10.1086/185972

Krishnamurthi, A., Pinsonneault, M. H., Barnes, S., & Sofia, S. 1997, ApJ, 480, 303, doi: 10.1086/303958

Kurucz, R. 1993, Robert Kurucz CD-ROM, 18

Lanzafame, A. C., & Spada, F. 2015, A&A, 584, A30, doi: 10.1051/0004-6361/201526770

Larson, R. B., & Demarque, P. R. 1964, ApJ, 140, 524, doi: 10.1086/147946

Ledoux, P. 1947, ApJ, 105, 305, doi: 10.1086/144905

Li, G., Deheuvels, S., & Ballot, J. 2024, A&A, 688, A184, doi: 10.1051/0004-6361/202449882

Li, Y. 2025, ApJ, 988, 179, doi: 10.3847/1538-4357/ade3c7

Li, Y., Bedding, T. R., Murphy, S. J., et al. 2022, Nature Astronomy, 6, 673, doi: 10.1038/s41550-022-01648-5

Li, Y., Huber, D., Ong, J. M. J., et al. 2025, ApJ, 984, 125, doi: 10.3847/1538-4357/adc737

Lodders, K. 2021, SSRv, 217, 44, doi: 10.1007/s11214-021-00825-8

Lodders, K., Bergemann, M., & Palme, H. 2025, SSRv, 221, 23, doi: 10.1007/s11214-025-01146-w

Maeder, A. 1997, A&A, 321, 134

Maeder, A., & Zahn, J.-P. 1998, A&A, 334, 1000

Magg, E., Bergemann, M., Serenelli, A., et al. 2022, A&A, 661, A140, doi: 10.1051/0004-6361/202142971

Mamajek, E. E., & Hillenbrand, L. A. 2008, ApJ, 687, 1264, doi: 10.1086/591785

Manchon, L., Deal, M., Marques, J. P. C., & Lebreton, Y. 2025, arXiv e-prints, arXiv:2511.02801, doi: 10.48550/arXiv.2511.02801

Mann, A. W., Feiden, G. A., Gaidos, E., Boyajian, T., & von Braun, K. 2015, ApJ, 804, 64, doi: 10.1088/0004-637X/804/1/64

Matt, S., & Pudritz, R. E. 2008, ApJ, 678, 1109, doi: 10.1086/533428

Matt, S. P., MacGregor, K. B., Pinsonneault, M. H., & Greene, T. P. 2012, ApJL, 754, L26, doi: 10.1088/2041-8205/754/2/L26

Metcalfe, T. S., van Saders, J. L., Basu, S., et al. 2020, ApJ, 900, 154, doi: 10.3847/1538-4357/aba963

Morel, P. 1997, A&AS, 124, 597, doi: 10.1051/aas:1997209

Morel, P., & Lebreton, Y. 2008, Ap&SS, 316, 61, doi: 10.1007/s10509-007-9663-9

Mosser, B., Goupil, M. J., Belkacem, K., et al. 2012, A&A, 548, A10, doi: 10.1051/0004-6361/201220106

Nguyen, C. T., Costa, G., Girardi, L., et al. 2022a, A&A, 665, A126, doi: 10.1051/0004-6361/202244166

Nguyen, C. T., Costa, G., Girardi, L., et al. 2022b, A&A, 665, A126

Nguyen, C. T., Costa, G., Bressan, A., et al. 2025a, A&A, 701, A258

Nguyen, C. T., Costa, G., Bressan, A., et al. 2025b, A&A, 701, A258, doi: 10.1051/0004-6361/202556005

Paczyński, B. 1970, AcA, 20, 195

Palla, F., & Stahler, S. W. 1991, ApJ, 375, 288, doi: 10.1086/170188

Paxton, B. 2004, PASP, 116, 699, doi: 10.1086/422345

Paxton, B., Bildsten, L., Dotter, A., et al. 2011, ApJS, 192, 3

Paxton, B., Cantiello, M., Arras, P., et al. 2013, ApJS, 208, 4

Paxton, B., Marchant, P., Schwab, J., et al. 2015, ApJS, 220, 15

Paxton, B., Schwab, J., Bauer, E. B., et al. 2018, ApJS, 234, 34, doi: 10.3847/1538-4365/aaa5a8

Paxton, B., Smolec, R., Schwab, J., et al. 2019, ApJS, 243, 10, doi: 10.3847/1538-4365/ab2241

Pérez Paolino, F., Bary, J. S., Hillenbrand, L. A., Horner, B., & Carvalho, A. 2025, ApJ, 990, 205, doi: 10.3847/1538-4357/adf6ad

Pinsonneault, M. H., Deliyannis, C. P., & Demarque, P. 1991, ApJ, 367, 239, doi: 10.1086/169623

Pinsonneault, M. H., Kawaler, S. D., Sofia, S., & Demarque, P. 1989, ApJ, 338, 424, doi: 10.1086/167210

Pinsonneault, M. H., Steigman, G., Walker, T. P., & Narayanan, V. K. 2002, ApJ, 574, 398, doi: 10.1086/340119

Pinsonneault, M. H., Zinn, J. C., Tayar, J., et al. 2025, ApJS, 276, 69, doi: 10.3847/1538-4365/ad9fef

Pizzolato, N., Maggio, A., Micela, G., Sciortino, S., & Ventura, P. 2003, A&A, 397, 147, doi: 10.1051/0004-6361:20021560

Planck Collaboration, Aghanim, N., Akrami, Y., et al. 2020, A&A, 641, A6, doi: 10.1051/0004-6361/201833910

Prather, M. J. 1976, PhD thesis, Yale University, Connecticut

Reyes, C., Stello, D., Hon, M., et al. 2024, MNRAS, 532, 2860, doi: 10.1093/mnras/stae1650

Rogers, F. J., & Nayfonov, A. 2002, ApJ, 576, 1064, doi: 10.1086/341894

Rogers, F. J., Swenson, F. J., & Iglesias, C. A. 1996, ApJ, 456, 902, doi: 10.1086/176705

Roquette, J., Matt, S. P., Winter, A. J., Amard, L., & Stasevic, S. 2021, MNRAS, 508, 3710, doi: 10.1093/mnras/stab2772

Salpeter, E. E. 1954, Australian Journal of Physics, 7, 373, doi: 10.1071/PH540373

Saumon, D., Chabrier, G., & van Horn, H. M. 1995, ApJS, 99, 713, doi: 10.1086/192204

Saunders, N., van Saders, J. L., Lyttle, A. J., et al. 2024, ApJ, 962, 138, doi: 10.3847/1538-4357/ad1516

Schwarzschild, M., & Härm, R. 1958, ApJ, 128, 348, doi: 10.1086/146548

Shu, F., Najita, J., Ostriker, E., et al. 1994, ApJ, 429, 781, doi: 10.1086/174363

Sills, A., & Pinsonneault, M. H. 2000, ApJ, 540, 489, doi: 10.1086/309306

Soderblom, D. R., Jones, B. F., Balachandran, S., et al. 1993, AJ, 106, 1059, doi: 10.1086/116704

Somers, G., Cao, L., & Pinsonneault, M. H. 2020, ApJ, 891, 29, doi: 10.3847/1538-4357/ab722e

Somers, G., & Pinsonneault, M. H. 2015, ApJ, 807, 174, doi: 10.1088/0004-637X/807/2/174

Somers, G., & Pinsonneault, M. H. 2016, ApJ, 829, 32, doi: 10.3847/0004-637X/829/1/32

Somers, G., Stauffer, J., Rebull, L., Cody, A. M., & Pinsonneault, M. 2017, ApJ, 850, 134, doi: 10.3847/1538-4357/aa93ed

Spada, F., & Lanzafame, A. C. 2020, A&A, 636, A76, doi: 10.1051/0004-6361/201936384

Spruit, H. C. 1982, A&A, 108, 356

Spruit, H. C., & Weiss, A. 1986, A&A, 166, 167

Stauffer, J. R., Jones, B. F., Backman, D., et al. 2003, AJ, 126, 833, doi: 10.1086/376739

Sweigart, A. V., & Demarque, P. 1973, in Astrophysics and Space Science Library, Vol. 36, IAU Colloquium 21: Variable Stars in Globular Clusters and in Related Systems, ed. J. D. Fernie, 221, doi: 10.1007/978-94-010-2590-4_32

Tayar, J., Beck, P. G., Pinsonneault, M. H., García, R. A., & Mathur, S. 2019, ApJ, 887, 203, doi: 10.3847/1538-4357/ab558a

Tayar, J., & Pinsonneault, M. H. 2013, ApJL, 775, L1, doi: 10.1088/2041-8205/775/1/L1

Tayar, J., & Pinsonneault, M. H. 2018, ApJ, 868, 150, doi: 10.3847/1538-4357/aae979

Tayar, J., Somers, G., Pinsonneault, M. H., et al. 2017, ApJ, 840, 17, doi: 10.3847/1538-4357/aa6a1e

Thoul, A. A., Bahcall, J. N., & Loeb, A. 1994, ApJ, 421, 828, doi: 10.1086/173695

Torres, G., Andersen, J., & Giménez, A. 2010, A&A Rv, 18, 67, doi: 10.1007/s00159-009-0025-1

van Groeningen, M. G. J., Castro-Ginard, A., Brown, A. G. A., Casamiquela, L., & Jordi, C. 2023, A&A, 675, A68, doi: 10.1051/0004-6361/202345952

van Saders, J. L., Ceillier, T., Metcalfe, T. S., et al. 2016, Nature, 529, 181, doi: 10.1038/nature16168

van Saders, J. L., & Pinsonneault, M. H. 2012, ApJ, 746, 16, doi: 10.1088/0004-637X/746/1/16

van Saders, J. L., & Pinsonneault, M. H. 2013, ApJ, 776, 67, doi: 10.1088/0004-637X/776/2/67

van Saders, J. L., Pinsonneault, M. H., & Barbieri, M. 2019, ApJ, 872, 128, doi: 10.3847/1538-4357/aafafe

Villante, F. L., Serenelli, A. M., Delahaye, F., & Pinsonneault, M. H. 2014, ApJ, 787, 13, doi: 10.1088/0004-637X/787/1/13

White, R., Pratt, J., & Rieutord, M. 2025, arXiv e-prints, arXiv:2509.20264, doi: 10.48550/arXiv.2509.20264

Woo, J.-H., & Demarque, P. 2001, AJ, 122, 1602, doi: 10.1086/322122

Wood, B. E., Müller, H. R., Zank, G. P., Linsky, J. L., & Redfield, S. 2005, ApJL, 628, L143, doi: 10.1086/432716

Xu, Y., Takahashi, K., Goriely, S., et al. 2013, NuPhA, 918, 61, doi: 10.1016/j.nuclphysa.2013.09.007

Yorke, H. W., & Sonnhalter, C. 2002, ApJ, 569, 846, doi: 10.1086/339264

Zahn, J. P. 1992, A&A, 265, 115

Zakai, A. 2011, in Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, OOPSLA '11 (New York, NY, USA: Association for Computing Machinery), 301–312, doi: 10.1145/2048147.2048224