Tools for Astrophysics: MESA and NuGrid

Pavel A. Denissenkov

Department of Physics & Astronomy, University of Victoria, P.O. Box 3055, Victoria, B.C., V8W 3P6, Canada; `pavelden@uvic.ca`

1 What Are These Tools For?

MESA is a collection of Fortran-95 **M**odules for **E**xperiments in **S**tellar **A**strophysics. Its main module `star` can be used for one-dimensional stellar evolution simulations of almost any kind. For instance, it can compute without any interruption the evolution of a solar-type star from the pre-MS phase through the He-core flash and thermal pulses on the AGB towards white-dwarf cooling. Other modules provide `star` with state-of-the-art numerical algorithms, e.g. for adaptive mesh refinement and timestep control, atmospheric boundary conditions, and modern input physics (see Marco Pignatari's tutorial on MESA/NuGrid physics packages).

NuGrid tools include codes and data enabling large-scale post-processing nucleosynthesis computations. These can be done either at one Lagrangian coordinate (with the one-zone code PPN), in which case changes of T and ρ with time at a chosen mass zone (the so-called "trajectory") have to be provided, or for an entire star (with the multi-zone code MPPNP), in which case T and ρ have to be given as functions of M_r for each evolutionary model ("cycle").

The MESA `star` has an option to output an evolutionary sequence of stellar models in a compressed format readable by the NuGrid MPPNP code. This interface allows to run `star` with a small reaction network, sufficient to properly take into account the nuclear energy generation rate, followed by an MPPNP run that can use stellar models prepared by `star` as a background for detailed nucleosynthesis computations.

The `star` code uses multi-threading to speed up its execution on multiple CPU cores, while MPPNP uses MPI to run in parallel on multiple CPUs.

2 What Should You Know if You Want to Use These Tools?

I would recommend first to read the MESA and NuGrid manifestos that can be found on their websites, `http://mesa.sourceforge.net/` and `www.astro.keele.ac.uk/nugrid`. These documents describe important "terms and conditions" (actually, a sort of fair-play rules) for accessing and using the tools.

MESA is free for download from its website that also gives some information on how to install and run the code. Its technical details, physical assumptions, and unique modeling capabilities are reported and copiously illustrated in the MESA main "instrument" paper by Paxton et al. (2011, ApJS, 192, 3).

To be successfully compiled and executed, MESA needs several libraries to be installed on your Linux or MAC computer. In particular, MESA will require a recent version of `ifort` or `gfortran` compiler, linear algebra BLAS and LAPACK libraries, version-5 Hierarchical Data Format (HDF5) library, PGPLOT graphics library, `ndiff` fuzzy comparison tool, and

SE library from the NuGrid project. You can find, download, and install them on your computer separately or, as an alternative, you can use a unified software development kit MESA SDK put together by Rick Townsend on the website `www.astro.wisc.edu/~townsend` that also explains how to build MESA. The most important next step is to correctly specify MESA installation parameters in the file `mesa/utils/makefile_header`. It contains step-by-step instructions that help to choose right options for the parameters. A special version of this file called `makefile_header.mesasdk` can be used by those who decided to take advantage of the MESA SDK kit.

To join the NuGrid collaboration and get an access to the NuGrid tools, you should send an email message expressing your interest to one of its active members, e.g. Marco Pignatari (`mpignatari@gmail.com`), Gabriel Rockefeller (`gaber@lanl.gov`), or Falk Herwig (`fherwig@uvic.ca`).

3 How to Run MESA Star?

During its installation, MESA conducts some tests. If results of these tests are correct, MESA reports about its successful installation, and it is now ready for use. The main working directory for running MESA `star` is `mesa/star/work`. It is a good idea to copy this template directory elsewhere. Then, at its new location, you should first change the directory path names that are assigned to the string variables `mesa_data_dir` and `MESA_DIR` at the tops of the files `work/inlist_project` and `work/make/makefile`, respectively. These changes will let your copy of the `work` directory know where the other MESA modules are located. After that, you should type `./mk` in your `work` directory to make the code, followed by the command `./rn` that launches the MESA `star`. As a result, the evolution of a star with predefined parameters will be calculated with a lot of various text and graphics output data sent to a number of files and popped-up windows.

In order to set up parameters for your own MESA `star` run, you have to use an `inlist` file or the `inlist_project` file, if the latter is read by the former. It contains three Fortran-95 parameter namelists: `star_job, controls`, and `pgstar`. Roughly speaking, the first namelist allows you to "build" your own stellar evolution code by specifying where an initial model is to be taken, what data from computed models you want to be extracted and saved in files, what input physics, in particular a MESA standard or your own customized nuclear network should be used, and other global parameters. In the second namelist, you assign values to parameters that are specific for an individual run, such as the initial chemical composition and mass of the star, conditions to stop the simulations, prescriptions for mass gain or loss at different evolutionary phases, a criterion for convective instability (Schwarzschild or Ledoux), and others. The `controls` namelist is also the place where you can add some non-standard physical assumptions to your stellar evolution simulations. At present, these include element diffusion, semiconvection and convective overshooting, as well as thermohaline, rotational and magnetic mixing. Finally, the third namelist customizes the graphics output.

The section "how to use MESA star" on the MESA website provides more details on setting up the parameters in the `inlist`. Although MESA `star` has a too large number of parameters to comprehend, all of them already have reasonable default values with which standard stellar evolution simulations can be done. The full lists of the star_job, controls, and pgstar parameters with their default values can be found in the files run_star_defaults.dek, star_defaults.dek, and pgstar_defaults.dek located in the directory mesa/star/public.

A few words of caution are necessary for those who want to experiment with the MESA non-standard physical assumptions. It is your responsibility to verify that a set of inlist parameters that you choose will result in correct modeling of the corresponding physical process. It is a matter of fact that non-standard additions to the MESA star are made on requests of its active and potential users who may not thoroughly test them afterwards. Sometimes, an option that worked in a previous release of MESA may not be available or produces a different result in its newer version. This can happen if the addition is considered non-standard, and therefore not included in MESA regression testing. The best way to solve such a problem is to contact a person who has already used the non-standard physical assumption you are interested in and ask for advice. For this, you can use the MESA-users mailing list or try the new MESA-user forum.

4 Application of MESA and NuGrid Tools for Simulations of Classical Nova Outbursts and Nucleosynthesis

If you want to reproduce someone else's stellar evolution simulations with MESA, you need to have a copy of the inlist file and know the version of the MESA package with which the simulations were run. If the inlist reads other project inlists or non-standard files specifying the initial model and its chemical composition, nuclear network, and output data lists, you will also need these. MESA itself has a special directory /mesa/star/test_suite that contains examples of a large number of its possible applications. You can use one of these tests, that best suits your problem, as a starting point for its solution.

Classical novae are the result of thermonuclear explosions of hydrogen occurring on the surfaces of white dwarfs (WDs) that accrete H-rich material from their low-mass main-sequence binary companions. In our MESA nova simulations, we have used inlists of two relevant test_suite cases:

- make_co_wd combines some "stellar engineering" procedures to create CO WD models from a range of initial masses,
- wd2 demonstrates the use of parameters that control accretion, as well as mass ejection options available for nova calculations.

Our extended nuclear network nova_ext.net includes 48 isotopes from H to ^{30}Si coupled by 120 reactions. The initial stellar models are the $1.0\,M_\odot$ and $1.15\,M_\odot$ CO WDs, and $1.15\,M_\odot$ and $1.3\,M_\odot$ ONe WDs. They have been prepared with inlists similar to the one in the directory make_co_wd.

Nova outbursts become stronger when WD's mass increases, while its initial central temperature and the accretion rate decrease, the latter being limited by the range $10^{-11}\,M_\odot/\mathrm{yr} \le \dot{M} \le 10^{-9}\,M_\odot/\mathrm{yr}$ for classical novae. The observed enrichment of the ejecta of novae in heavy elements (C, N, O, and Ne) is believed to be a signature of mixing between the accreted envelope and WD. Like in most other 1D nova simulations, we do not model this mixing explicitly but, instead, assume that the WD already accretes a pre-enriched mixture of equal amounts of its core and solar-composition materials.

As a representative case demonstrating the application of MESA and NuGrid tools for simulations of nova outbursts and nucleosynthesis, we have chosen our model of nova occurring on the $1.3\,M_\odot$ ONe WD with the initial central temperature $T_{\rm WD} = 12$ MK and accretion rate $\dot{M} = 2\times 10^{-10}\,M_\odot$/yr. Its evolutionary track is plotted in Fig. 1 (the blue curve in the left panel) from the start of accretion till the star has expanded to several solar radii as a result of explosion (the dashed black lines are the *loci* of constant R). The three right panels show the internal profiles of T and ρ (upper), mass fractions of some isotopes (middle), and temperature gradients (logarithmic and with respect to pressure) (lower) in the envelope of a model (the red star symbol in the left panel) in which the temperature has reached its peak value, $T_{\rm max} \approx 355$ MK. These computations have been done with the inlists presented in Figs. 2 – 6 using the version 3611 of MESA. In order to reproduce them, you will need the inlists, the initial WD model (`ne_wd_1.3_12_mixed.mod`), the chemical composition of the accreted envelope (`ne_nova_1.3_mixed_comp`), the nuclear networks `nova_ext.net` and `nova.net`, the latter having a shorter list of isotopes with which the WD models have been generated, and the file `my_log_columns.list` specifying the output data for each model. You will also have to use our modified fortran subroutine `run_star_extras.f` that should replace its standard copy in the directory `work/src`. All these files can be downloaded from the webpage `http://astrowww.phys.uvic.ca/~dpa/MESA_NuGrid_Tutorial.html`.

The `star_job` namelist has the parameter `set_se_output` in its output section (Fig. 3). When this parameter is set to `.true.`, the internal structure (the T and ρ profiles, chemical composition, diffusion coefficients corresponding to different mixing mechanisms, etc.) of each model (called "cycle" in NuGrid) will be written to a disk in the compressed `hdf5` format (the file `se.input` defines a prefix to names of the model structure files, as well as a name of the directory where they are stored). The NuGrid MPPNP code can read these files and use them as a background for multi-zone post-processing nucleosynthesis computations. A list of isotopes considered by the code can be changed. In our nova post-processing nucleosynthesis computations, we use 147 isotopes from H to Ca. The initial composition is the same as the one used in our MESA nova simulations, except that the number of isotopes has been increased from 48 to 147. Because the NuGrid tools are not free for download yet, we will not explain in detail how to run MPPNP and PPN codes. The interested reader is referred to the NuGrid Code Book.

The NuGrid tools have a library of Python scripts that can be used to plot MPPNP and PPN results. We have used them to produce Figs. 7, 8, and 9. Fig. 7 shows the isotopes and their initial abundances that we have used in our MPPNP computations. Fig. 8 corresponds to the evolved model with $T_{\rm max} \approx 355$ MK (the red star symbol in the left panel in Fig. 1). Note the accumulation of the β^+-unstable isotopes ^{14}O, ^{15}O, and ^{17}F, also seen in the middle right panel in Fig. 1, which is typical for H-burning in the hot CNO cycle in novae. The upper panel in Fig. 9 displays the abundances of stable isotopes divided by their corresponding solar values in the expanding envelope of our final ONe nova model. They are compared with the abundance ratios from a similar nova model ($1.35\,M_\odot$ ONe WD with 50% mixing, $\dot{M} = 2\times 10^{-10}\,M_\odot$/yr) reported by the Barcelona group (José, J., & Hernanz, M., 1998, ApJ, 494, 680). In spite of the different input physics and computer codes, there is a good agreement between the two plots.

Finally, Fig. 10 shows the trajectory (T and ρ as functions of time) for a Lagrangian coordinate at which the temperature profile has its maximum. It can be used by the NuGrid PPN code to carry out the one-zone post-processing nucleosyntheis computations. The PPN code runs much faster than MPPNP, while producing qualitatively similar results (compare Figs. 8 and 11), therefore it can be used for comprehensive numerical analysis of parameter space, when the isotope abundances and reaction rates are varied within their observationally and experimentally constrained limits.

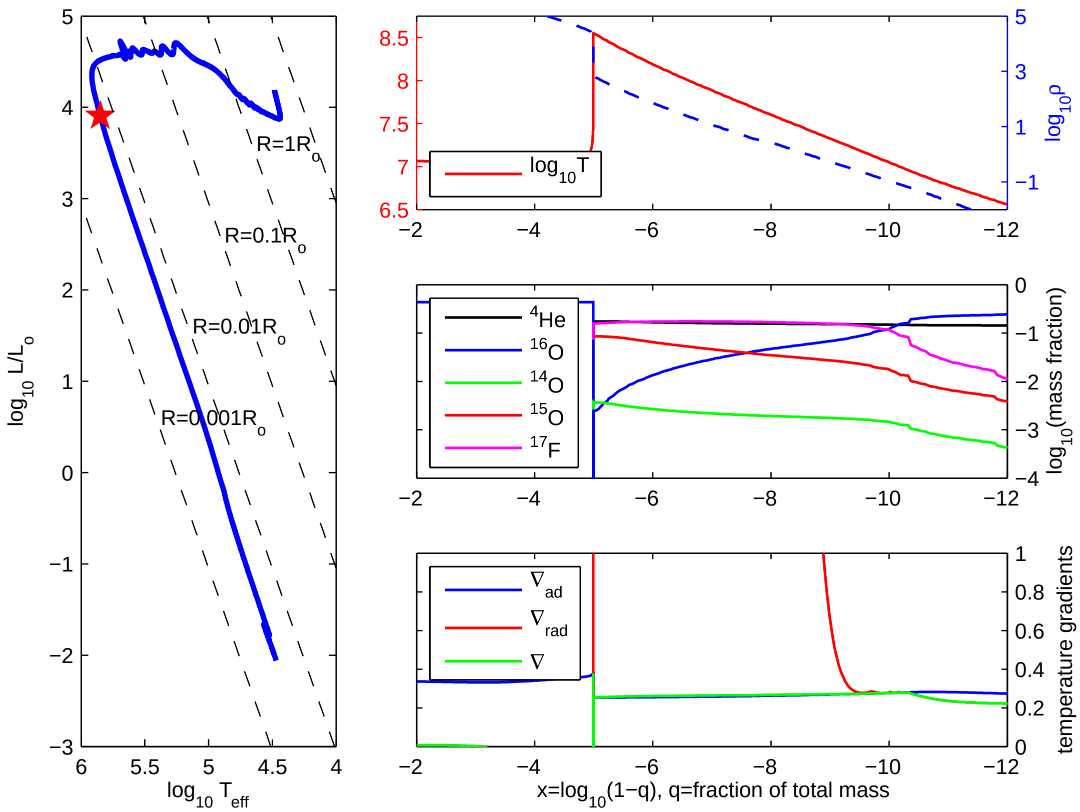

Figure 1: The results of our MESA simulations of the nova outburst occurring on the 1.3 $M_{\odot}$ ONe WD with the initial central temperature 12 MK and accretion rate 2×10^{-10} $M_{\odot}$/yr. It is assumed that the WD accretes a pre-enriched mixture of equal amounts of WD's core and solar-composition materials. Left panel: the evolutionary track (the blue curve) and location of the model with the maximum temperature at the base of the H-rich envelope (the red star symbol). The three right panels show the internal profiles of T and ρ (upper), mass fractions of some isotopes (middle), and temperature gradients (lower; the region where $\nabla_{\rm rad} > \nabla_{\rm ad}$ is convectively unstable) in the envelope of the model with the maximum T.

```
! main inlist file for simulations of ONe novae with MESA

&star_job

      mesa_data_dir = '/local/astro/mesa_3611/data'

      read_extra_star_job_inlist1 = .true.
      extra_star_job_inlist1_name = 'inlist_ne_nova'

/ ! end of star_job namelist


&controls

      read_extra_controls_inlist1 = .true.
      extra_controls_inlist1_name = 'inlist_ne_nova'

/ ! end of controls namelist


&pgstar

      read_extra_pgstar_inlist1 = .true.
      extra_pgstar_inlist1_name = 'inlist_ne_nova_pgstar'

/ ! end of pgstar namelist
```

Figure 2: The main `inlist` file used in our simulations of ONe novae with MESA. It reads the project `inlist_ne_nova` file that is shown in Figs. 3 – 6. It also reads the `inlist_ne_nova_pgstar` file that specifies the graphics output.

```
! inlist_ne_nova


&star_job

   ! output

      show_log_description_at_start = .false.
         ! set this false if you want to skip the initial terminal output

      show_net_reactions_info = .true.
         ! if true, then output a list of the reactions in the current net (nova.net)

      pgstar_flag = .true.
      use_se_output = .true. ! se is based on hdf5.  ask Falk Herwig about it.

      log_columns_file = 'my_log_columns.list'
         ! if null, then use standard, mesa/data/star_data/log_columns.list

      profile_columns_file = ''
         ! if null, then use standard, mesa/data/star_data/profile_columns.list

   ! starting model

      load_saved_model = .true.
      saved_model_name = 'ne_wd_1.3_12_mixed.mod'
         ! if load_saved_model is true, then use this initial model

   ! when to stop

      steps_to_take_before_terminate = 2100
         ! if > 0, stop after taking this many steps
         ! sets max_model_number = model_number + steps_to_take_before_terminate

   ! modifications to model

         ! controls that only apply to the first model have 'initial' in their names.
         ! they are ignored for restarts.

      set_initial_age = .true.
      initial_age = 0 ! in years

      set_tau_factor = .true. ! change tau_factor without reconverging.
      set_to_this_tau_factor = 25

! to be continued...
```

Figure 3: The `inlist_ne_nova` file used in our MESA simulations of the classical nova outburst occurring on the $1.3\,M_\odot$ ONe WD. The WD's initial central temperature is $T_{\rm WD} = 12$ MK, and the accretion rate is $\dot{M} = 2\times 10^{-10}\,M_\odot$/yr. The accreted material is assumed to be a mixture of equal amounts of the WD and solar-composition materials. This project inlist file is read by the main inlist shown in Fig. 2.

```
! inlist_ne_nova (continued)


   ! velocity variables

         ! NOTE: at present, mesa/star can deal with large velocities,
         !       but not with shocks or other "violent" hydrodynamic events.

      change_v_flag = .true.
      new_v_flag = .true.

   ! nuclear reactions

      change_net = .true. ! switch nuclear reaction network
      new_net_name = 'nova_ext.net'

      set_rates_preference = .false. ! for use by net + rates modules
      new_rates_preference = 1
         ! 1 = NACRE rates -- this is the default
         ! 2 = jina reaclib rates

/ ! end of star_job namelist

&controls

   ! controls for output

      photostep = 100
      profile_interval = 100
      log_cnt = 1
      terminal_cnt = 1
      write_header_frequency = 10

   ! when to stop

      min_timestep_limit = 1d-12 ! (seconds)
         ! stop if need timestep smaller than this limit

   ! atmosphere boundary conditions

      which_atm_option = 'grey_and_kap'

! to be continued...
```

Figure 4: The `inlist_ne_nova` file from Fig. 3 continued.

```
! inlist_ne_nova (continued)

   ! mass gain or loss

      mass_change = 2d-10 ! rate of accretion (Msun/year). negative for mass loss.
         ! this only applies when the current wind scheme = no_automatic_wind

      RGB_wind_scheme = '' ! empty string means no RGB wind
      AGB_wind_scheme = '' ! empty string means no AGB wind

         ! these params provide the option to turn off mass change when have small
         ! timesteps.
         ! mass change doesn't do much in such cases except make convergence harder.
      mass_change_full_on_dt = 1d0 ! (seconds)
      mass_change_full_off_dt = 1d-1 ! (seconds)
         ! between these limits, mass change is gradually reduced

   ! composition of accreted material

      accrete_same_as_surface = .false.
         ! if true, composition of accreted material is identical to the current
         ! surface composition.
      accrete_given_mass_fractions = .true.
         ! if true, use the following mass fractions -- they must add to 1.0

      read_extra_controls_inlist2 = .true.
      extra_controls_inlist2_name = 'ne_nova_1.3_mixed_comp'

   ! mesh adjustment

      min_center_cell_dq = 1d-7
      max_center_cell_dq = 1d-6
      max_surface_cell_dq = 1d-12

      xa_function_species(1) = 'h1'  ! name of nuclide as defined in chem_def
               xa_function_weight(1) = 20
               xa_function_param(1) = 1d-6
      xa_function_species(2) = 'he4'  ! name of nuclide as defined in chem_def
               xa_function_weight(2) = 20
               xa_function_param(2) = 1d-2

      mesh_delta_coeff = 2.0

   ! nuclear reactions

      net_logTcut_lo = 5.3d0 ! rates are zero logT < logTcut_lo
      net_logTcut_lim = 5.4d0 ! rates cutoff smoothly for logT < logTcut_lim

! to be continued...
```

Figure 5: The `inlist_ne_nova` file from Fig. 3 continued.

```
! inlist_ne_nova (continued)

   ! structure equations

         ! artificial viscosity -- only applies when using velocity variables
      use_artificial_viscosity = .true.
      l1_coef = 0.1 ! increase to as much as 1d3 to suppress pulsations

   ! controls for timesteps

      varcontrol_target = 1d-3
         ! this is the target value for relative variation in the structure from
         ! one model to the next.
         ! the default timestep adjustment is to increase or reduce the timestep
         ! depending on whether
         ! the actual variation was smaller or greater than this value.

         ! if exceed a "limit", then reduce the timestep for the next step.
         ! if exceed a "hard_limit", then retry the current step with a reduced
         ! timestep.

      delta_lgL_H_limit = 0.05 ! limit for magnitude of change in lgL_H
      delta_lgL_H_hard_limit = 0.5
      lgL_H_burn_min = 1.5 ! ignore changes in lgL_H if value is less than this

      delta_lgL_He_limit = 0.025 ! limit for magnitude of change in lgL_He
      delta_lgL_He_hard_limit = 1
      lgL_He_burn_min = 2.5 ! ignore changes in lgL_He if value is less than this

      delta_lgRho_limit = 1 ! limit for magnitude of max change in log10 density
                            ! at any point
      delta_lgRho_hard_limit = -1

      ! limit for magnitude of change in log10(L/Lsun)
      delta_lgL_limit = 0.05
      delta_lgL_hard_limit = 0.15

      ! limit for magnitude of max change in log10 temperature at any point

      delta_lgT_limit = 0.5
      delta_lgT_hard_limit = 1

      ! limit for magnitude of max change in log10 temperature at photosphere

      delta_lgTeff_limit = 0.10
      delta_lgTeff_hard_limit = 0.30

/ ! end of controls namelist
```

Figure 6: The `inlist_ne_nova` file from Fig. 3 continued.

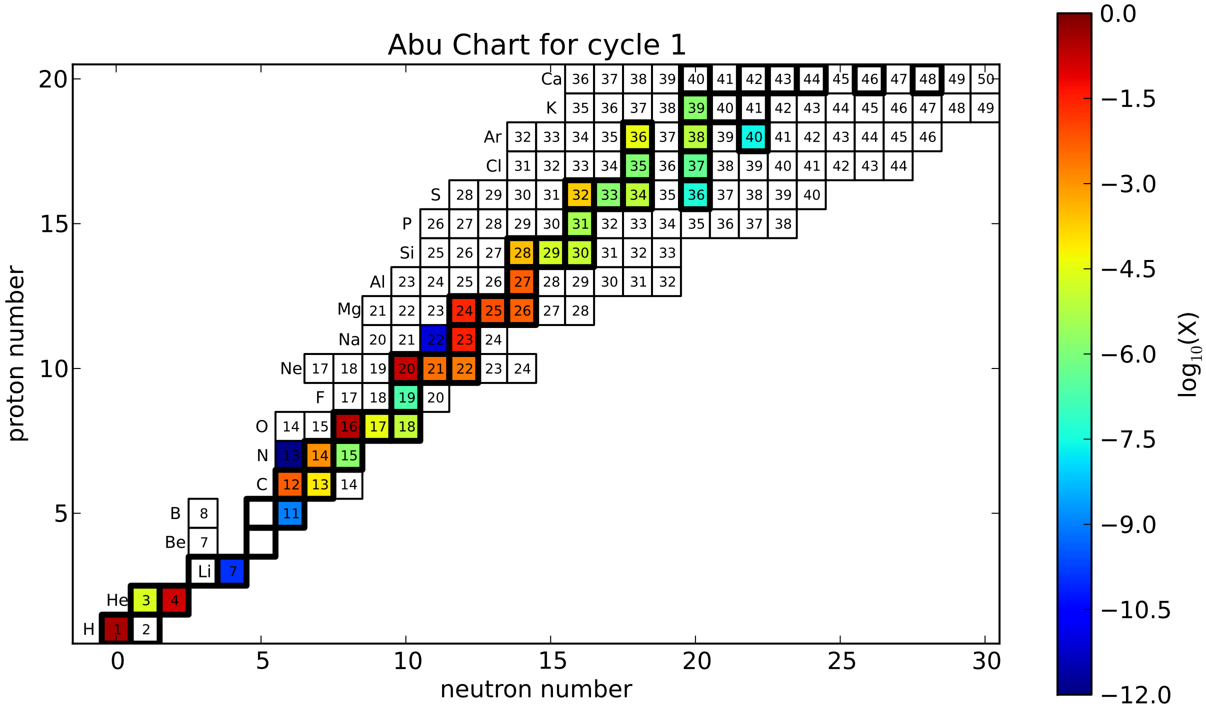

Figure 7: The isotopes and their initial abundances in the H-rich envelope of our $1.3\,M_\odot$ ONe nova model used in our post-processing nucleosynthesis computations.

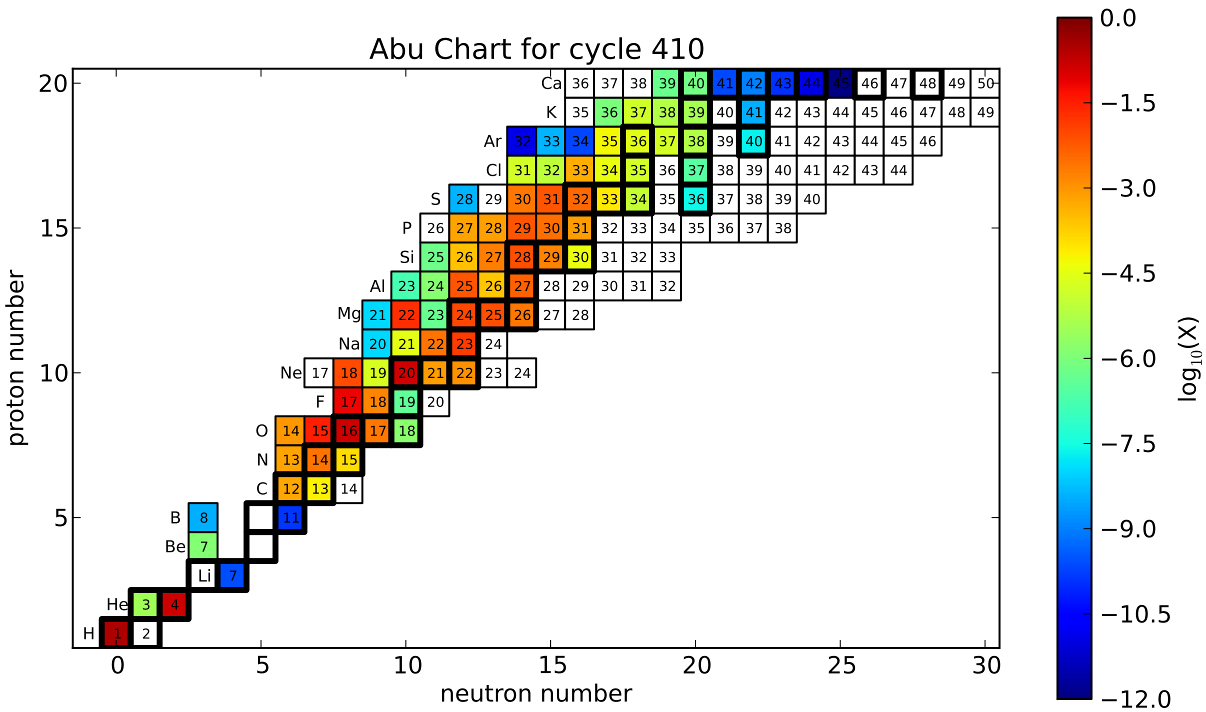

Figure 8: The isotope abundance distribution in the model with the maximum internal temperature (the red star symbol in Fig. 1). Note the accumulation of the β^+-unstable isotopes ^{14}O, ^{15}O, and ^{17}F, also seen in the middle right panel in Fig. 1, which is typical for H-burning in the hot CNO cycle in novae.

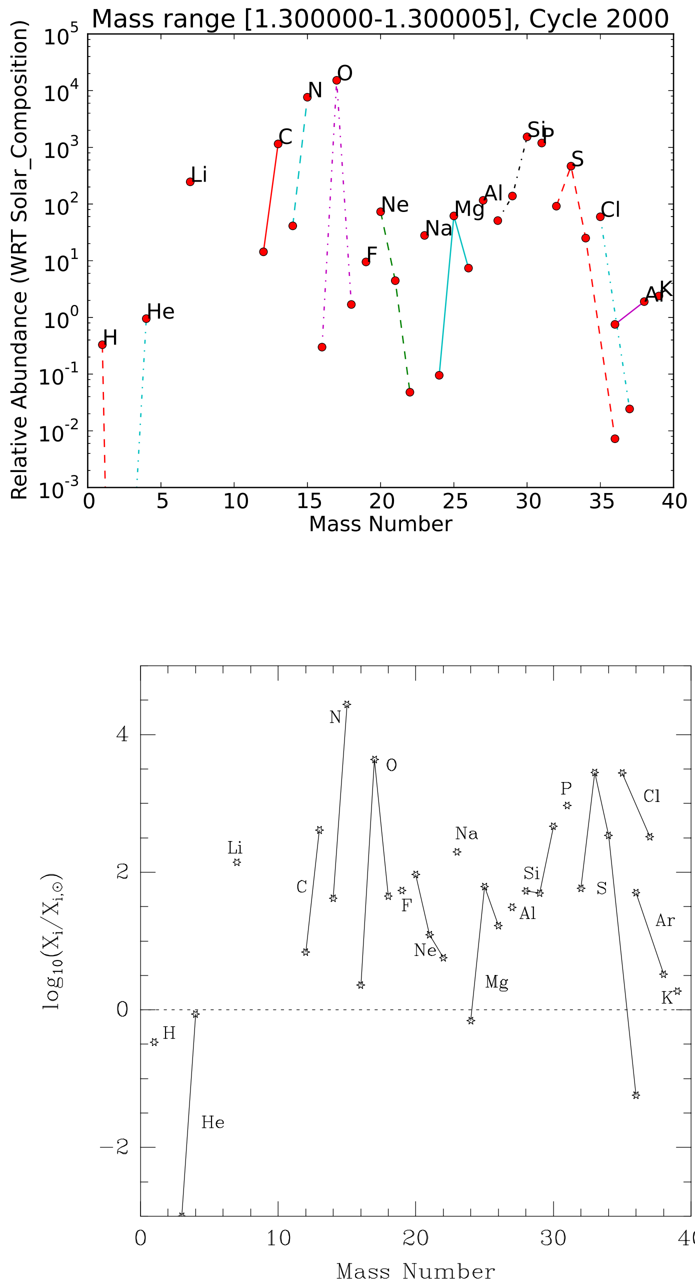

Figure 9: Upper panel: the abundances of stable isotopes divided by their corresponding solar values in the expanding envelope of our final ONe nova model calculated with the NuGrid MPPNP code. Lower panel: the abundance ratios from a similar nova model (1.35 $M_{\odot}$ ONe WD with 50% mixing and accretion rate 2×10^{-10} $M_{\odot}$/yr) reported by the Barcelona group (José, J., & Hernanz, M., 1998, ApJ, 494, 680).

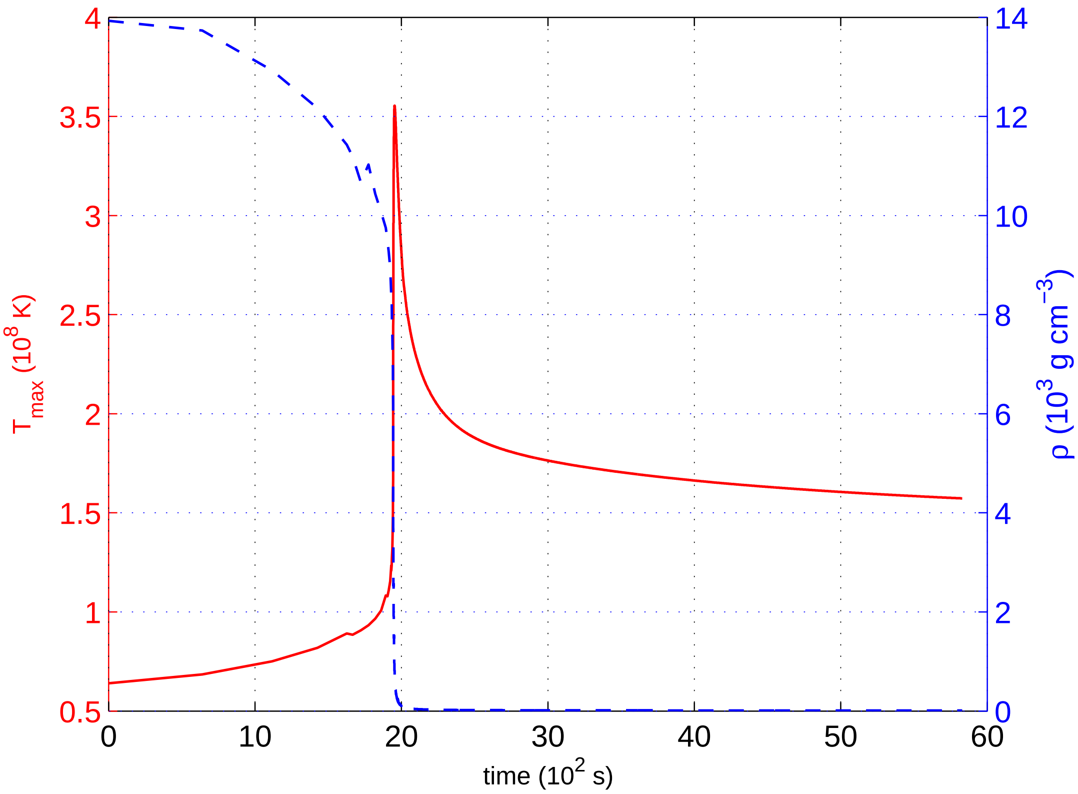

Figure 10: The trajectory (T and ρ as functions of time) for the Lagrangian coordinate at which the temperature profile has its maximum (e.g., see the red curve in the upper right panel in Fig. 1), extracted from our ONe nova simulations (from the file `work/LOGS/star.log`).

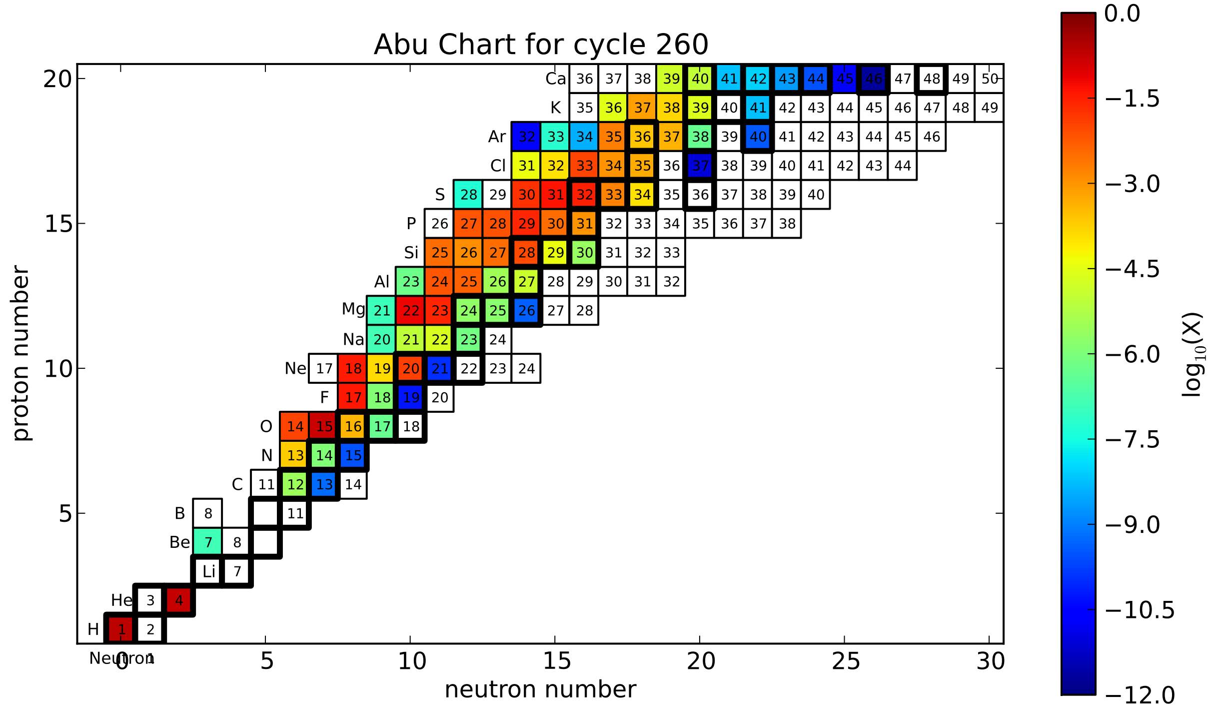

Figure 11: The distribution of isotope abundances at the time when T reaches its maximum on the trajectory from Fig. 10, obtained with the NuGrid PPN code. Compare this distribution with the one from Fig. 8.