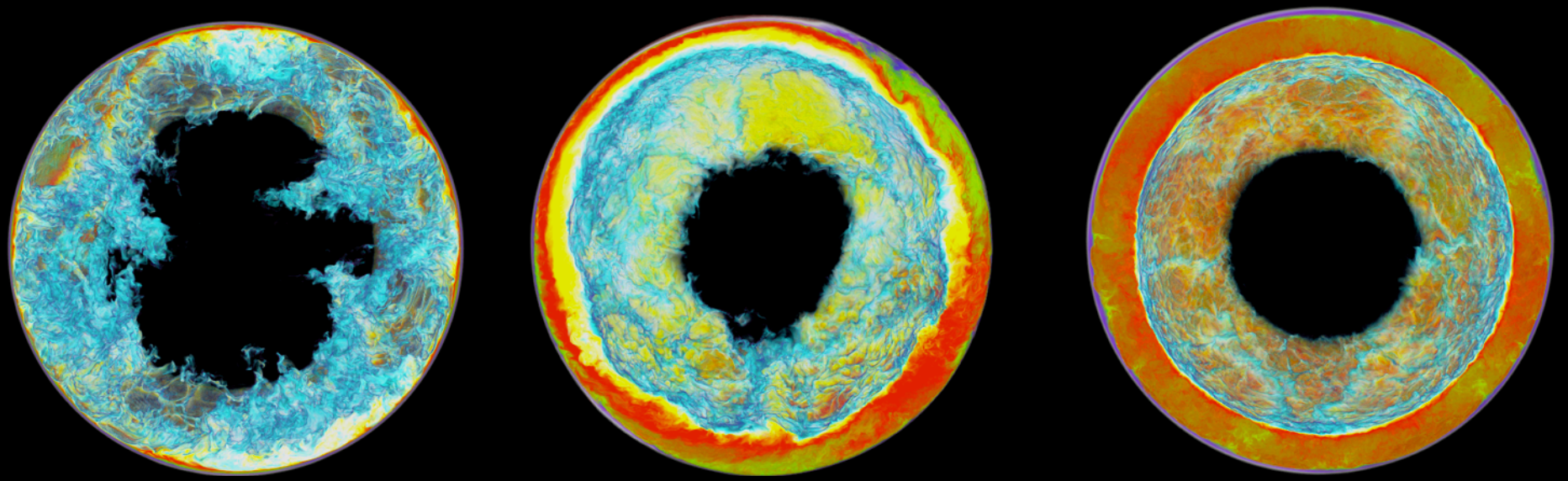
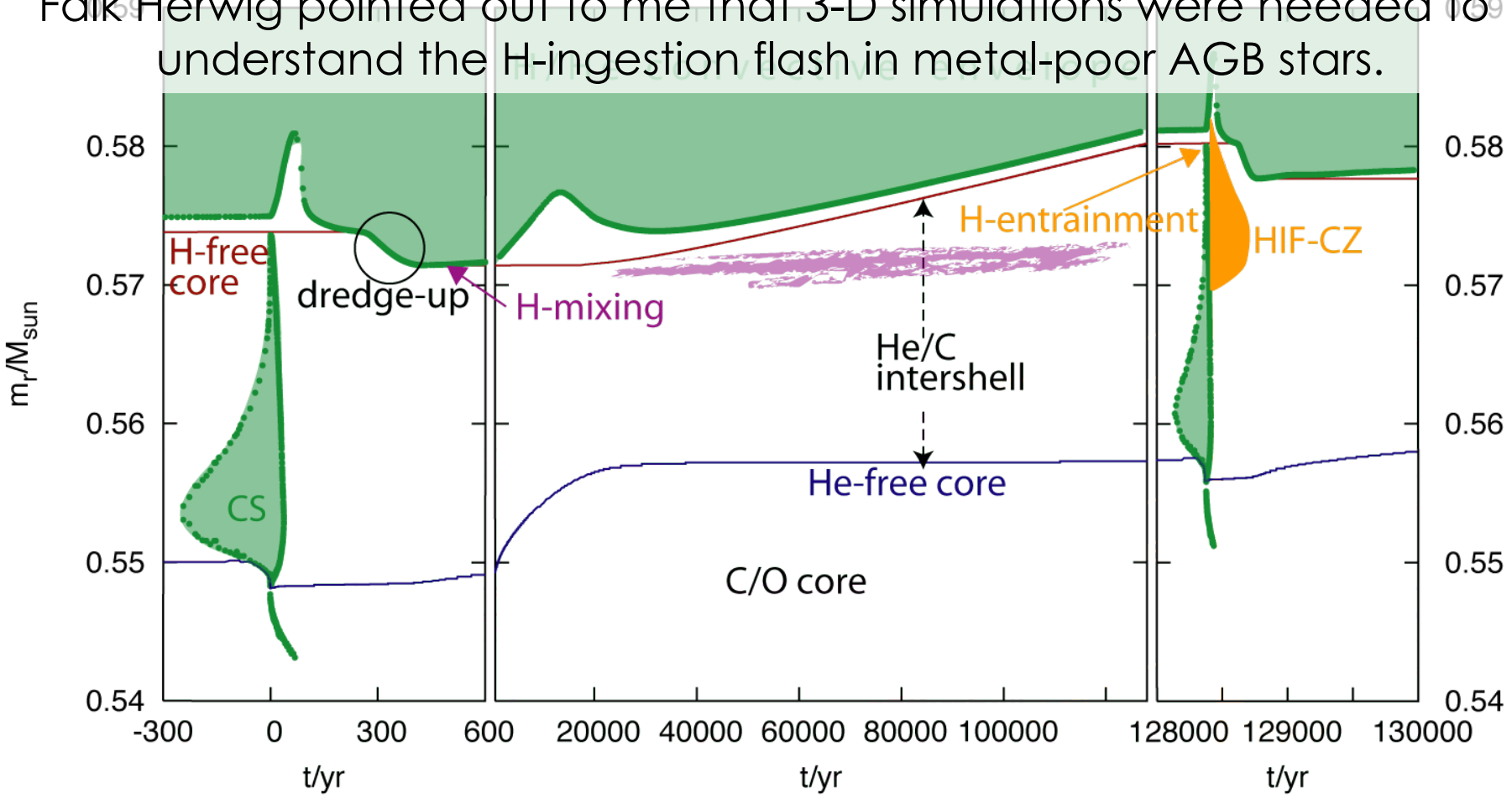


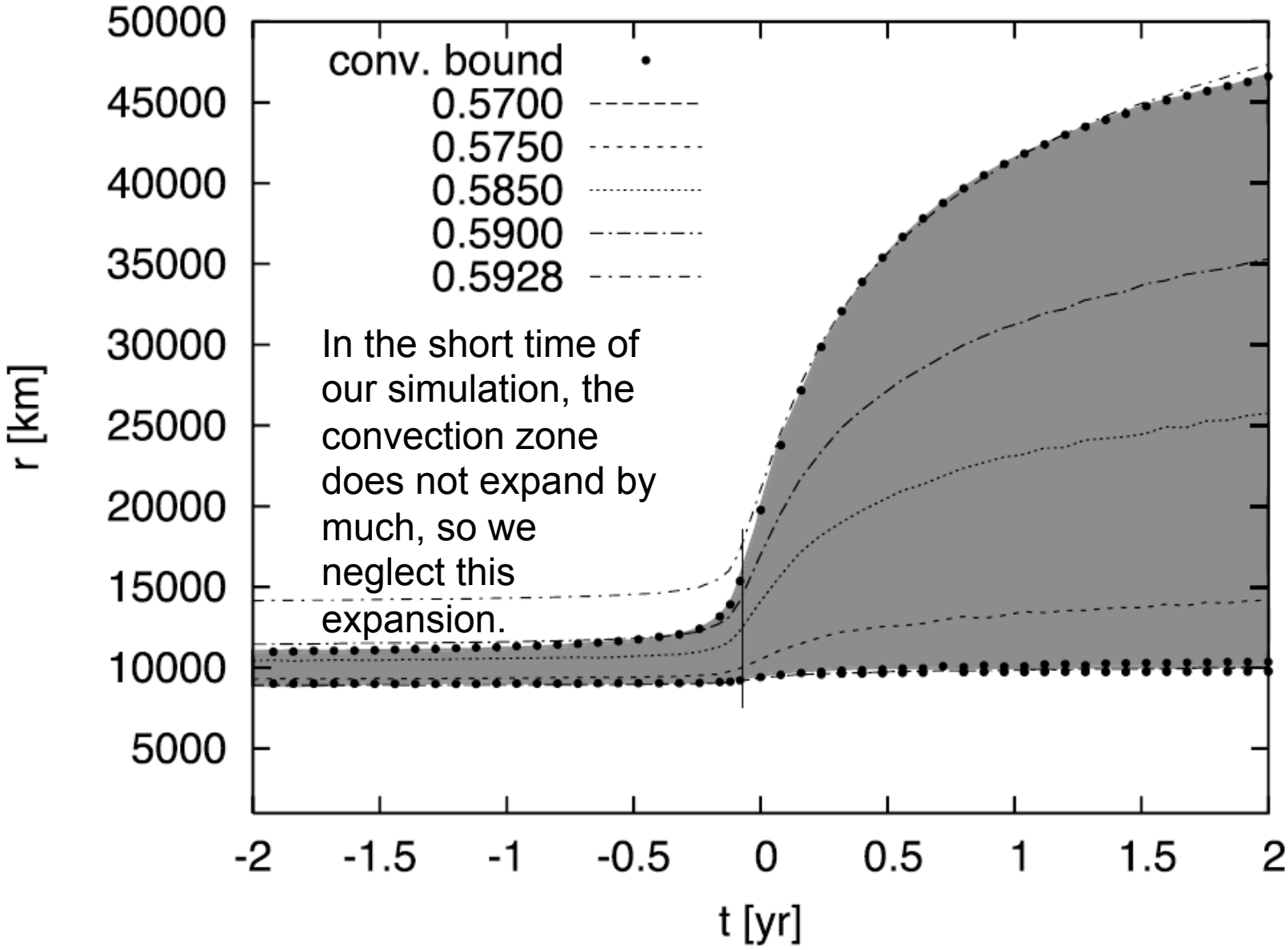
The Hydrogen Ingestion Flash in a Low-Z AGB Star of the Early Universe and the Special Challenges it Presents to Computation

Paul Woodward, Stou Sandalski, Huaqing Mao, Aaron D'Sa
Laboratory for Computational Science & Engineering
University of Minnesota
working with UVic team:
Falk Herwig, Pavel Denisenkov, Christian Ritter

Falk Herwig pointed out to me that 3-D simulations were needed to understand the H-ingestion flash in metal-poor AGB stars.

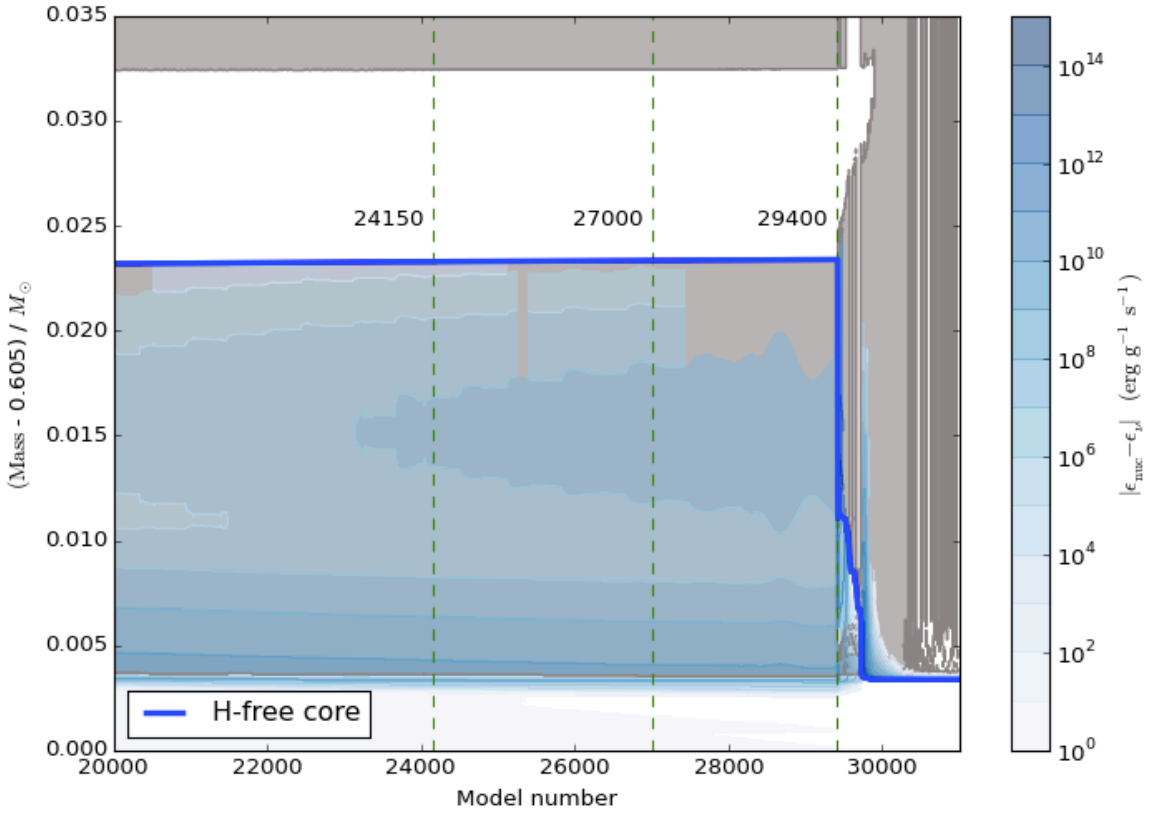


Time evolution of convective mixing and nuclear burning processes in He-shell flash AGB stars. Green regions indicate convectively unstable zones. CS is the He-shell flash convection zone. During and at the end of dredge-up, H-mixing (purple) into the C-rich intershell material can lead to formation of the n-source ^{13}C for the s-process (pink shaded region). H-entrainment into the CS leads to a H-ingestion flash convection zone (HIF-CZ), shown schematically in orange for the second He-flash. Adapted from Fig 3 in Herwig 2005.



Time evolution of the radial location of the He-shell flash convection zone based on the 1-D stellar evolution model of Herwig. Time is set to 0 at the peak of the He-burning luminosity. Dots represent individual time steps. Lagrangian lines at different mass fractions are shown. The convection zone grows both in radius and in mass fraction over the 2-year interval shown. Our simulation is performed at about time 0.2 yr on this slide.

Here is the Kippenhahn diagram for the 1-D MESA simulation. We have zoomed into the short time interval before the 1-D code ingests H into the He-shell-flash convection zone.

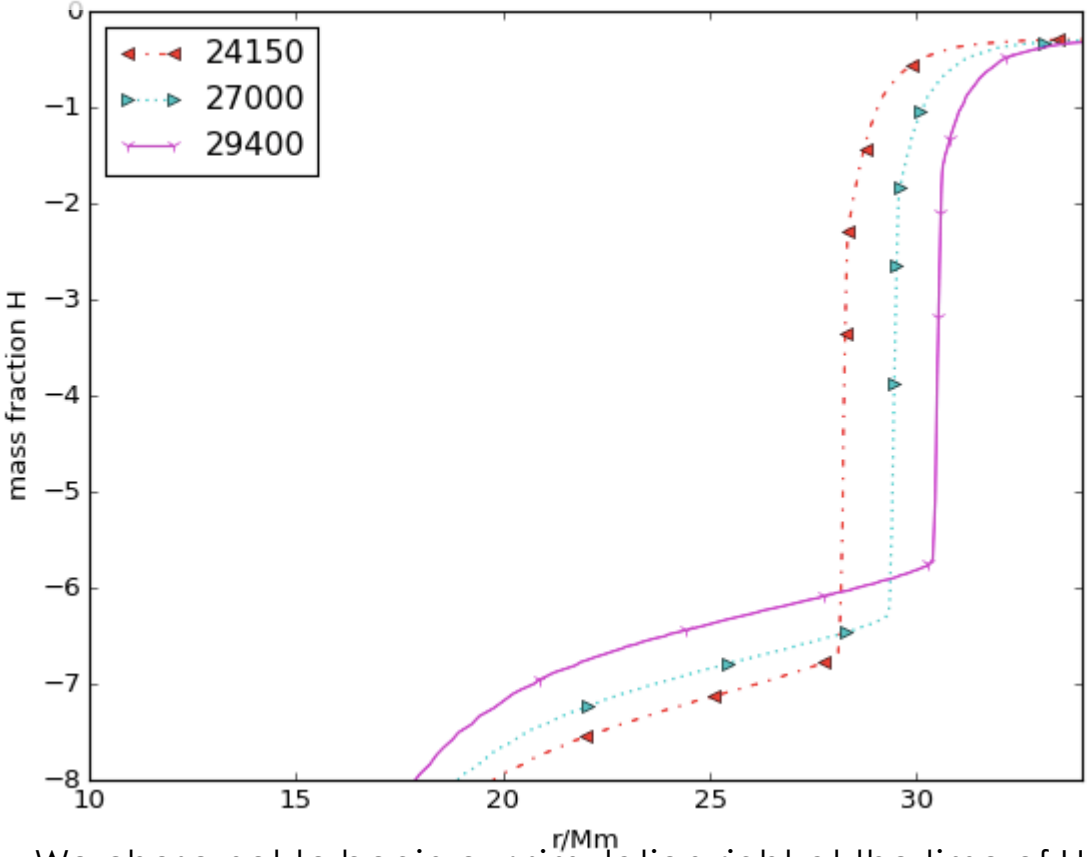


We begin from a 1-D stellar evolution calculation for a 2 solar mass AGB star with $Z = 10^{-5}$. The low entropy barrier causes H to be ingested, producing a new convection zone above the new H-burning shell. This does not happen in 3D

We chose not to begin our simulation right at the time of H-ingestion in the 1-D run, but instead at about the middle of this diagram's time interval, when the energy release near the middle of the convection zone shows that some very small ingestion must be happening in the 1-D run. We want to get the ingestion process "right" in 3-D, but cannot afford to integrate in 3-D for the long time interval the 1-D simulation indicates is needed.

Consequently, we accelerate the time evolution of the H-ingestion without, we think, falsifying its basic dynamics by increasing the He-shell flash luminosity by a factor of 22.5.

Here is ingested H mass fraction diagram for the 1-D MESA simulation. We have zoomed into the short time interval before the 1-D code ingests H into the He-shell-flash convection zone.

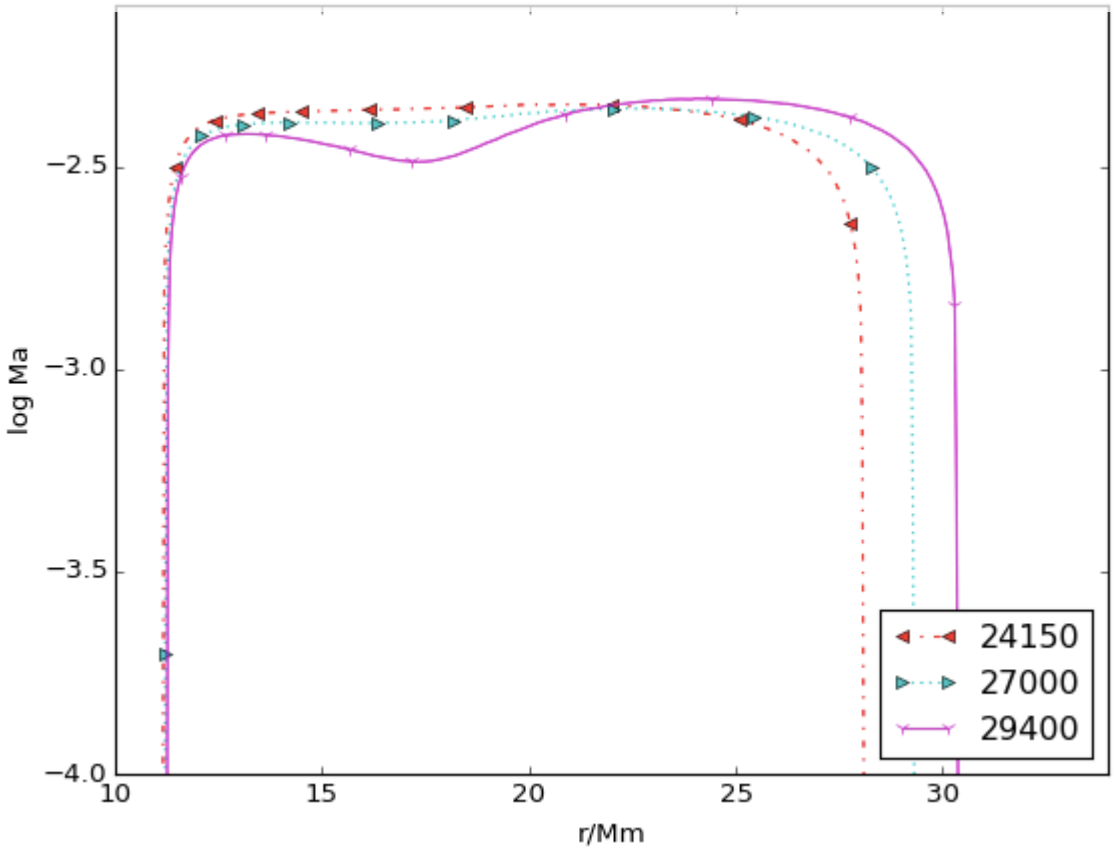


We begin from a 1-D stellar evolution calculation for a 2 solar mass AGB star with $Z = 10^{-5}$. The low entropy barrier causes H to be ingested, producing a new convection zone above the new H-burning shell. This does not happen in 3D

We chose not to begin our simulation right at the time of H-ingestion in the 1-D run, but instead at about the middle of this diagram's time interval, when the energy release near the middle of the convection zone shows that some very small ingestion must be happening in the 1-D run. We want to get the ingestion process "right" in 3-D, but cannot afford to integrate in 3-D for the long time interval the 1-D simulation indicates is needed.

Consequently, we accelerate the time evolution of the H-ingestion without, we think, falsifying its basic dynamics by increasing the He-shell flash luminosity by a factor of 22.5.

Here is the Mach number for the 1-D MESA simulation.
We have zoomed into the short time interval before the 1-D code ingests H into the He-shell-flash convection zone.

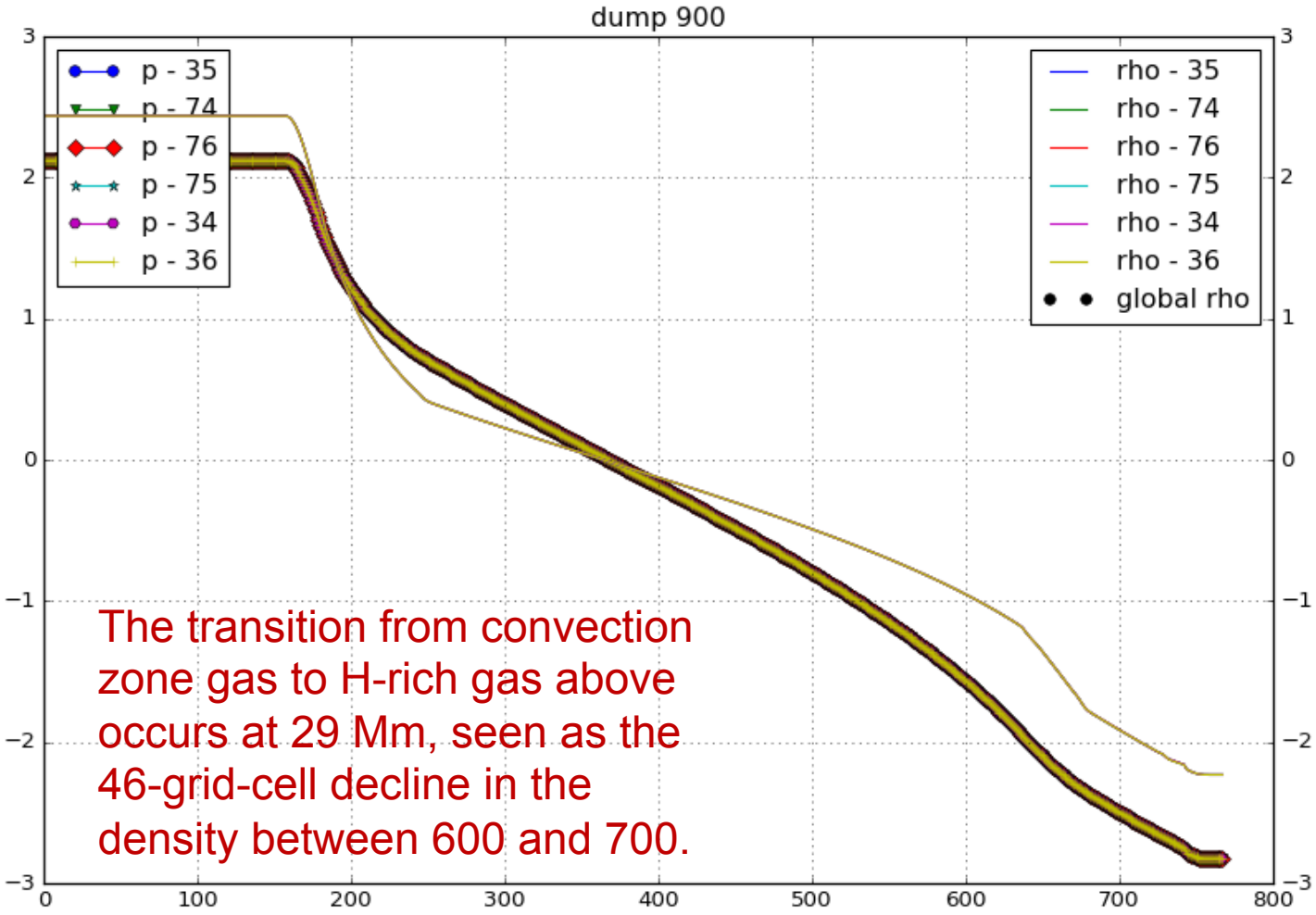


We begin from a 1-D stellar evolution calculation for a 2 solar mass AGB star with $Z = 10^{-5}$. The low entropy barrier causes H to be ingested, producing a new convection zone above the new H-burning shell. This does not happen in 3D

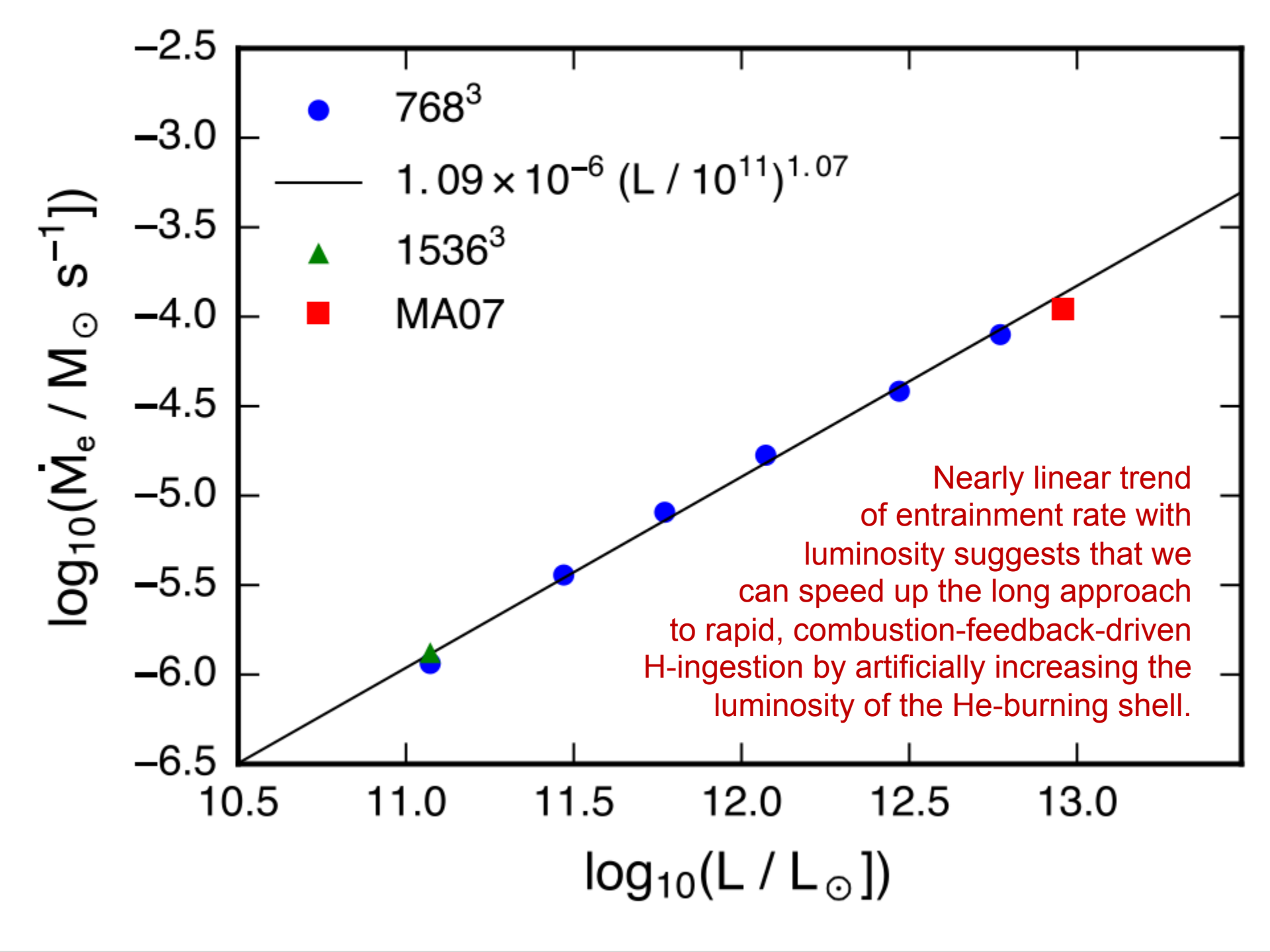
We chose not to begin our simulation right at the time of H-ingestion in the 1-D run, but instead at about the middle of this diagram's time interval, when the energy release near the middle of the convection zone shows that some very small ingestion must be happening in the 1-D run. We want to get the ingestion process "right" in 3-D, but cannot afford to integrate in 3-D for the long time interval the 1-D simulation indicates is needed.

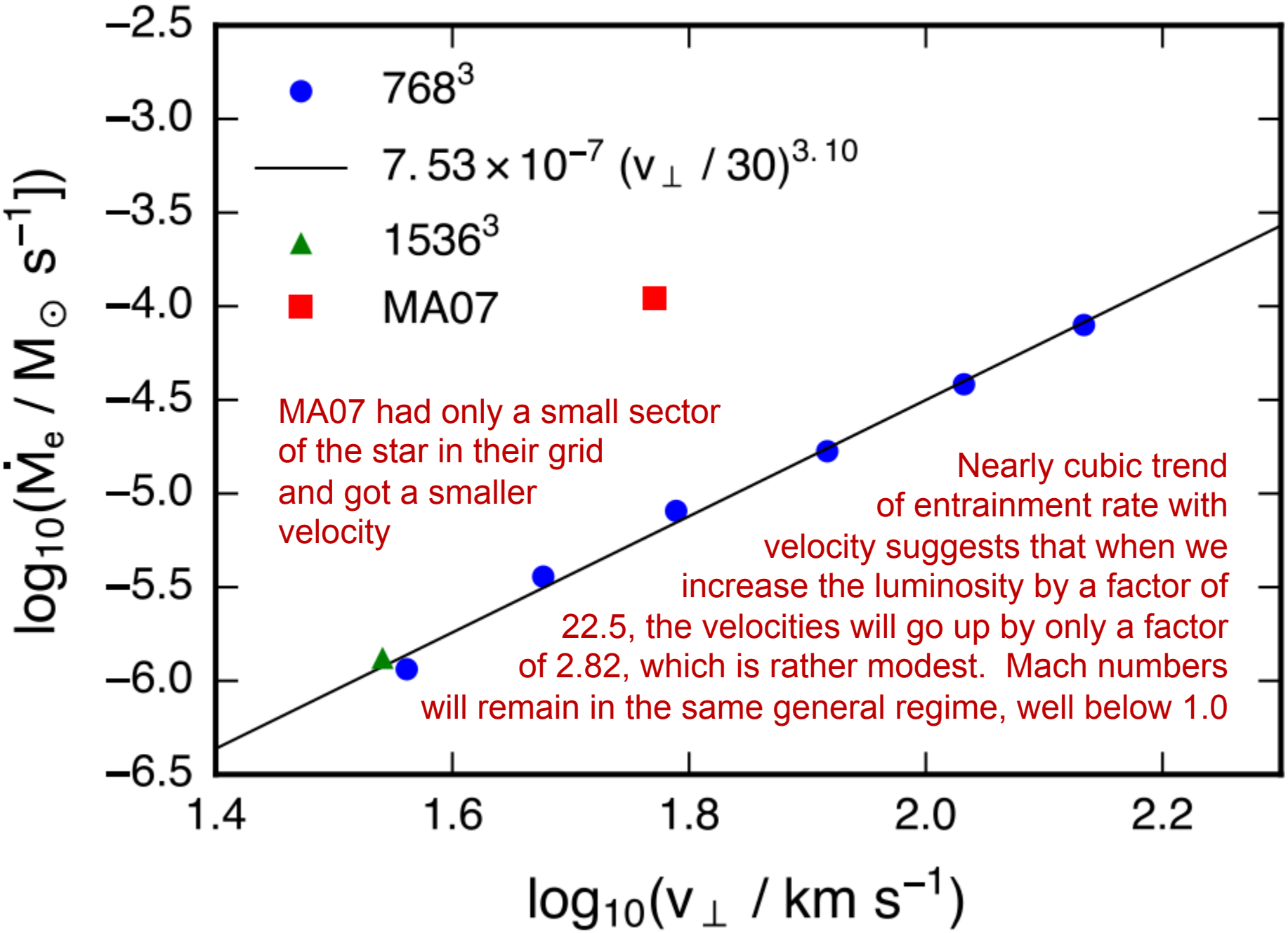
Consequently, we accelerate the time evolution of the H-ingestion without, we think, falsifying its basic dynamics by increasing the He-shell flash luminosity by a factor of 22.5.

Simulation of the Low-Z AGB Star at 1536^3 grid resolution.

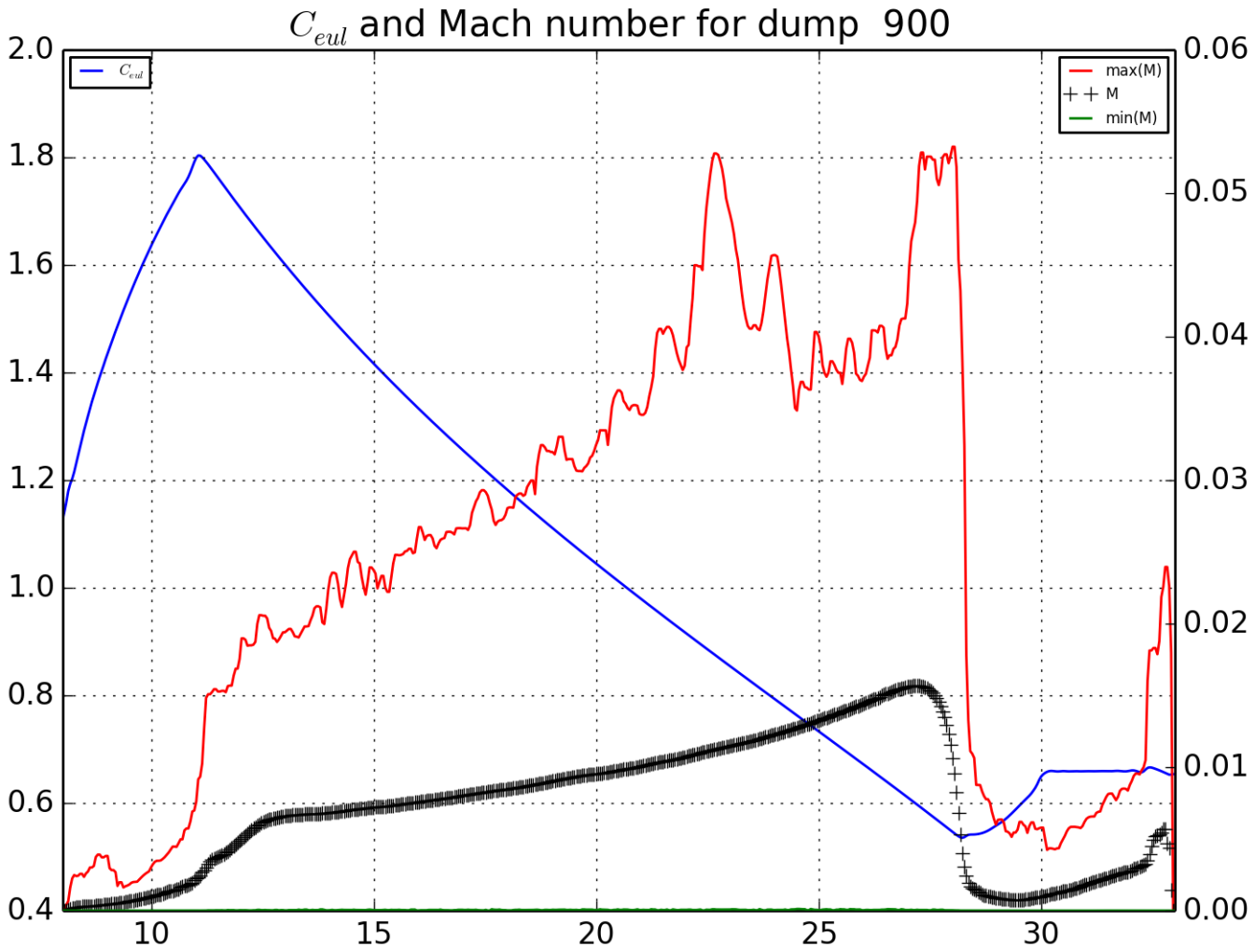


These are the pressure and density values in 6 different "bucket" directions at dump 900 as a function of radial grid cell number. These 6 plots are essentially on top of each other. The transition from the convection zone gas to the H-rich gas above is evident in the density plot, and the upper, essentially isothermal region above that is also easily made out.


768^3
$1.09 \times 10^{-6}\ (L / 10^{11})^{1.07}$
1536^3
MA07
Nearly linear trend of entrainment rate with luminosity suggests that we can speed up the long approach to rapid, combustion-feedback-driven H-ingestion by artificially increasing the luminosity of the He-burning shell.
$\log_{10}(\dot{M}_e / M_\odot\ s^{-1}])$
$\log_{10}(L / L_\odot])$
−2.5
−3.0
−3.5
−4.0
−4.5
−5.0
−5.5
−6.0
−6.5
10.5
11.0
11.5
12.0
12.5
13.0

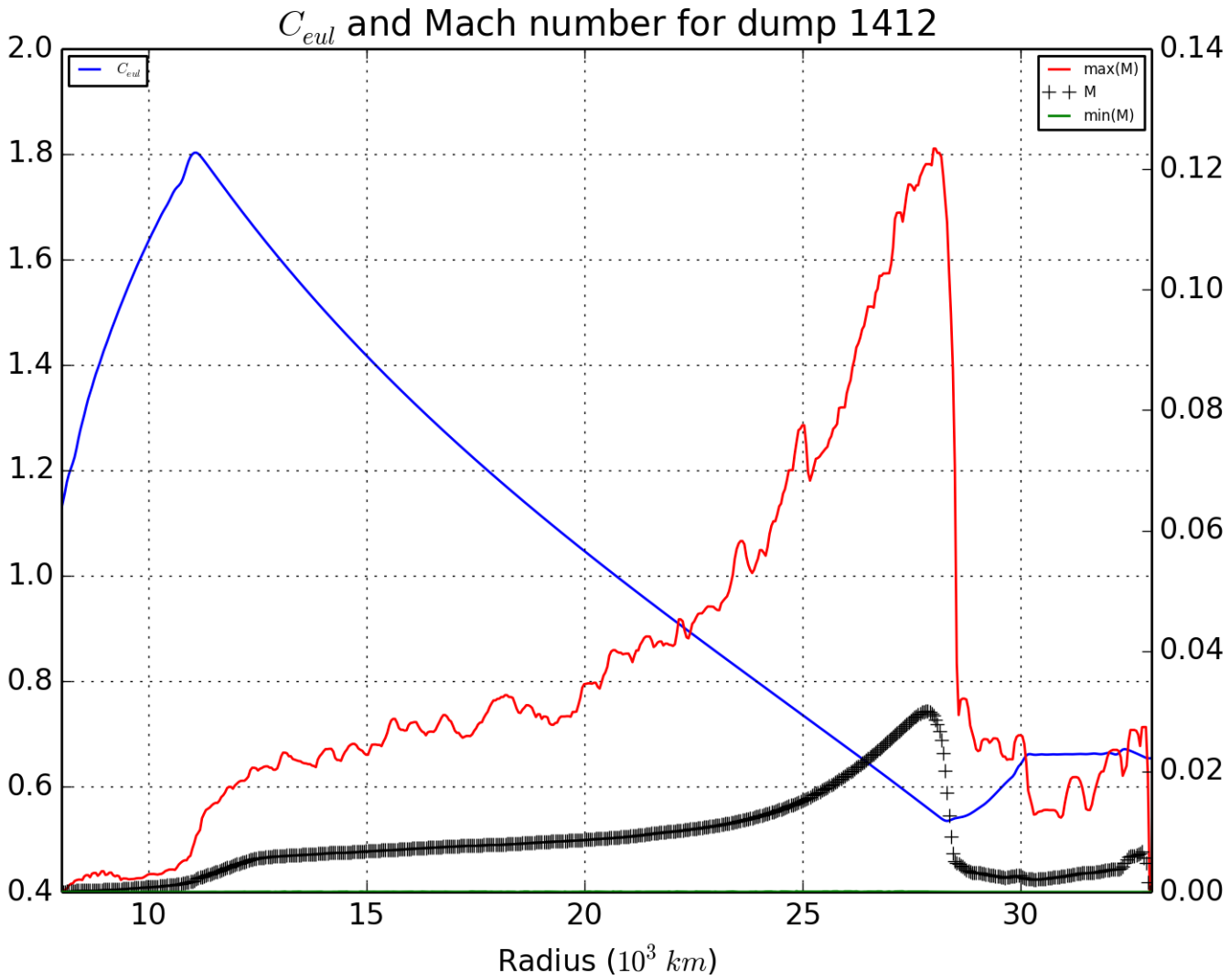

768^3
$7.53 \times 10^{-7}\ (v_\perp / 30)^{3.10}$
1536^3
MA07
MA07 had only a small sector of the star in their grid and got a smaller velocity
Nearly cubic trend of entrainment rate with velocity suggests that when we increase the luminosity by a factor of 22.5, the velocities will go up by only a factor of 2.82, which is rather modest. Mach numbers will remain in the same general regime, well below 1.0
$\log_{10}(\dot{M}_e / M_\odot\ s^{-1})$
$\log_{10}(v_\perp / km\ s^{-1})$
−2.5
−3.0
−3.5
−4.0
−4.5
−5.0
−5.5
−6.0
−6.5
1.4
1.6
1.8
2.0
2.2

Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 1298 min.



These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 900 as a function of radius (Mm). Not much has changed in these radial profiles since dump 200, except right up close to the artificial spherical reflecting boundary at the top, which has not affected the dynamics.

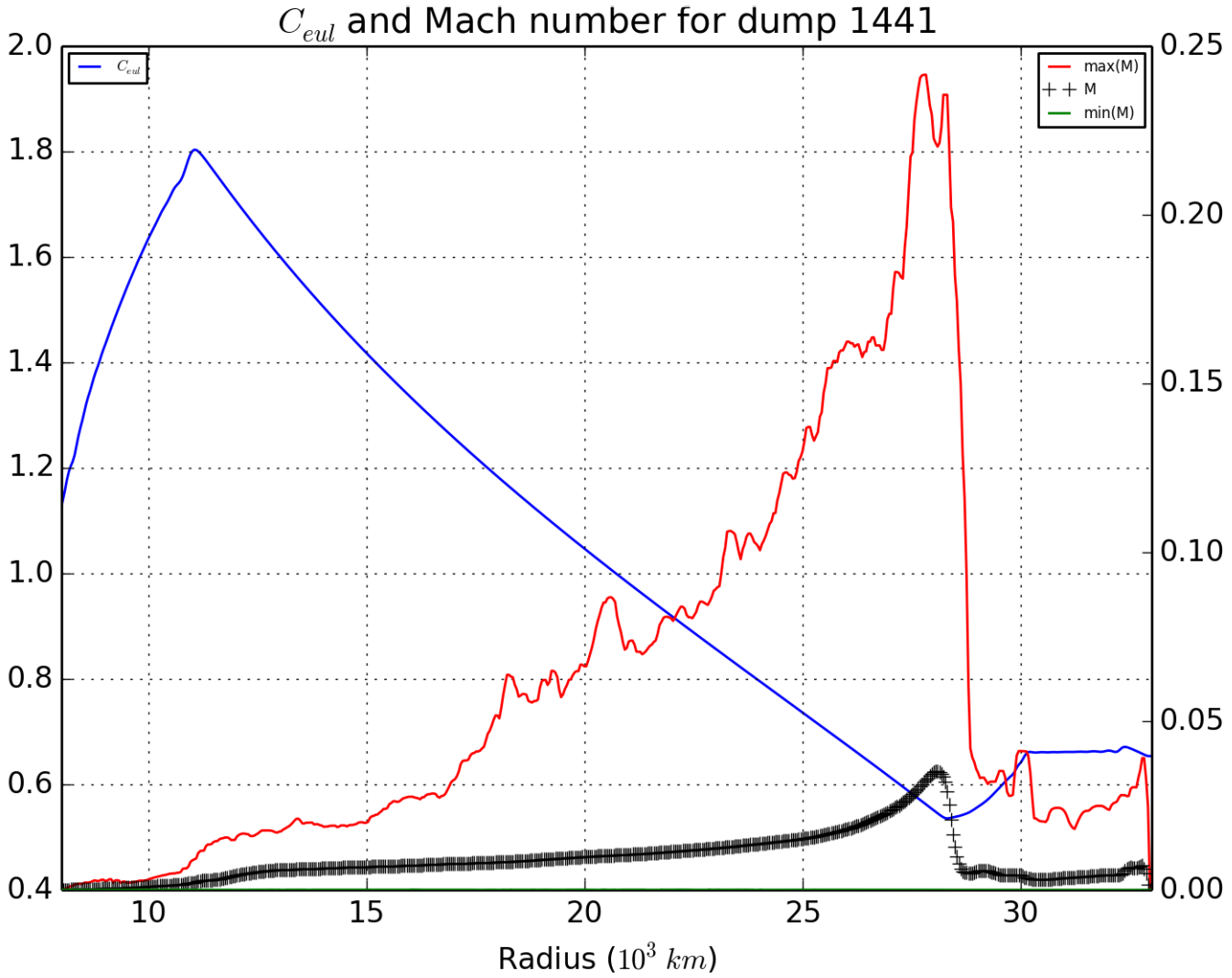
Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2036.8 min.



These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1412 as a function of radius (Mm). Not much has changed in the global radial profiles since dump 200 below 20 Mm, but the Mach numbers, which reflect tsunami-like wave activity, are very different above 20 Mm.

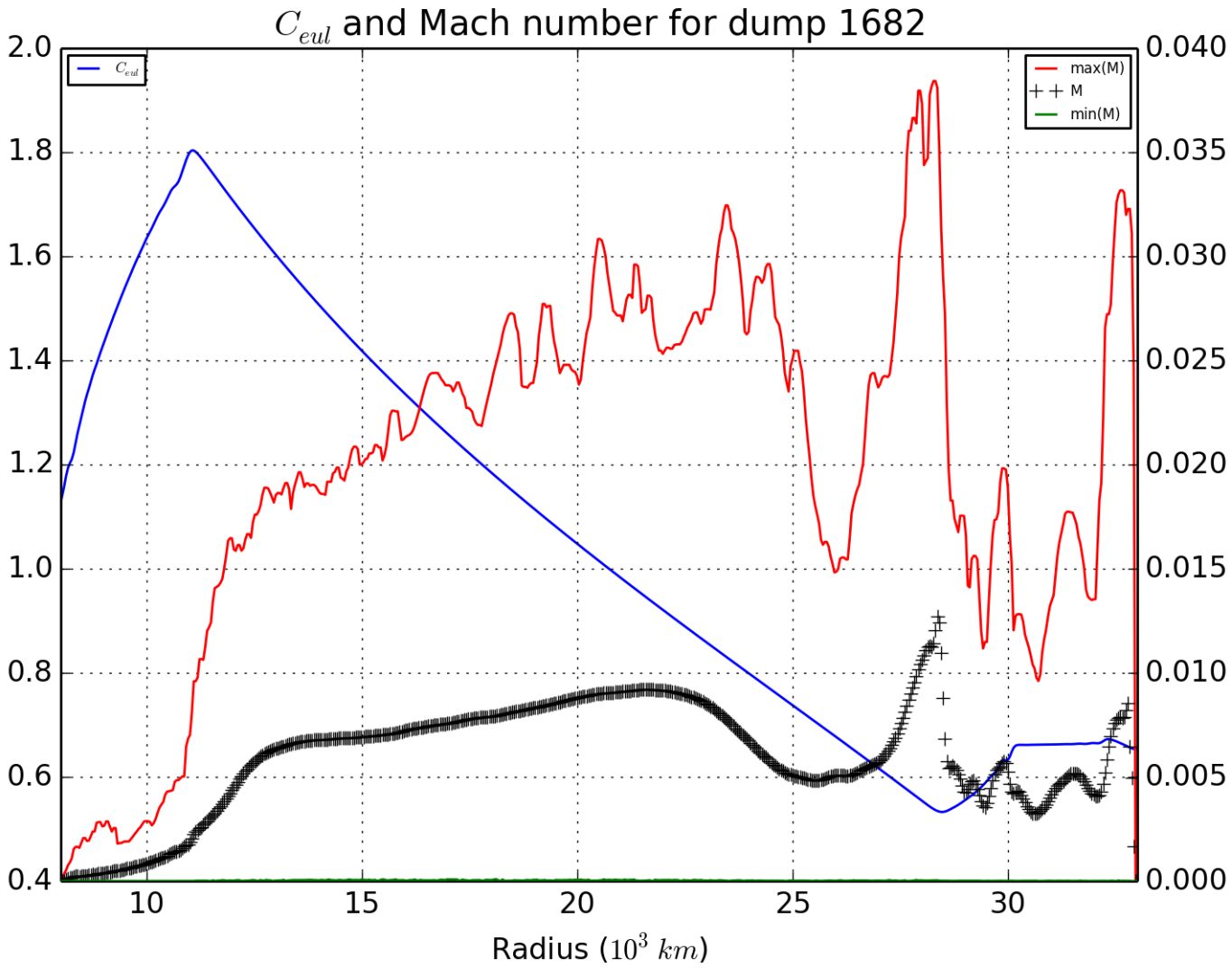
Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2078.6 min. Here we see the onset of the GOSH.

Because we increased the luminosity by 22.5 times, 2079 minutes scales to 32.5 days for the star. The one-month approach to this violent H-ingestion event would be very expensive to compute directly.



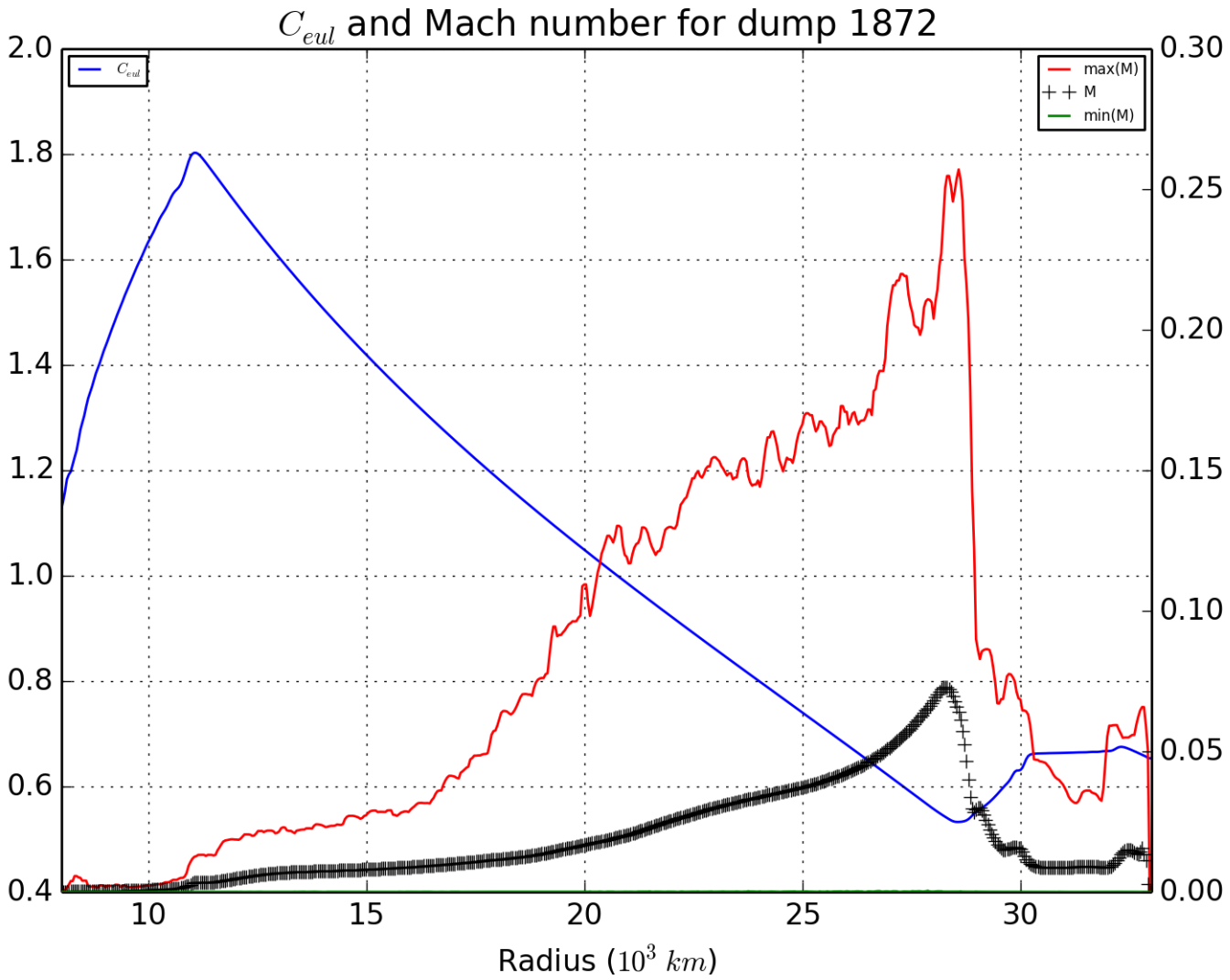
These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1441 as a function of radius (Mm). Not much has changed in the global radial profiles since dump 200 below 17 Mm, but the Mach numbers, which reflect tsunami-like wave activity, are very different above 17 Mm.

Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2426.3 min.



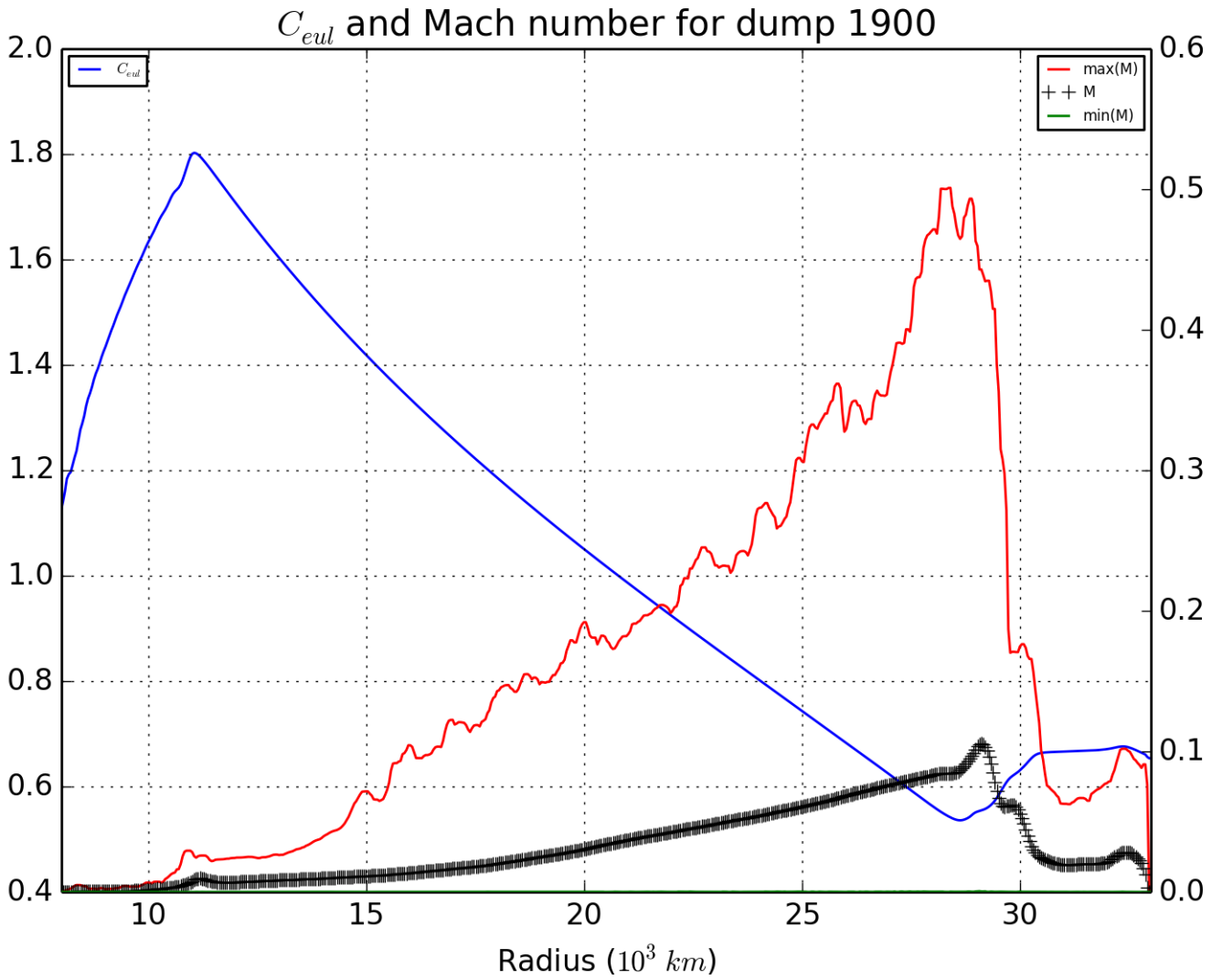
These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1682 as a function of radius (Mm). Now an H-enriched shell has formed above 23 Mm, with H concentrations between 10^{-4} and 10^{-3} by volume, and the violent motions above that radius have died down.

Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2668.6 min.



These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1872 as a function of radius (Mm). Now the H-enriched shell that formed above 23 Mm, with H concentrations between 10^{-4} and 10^{-3} by volume, has been itself entrained, and very violent motions have resulted.

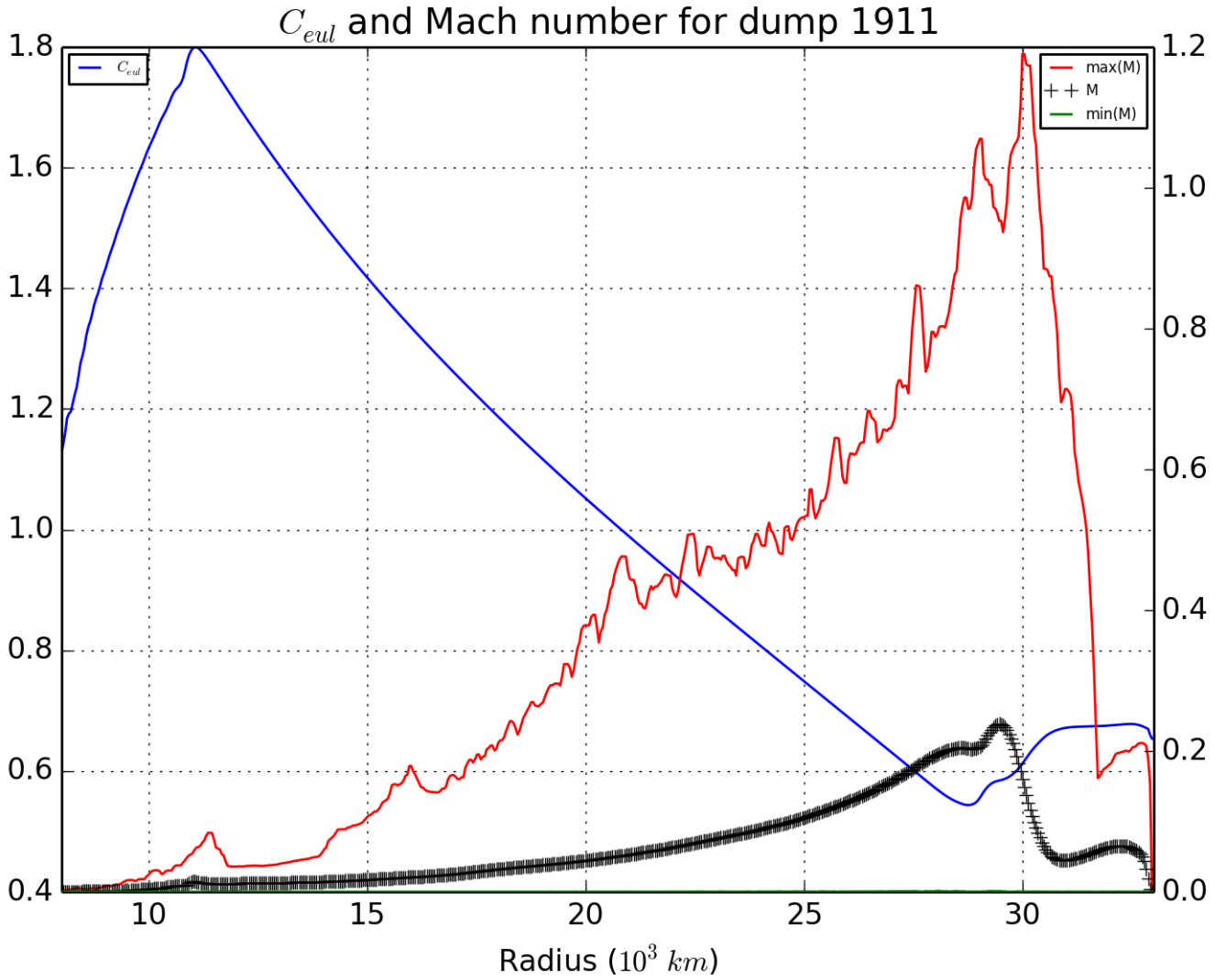
Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2688.8 min.



These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1900 as a function of radius (Mm). Now the H-enriched shell that formed above 23 Mm, with H concentrations between 10^{-4} and 10^{-3} by volume, has been itself entrained, and extremely violent motions have resulted.

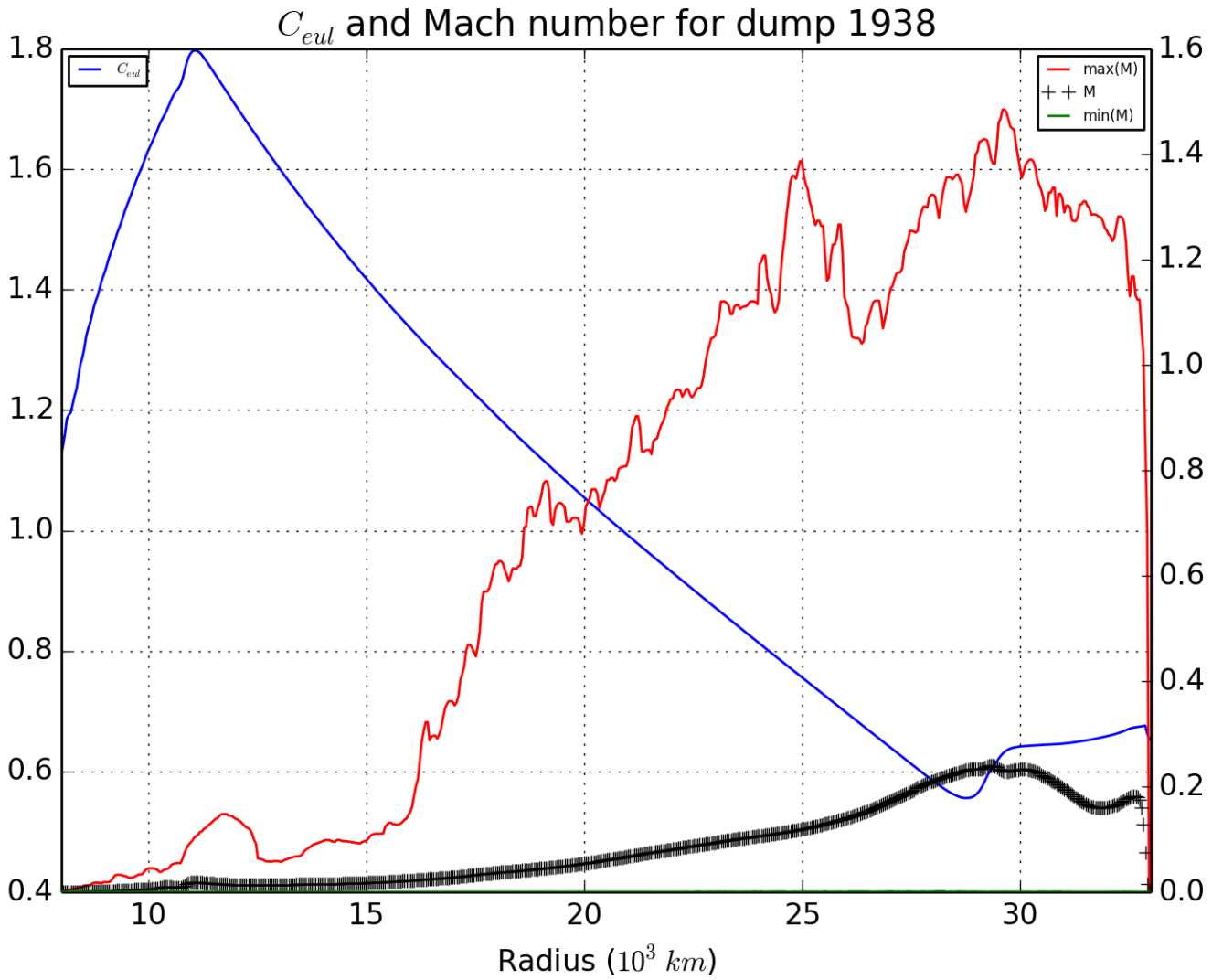
Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2695.8 min.

Because we increased the luminosity by 22.5 times, 2700 minutes scales to 42.2 days for the star. In the future we will simulate the long approach to the H-ingestion outburst using a 1D-3D technique.



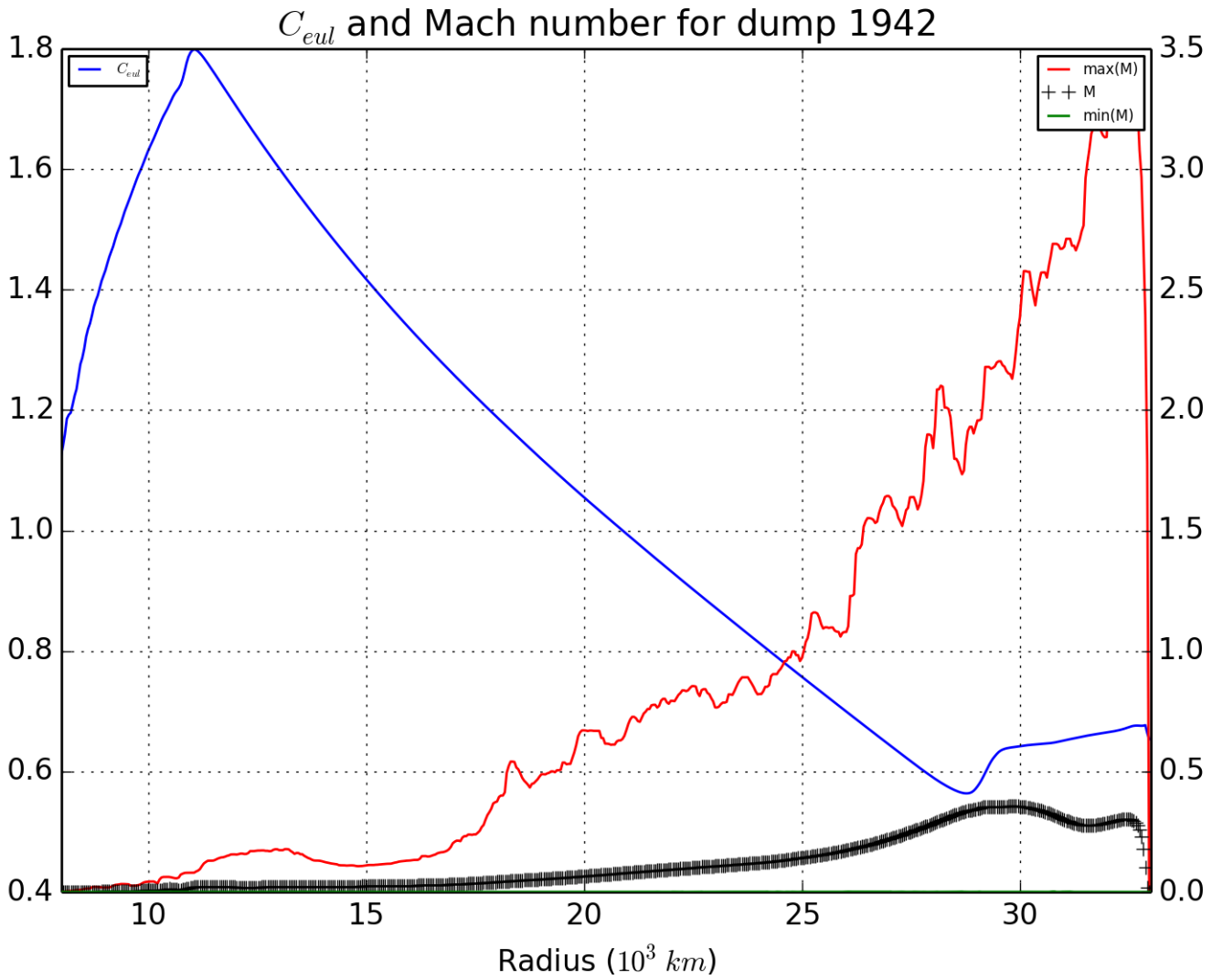
These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1911 as a function of radius (Mm). Now the H-enriched shell that formed above 23 Mm, with H concentrations between 10^{-4} and 10^{-3} by volume, has been itself entrained, and extremely violent motions have resulted.

Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2700.7 min.



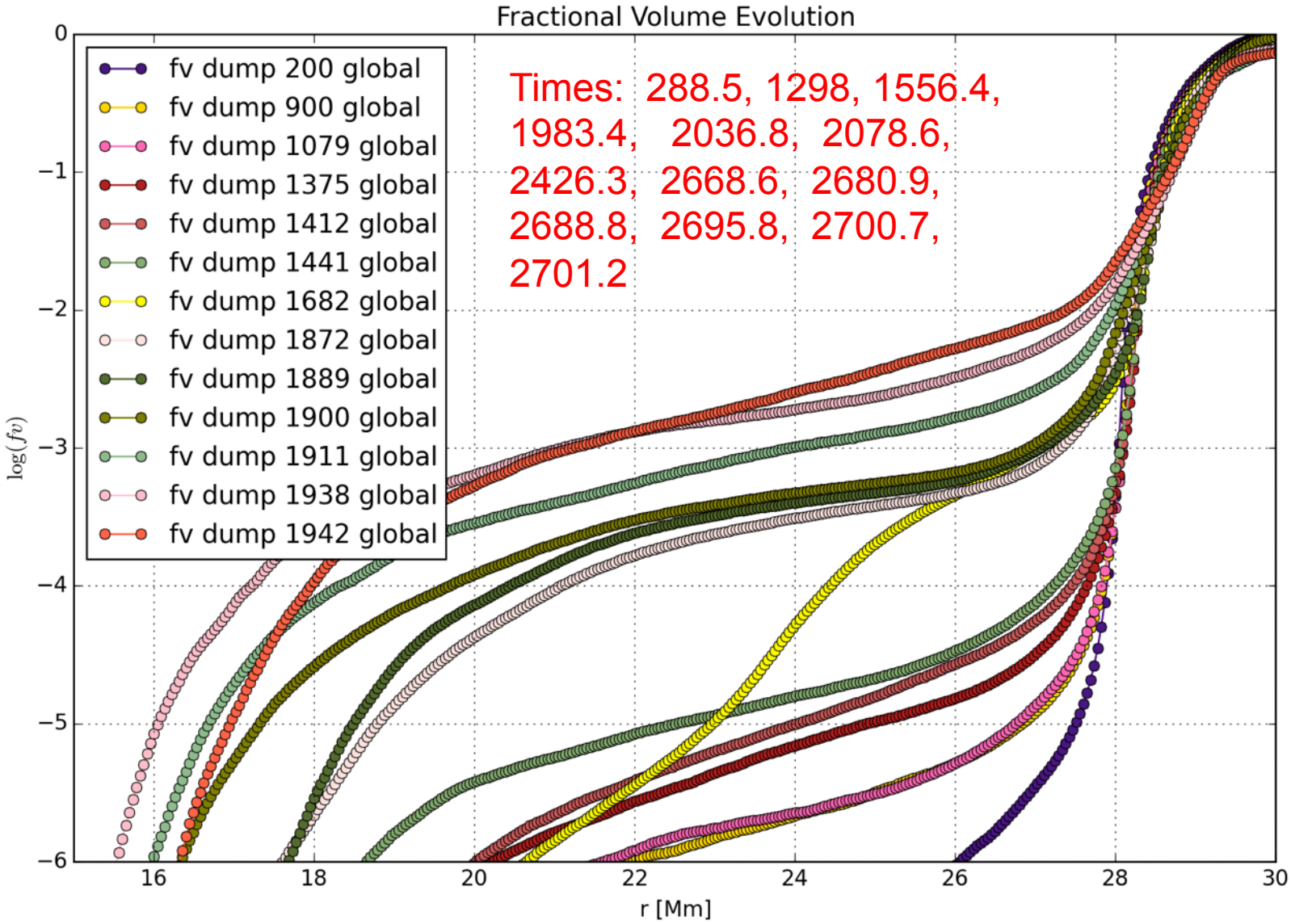
These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1938 as a function of radius (Mm). Now the violent motions driven by H-burning have penetrated thoroughly into the region above the convective boundary, and the temperature there is beginning to rise. The H-concentration is 10^{-4} as deep down as 17 Mm.

Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 2701.2 min.

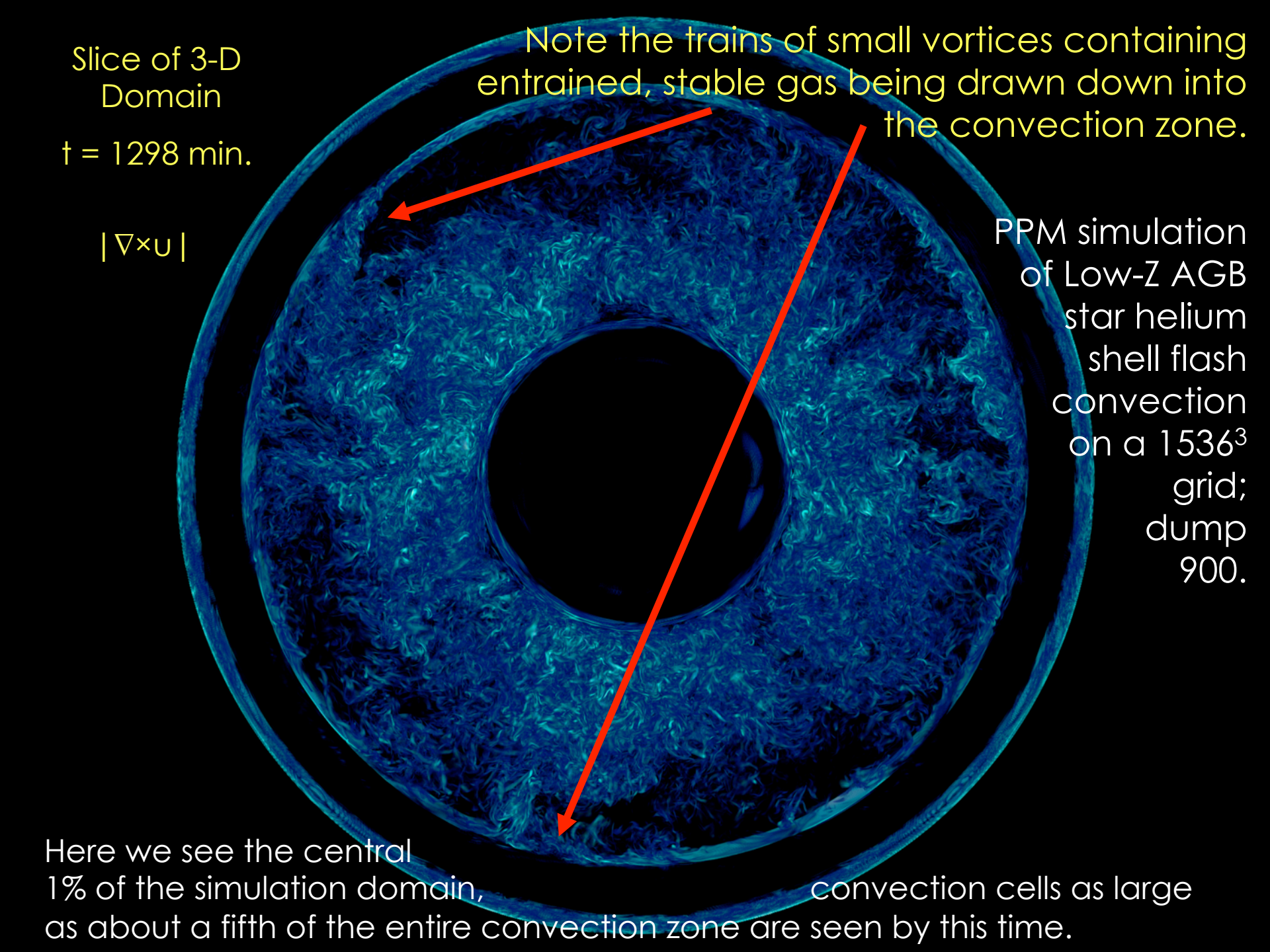


These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 1942 as a function of radius (Mm). Now the violent motions driven by H-burning have penetrated thoroughly into the region above the convective boundary, and the temperature there is beginning to rise. The H-concentration is 10^{-4} as deep down as 17 Mm.

Simulation of the Low-Z AGB Star at 1536^3 grid resolution @ 288.5 min.



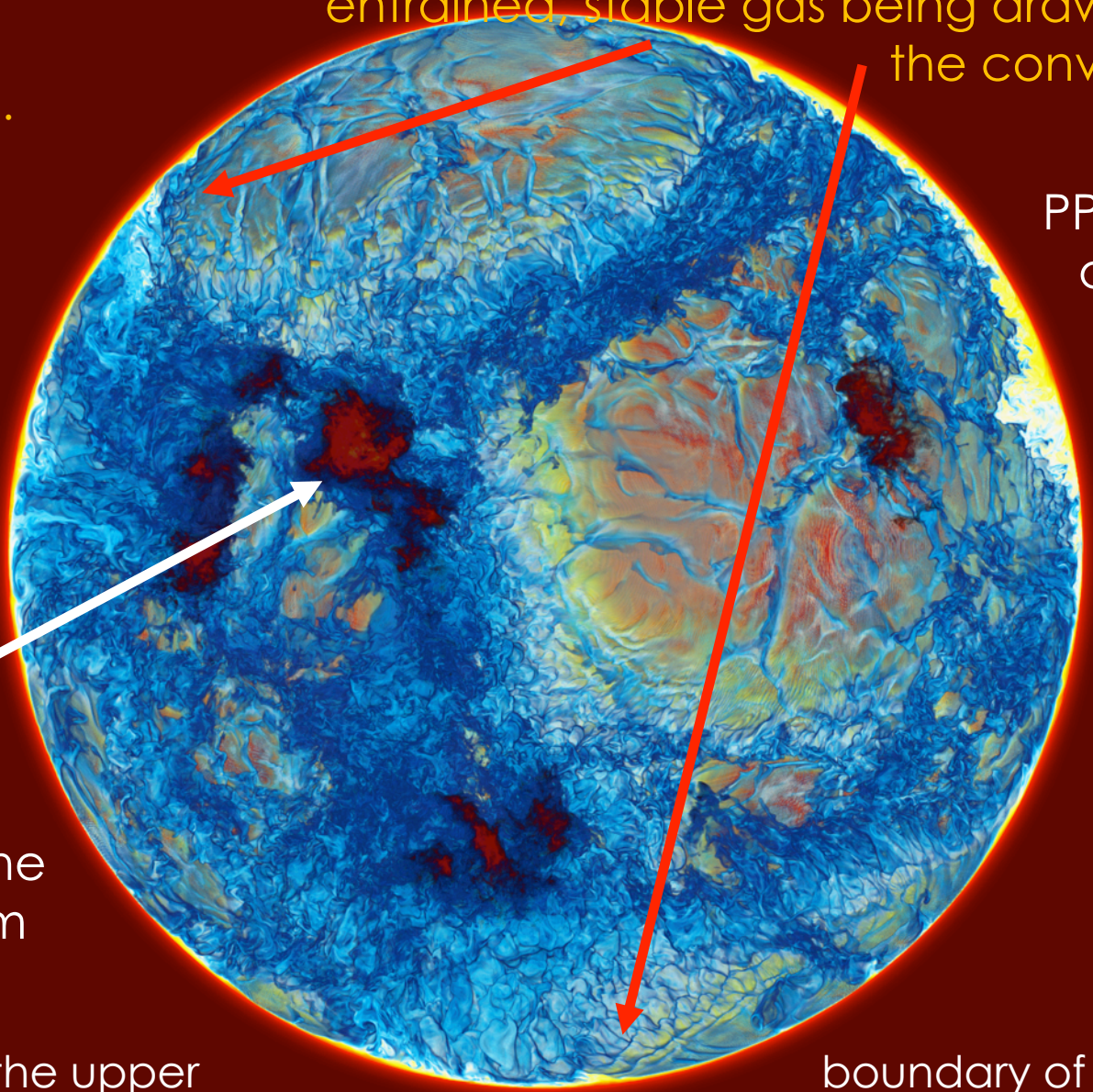
These are the globally averaged radial distributions of Eulerian sound speed and Mach number, as well as maximum Mach number, at dump 200 as a function of radius (Mm). At this relatively early time in the simulation, the initial readjustments of the initial state have had time to give way to a statistically nearly steady continued evolution.


Slice of 3-D Domain
t = 1298 min.
|∇×u|
Note the trains of small vortices containing entrained, stable gas being drawn down into the convection zone.
PPM simulation of Low-Z AGB star helium shell flash convection on a 1536^3 grid; dump 900.
Here we see the central 1% of the simulation domain, as about a fifth of the entire convection zone
convection cells as large are seen by this time.

Half of 3-D Domain

t = 1298 min.

FV_{H+He}

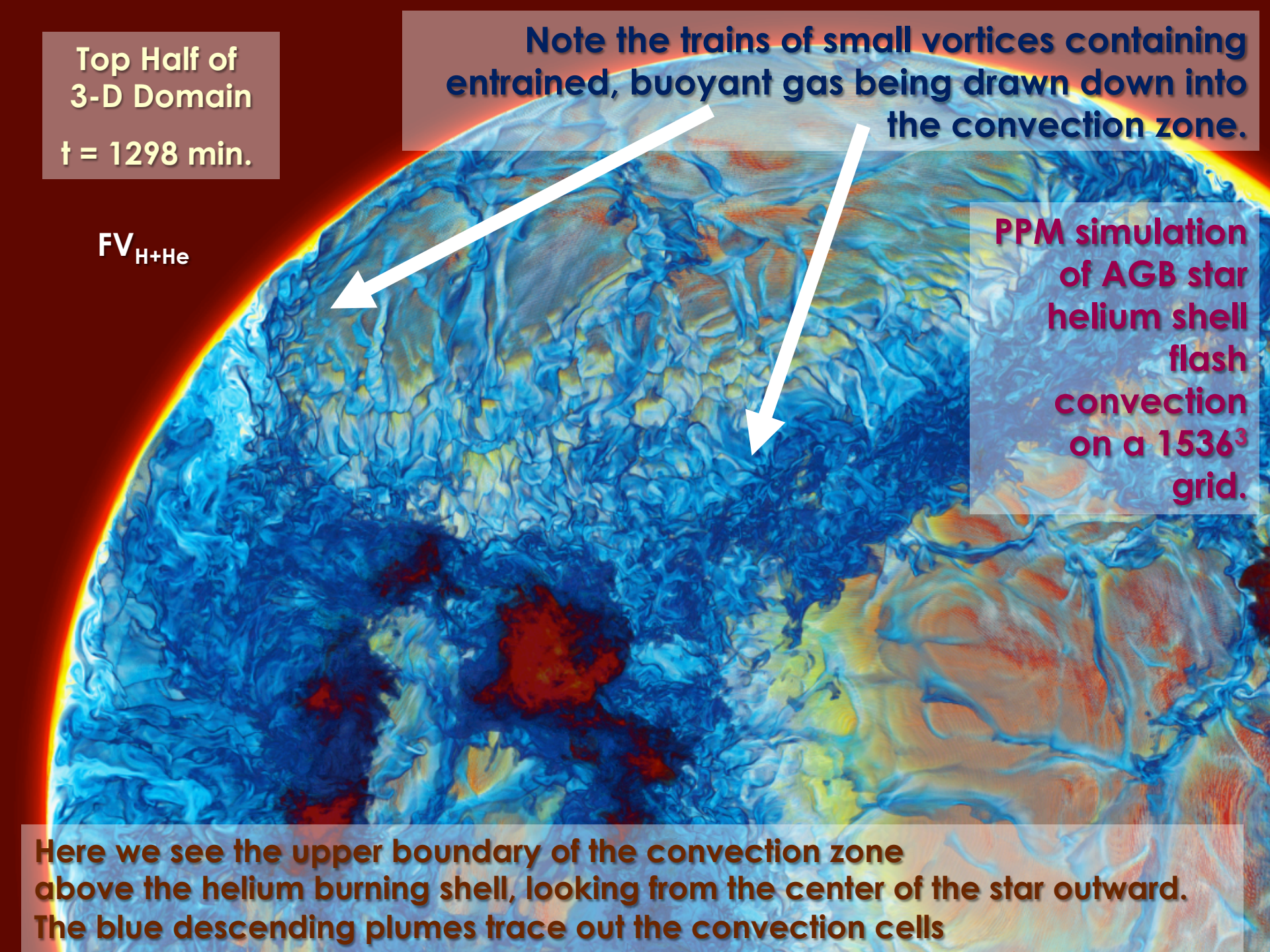


Note the trains of small vortices containing entrained, stable gas being drawn down into the convection zone.

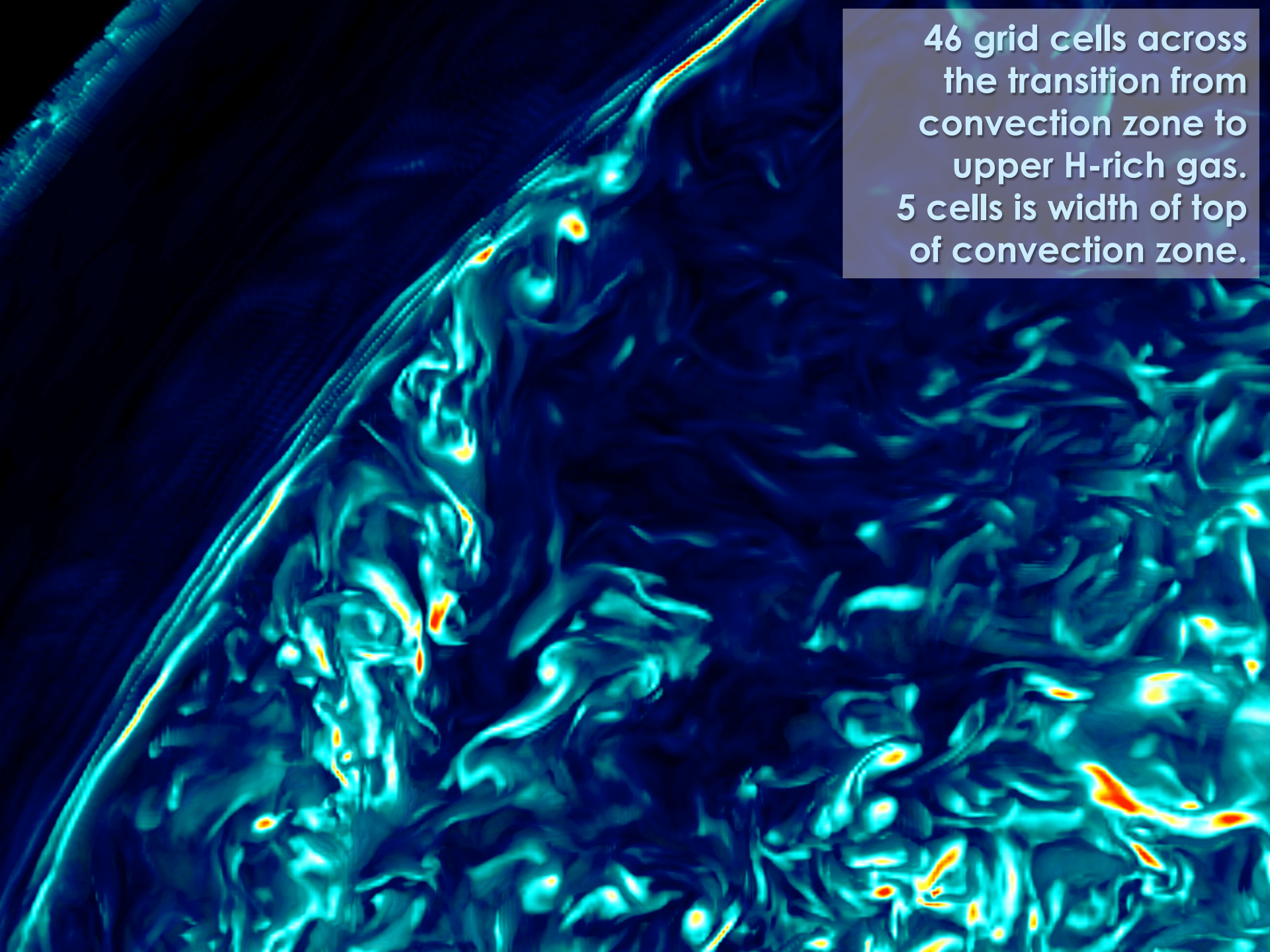
PPM simulation of Low-Z AGB star helium shell flash convection on a 1536^3 grid; dump 900.

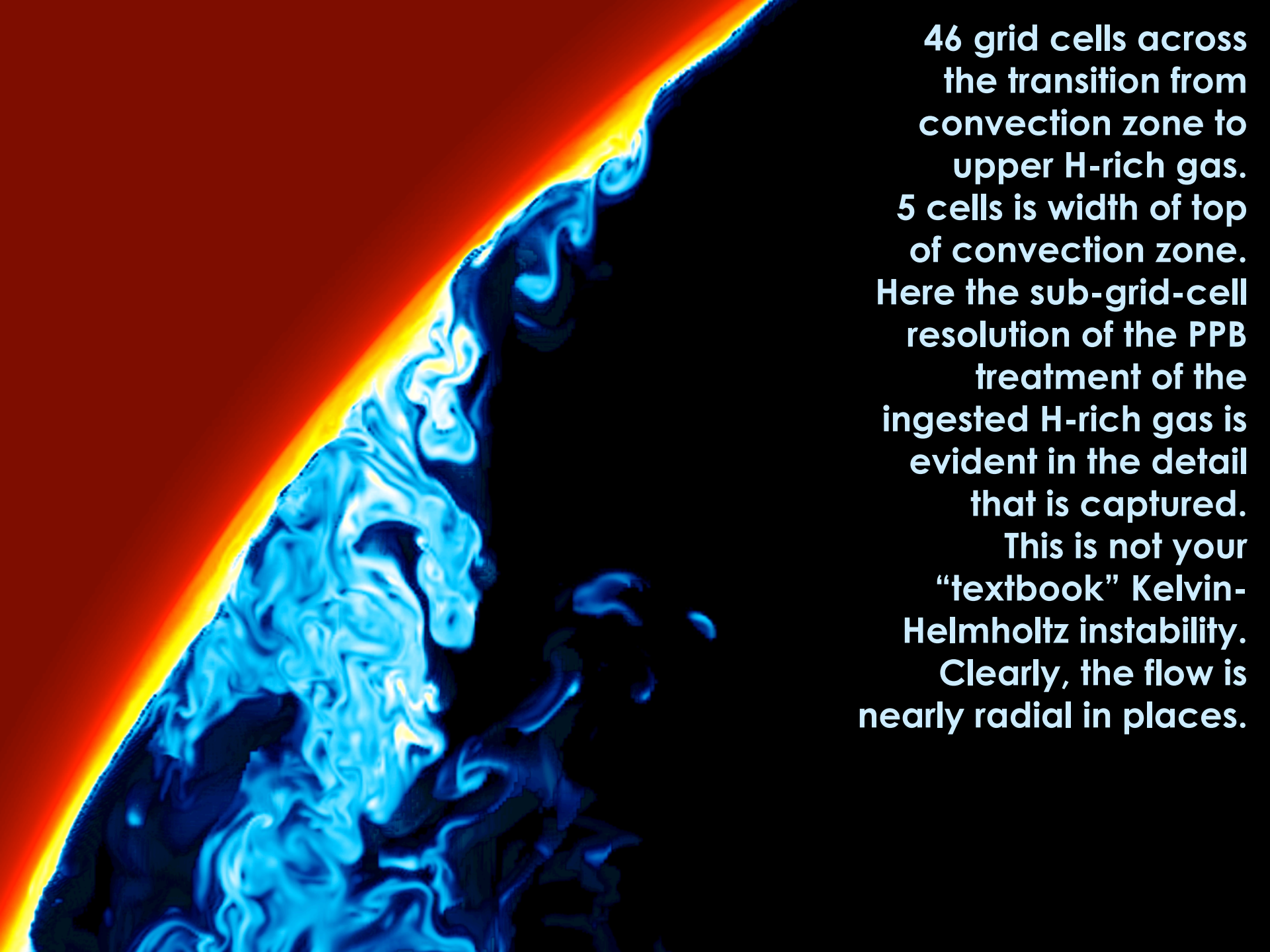
Locations where ingested H burns are seen in the purple & red flames denoting the energy from H burining.

Here we see the upper boundary of the convection zone above the helium burning shell, looking from the center of the star outward. The blue descending plumes trace out the convection cells


Top Half of
3-D Domain
t = 1298 min.
FV H+He
Note the trains of small vortices containing entrained, buoyant gas being drawn down into the convection zone.
PPM simulation of AGB star helium shell flash convection on a 1536³ grid.
Here we see the upper boundary of the convection zone above the helium burning shell, looking from the center of the star outward. The blue descending plumes trace out the convection cells

Zoomed in views of vorticity magni-tude (1st) and log of the fractional volume of H-rich fluid (2nd) in a thin slice through the center of a 2 solar mass star model. We see resolved, breaking Kelvin-Helmholtz waves where the flow separates from the top of the convection zone ("top" is up and to the left in this view) because it meets oppos-itely directed flow in an adjacent large convection cell. The level of detail is greater in the mixing fraction of entrained gas in the 2nd image, because we have exploited PPB's 10 moments in each cell of this 1536^3 grid to display this variable at twice the grid resolution. The 2 Mm initial thickness of the transition layer from convection zone gas to more buoyant, H-rich gas above the convection zone is 46 grid cell widths. The smallest vortices are about 5 cells across. Clearly, our PPMstar code in this simulation (see [1]) is resolving features smaller than the transition layer, which is likely to be one reason that we observe the entrainment rate to converge with mesh refinement.


46 grid cells across
the transition from
convection zone to
upper H-rich gas.
5 cells is width of top
of convection zone.



46 grid cells across the transition from convection zone to upper H-rich gas. 5 cells is width of top of convection zone. Here the sub-grid-cell resolution of the PPB treatment of the ingested H-rich gas is evident in the detail that is captured. This is not your "textbook" Kelvin-Helmholtz instability. Clearly, the flow is nearly radial in places.

The Code must Scale and it must be Fast:

1. **For this case, need to cover 2 months for the star:**
   a. We speed things up by increasing the luminosity.
   b. Entrainment and velocity**3 scale with luminosity.
   c. Even reducing time by factor 22.5, is still 2 days.
   d. Turn-over time is 32 Mm/40 km/sec = 800 sec = 14 min.
   e. Need 9 million time steps for the 2-day integration.
2. **This is no problem if the code can scale, and if it is fast.**
   a. 4 threads per MPI process, updating 32**3 cells.
   b. Pack 8 to a node to account for 4 separate memories.
   c. 216 MPI processes per “team”, with 512 teams.
   d. 13,952 nodes.
   e. 26 time steps per second, continuously for 4 days.
   f. Data dump of 46 GB every 3 minutes.
3. **1D-3D.**
   a. Compute for a few hours, or until statistically steady.
   b. Determine coefficients of an appropriate 1-D model
   c. Go forward a few months in 1D, and reassess.

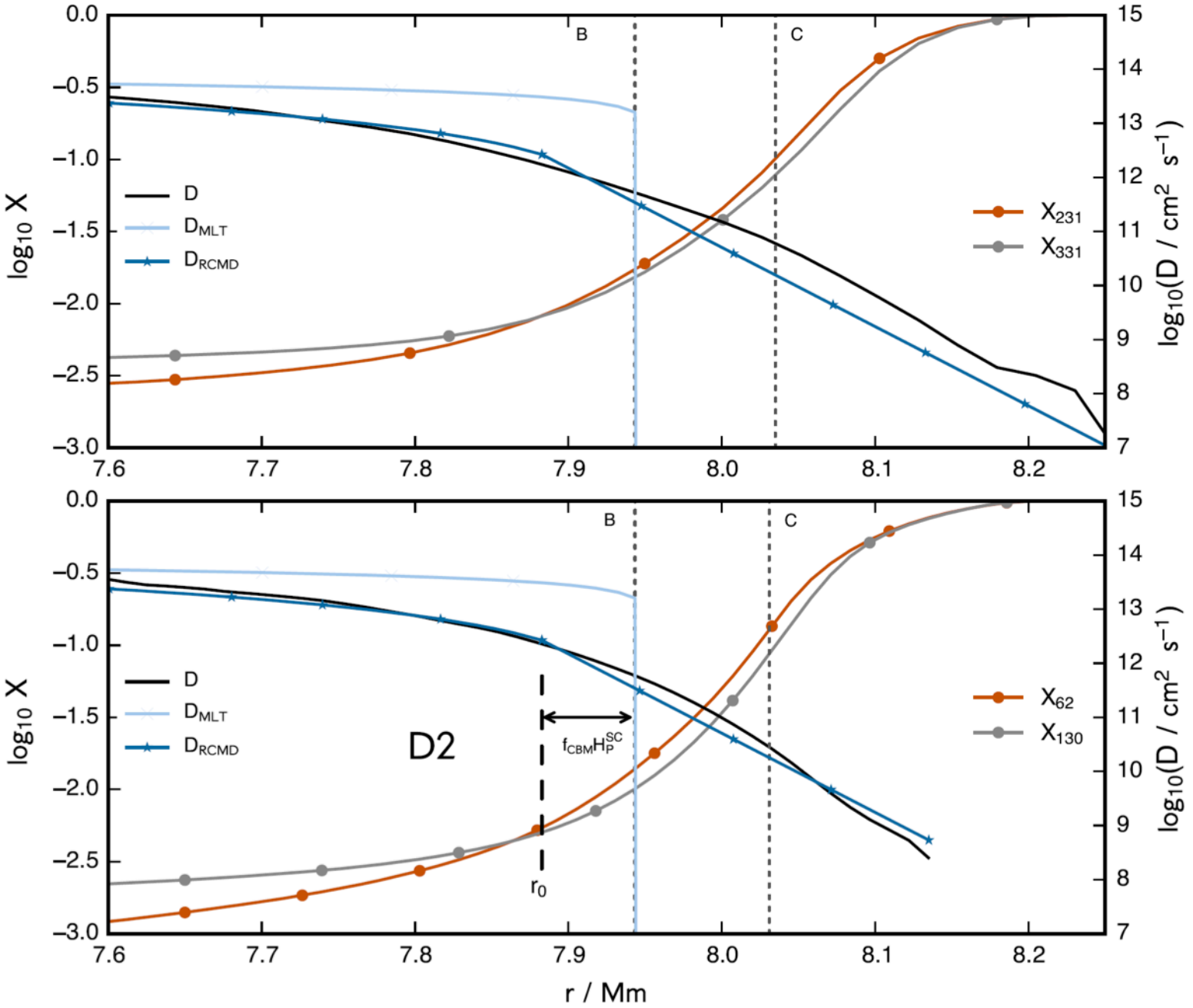

B
C
D
D_{MLT}
D_{RCMD}
X_{231}
X_{331}
$\log_{10} X$
$\log_{10}(D / cm^2\ s^{-1})$
D2
$f_{CBM}H_P^{SC}$
r_0
X_{62}
X_{130}
r / Mm

Figure 22. Results of the 3D–1D diffusion analysis at the upper convective boundary of the D1 (768^3 grid) and D2 (1536^3 grid) simulations (see Table 1). The vertical dotted lines represent the upper boundary of the convection zone. B is where the entropy gradient becomes positive in our PPMSTAR setup (equivalent to the Schwarzschild criterion); C is where the radial gradient of the tangential component of the fluid velocity is steepest after 46.7 (16.0) minutes of simulated time for simulation D1 (D2). We also give the MESA model upon which these simulations were based; it has been aligned so that the convective boundary according to the Schwarzschild criterion is located at B. X is the spherically-averaged mass fraction of the overlying fluid and is plotted at a simulated time indicated by the subscript in tens of seconds. $D_{\rm MLT} = \frac{1}{3} v_{\rm MLT} \alpha H_P$ is the diffusion coefficient computed in the framework of mixing length theory with $\alpha = 1.6$. D (solid black line) is the derived diffusion coefficient that gives the same net mixing as the 3D hydrodynamic simulation when its output is spherically averaged. $D_{\rm RCMD}$ is the recommended diffusion coefficient to use in a 1D code given by $D_{\rm RCMD} = v_{\rm MLT} \times \min(\alpha H_P, |r - r_{SC}|)$, where r_{SC} is the radial coordinate of the Schwarzschild boundary at B, as described in Section 3.6 of the text, with an exponentially decaying convective boundary mixing from radius $r_0 = r_{SC} - f_{\rm CBM} H_P$ with $f_{\rm CBM} = 0.03$, as formulated by Freytag, Ludwig & Steffen (1996, see Section 3.6).

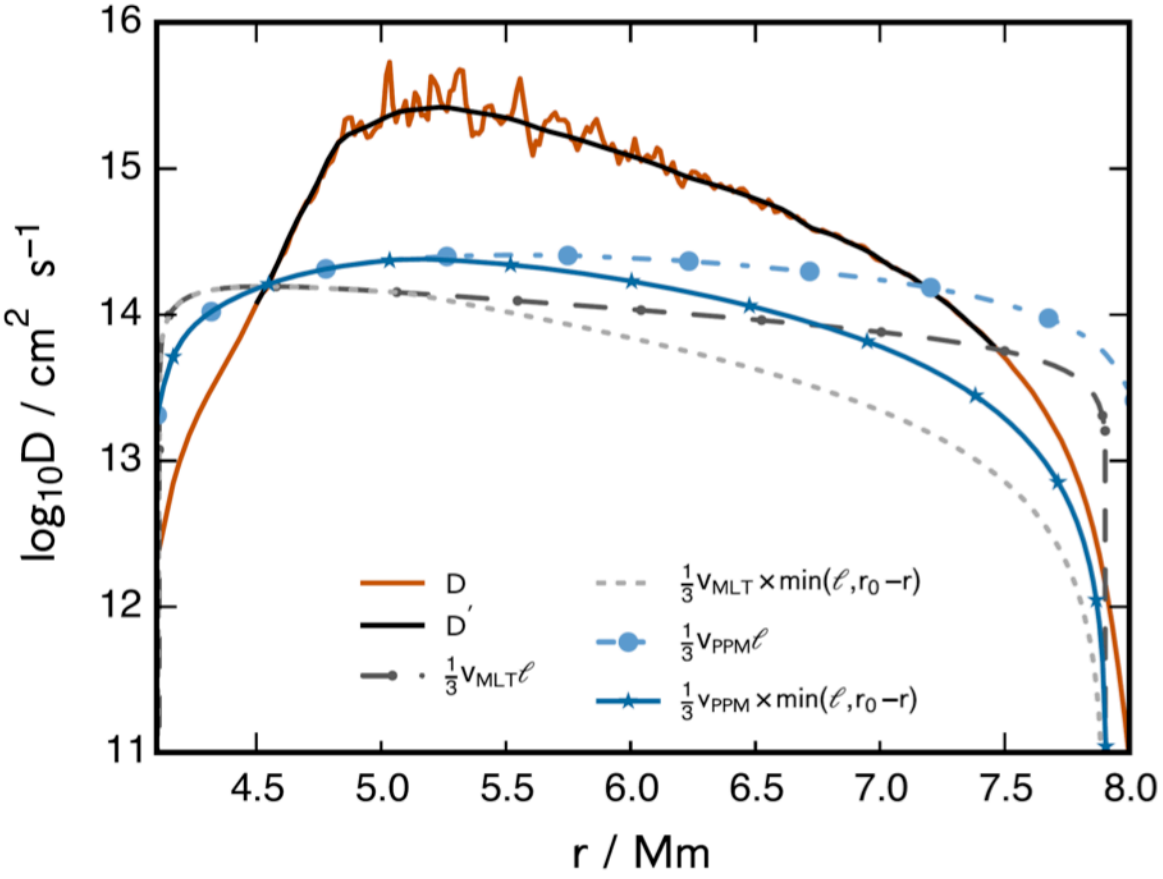
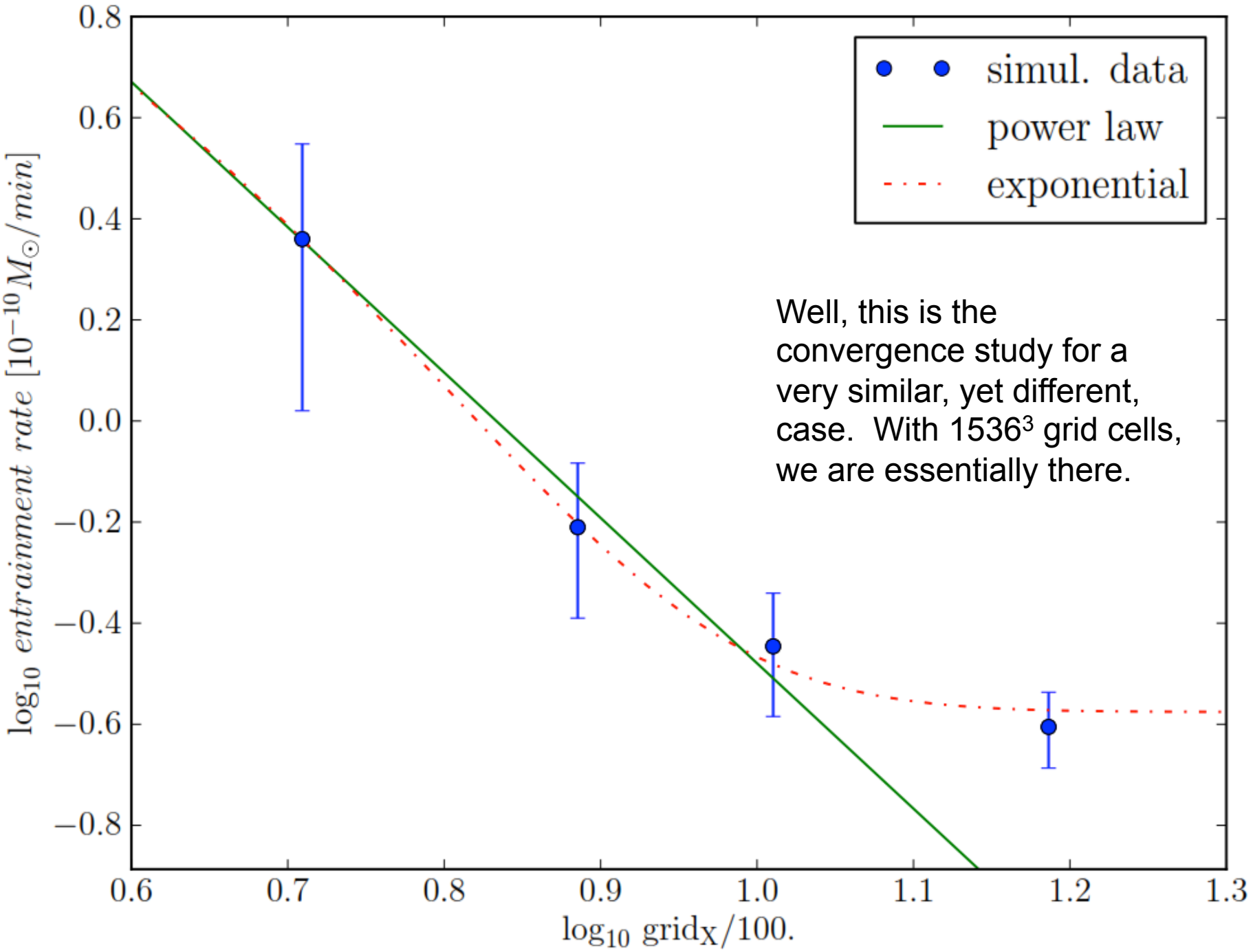


Figure 21. Time-averaged radial diffusion coefficient profile calculated from the spherically-averaged abundance profiles by the method described in Section 3.5 (brown solid line; black solid line is a fit to the noisy region). The convective velocities computing using MLT agree with the spherically-averaged 3D velocities to within about a factor of 2 inside the convection zone but are too large in the vicinity of the convective boundary, resulting in an overestimation of the diffusion coefficient there. Limiting the mixing length to the distance from the convective boundary reproduces the fall-off of the diffusion coefficient inside the convection zone approaching the boundary that is seen in the spherically averaged 3D simulation results.



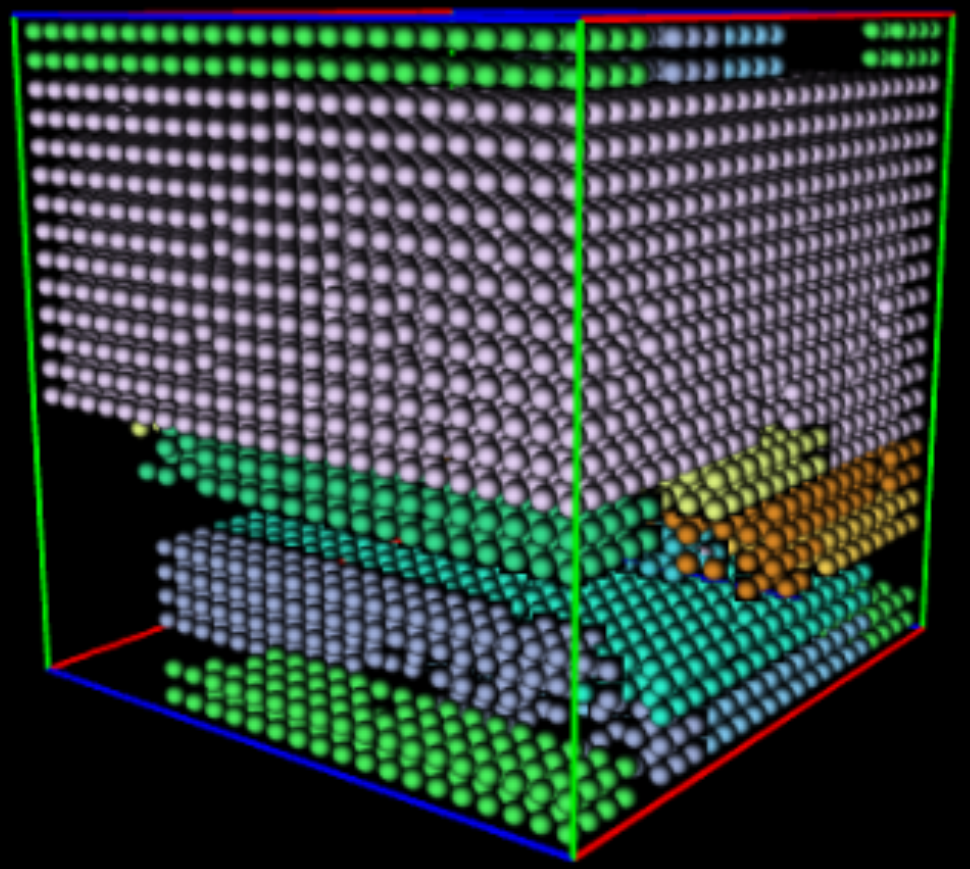
Fitting a model of exponential convergence or power law non-convergence to results.

Placement of 10 largest running jobs on the Gemini torus. Mon 27.01.2014 at 04:34:48 PM CST

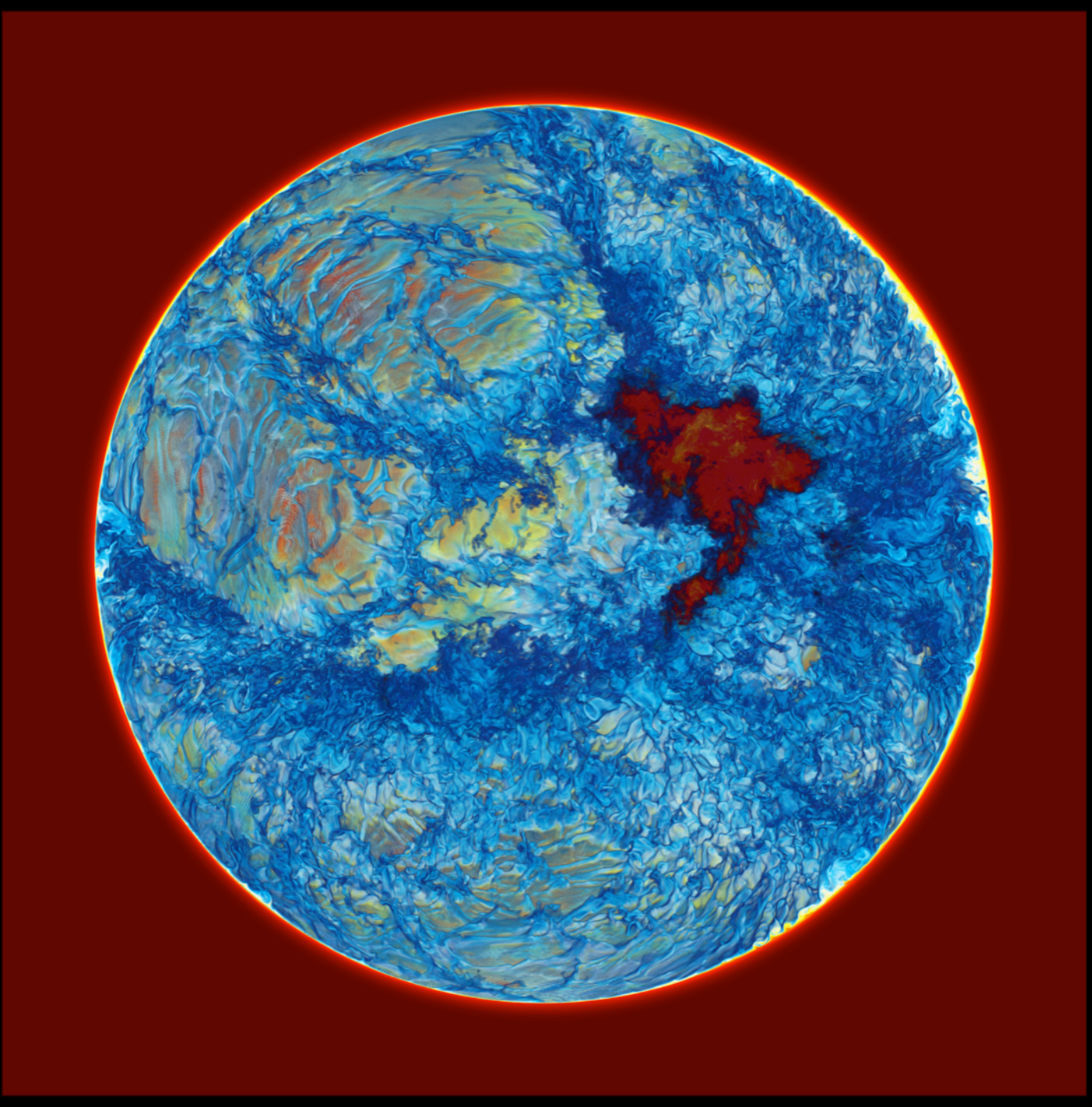
We have a modest grant on a world-class computing system. When we run, we use half the machine, but we don't run so much that anyone would feel squeezed out by us.

JOBID	USERID
564751	knox0043
565243	lawrenz
564004	lawrenz
565450	harshith
563257	bstein
564927	yanxinl
565368	redwards
565370	redwards
565456	jejoong
565163	jejoong

We can simulate the interior of a star on roughly half the Blue Waters machine, doing 9 million time steps in about 4 days running at about 0.4 Pflop/s



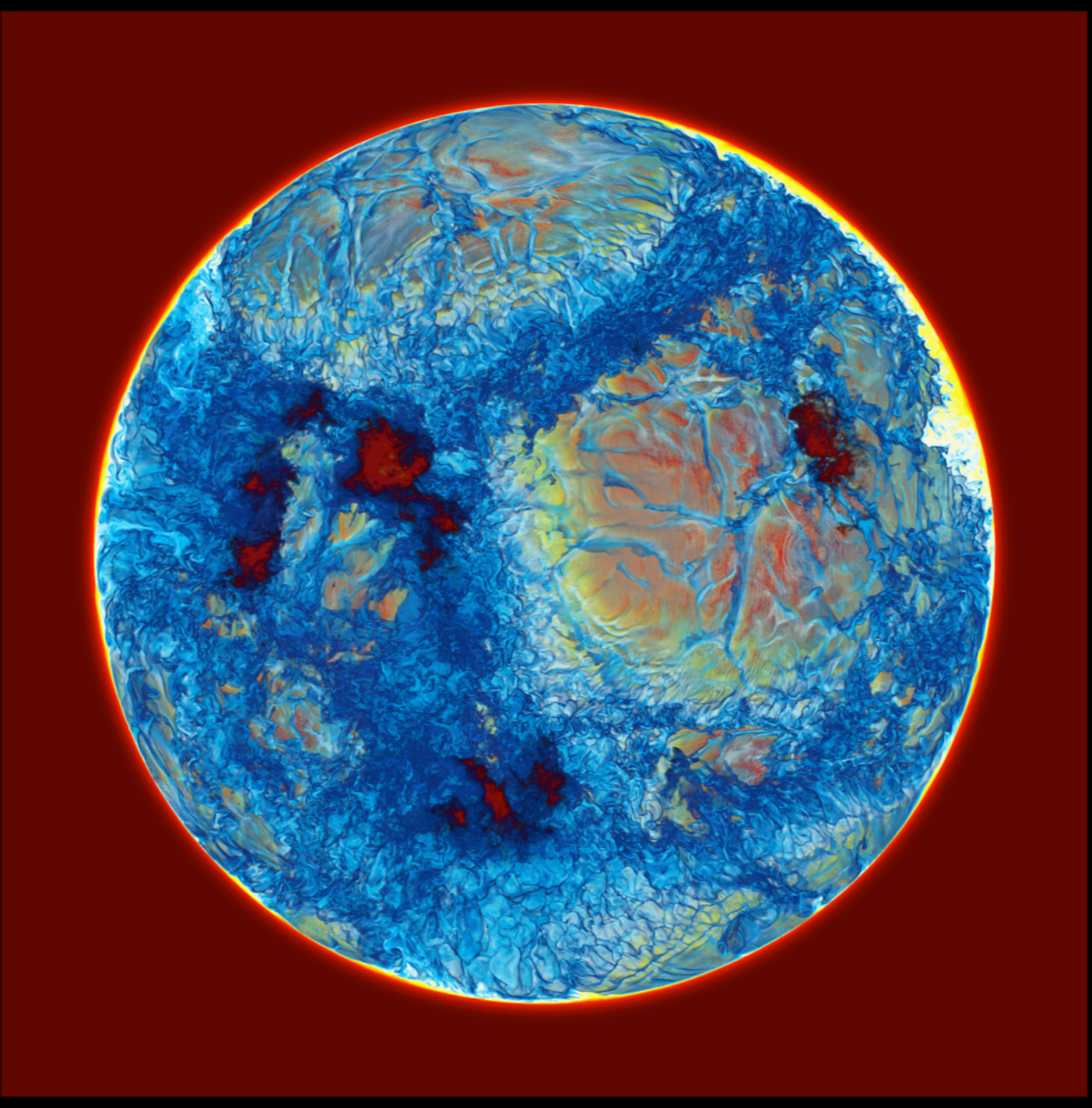
This computation would be easy on the Swiss Cray's interconnect.



2 M_{sun}, $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Aug., 2015, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

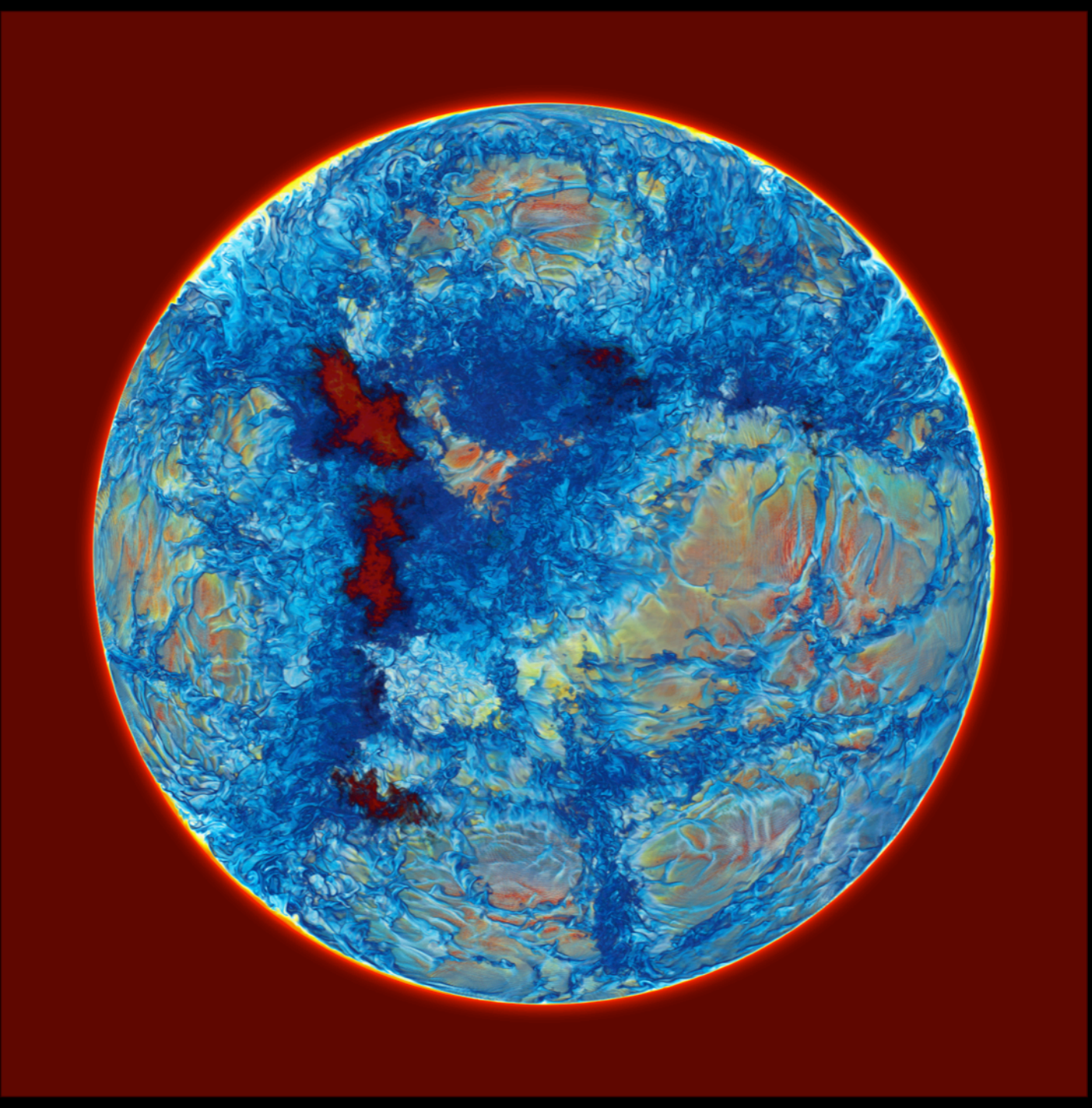
t = 887 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

Burning of ingested hydrogen is highly localized.

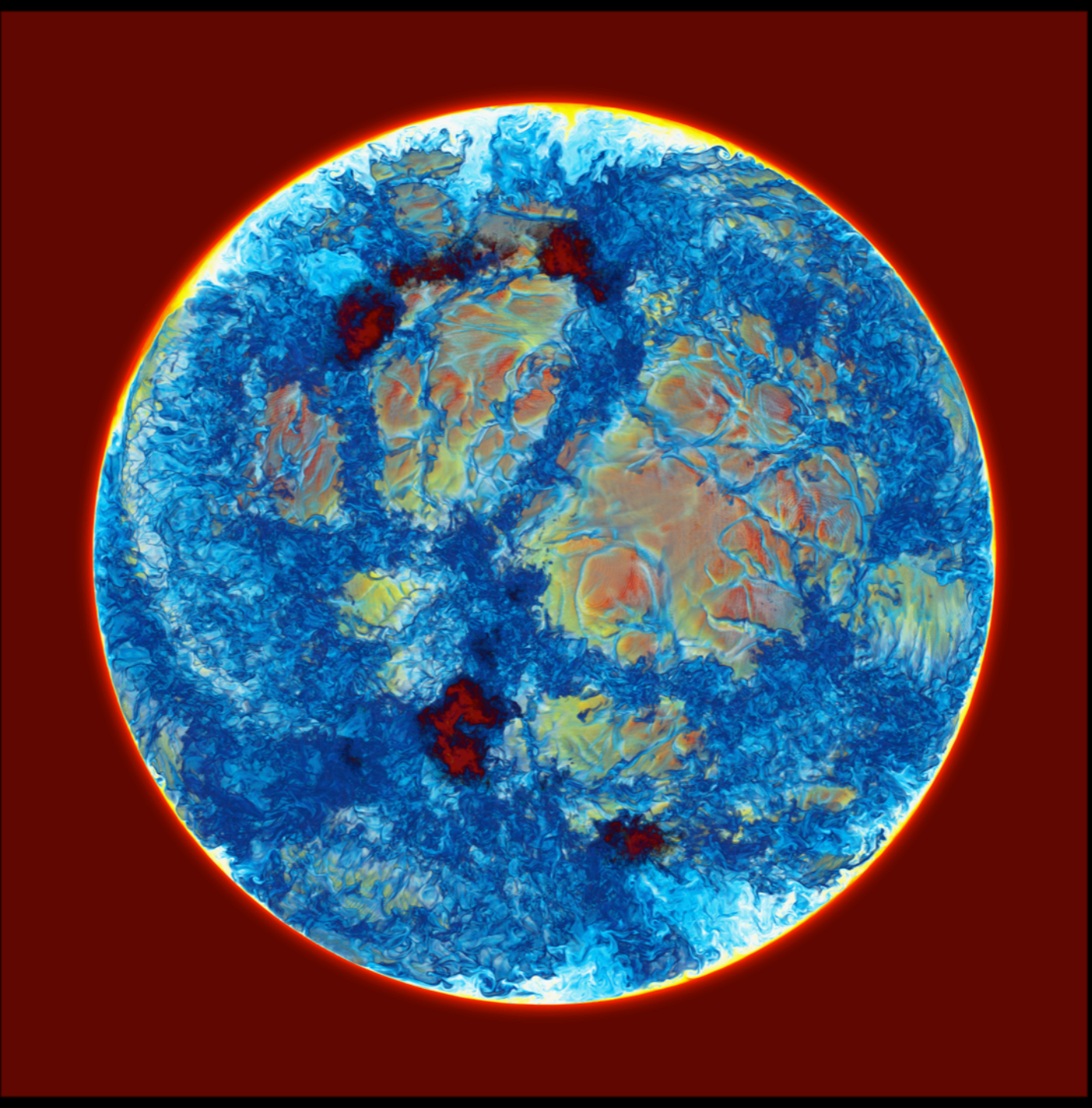
t = 1298 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

Burning of ingested hydrogen is highly localized.

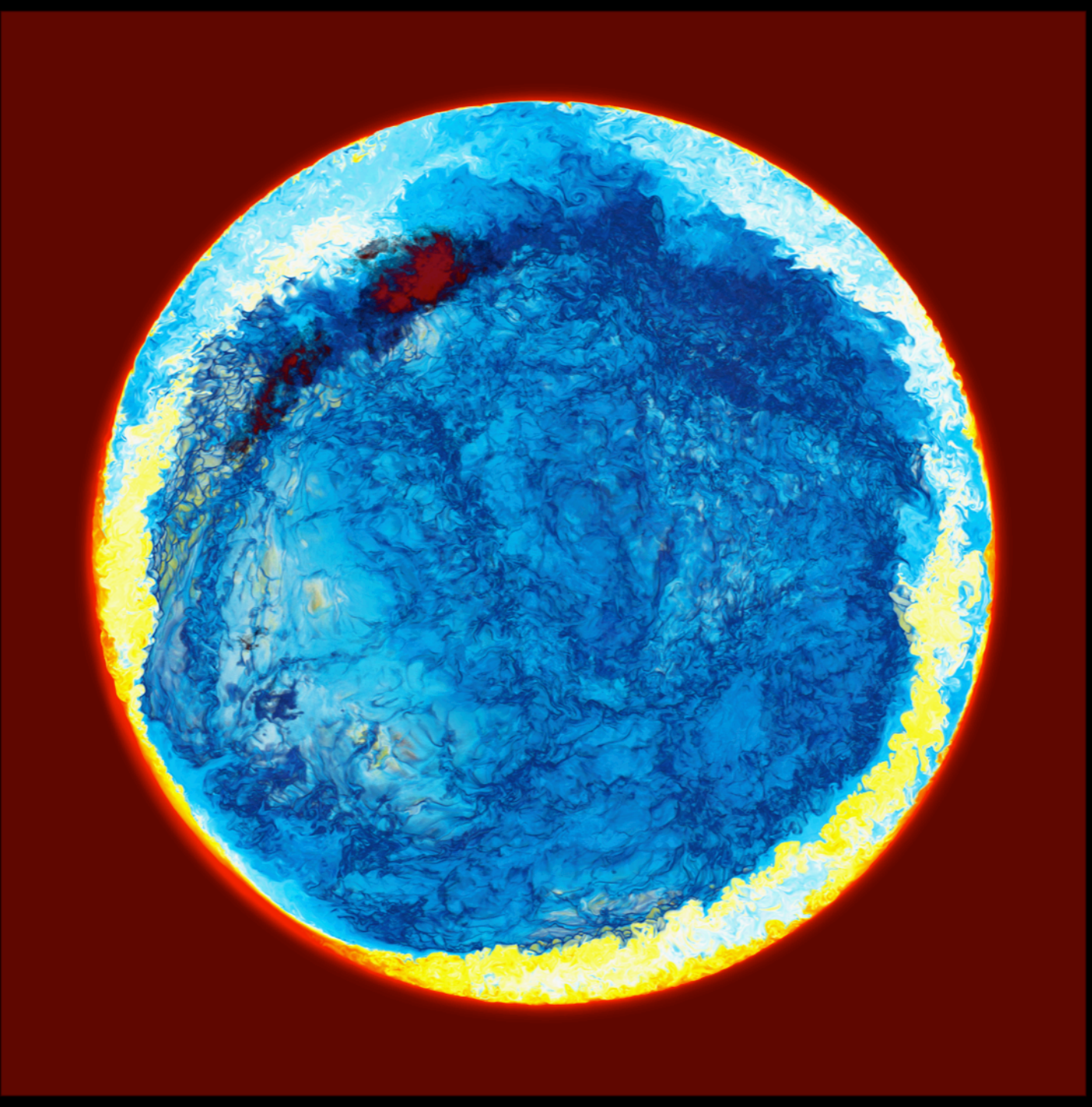
t = 1442.5 min.



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

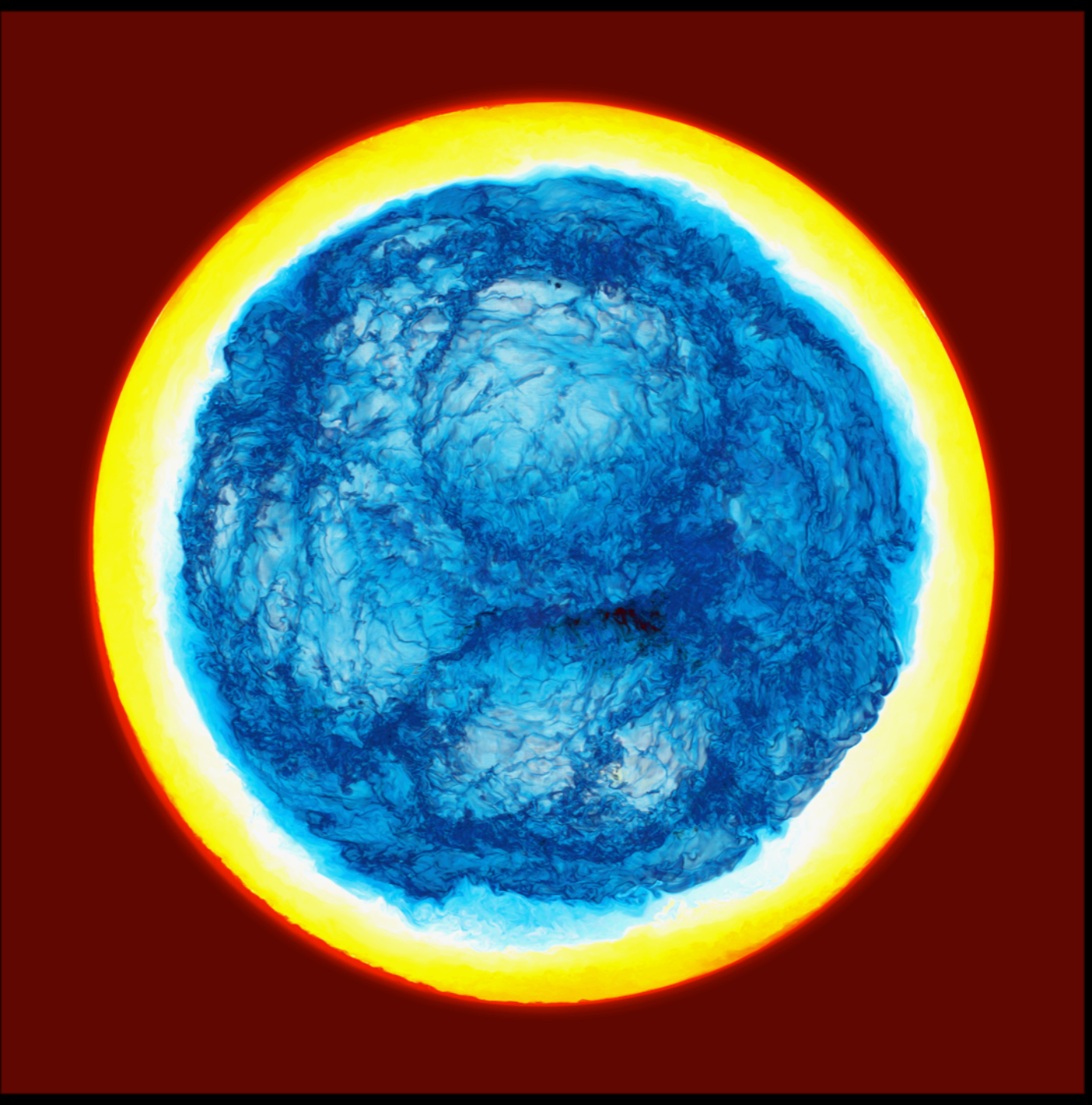
t = 1777 min.



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

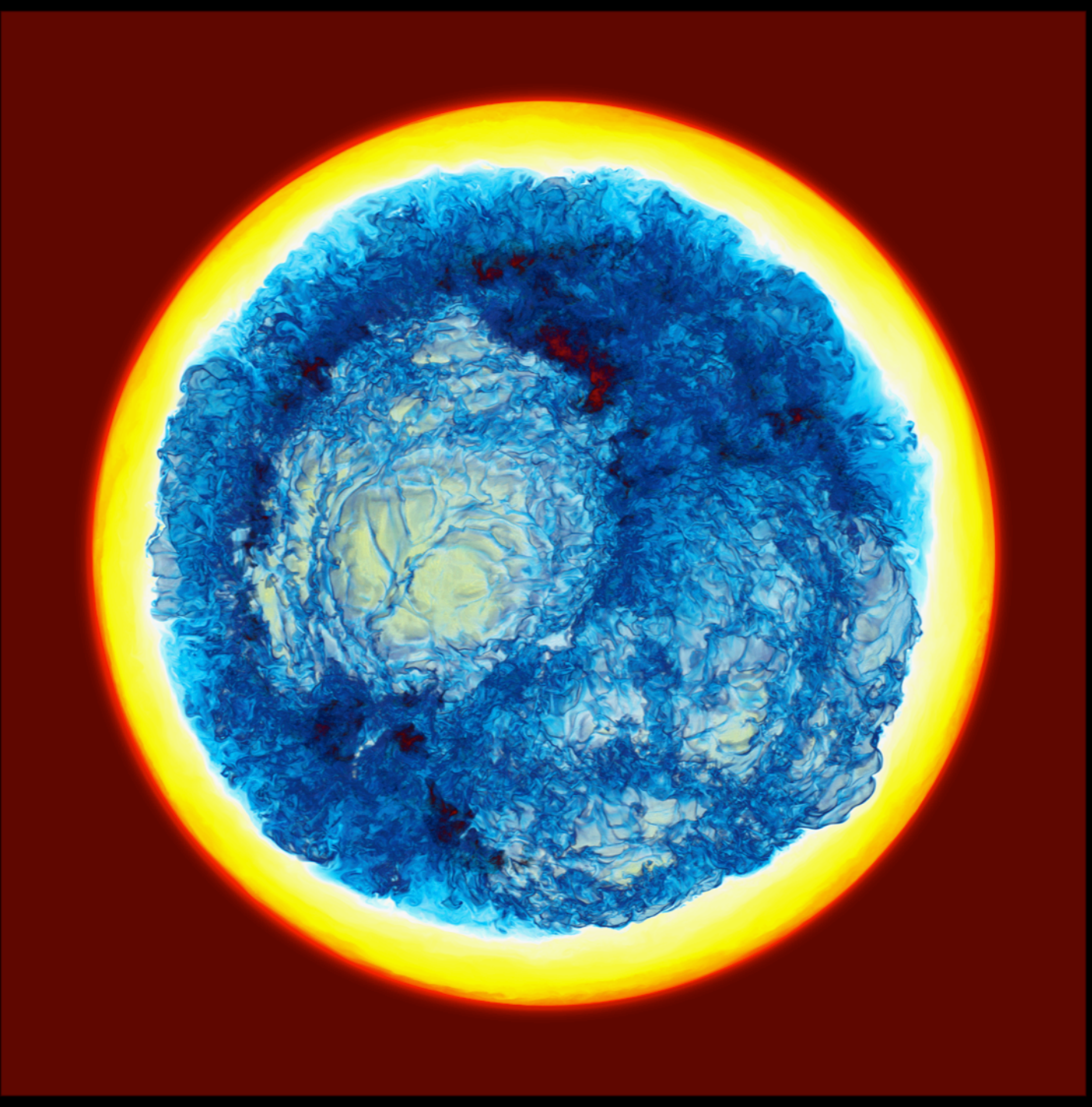
t = 2087.3 min.



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

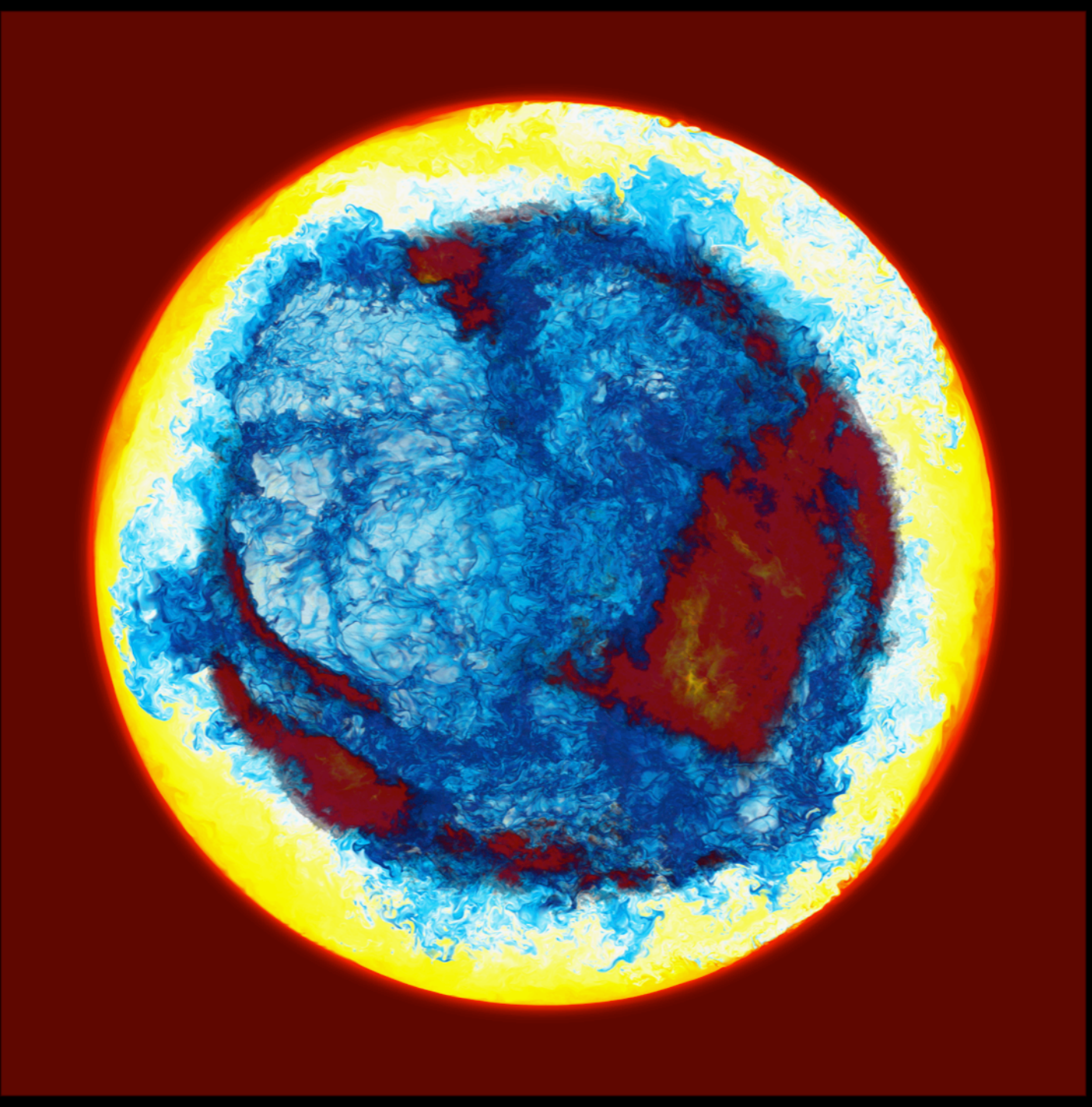
t = 2231.5 min.



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

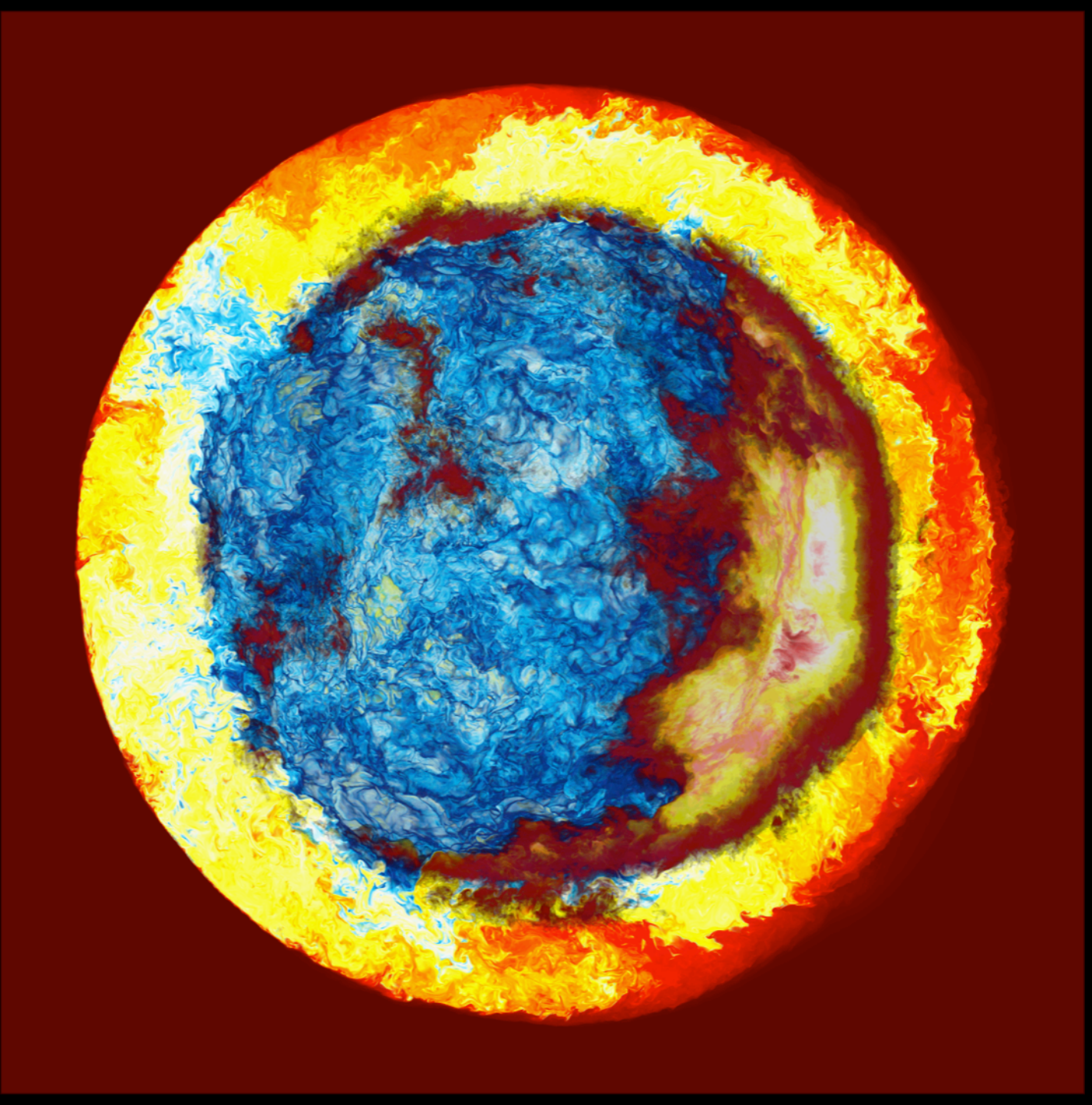
t = 2452 min.



2 M_{sun}, $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

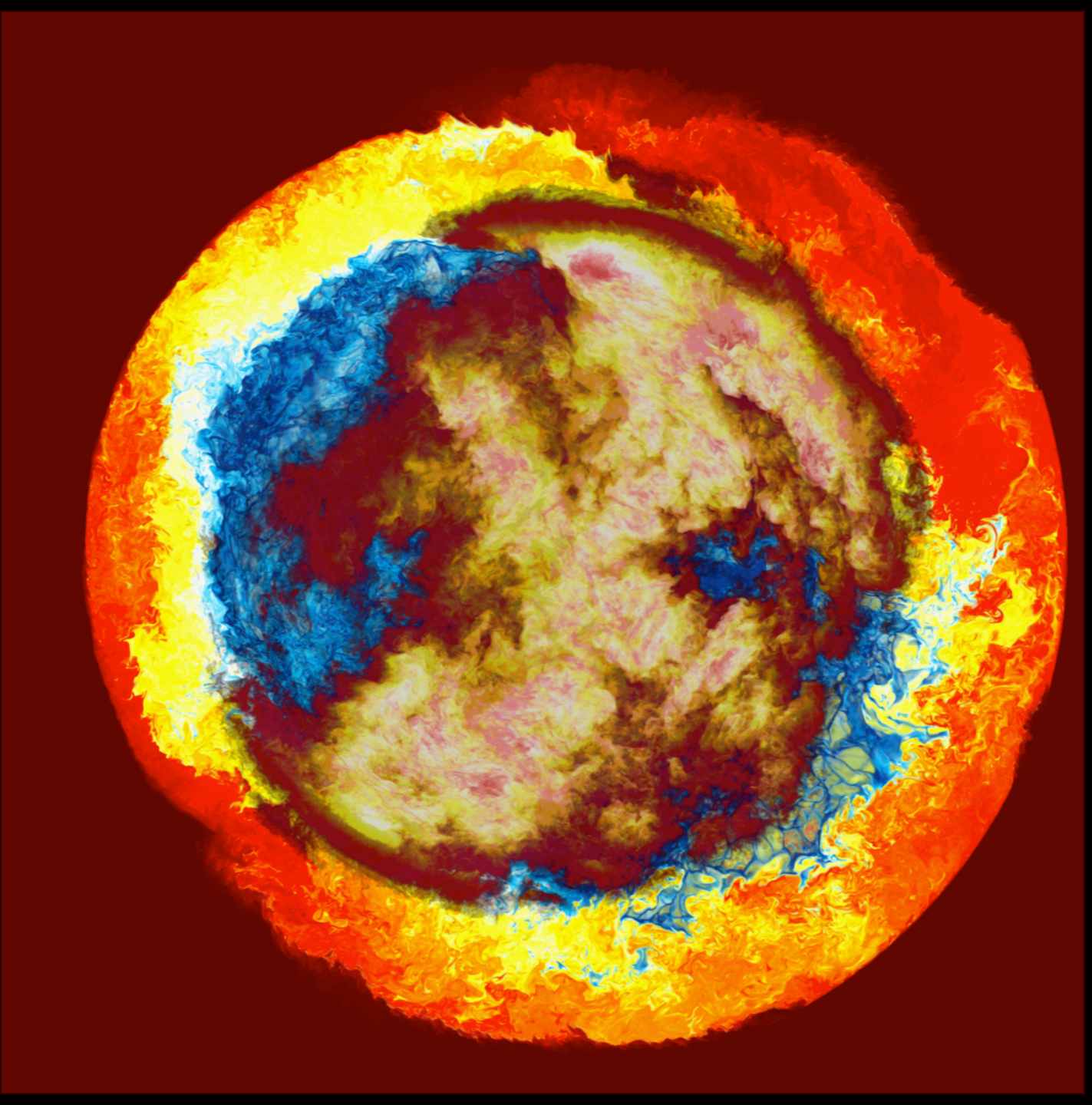
t = 2618.1 min.



2 M_{sun}, $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

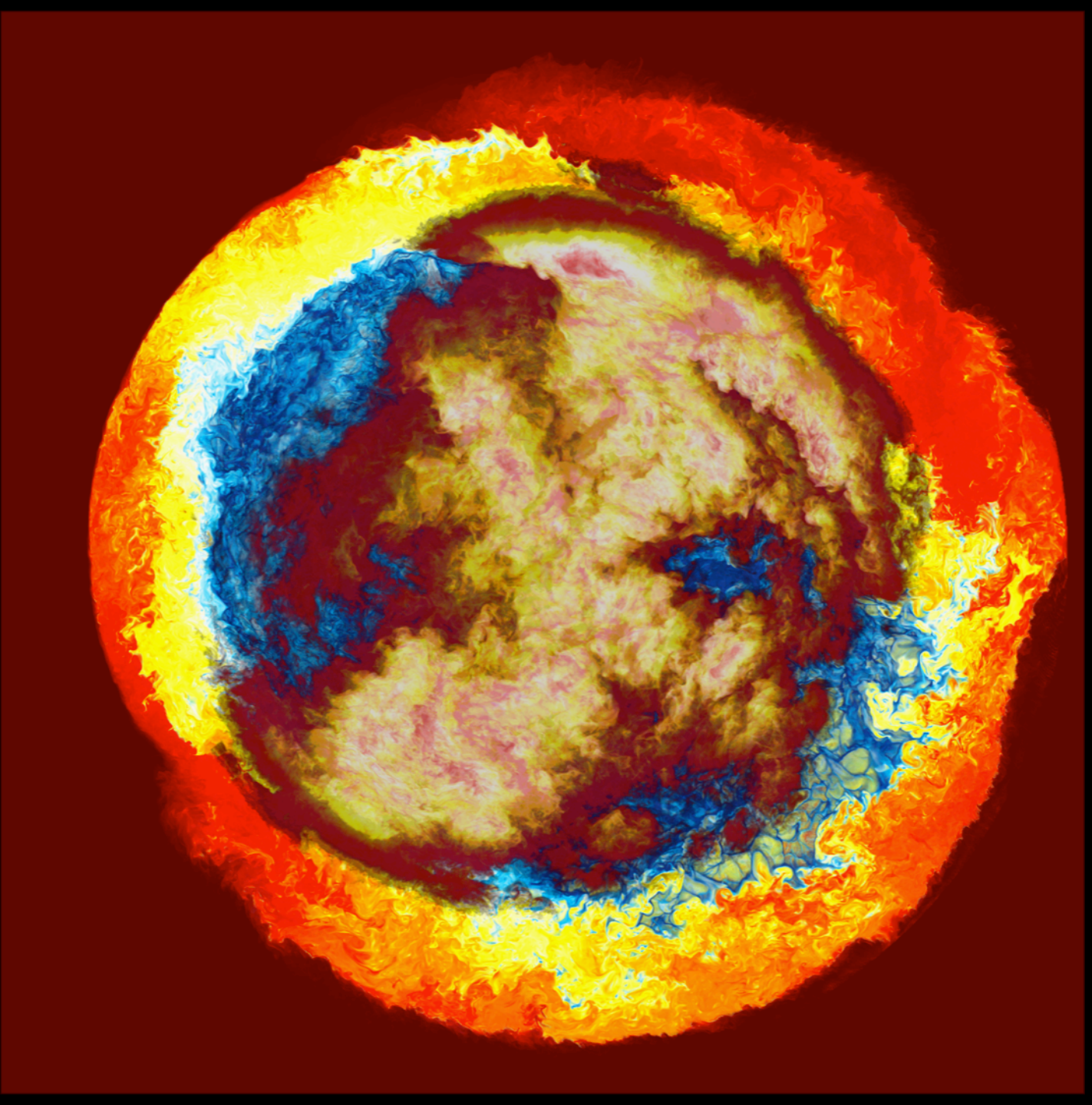
t = 2699.3 min.



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

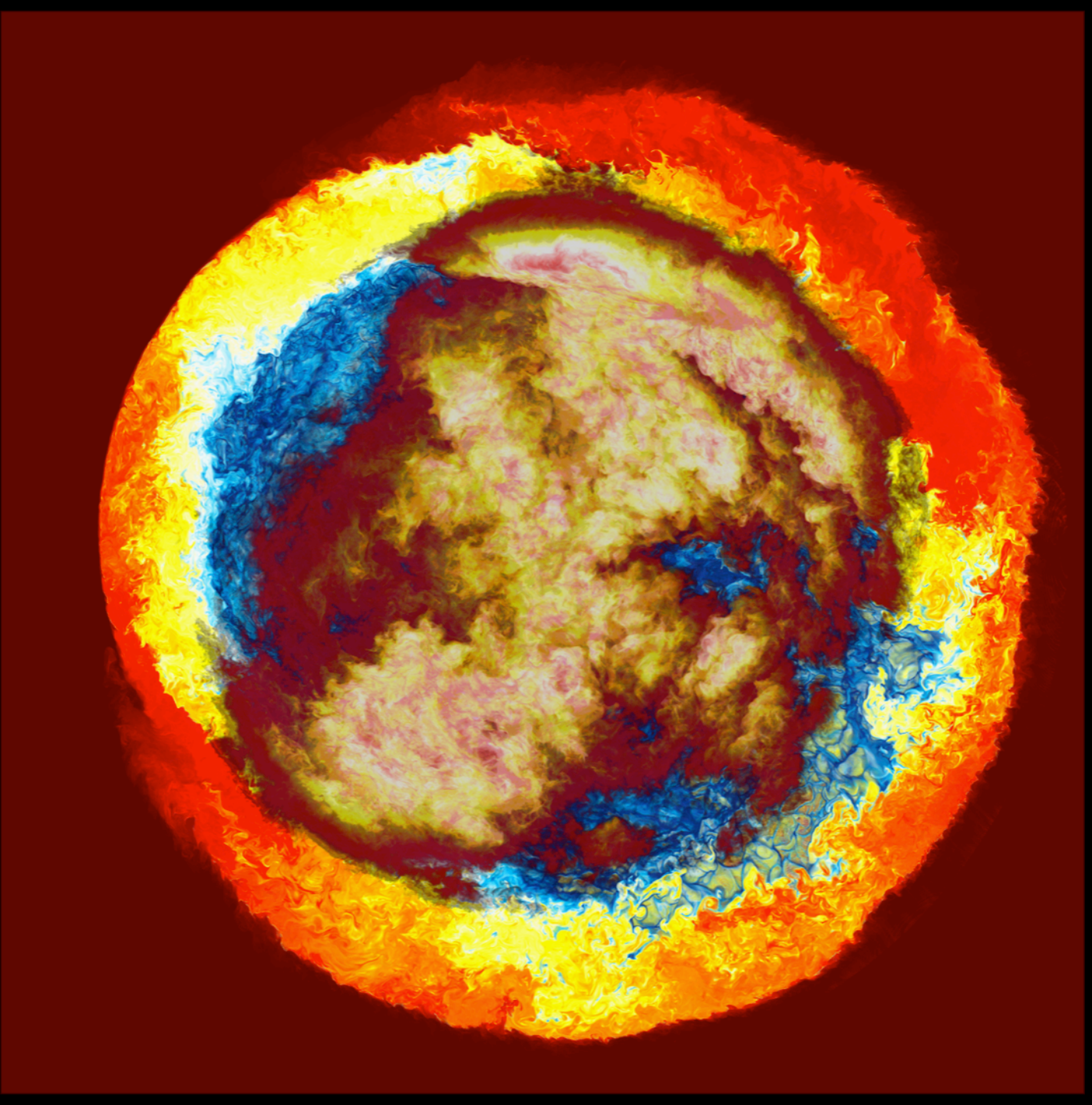
t = 2702.6 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

Burning of ingested hydrogen is highly localized.

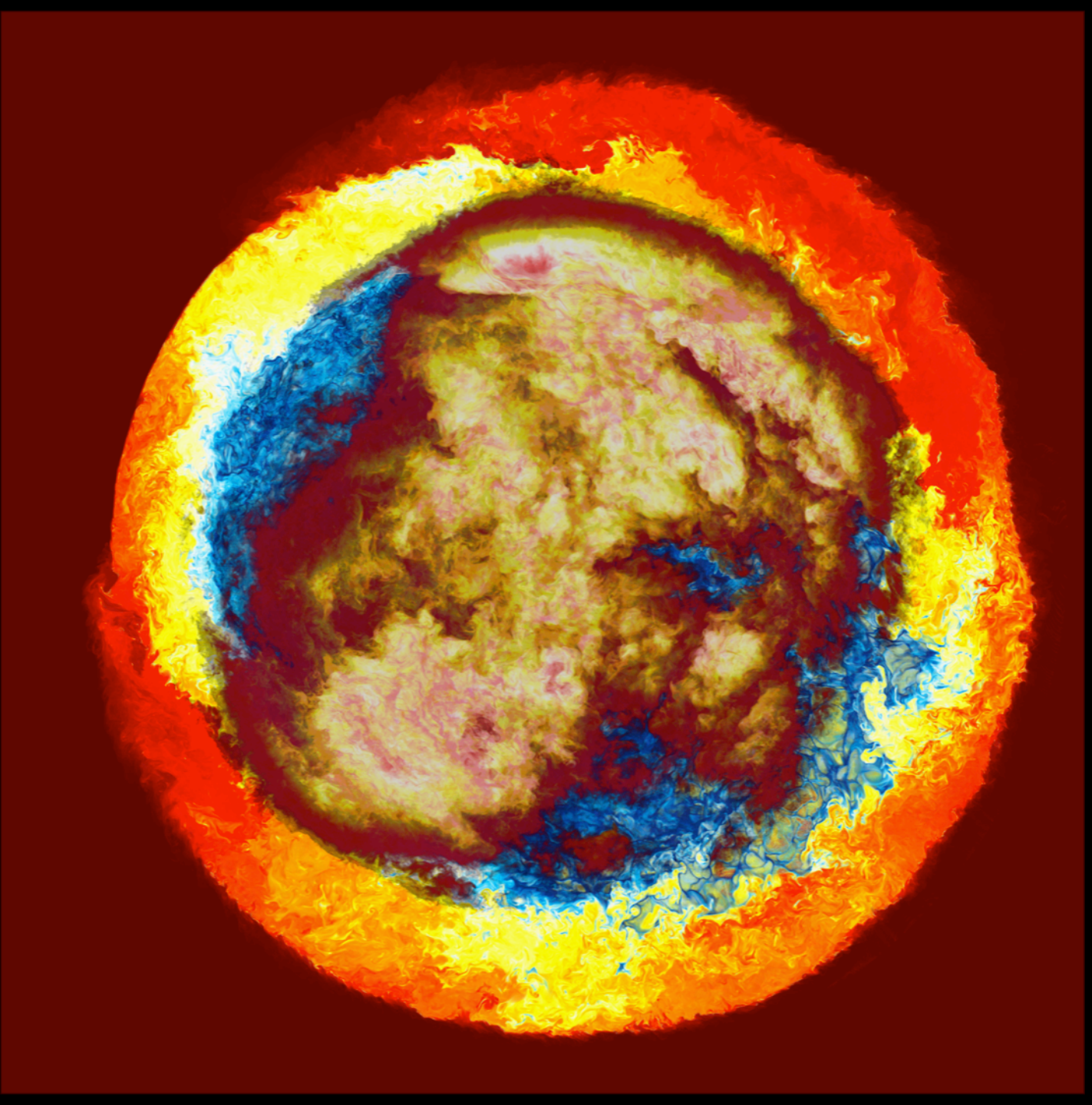
t = 2702.8 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

The very strong energy release at the bottom-right produces a violent updraft there, which sets off a Global Oscillation of Shell Hydrogen ingestion (GOSH), unmistakable in the next few images.

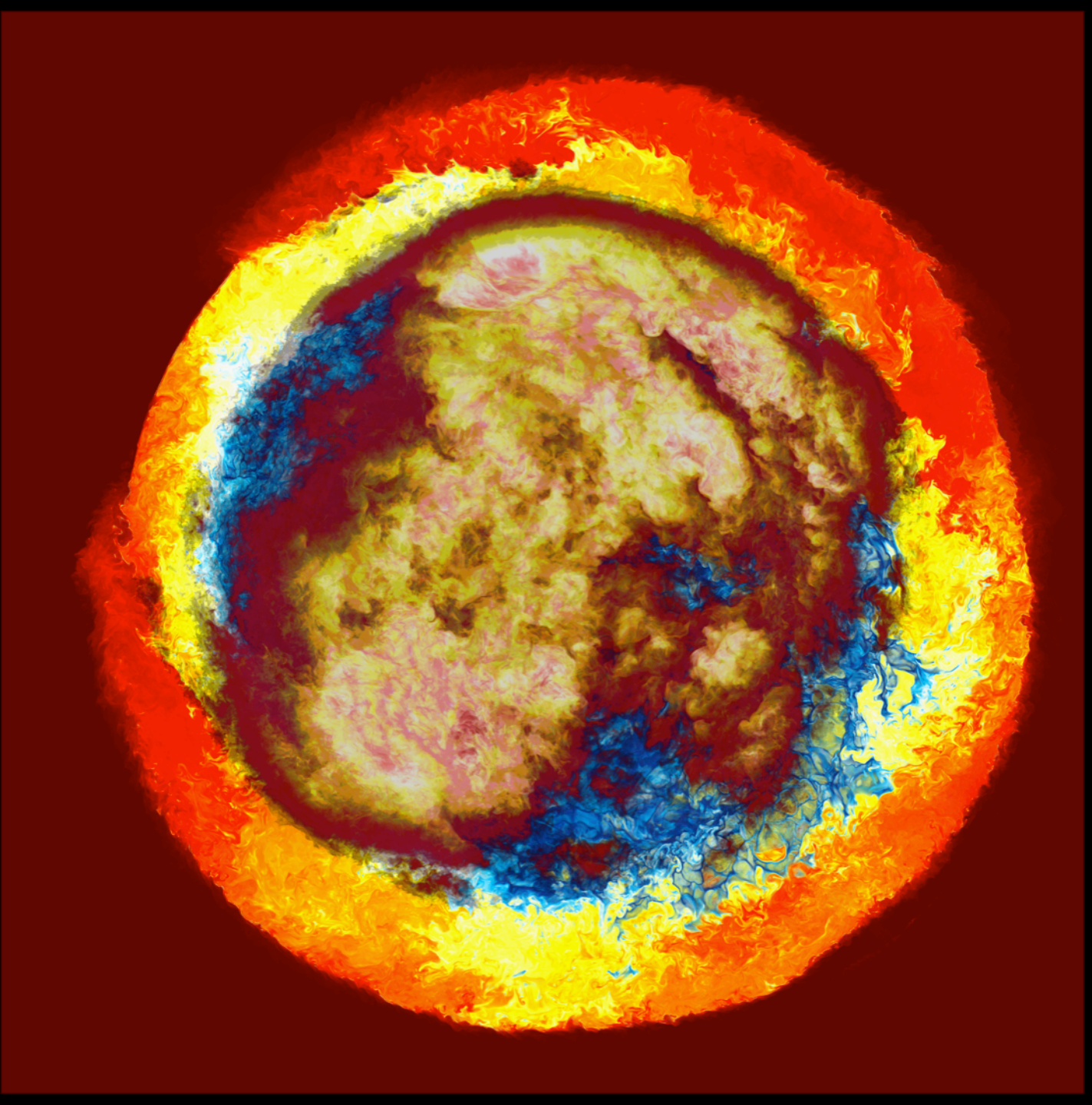
t = 2703.0 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion
simulation on Blue
Waters machine in
Jan., 2014, on a
grid of 1536^3 cells.

The regions of most powerful energy release are moving outward as a wave from the earlier site at the bottom right.

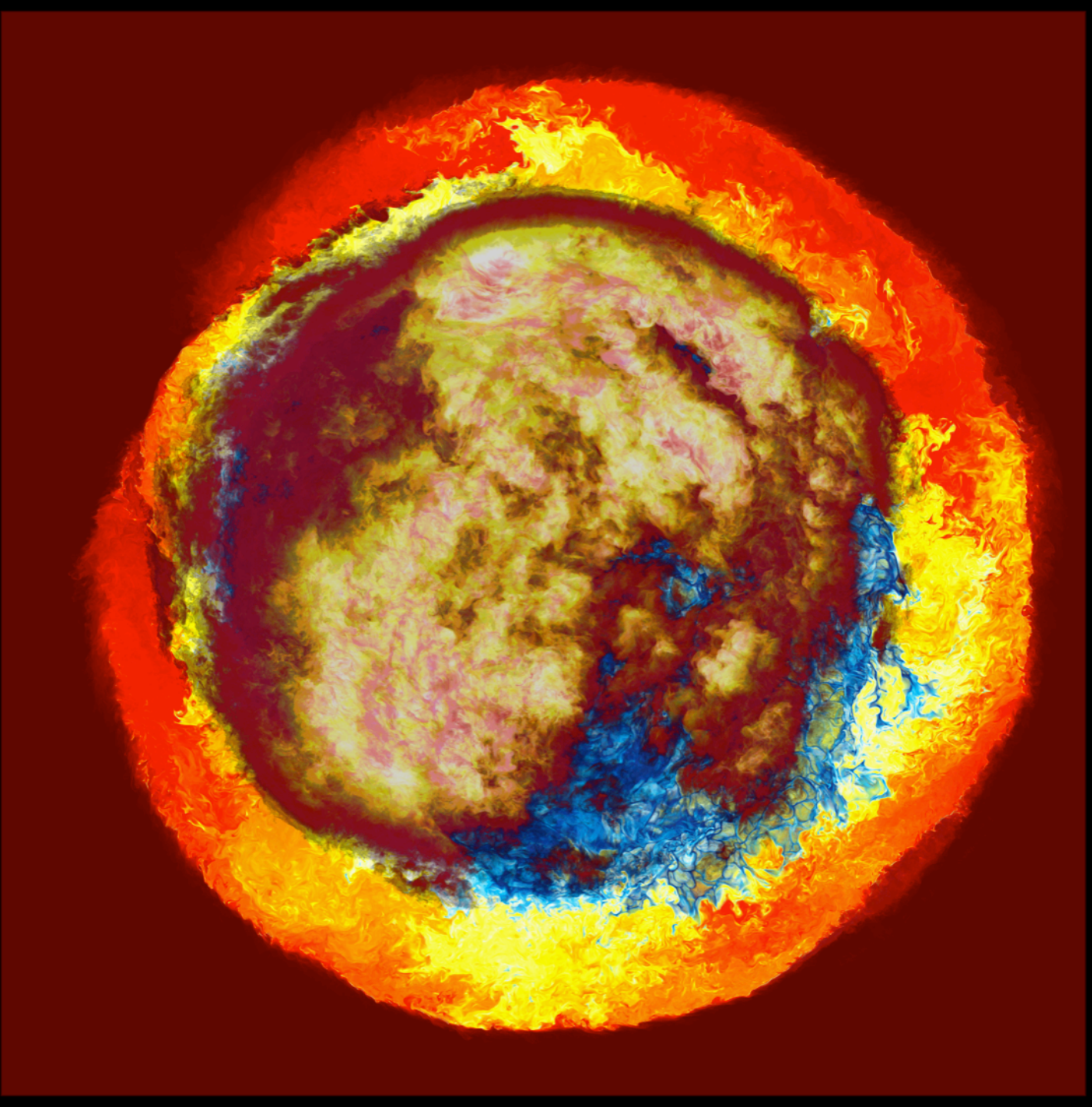
t = 2703.1 min.



2 M_{sun}, $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

As the front where hydrogen burns most rapidly advances, it drives ahead of it a ring of violent hydrogen ingestion, visible here in cross section. The averaged entrainment rate has jumped up from its early level by about 2 orders of magnitude.

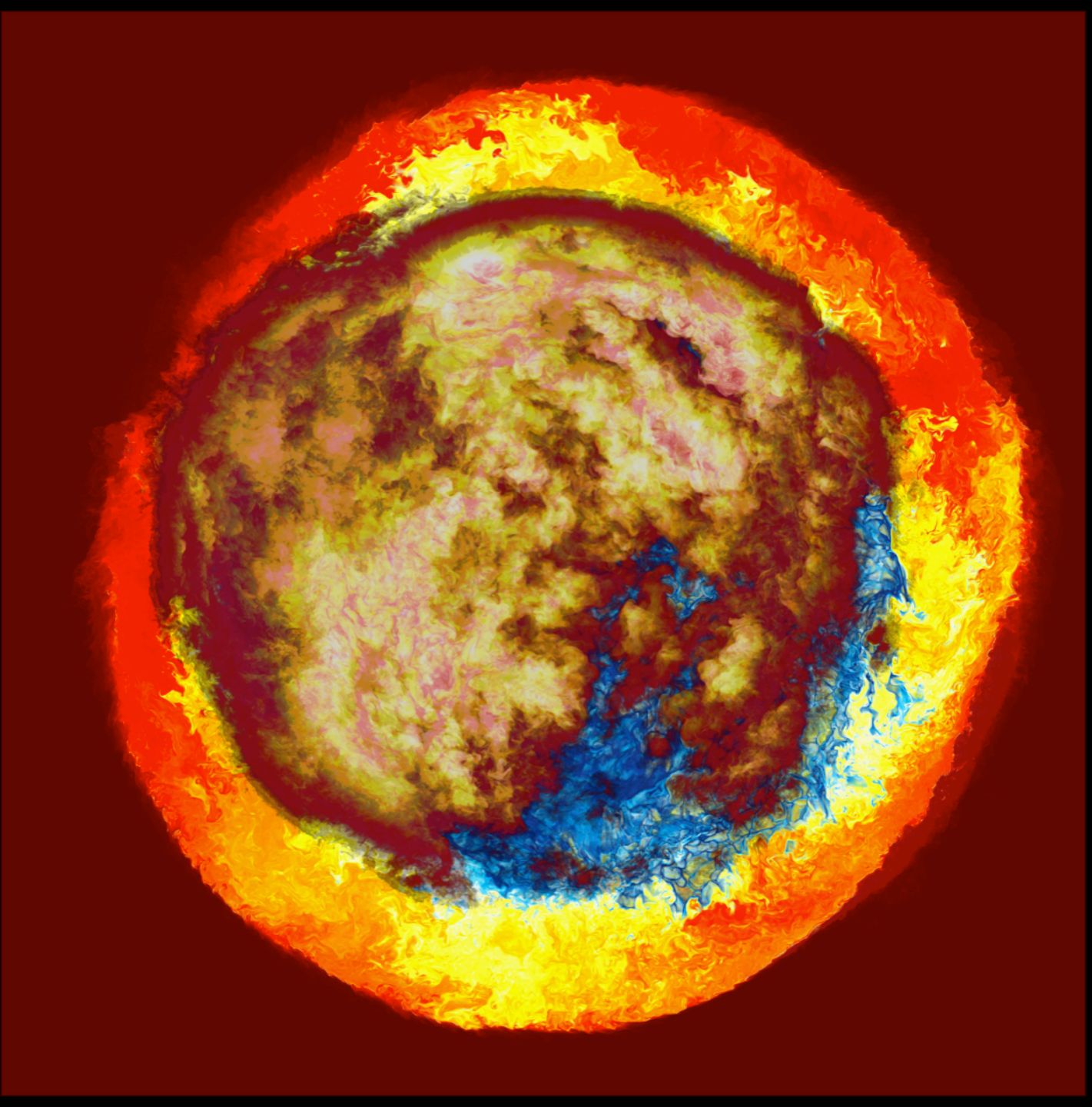
t = 2703.3 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

The burning front has now reached the antipode, where violent, localized energy release drives the oscillation back to its original site.

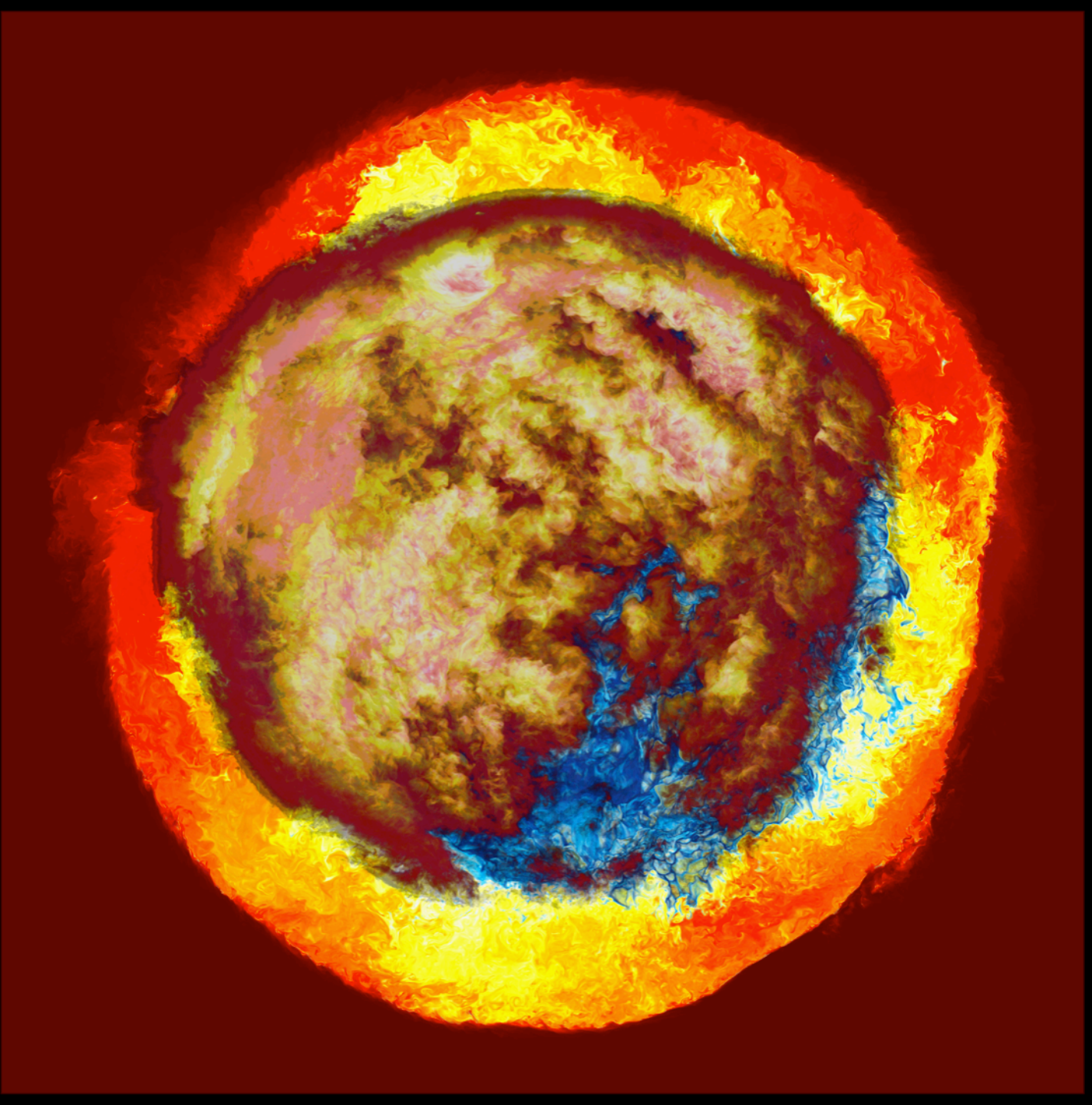
t = 2703.5 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

The GOSH is indeed global. This flow has a 1-D average, but it is by no means a 1-D phenomenon. Blue Waters makes it possible to see the GOSH in its full 3-D complexity.

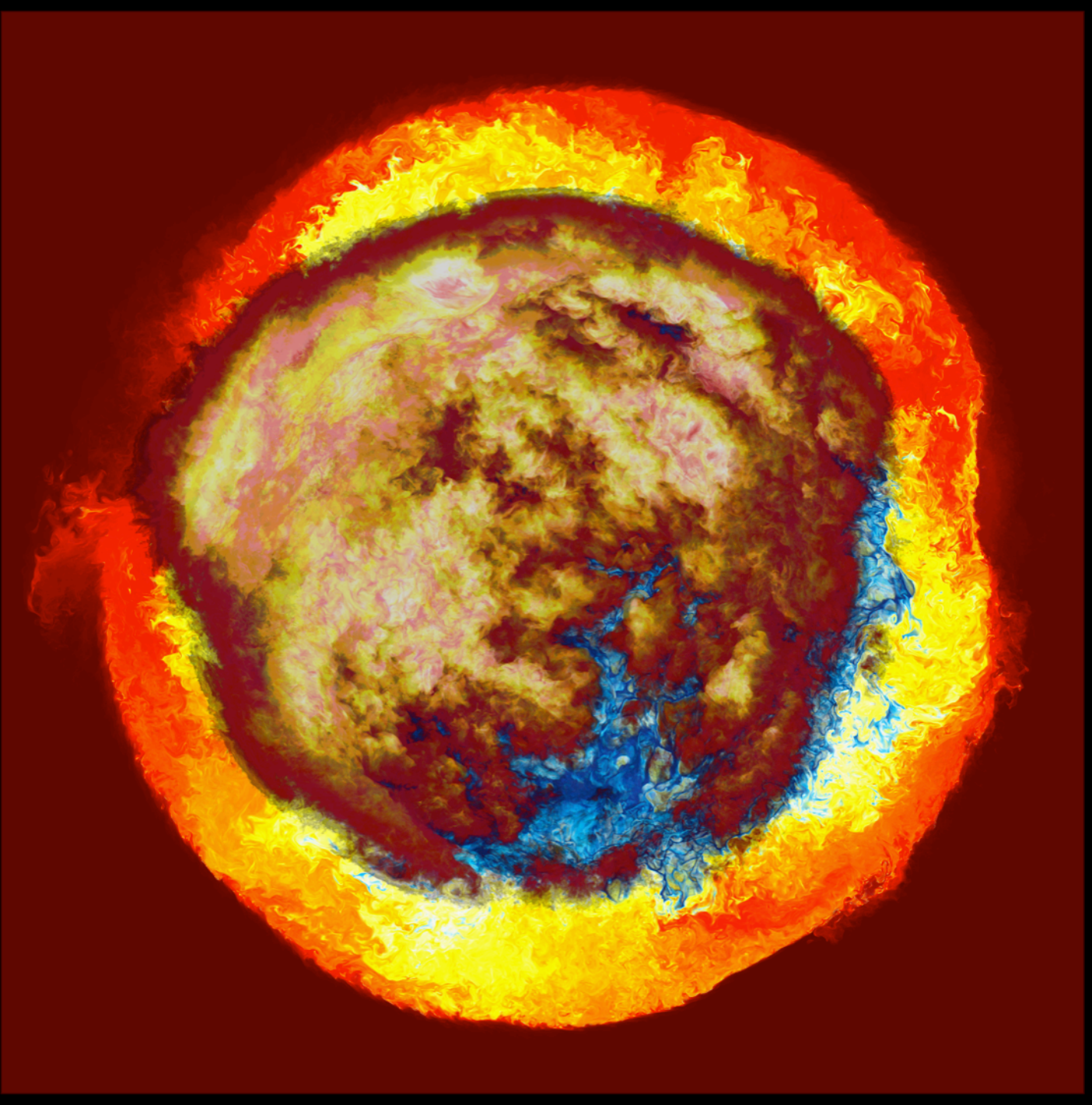
t = 2703.7 min.



2 M_{sun} , $Z = 10^{-5}$
AGB star
H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

Once the GOSH quiets down, after about a day in the life of this star, we can be well justified in carrying our description of the star forward with a 1-D stellar evolution code, suitably modified.

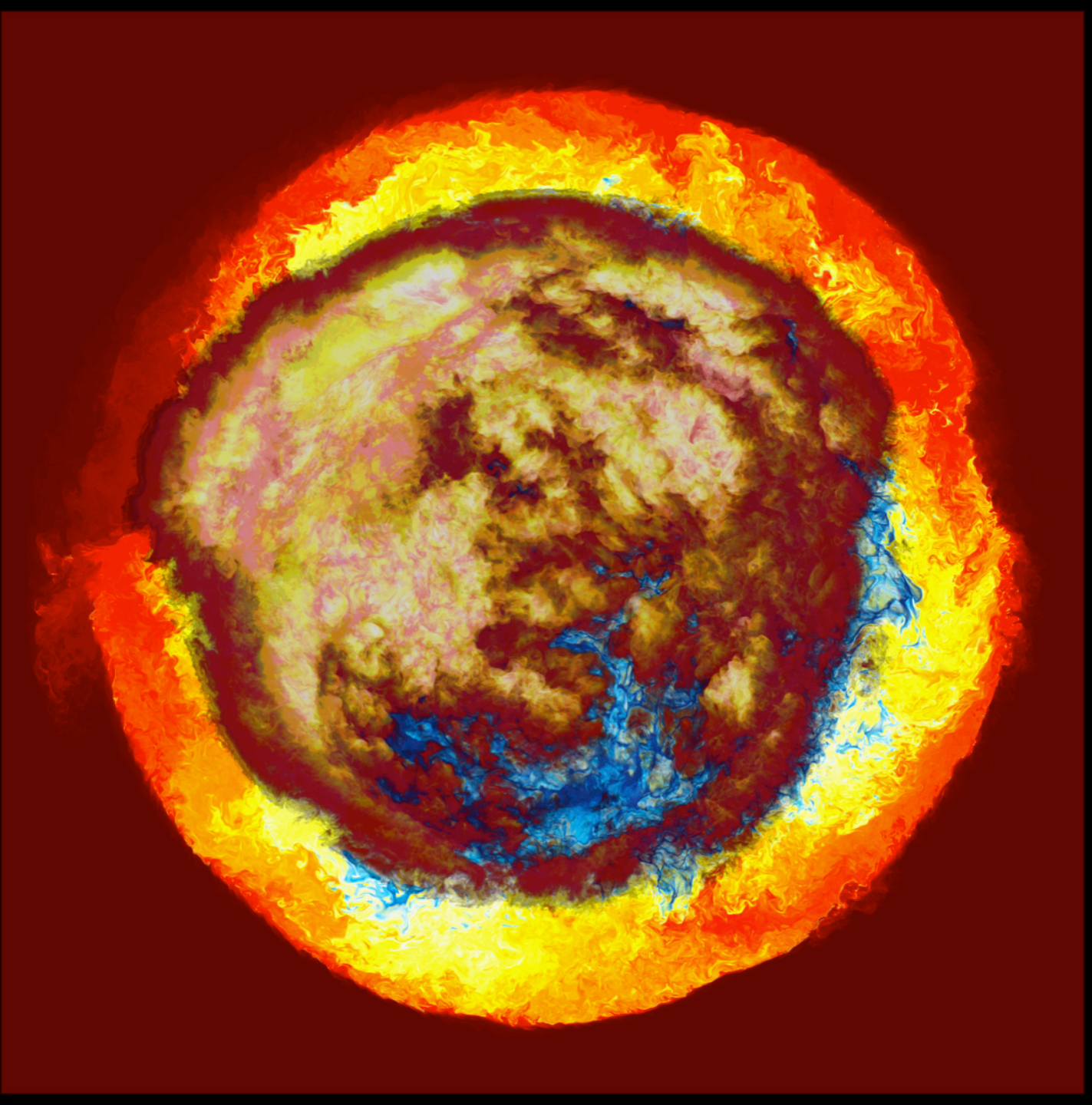
t = 2703.9 min/



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

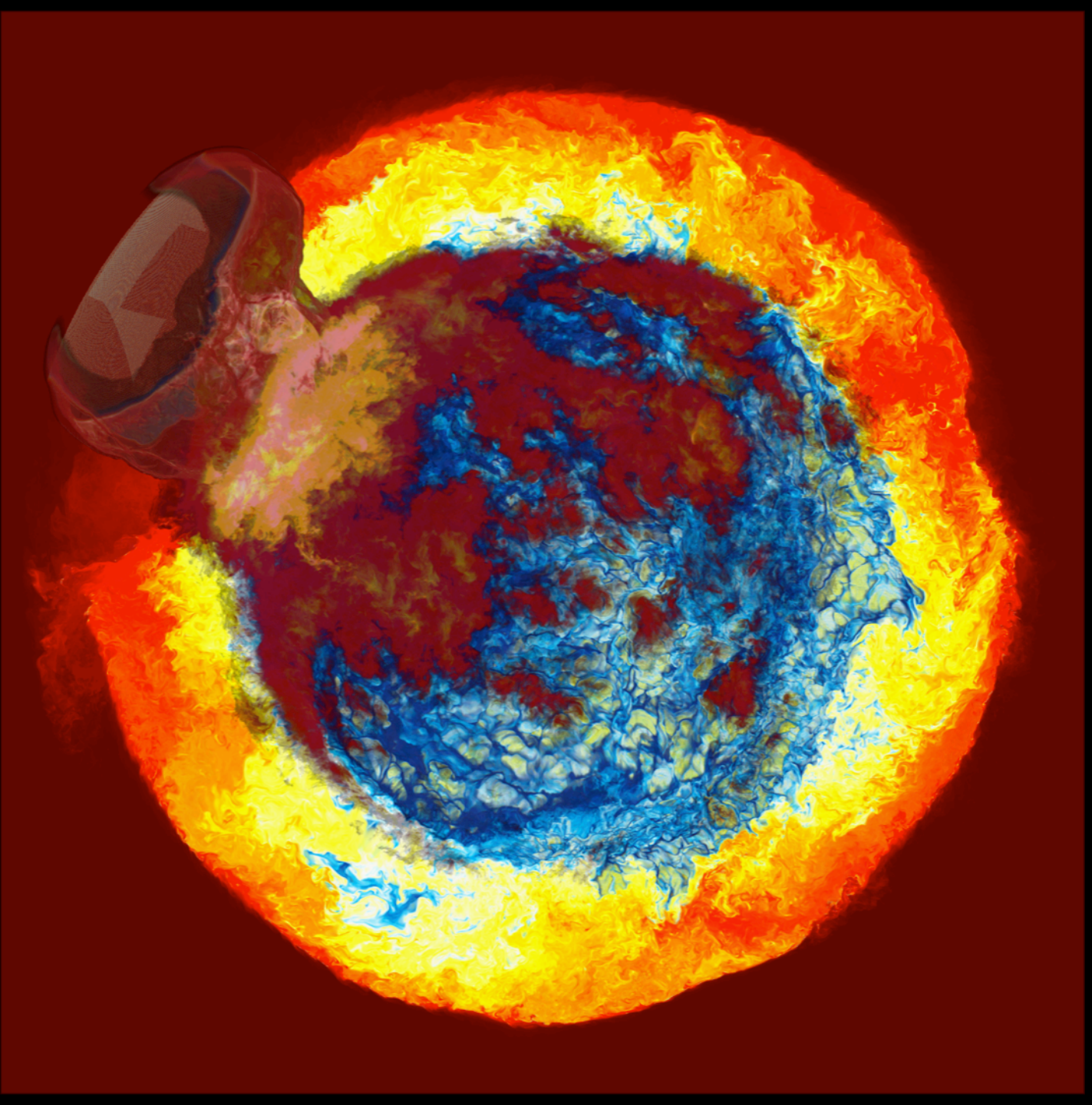
t = 2704.0 min.



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

t = 2704.2 min.



2 M_{sun} , $Z = 10^{-5}$ AGB star H-ingestion simulation on Blue Waters machine in Jan., 2014, on a grid of 1536^3 cells.

We see a hemisphere and make only mixtures of entrained hydrogen-rich gas with gas of the helium shell flash convection zone visible. The energy release rate from burning ingested H is shown in very dark blue, yellow, and white.

t = 2704.4 min.

The Global Oscillation of Shell H-ingestion (GOSH):

1. First produces a relatively stably stratified layer at the top of the convection zone.
2. Higher entropy of ingested H-rich gas shuts off ingestion, but only temporarily.
3. H still burns less violently at bottom of this upper layer.
4. Continues to stimulate global oscillatory burning behavior.

After helium burning for a few more hours adds enough entropy to the convection zone to match that of the H-enriched layer:

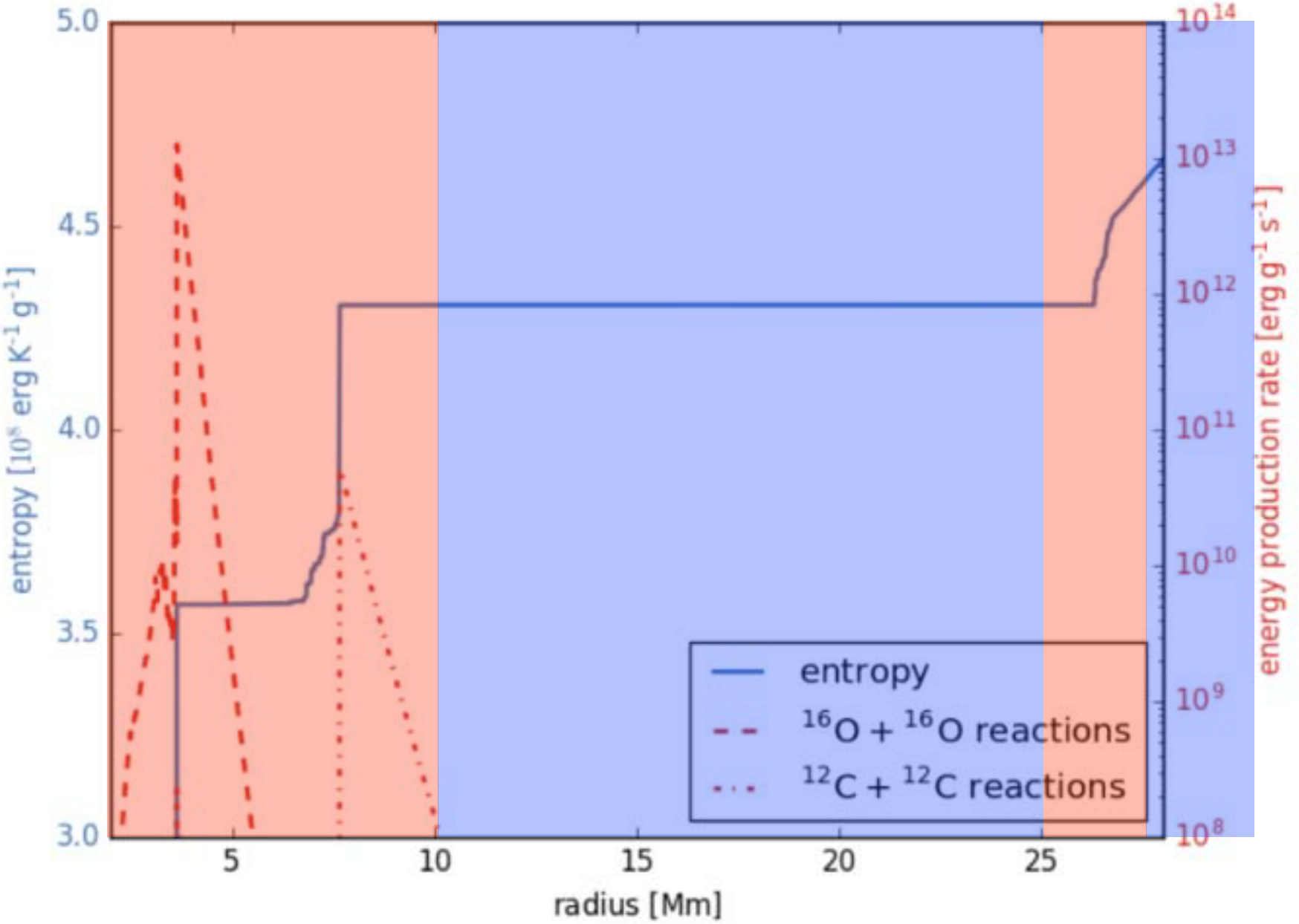
1. H-enriched gas is pulled down in large globs.
2. Subsequent violent burning drives second GOSH.
3. This eruption far more violent than the last.
4. Does not settle down with a new, H-burning convection zone, because is not 1-D. H-burning is local, and drives unstable behavior. To follow this correctly, we need to move our outer boundary much further outward with AMR

Applications:

1. i-process site. Do we need to pull the products of burning ingested H all the way down to the helium-burning region? Could have quite a different scenario than Herwig's 2011 analysis in 1D with its split convection zone suggests for Sakurai's object.
2. Effects on chemical evolution of galaxies through nucleo-synthesis post-processing in new way from 3-D simulation data for early-generation AGB stars.
3. Massive stars.
   a) H-ingestion events and i-process nucleosynthesis.
   b) Mergers of overlying nuclear burning shells.
4. Novae.
5. Falk keeps coming up with more and more situations in which we can perform 3-D simulations to attempt to resolve problems or doubts about 1-D models.
6. These drive a never-ending series of code enhancements.

Scaling this application on Blue Waters:

1. We have only 64 grid cells being update by 32 threads on 32 CPU cores of each node.
2. Restartable context per node is 40 MB. (out of 64 GB!!)
3. 26 time steps per second, continuously, without a pause.
4. 64 nodes is a "team" updating a region of the domain.
5. Each team has a dedicated I/O and global reduction process.
6. This impresses on the machine a hierarchical structure.
7. All I/O is completely asynchronous and causes no delay.
8. All global reductions to find time step values are one round out-of-date, so there is no delay.
9. All messages are sent while other parts of the grid bricks are updated, so no delay.
10. The nodes are simply worked to death, but no complaints.
11. When this code runs on Blue Waters, it consumes 14 Mwatt This is as much as LinPack!
12. But only 0.42 Pflop/s sustained, averaged over whole application.



The positions of the oxygen and carbon burning shells and the convection zones above them (regions of constant entropy) in a 1-D simulation of a 25 solar mass star shortly before silicon core burning. A 3-level AMR grid might use its fine grid level (red shaded radii) inside a radius of 10 Mm, its medium level (blue) to capture the convection zone above the carbon burning shell out to 30 Mm, with the fine level capturing stable gas entrainment near 26 Mm, and its coarse level to resolve the context from 30 Mm on out to 50 Mm. (MESA simulation provided by Falk Herwig and Christian Ritter.)

Example: **Merger of O- & C-burning shells**:

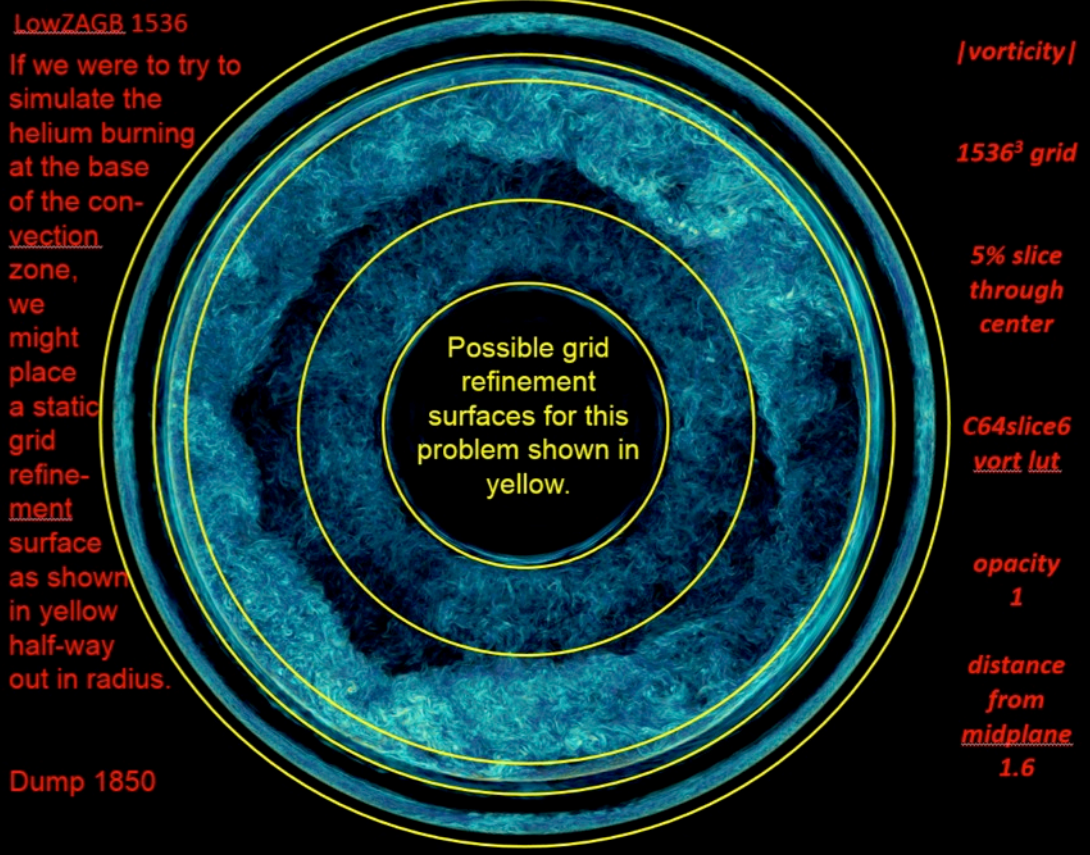
1. Must handle very different spatial and temporal scales.
2. AMR.
   a) Fine grid for O-burning shell and convection zone.
   b) Fine grid must continue through base of C-burning shell.
   c) Medium grid in C-burning convection zone & bit above.
   d) Coarse grid to allow expansion into outer part of star.
3. **3-level grid with roughly 21200, 91900, 886900 Mm^3** on the fine, medium, and coarse grids.
4. For a future machine:
   a) 8^3 regions of 16^3 bricks each, **each brick 96^3 fine cells**.
   b) Over 45,000 fine grid bricks, assigned 2 or 4 to a node.
   c) 2,097,152 bricks in all, plus 1,024 executive MPI ranks.
5. For a machine today:
   a) 8^3 regions of 8^3 bricks each, **each brick 192^3 fine cells**.
   b) Over 5,600 fine grid bricks, assigned 2 to a node.
   c) 262,144 bricks in all, plus 1,024 executive MPI ranks.

Example: **Merger of O- & C-burning shells**:

1. Must do at least twice as many time steps on the fine grid as we did for the H-ingestion simulation shown earlier.
2. Update 4×96^3 cells per time step rather than 64^3 cells.
   a) **9.7 × more work/node/Δt on about same # of nodes**.
   b) Fine grid must continue through base of C-burning shell.
   c) Medium grid in C-burning convection zone & bit above.
   d) Coarse grid to allow expansion into outer part of star.
3. 3-level grid with roughly 21200, 91900, 886900 Mm^3 on the fine, medium, & coarse grids**. > 60% of work is on fine grid**.
4. For a future machine:
   a) 8^3 regions of 16^3 bricks each, each brick 96^3 fine cells.
   b) Over 45,000 fine grid bricks, assigned 2 or 4 to a node.
   c) 2,097,152 bricks in all, plus 1,024 executive MPI ranks.
5. For a machine today (still need that 9.7× speed-up):
   a) 8^3 regions of 8^3 bricks each, each brick 192^3 fine cells.
   b) Over 5,600 fine grid bricks, assigned 2 to a node.
   c) 262,144 bricks in all, plus 1,024 executive MPI ranks.

Example: **Merger of O- & C-burning shells**:

1. **We need a factor of 10 in performance over Blue Waters**.
2. We have the following factors of:
   a) **2 ×** Remove base state and go to 32-bit precision.
   b) **3.5 ×** Go from Interlagos AMD to Haswell Intel CPU
   c) **2 ×** Go from CPU to Xeon Phi, K80 or other GPU.
   d) **1.7 ×** Generate extra ILP by unrolling entire code.
3. If we stay on Blue Waters, without upgrade, then only (a) and (d) are available.
4. **We cannot just wait for machine to get faster by itself**.
   a) We must revise the code substantially to make it equally accurate with **32-bit precision** – a student this summer.
   b) We must restructure the computational section extensively to **enable use of GPUs** – essentially done.
   c) **Unrolling the entire computationally intensive section is a major code transformation**.
   It can be automated, but it is not automatic yet.
   Pei-Hung Lin, at LLNL, says he will do it. **2×2×1.7 = 6.8**

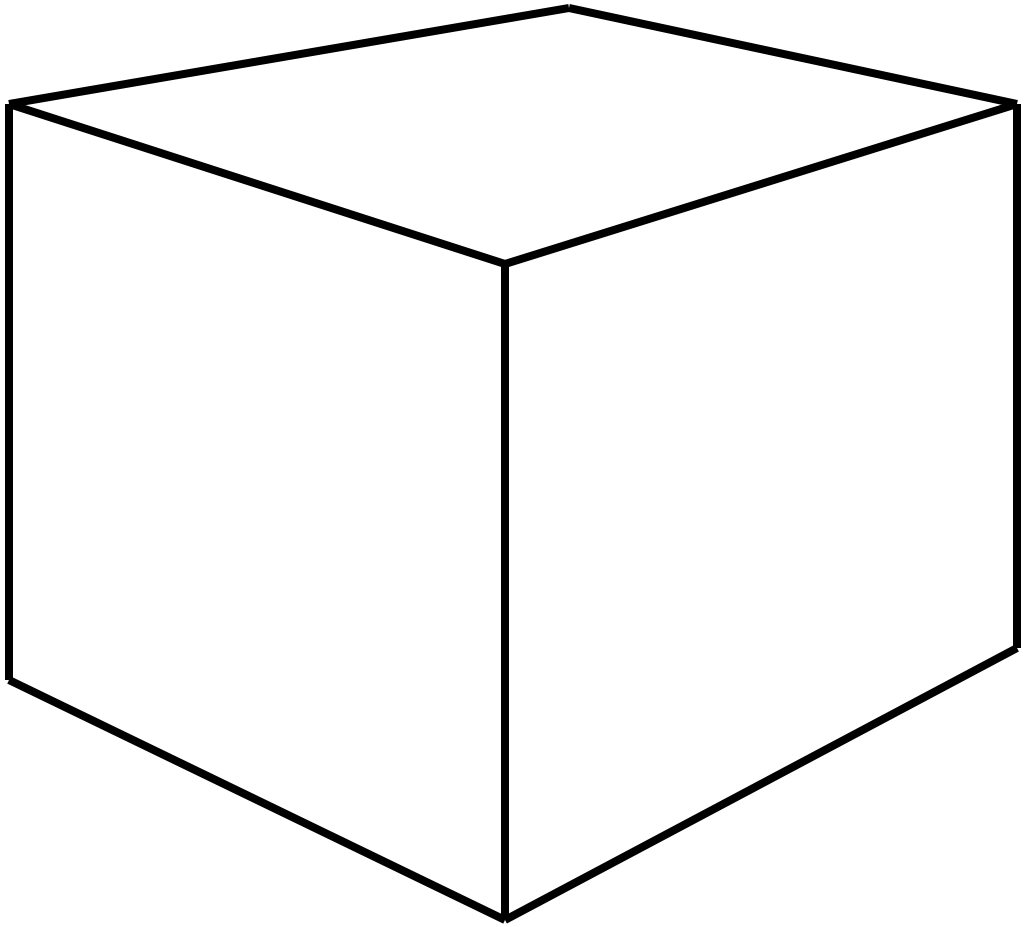


This is a slice through a grid of only 1536^3 cells that was decomposed into 110,592 bricks of 32^3 cells each and run on 13,952 nodes of Blue Waters.
26 Δt/sec @ 0.42 Pflop/s.
Data dump of 3072 files / 3 min.
This is 47 GB every 3 minutes continuously for 4 days.
Restart data = 40 MB/node.

Speed comes from:
14000 nodes
30 Gflop/s (64-bit) / node

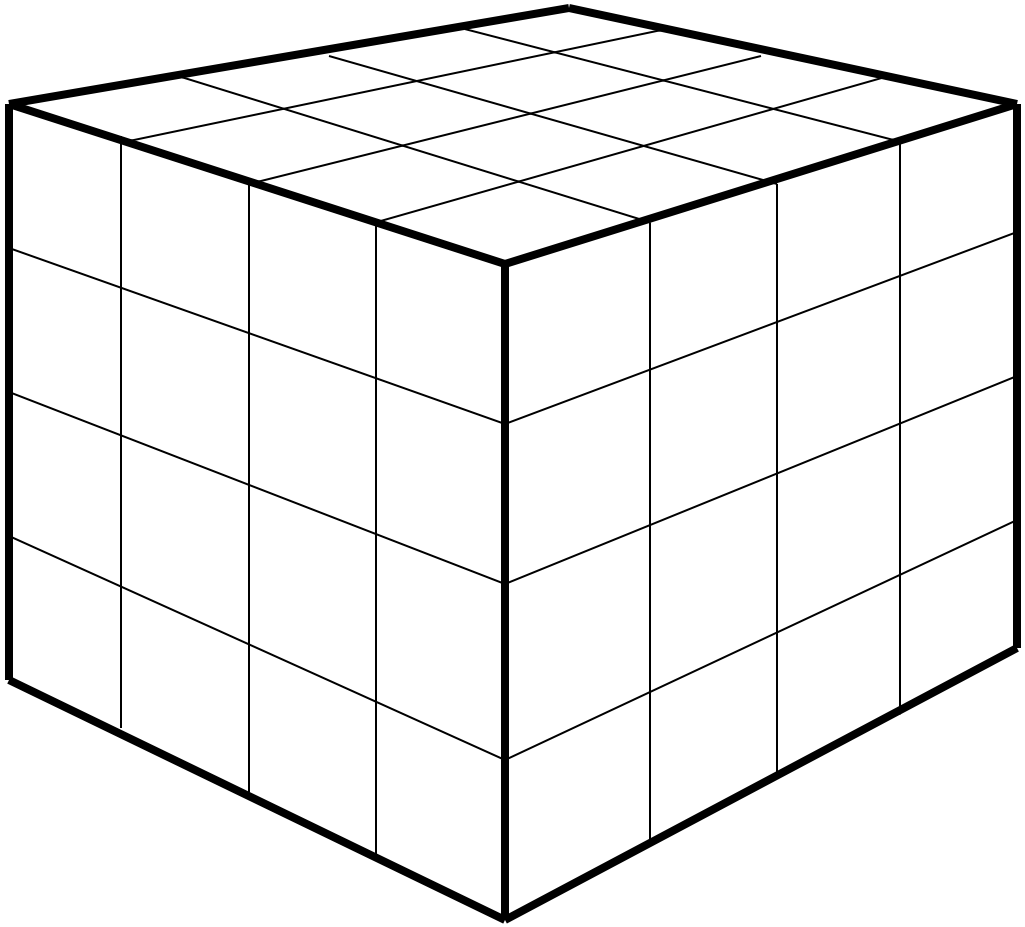
We need a factor of 9.7 in performance to run the O- and C-burning shell merger problem at roughly the same cost in computing time.
(64-bit ---> 32-bit) + (AMD Interlagos ---> Intel Haswell) = factor 7.0
Get the rest from increasing ILP by automating massive outer loop unrolling.

Let’s start small and work outward. This is a grid cell.



(Let’s see how our code gets its speed.)

We will subdivide it evenly into a grid briquette.



We force a minimal degree of uniformity at the microscale to accommodate needs of SIMD engine.

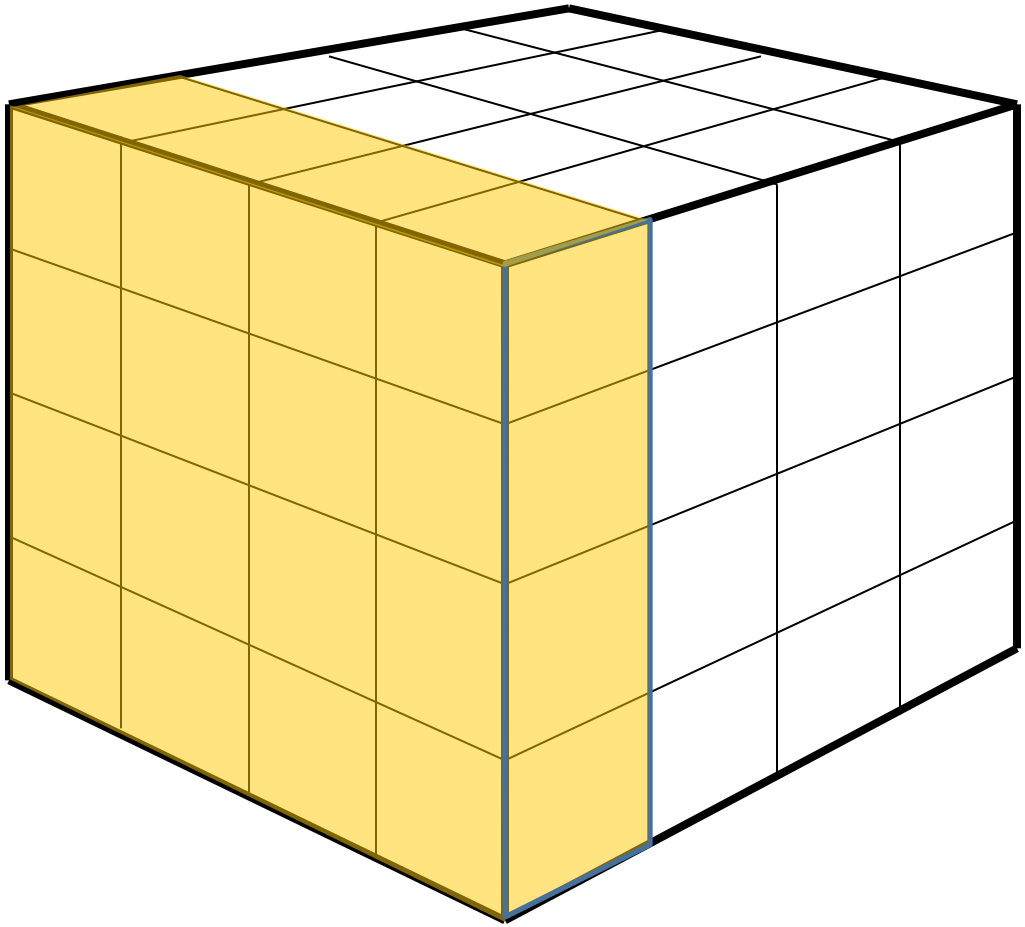
The highlighted “grid plane” will consist of either:

4 quadwords (Cell, Power7, Opteron, Nehalem),

2 octowords (Intel Sandy Bridge), or

1 hexadecaword (Intel MIC, Nvidia Fermi).

Process 2 grid planes at once for 32-wide SIMD of Nvidia Kepler.



Briquettes & Pipelining-for-reuse

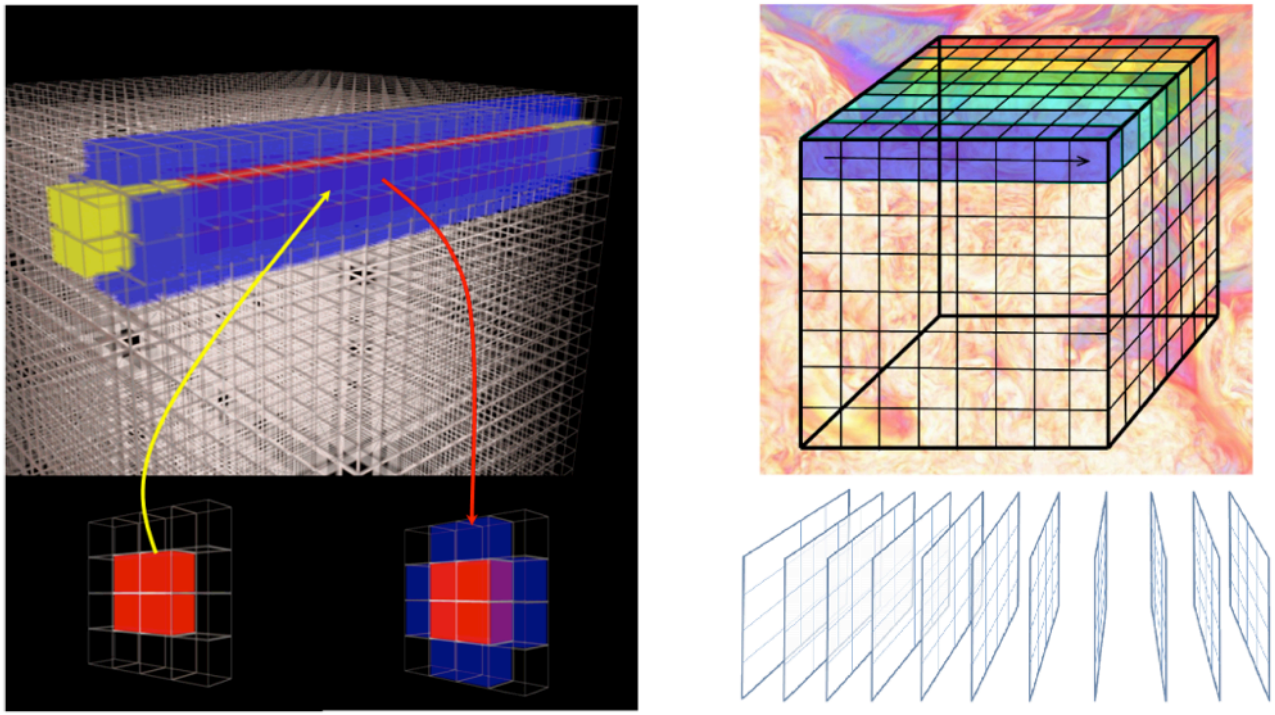
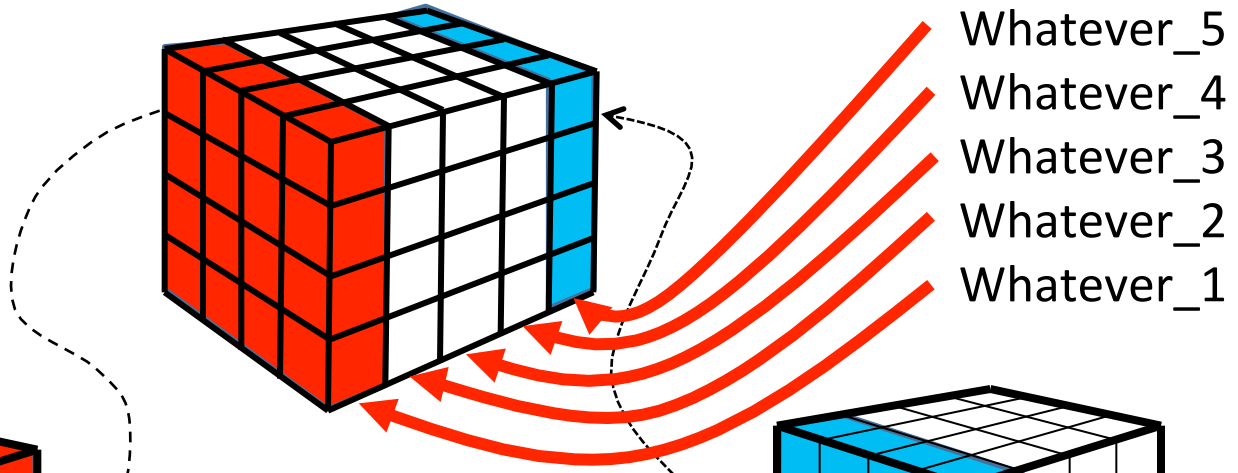


Figure 1. At the left, a grid pencil is indicated within its larger grid brick data structure. It consists of a core of $2^2 \times 16$ grid briquettes, shown in red, surrounded by transverse "ghost briquettes" shown in blue and with longitudinal ghost briquettes shown in yellow. A CPU core works its way down this grid pencil (from left to right in the figure) pulling into its cache memory groups of 4 core and 8 transverse ghost briquettes as shown. These are unpacked to produce vectors representing the 16 cells of single transverse core grid planes as indicated at the bottom right. These form the working data in the L1 cache for 16-long, aligned vector operations. Differences to obtain derivatives in this 1-D pass are formed by subtracting neighboring grid plane vectors along the longitudinal direction of the pass. The transverse ghost briquettes are used to construct, in the cache, 16-long vectors representing grid planes offset by one or two cells in a transverse direction. At the top right in the Figure, we illustrate how 8 processor cores may simultaneously update neighboring grid pencils. The ghost briquettes of one such grid pencil may be core briquettes of a neighboring one, so that they need be fetched from main memory only once into a shared on-chip data cache. This is a streaming paradigm of vector computation that works exceptionally well on modern multicore CPUs.

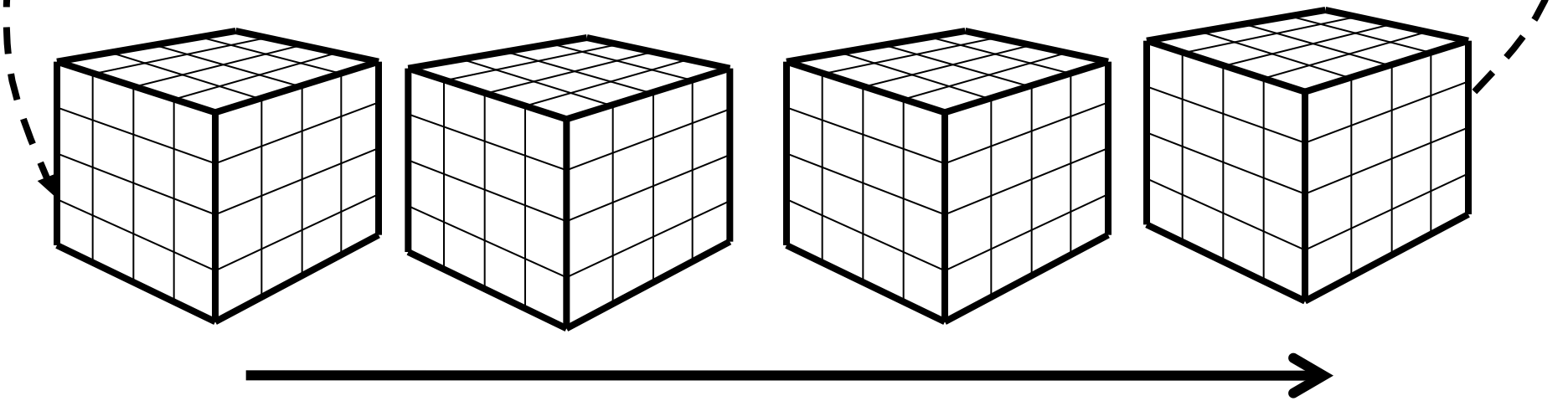
Woodward, P. R., J. Jayaraj, P.-H. Lin, P.-C. Yew, M. Knox, J. Greensky, A. Nowatzki, and K. Stoffels, "Boosting the performance of computational fluid dynamics codes for interactive super-computing," Proc. Intntl. Conf. on Comput. Sci., ICCS 2010, Amsterdam, Netherlands, May, 2010

In the on-chip cache workspace, we have many short segments of grid planes, each holding one variable and none > 5 planes.



These briquettes are in transit between main memory and the cache.

The computation proceeds along a sequence of briquettes at same grid level.



Overcoming main memory bandwidth limitation

- We need about **220 temporary arrays per thread** to update the problem state
- Through our optimizations, we reduced the workspace containing all the 220 temporaries to **just 45.09 KB per thread** *(this is **now 29 KB**).*
- Text segment for computation region is **91 KB**.
- On-chip data **fits onto chip, CPU or GPU**.
- **34 flops per off-chip word read or written**.
- Not memory bandwidth limited on any device.

Performance gains

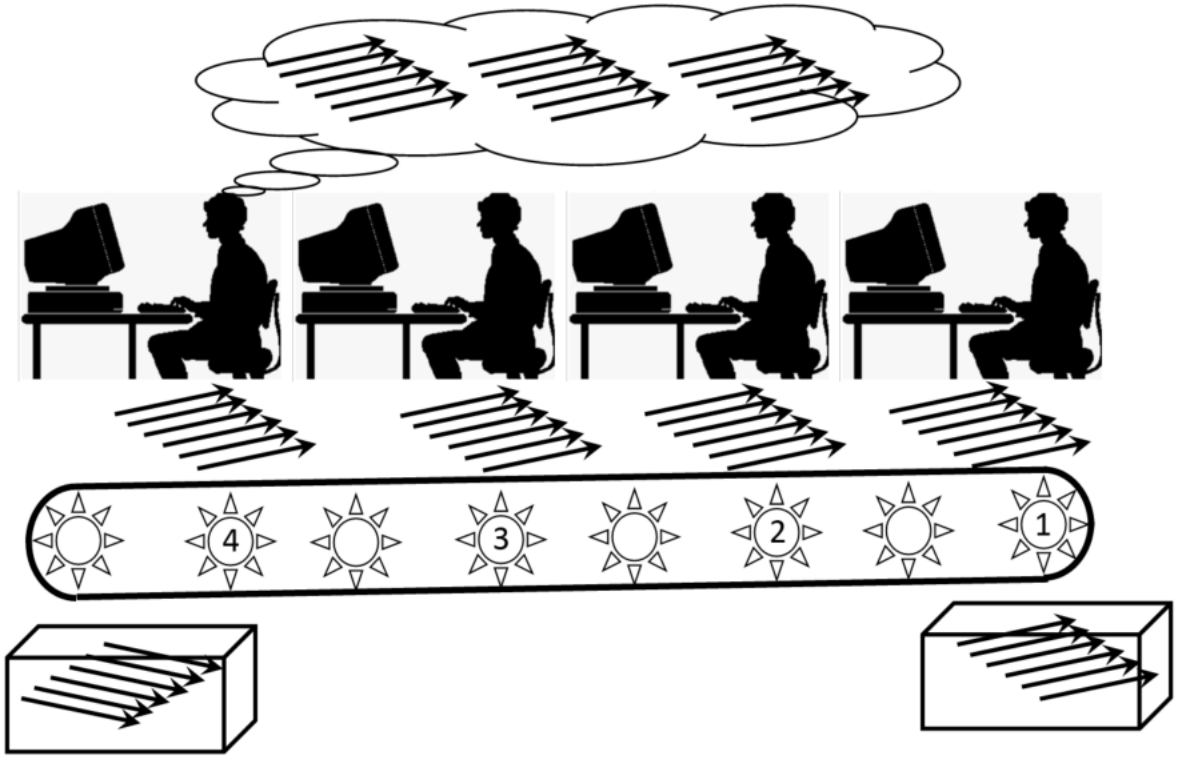
Redundancy in computation eliminated

	Workspace / thread (KB)	flop/cell		
		Fortran-W	Pipelined	% redundancy
RK-adv	16.59	379.89	162.92	133.16
PPM-adv	19.2	454.61	273.31	66.34
tp3	208.28	5195.77	3218.67	61.43

Performance gains for PPM-adv

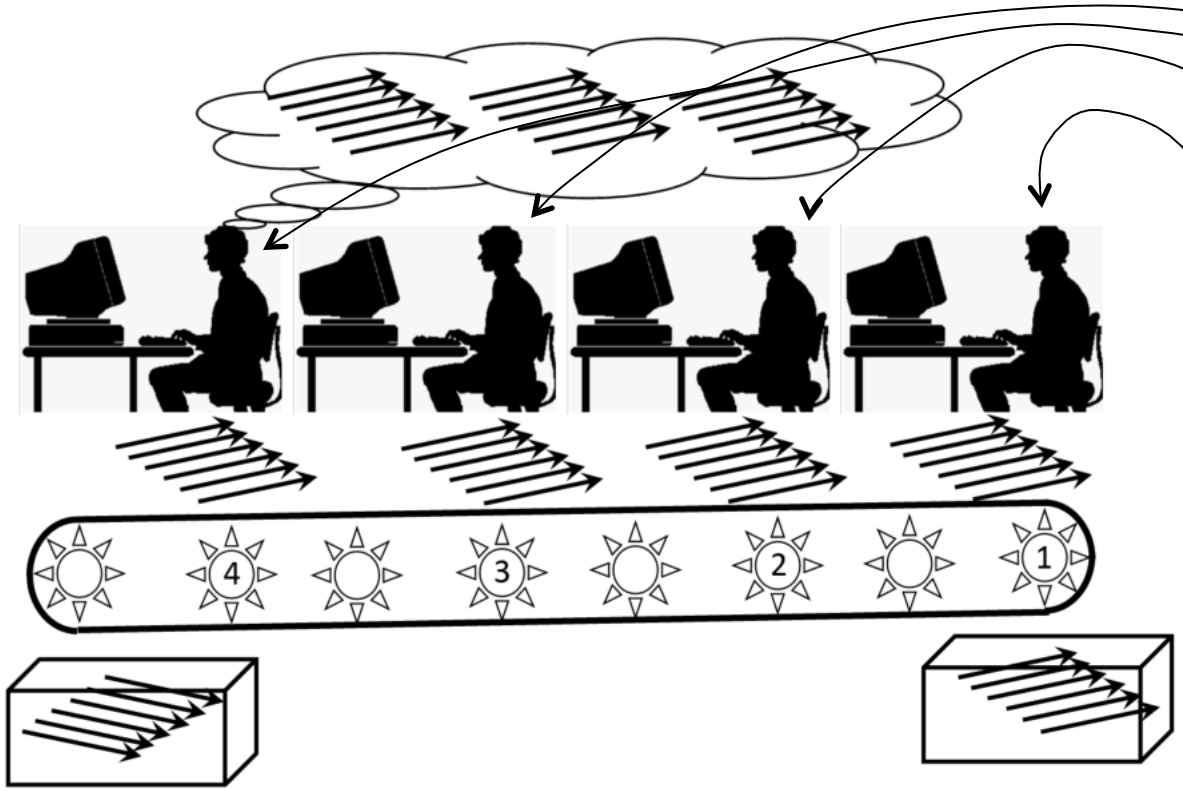
	Speed-up from		
	briquettes	pipelining-for-reuse & memory reduction	both
Nehalem	2x	3.33x	6.69x
Sandy Bridge	3.78x	1.66x	6.28x

Nehalem : Xeon 5570 ; Intel 9 Fortran Compiler; 16 OpenMP threads running on two sockets
Dual-socket, 4-core @ 2.93GHz, SSE-4.2 (128-bit)
Sandy Bridge : Xeon ES-2670; Intel 13 Fortran Compiler; 32 OpenMP threads running on two sockets
Dual-socket, 8-core @ 2.6GHz, AVX (256-bit)
Expect performance to double by number of cores and double by increased vector widths (for vectorized sections), and decrease by 11 % for clock-frequency of Nehalem is higher (3.54x in total)



Production line analogy of pipelined processing of briquettes in mPPM:

- New briquette record fed in at right from off-chip memory.
- Unpack record to produce 16-word or 32-word aligned vectors of physical state variables.
- At station 1, do work that requires only these new vectors.
- At station 2, do work requiring only products of station 1 for this and previous chimes.
- At station 3, do work requiring only products of station 1 for this and 2 previous chimes, as well as products of station 2 for this and 1 previous chime.
- At station 4, do work requiring only . . . . And produce fully updated physical state vectors.
- Work at each station is independent, if insert 1-chime delays between stations.
- Performance is 254 Gflop/s on Trinity node – 11% of the peak performance.



Production line analogy of pipelined processing of briquettes in mPPM:

- Fetch rho_2, p_2, fv_2, ux_2, uy_2, uz_2 at station #1.
  - Generate ceul_2, spdpls_2, spdmns_2, rhoair_2, e_2, etc.
- Fetch fvx_1, fvy_1, fvz_1, fvxx_1, fvxy_1, fvxz_1, fvyy_1, fvyz_1, fvzz_1 at station #2.
  - Generate interpolation parabola in cells of _1 for all variables.
  - First find left- and right-states at left-interfaces of cells of _1 ---> uxavl_1, pavl_1 (get only 1 plane at first)
- Solve Riemann problems to obtain fluxes at right-hand interfaces of cells of _0 at station #3
  - At right-interfaces of cells _0, do Lagrangian update of advected chunk to get flux.
- Apply fluxes with conservation laws to get new fluid states in cells of _0 at station #4.
- Fully updated briquette record produced on chime #6, when pipe is fully primed.

Table 1. multifluid PPM 1-D Grid Pencil Update

Device	GHz	Threads of control / cores	Gflop/s (32-bit)	GB/sec
dual AMD Interlagos node	2.3	32 / 32	71.7	8.30
dual Intel Sandy Bridge	2.0	32 / 16	84.7	9.80
dual Intel Haswell node	2.3	64 / 32	254	29.4
Nvidia K20	0.73	168 / 14	121	38.9
Nvidia K40	0.75	180 / 15	136	44.5
Nvidia K80	0.82	416 / 26	432	67.4

- Have transformed multifluid PPM module for GPUs.
- Pei-Hung Lin developed a code translator taking this Fortran to CUDA, with above results.

Table 2. Characterizing mPPM's 4 Episodes of Computation

Episode	Vect-ors In/Out	Vectors Required on Chip	Flops per Vectors in+out	Flops / Off-Chip Vectors In+Out
1	34 / 53	68+85	413 / 87	413 / 8
2	35 / 38	68+90	219 / 73	219 / 8
3	56 / 40	68+95	231 / 96	231 / 8
4	73 / 16	68+92	233 / 89	233 / 8

- Have transformed multifluid PPM module for GPUs.
- Pei-Hung Lin developed a code translator taking this Fortran to CUDA, with above results.
- The 4 episodes of computation are the 4 off-chip data accesses and the work done in them.
- Making each episode into a full-fledged "kernel" results in an increase in the off-chip memory bandwidth requirement by roughly an order of magnitude.
- Just as with the smaller subroutines in the mPPM module, structure as separate kernels is inefficient. Working from an on-chip cache gives the best performance on all devices.

ILP flag	dual Haswell Gflop/s	with mm_prefetch, hint=3	best reads with no writes	with no reads or writes	dual Haswell 1 thd/core	with mm_prefetch, hint=3	same with no reads or writes
0	244	273	314	324	135	134	263
1	301	334	392	399	211	222	343
2	307	322	363	390	221	228	374
3	323	298	331	362	291	276	367
5	257	256	278	303	274	271	358

- This is the PPM advection algorithm for a single variable – 81 flops/cell, 40 flops/word.
- Unrolled 0, 1, 2, 3, or 5 times from code to process 32-word pair of grid planes at a time.
- mm_prefetch does not help much.
- Compiler must not be generating “streaming stores.”
- Hope we can fix these things, but might have to convert to C and call intrinsic functions.
- Delivered performance is: 244 Gflop/s = 10.4% of peak.
- Potential is: 392 Gflop/s = 16.6% of peak.
- This potential is what we realized on Sandy Bridge. We want the lost % of peak back!

ILP flag	Sandy Bridge 2.0 GHz Gflop/s	with mm_prefetch, hint=3	with no writes	with no reads or writes	Sandy Bridge1 thread per core	with mm_prefetch, hint=3	same with no reads or writes
0		55.4		63			
1	89.3	92		105.1			
2		72		101.7			
3		83.9		92.3			
5		66.8		74.3			

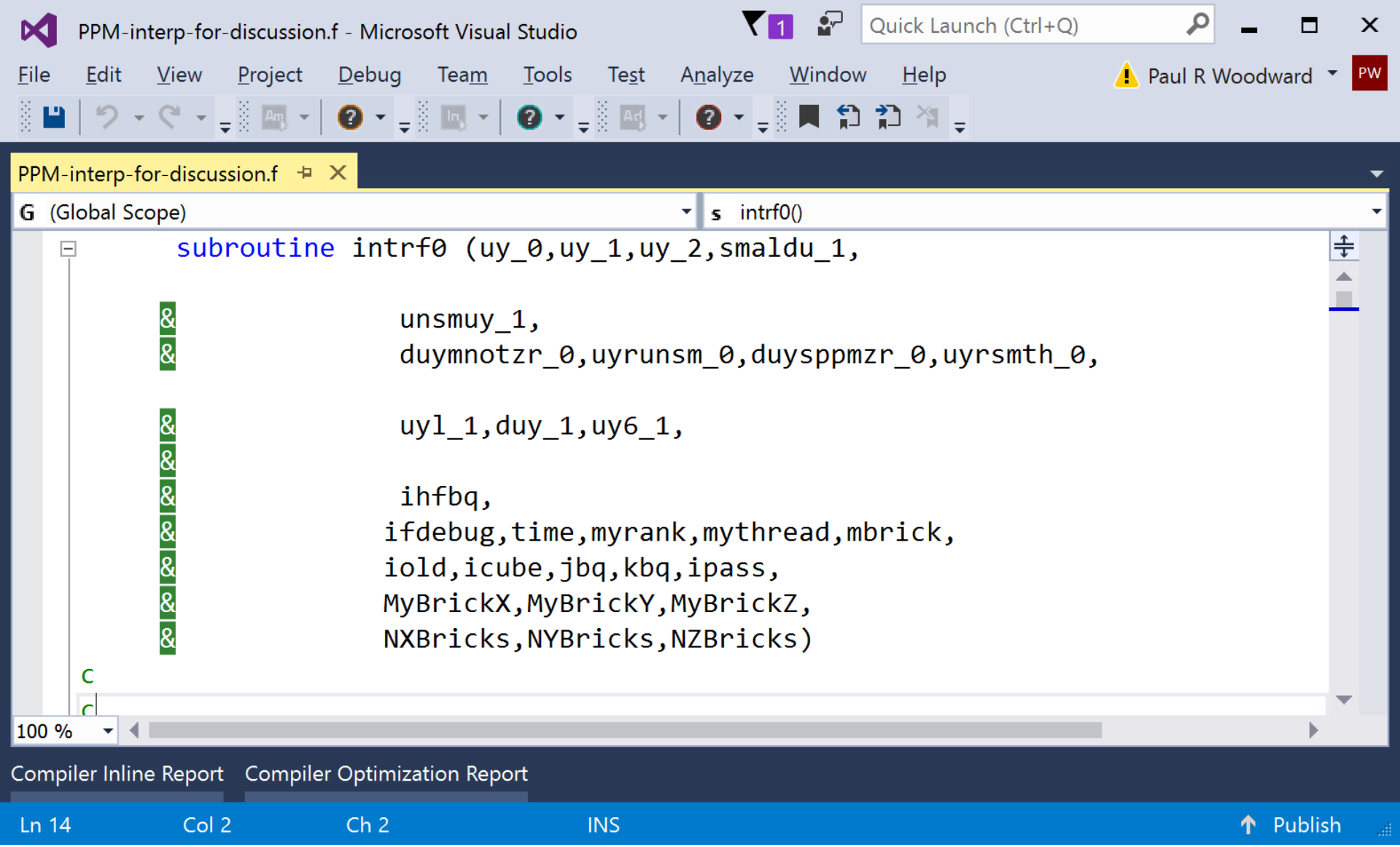
There is basically little advantage to be had on Sandy Bridge CPUs from reading and writing nothing at all. But more ILP helps.

ILP flag	dual Haswell 2.3 GHz Gflop/s 36 cores	secs	with no writes	with no reads or writes	K40 Gflop/s	K40 secs	same with no reads or writes
0					134.88	13.82	
1					210.23	8.862	
2					241.76	7.707	
3					257.31	7.241	
5							

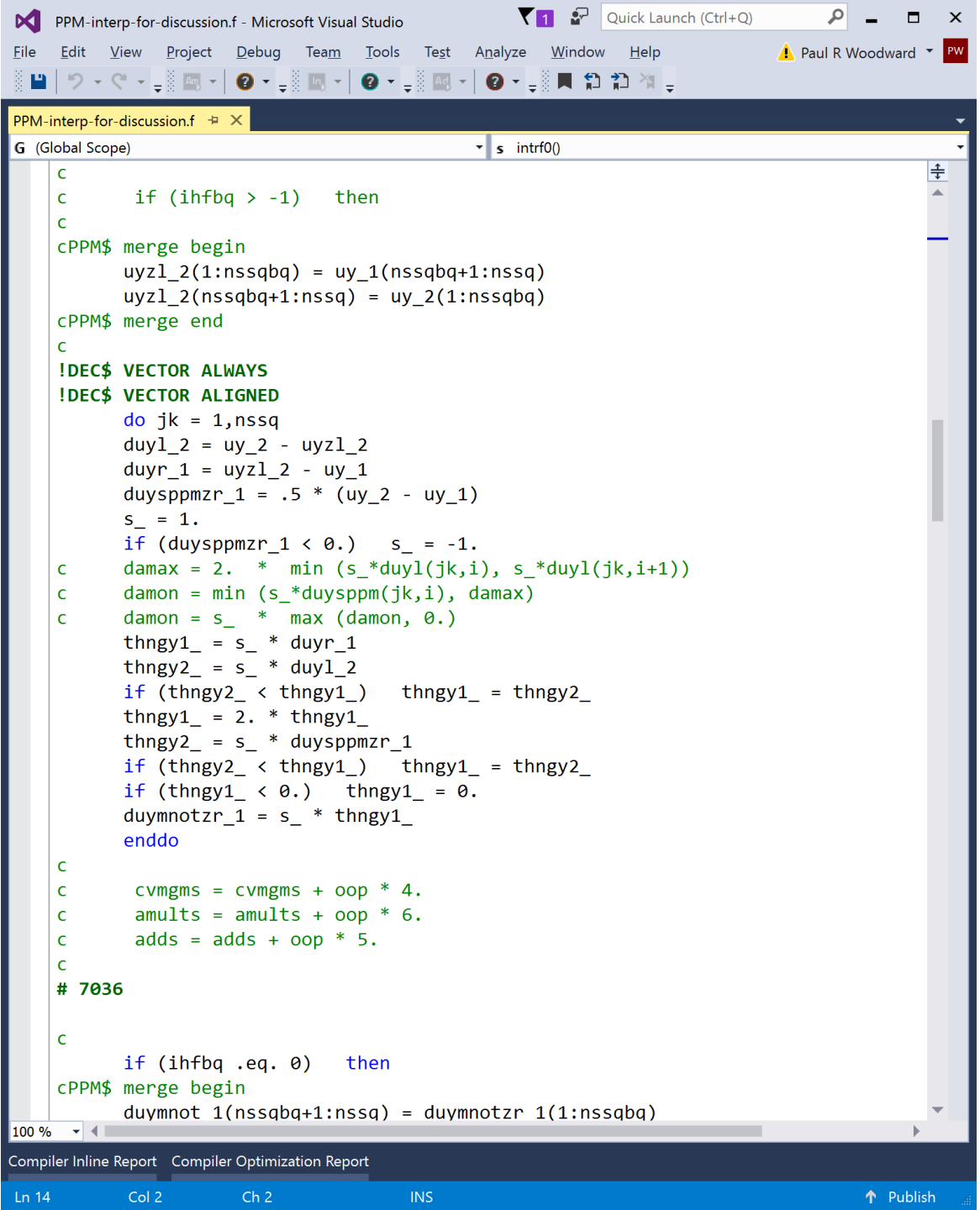
On Nvidia GPUs, there is a factor of 2 to be gained by generating enough independent instructions to keep the arithmetic going.

Conclusion from the ILP tests:

1. Despite producing thousands of aligned, 32-long vector operations, clearly this is not enough for the latest hardware.
2. **For GPUs and for CPUs, it is worth unrolling once**.
   a) This is easy to do, through *a mechanical transformation*.
   b) No compiler will ever do this for you.
   c) Pei-Hung Lin will write a pre-compiler to do this.
   d) I have written a pre-pre-compiler to lessen his task.
   e) Final output code will be either Fortran or CUDA.
3. This approach of a custom code translation tool is being taken by many teams in Europe as well as in the US.
4. I will be comparing notes this summer with some of these efforts to see what techniques can be borrowed or loaned.
   a) Common goal of easy initial code expression.
   b) Common goal of near optimal performance on all targets.
   c) Eventually, these code transformations will end up in languages and compilers, but could take 20 years (CAF).
   d) Best, it seems, to keep it as simple as possible.



Here is roughly 60% of the code in the ILP test, to illustrate the technique of algorithm unrolling. 6 grid planes of uy values are input, and 2 grid planes of parabola are output.



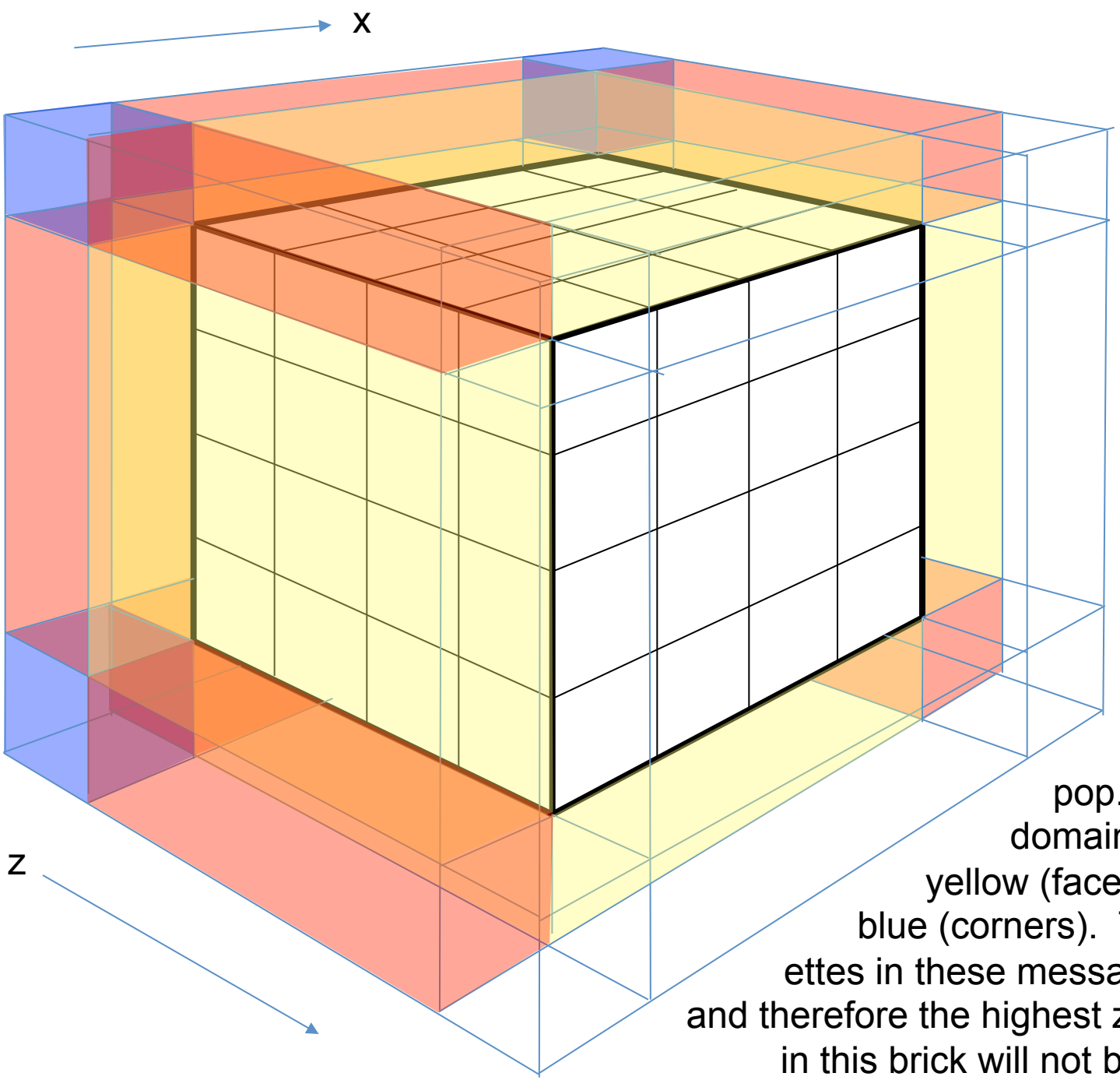
Here is the construction of differences of uy across cells, based either on the assumption that uy is smooth or that it is not. Note that I cannot construct s_ until I have duysppmzr_1. This forces a serialization of these 2 instructions. Similar comments apply to the thngys. If I also build duymnotzr_2, this problem disappears.

Challenges to Scalability for AMR:

1. **AMR reduces grid size**, so there is a lot less parallelism.
2. Coarser grids sit idle while finer ones are updated, so there is **even *less* parallelism**.
3. Can solve these 2 problems with a **bigger scientific problem**
   a. Make problem big enough to fill machine anyway.
   b. But then have some fraction of tiny cells.
   c. Tiny cells have tiny time steps.
   d. Damn. **Fill machine, but runs forever**.
4. Can solve this new problem with a **faster machine**.
   a. This is great unless the machine is also bigger.
   b. The machine might also be too expensive, and they won't let you on it.
5. Can solve this problems with **a faster code**.
   a. This is great unless the code outruns the interconnect.
   b. Obviously, one needs faster interconnects.
   c. You could also pass less info and compute more.
6. **My solution: All of the above**.

Some Ideas for a Faster AMR Code:

1. Limit yourself to **just 3 grid levels**.
    a. This is a factor of 256 in required computational effort, and if that saving is not enough, you probably should think again.
    b. This statement is obviously problem dependent, but if you have tenure, you can pick your problem.
2. **Make all your subdomains the same size**.
    1. Your friends may laugh at you, but this could really save you a lot of work.
    2. Your easiest subdomain will cost 256 times less.
        1) This is no problem, just give it 256 times fewer threads. After all, the hardware is forcing you to run more than this number of threads on each node.
    3. All your domain faces, edges, and corners will line up.
        1) This is an incredible simplification, exploit it.
3. **Slice the domain in 2, and update the subdomains on the slicing surface twice per update round**. This is really helpful.



Here we show a grid brick of only 32^3 cells, with its interior briquettes of 4^3 cells indicated. The MPI ghost cell messages that are received before the pop. III update with pop. I bricks in the $z < 0$ domain are shown shaded yellow (faces), red (edges), and blue (corners). The highest z briquettes in these messages are not correct, and therefore the highest z plane of briquettes in this brick will not be updated correctly.

Work supported by:

1. NSF through research grants and grants of computer time on the Blue Waters machine at NCSA.

2. DoE labs through contracts from Los Alamos and Sandia

3. Blue Waters project at NCSA, through code improvement subcontracts.

4. Very enjoyable visits to the University of Zurich.

5. Very interesting experience gained with the IBM Cell processor on the LANL Roadrunner machine.