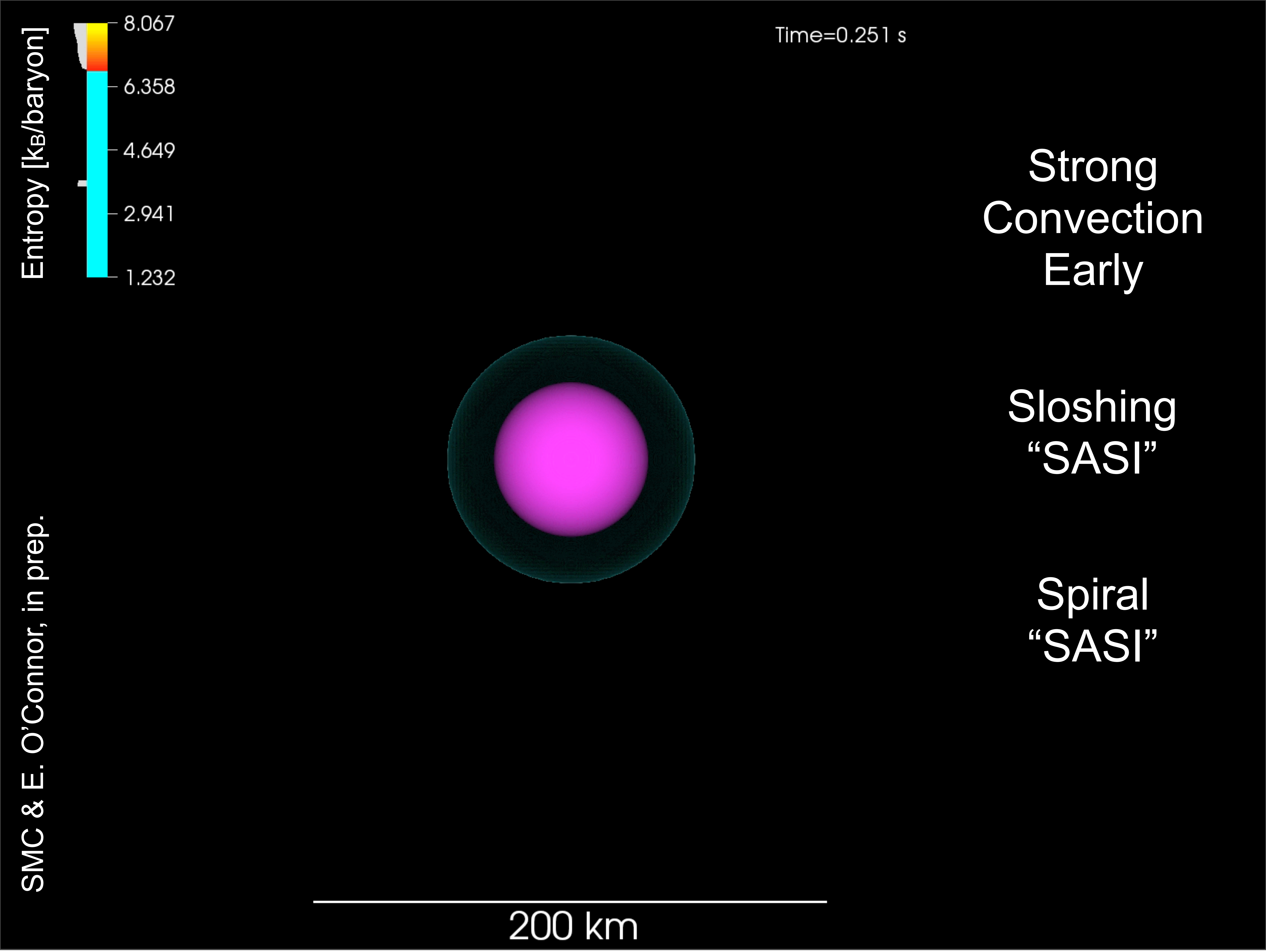
Petascale Post-doctoral Computing or: How I Learned to Stop Worrying and Blow Up Stars

Sean M. Couch
Hubble Fellow, University of Chicago
smc@flash.uchicago.edu

Argonne Training Program on Extreme-Scale Computing
St. Charles, IL, 30 July 2013

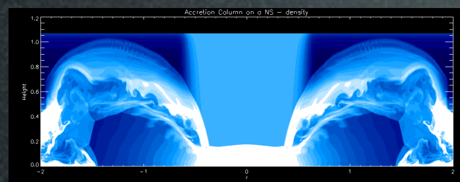
Entropy [k_B/baryon]

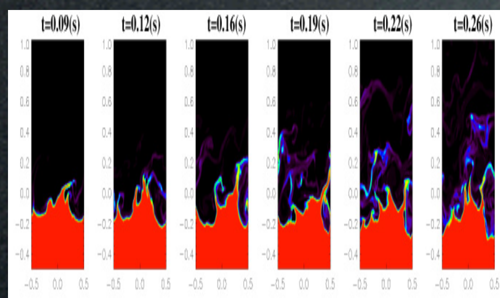
SMC & E. O'Connor, in prep.


Entropy [kB/baryon]
8.067
6.358
4.649
2.941
1.232
Time=0.251 s
Strong Convection Early
Sloshing "SASI"
Spiral "SASI"
SMC & E. O'Connor, in prep.
200 km

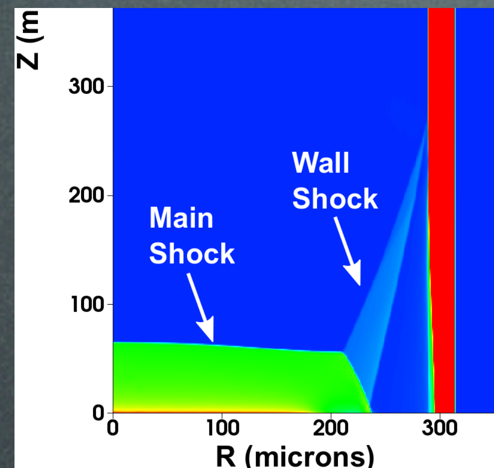
FLASH: A Multiphysics Simulation Framework

Relativistic accretion onto NS

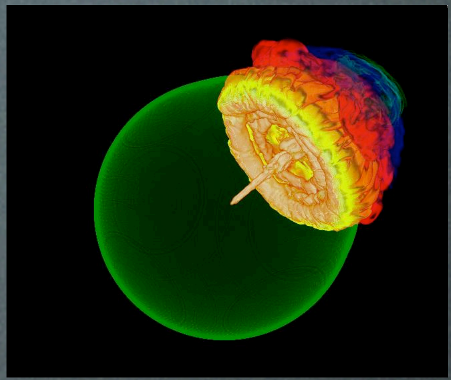




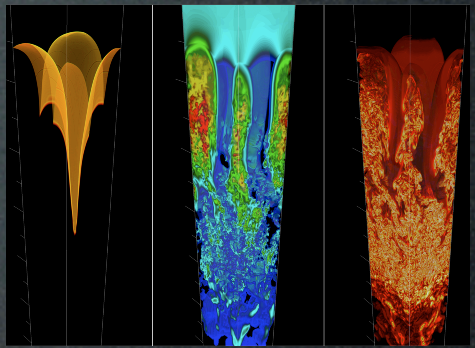
Wave breaking on white dwarfs



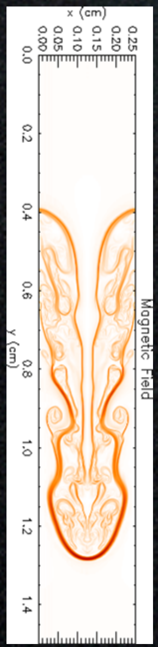
radiative shock experiment



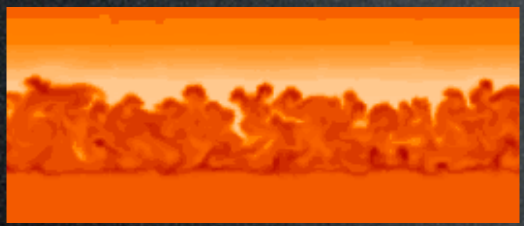
Gravitationally confined detonation



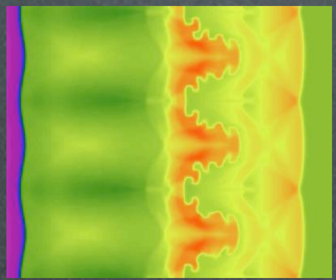
Turbulent Nuclear Burning



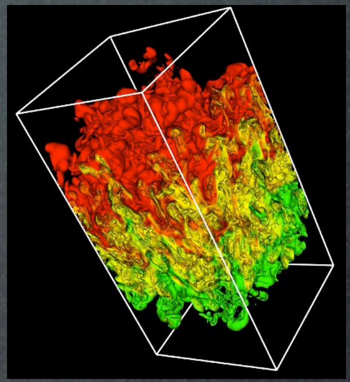
Magnetic Rayleigh-Taylor



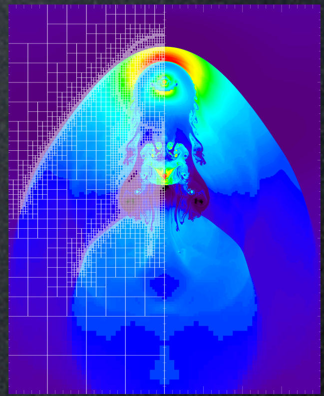
Nova outbursts on white dwarfs



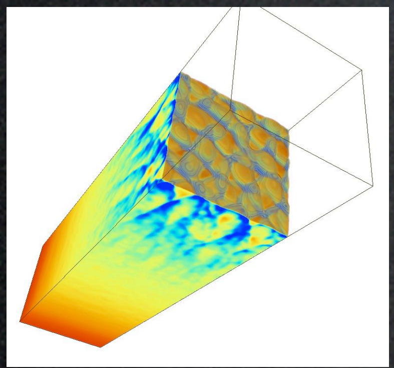
Laser-driven shock instabilities



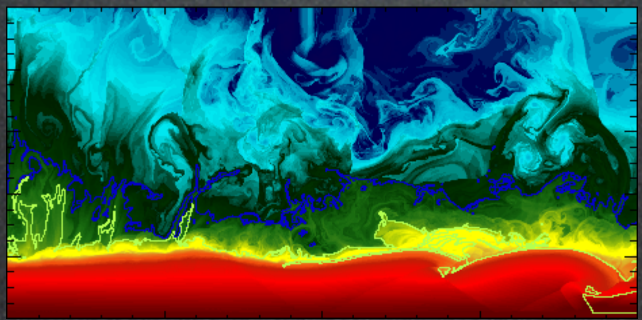
Rayleigh-Taylor instability



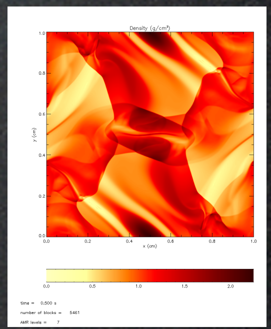
Intracluster interactions



Cellular detonation



Helium burning on neutron stars

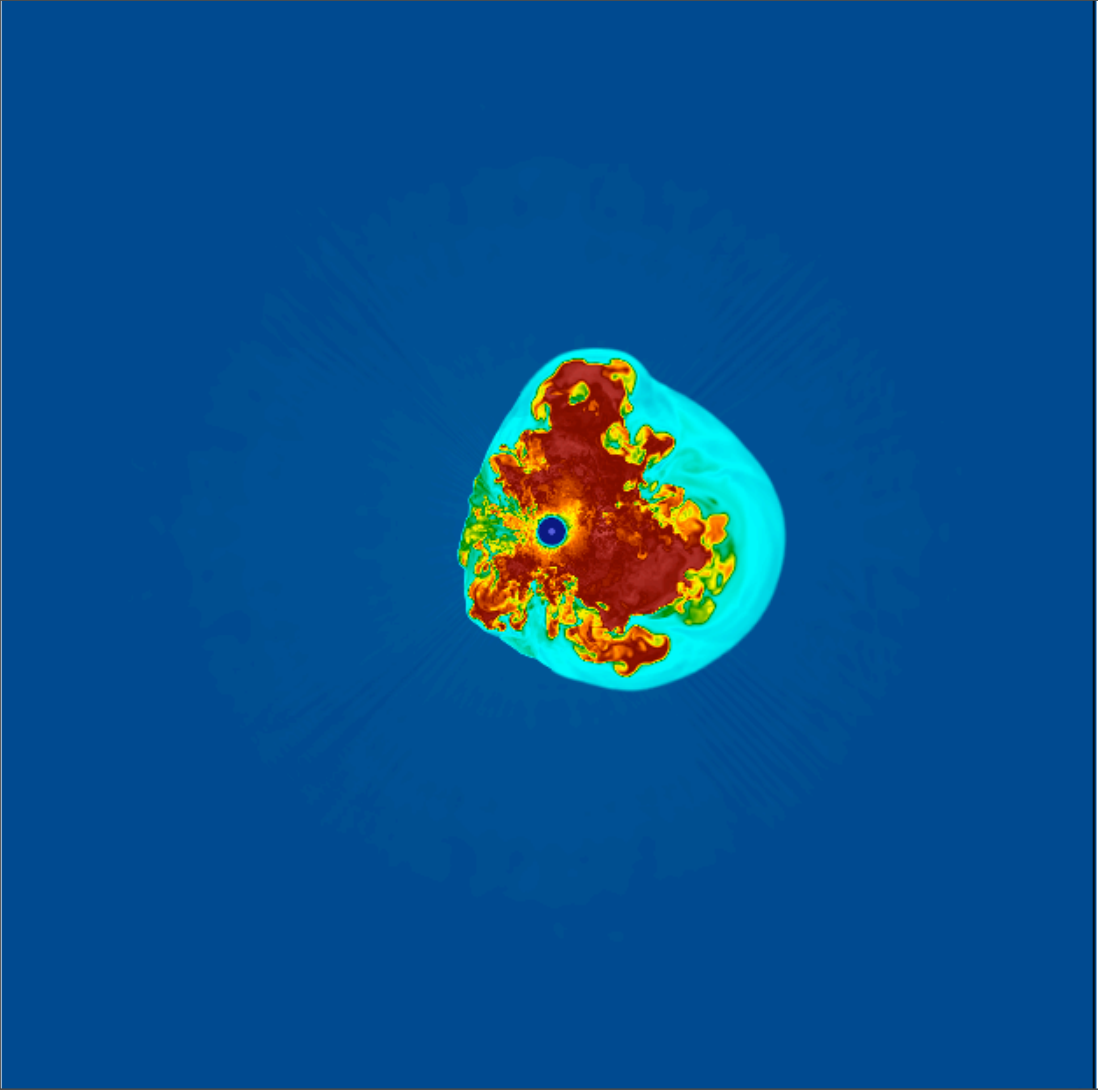


Orzag/Tang MHD vortex

Richtmyer-Meshkov instability

FLASH: A Multiphysics Simulation Framework

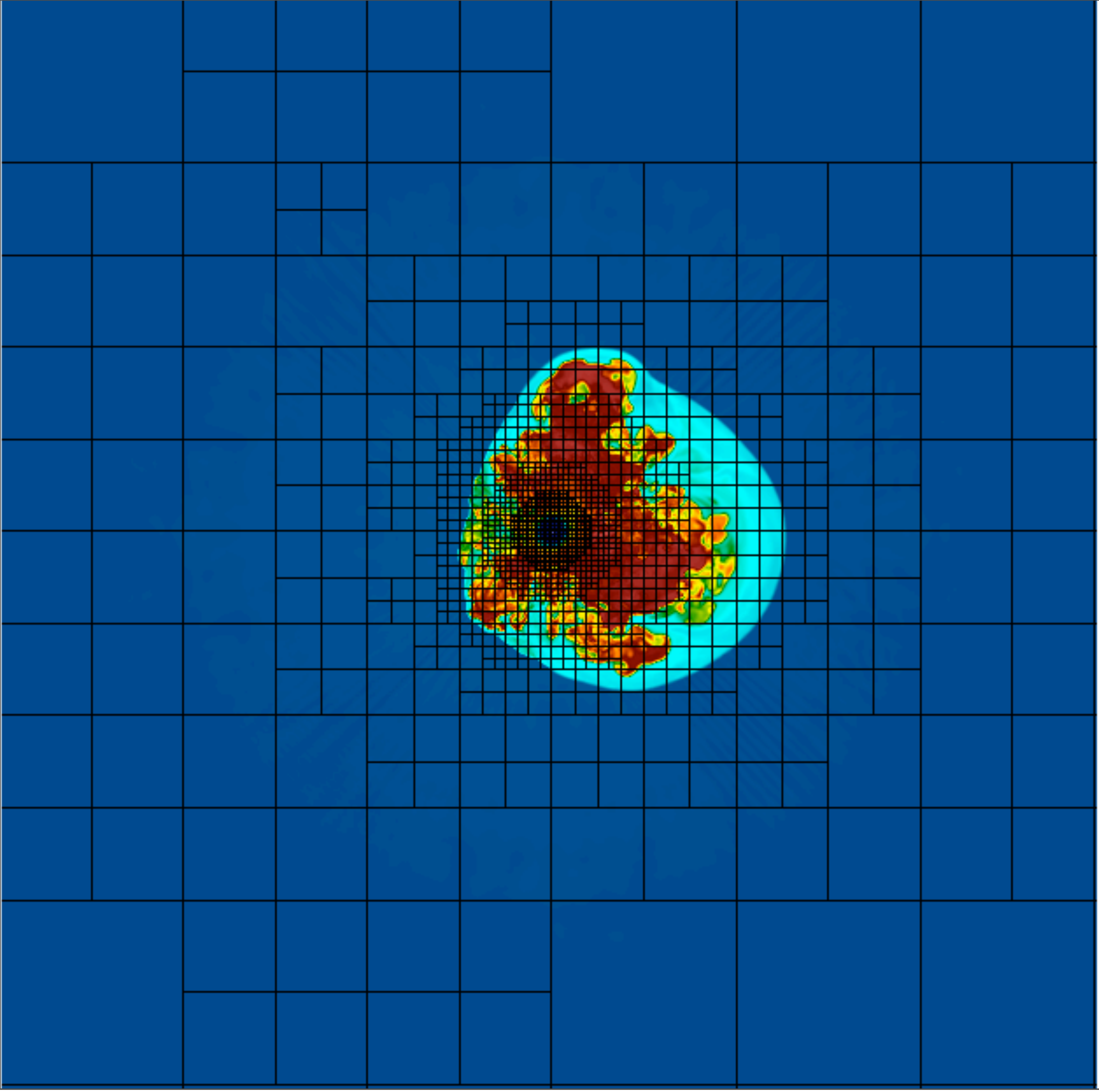
- Solves (hyperbolic) Euler equations in time-explicit, high-order Godunov approach.
- Self-gravity via solution of Poisson's eqn. (elliptic).
- Realistic table-based EOS
- Neutrino "leakage" in ray-by-ray approx.



AMR

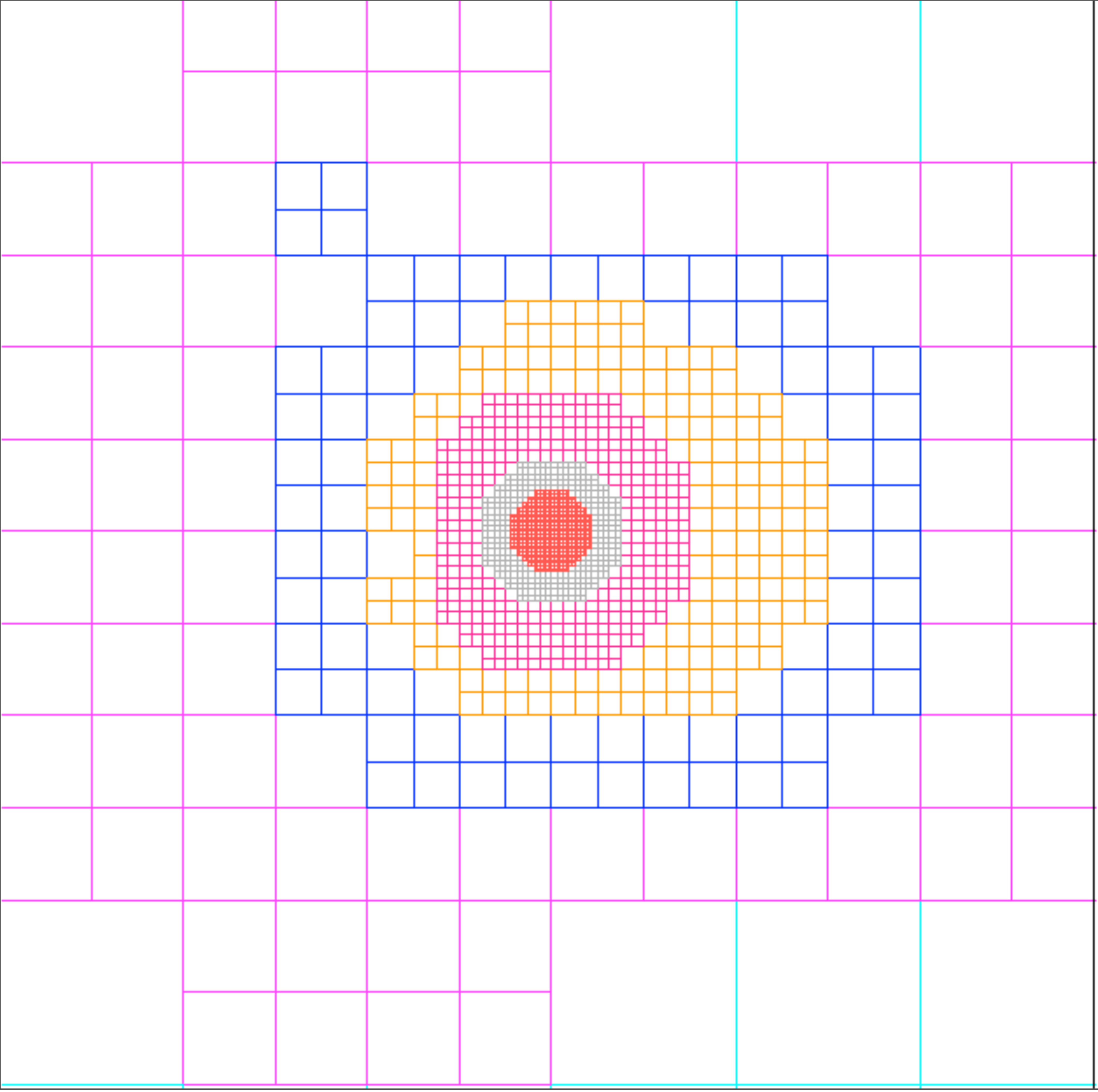
S.M. Couch

AMR



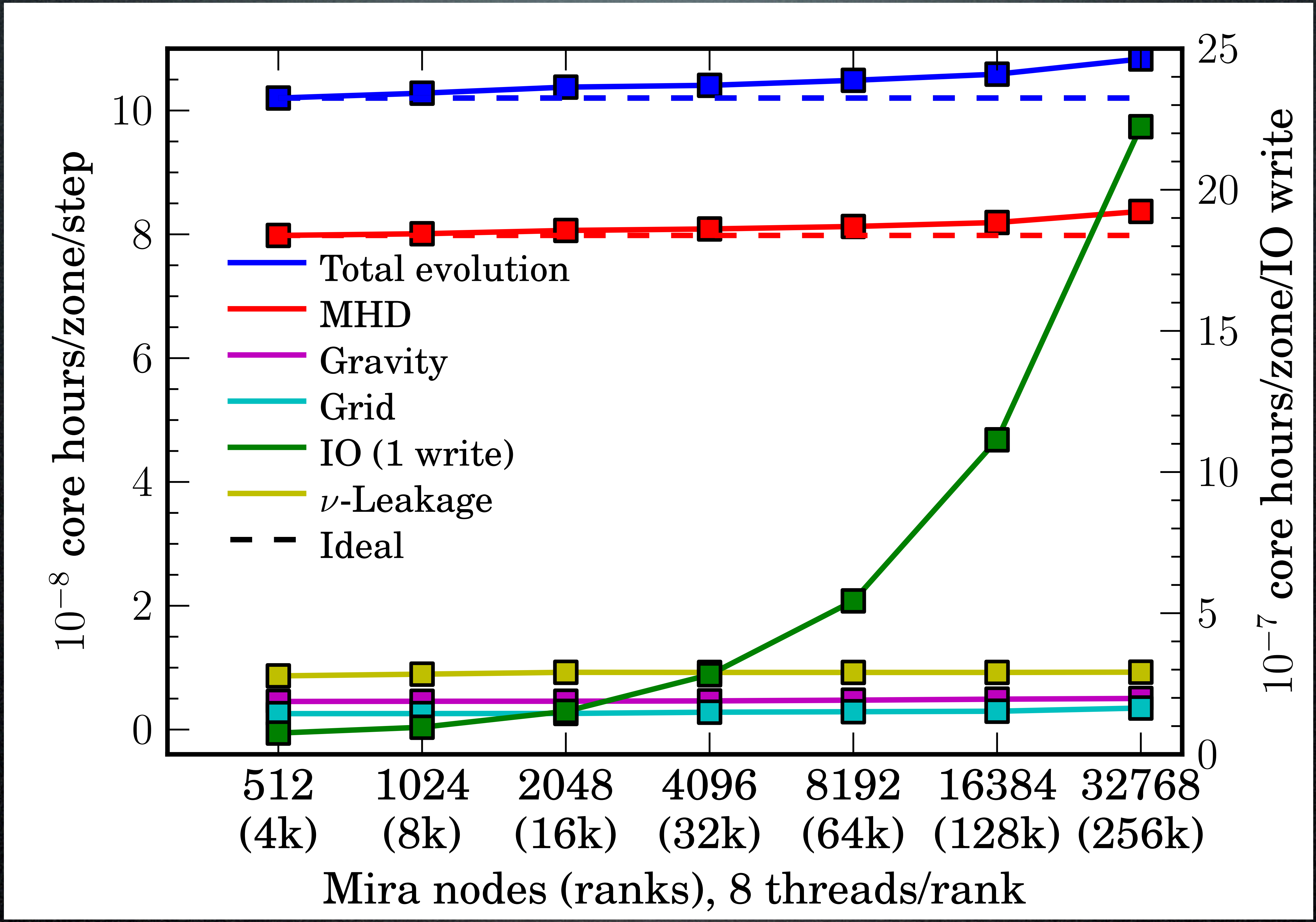
S.M. Couch

AMR



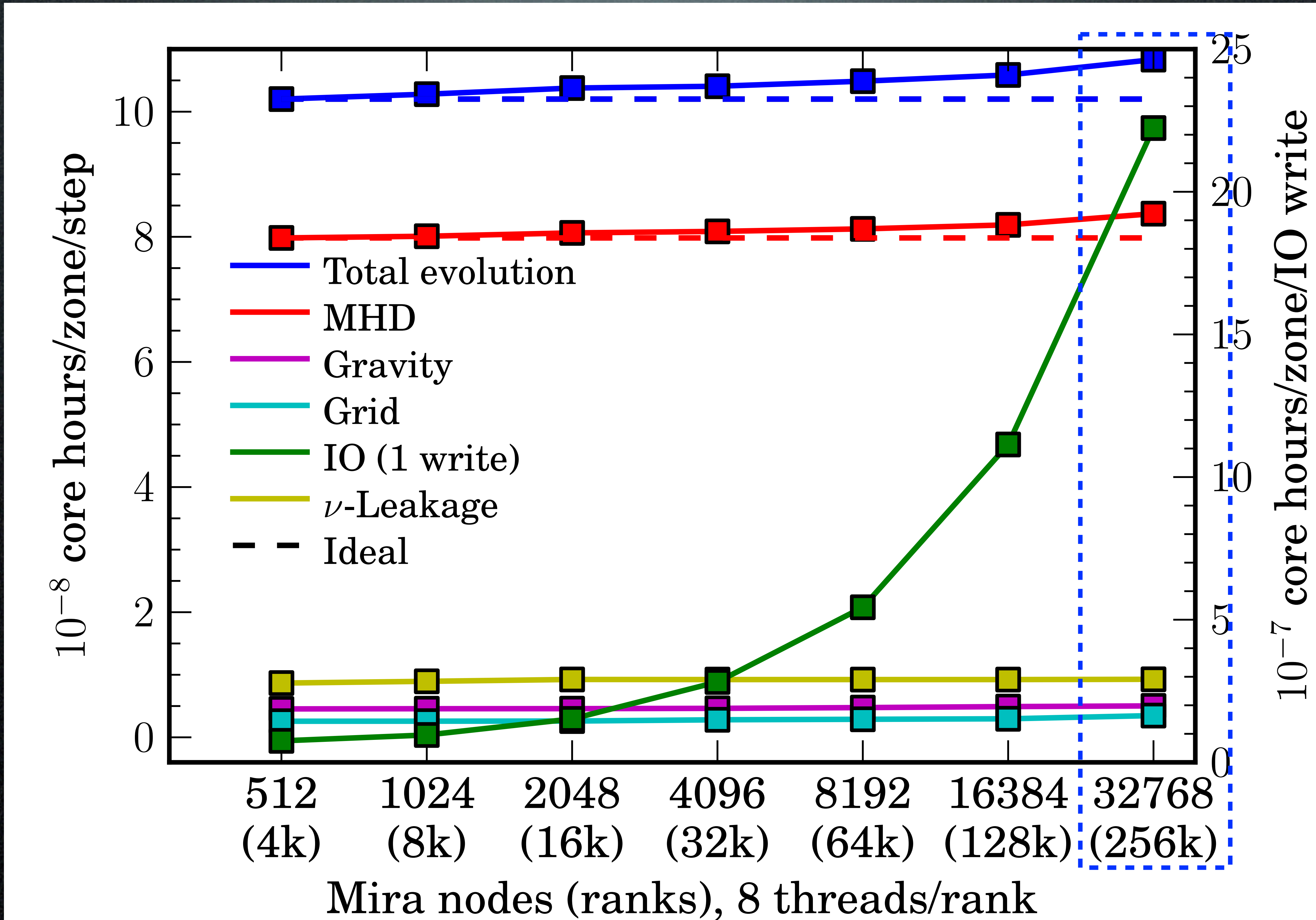
S.M. Couch

FLASH Scales!



FLASH Scales!

2,100,000 threads!



Involvement in Code Devel.

- Early (grad student): little, but some. Extending existing code features. New problems.

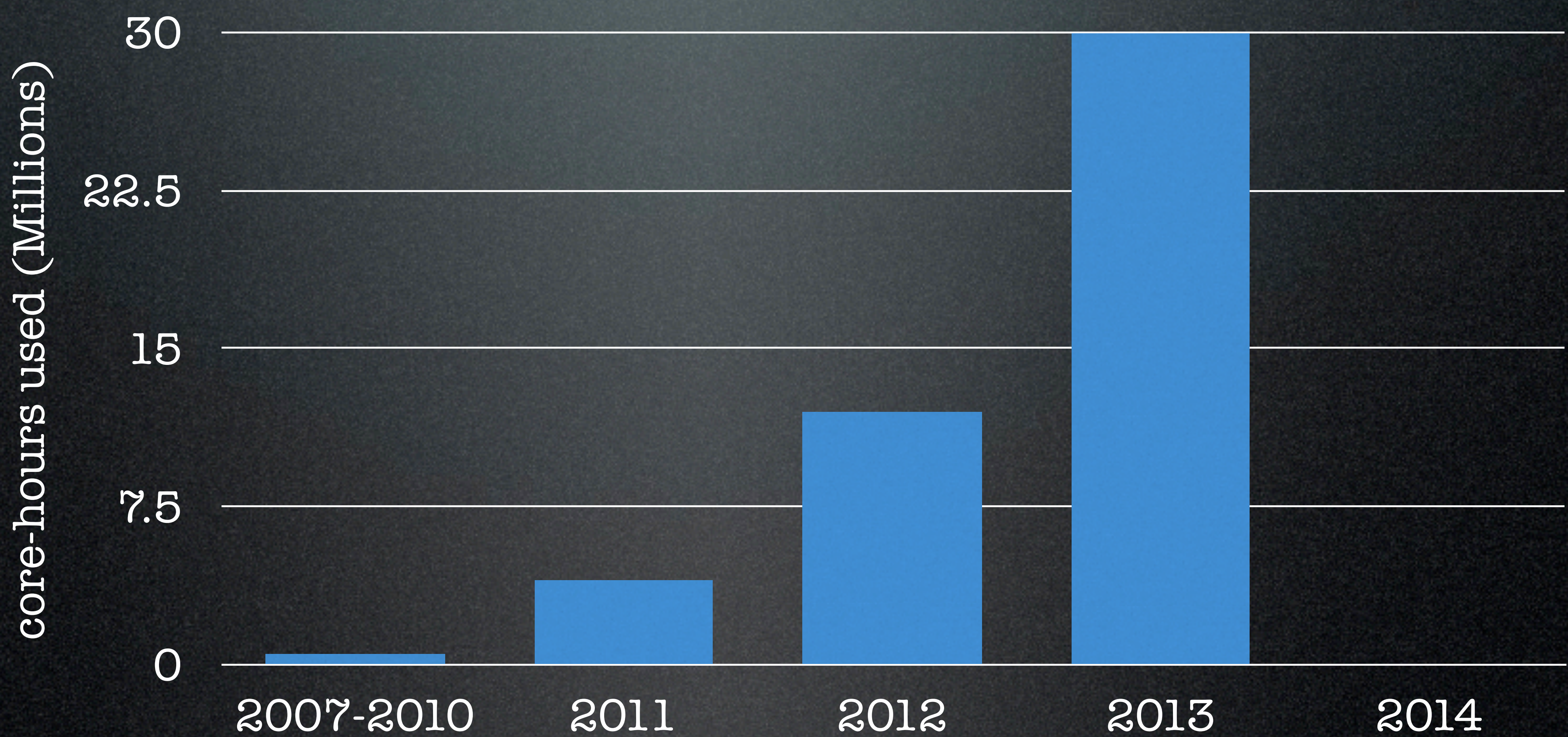
- Later (sr. grad): new physics capabilities based on existing code units.

- Post-doc: completely new physics models.

- Now: extensive. Development of new algorithms for gravity/hydro/MHD/radiation, threading, etc.

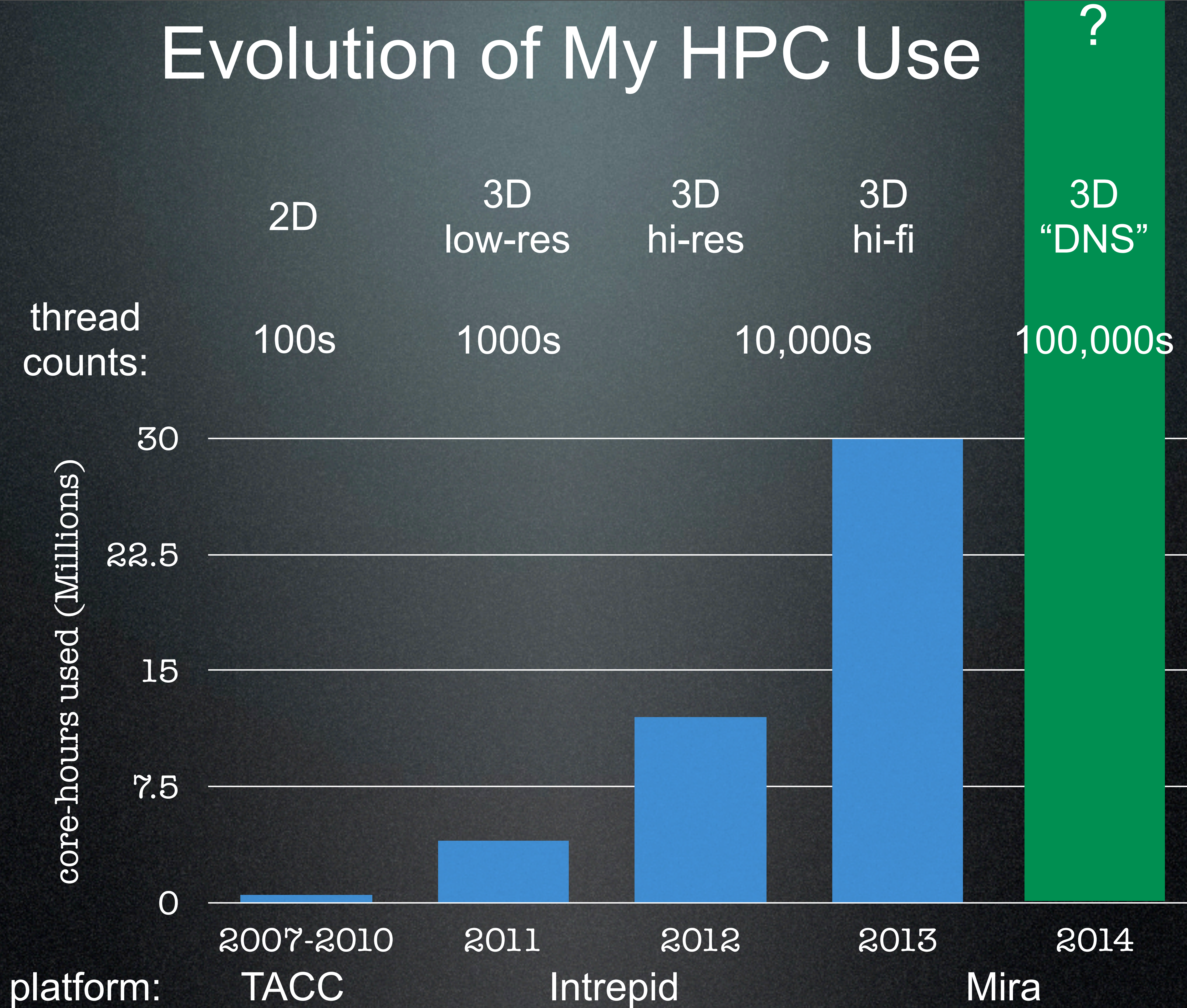
Evolution of My HPC Use

	2D	3D low-res	3D hi-res	3D hi-fi
thread counts:	100s	1000s	10,000s	



platform: TACC Intrepid Mira

Evolution of My HPC Use



Breaking into Extreme Scale

- Early: medium scale. E.g., TACC, NERSC, NSF Tera-Grid, etc.
- Resources from UChicago/Flash Center
- Discretionary allocation(s)

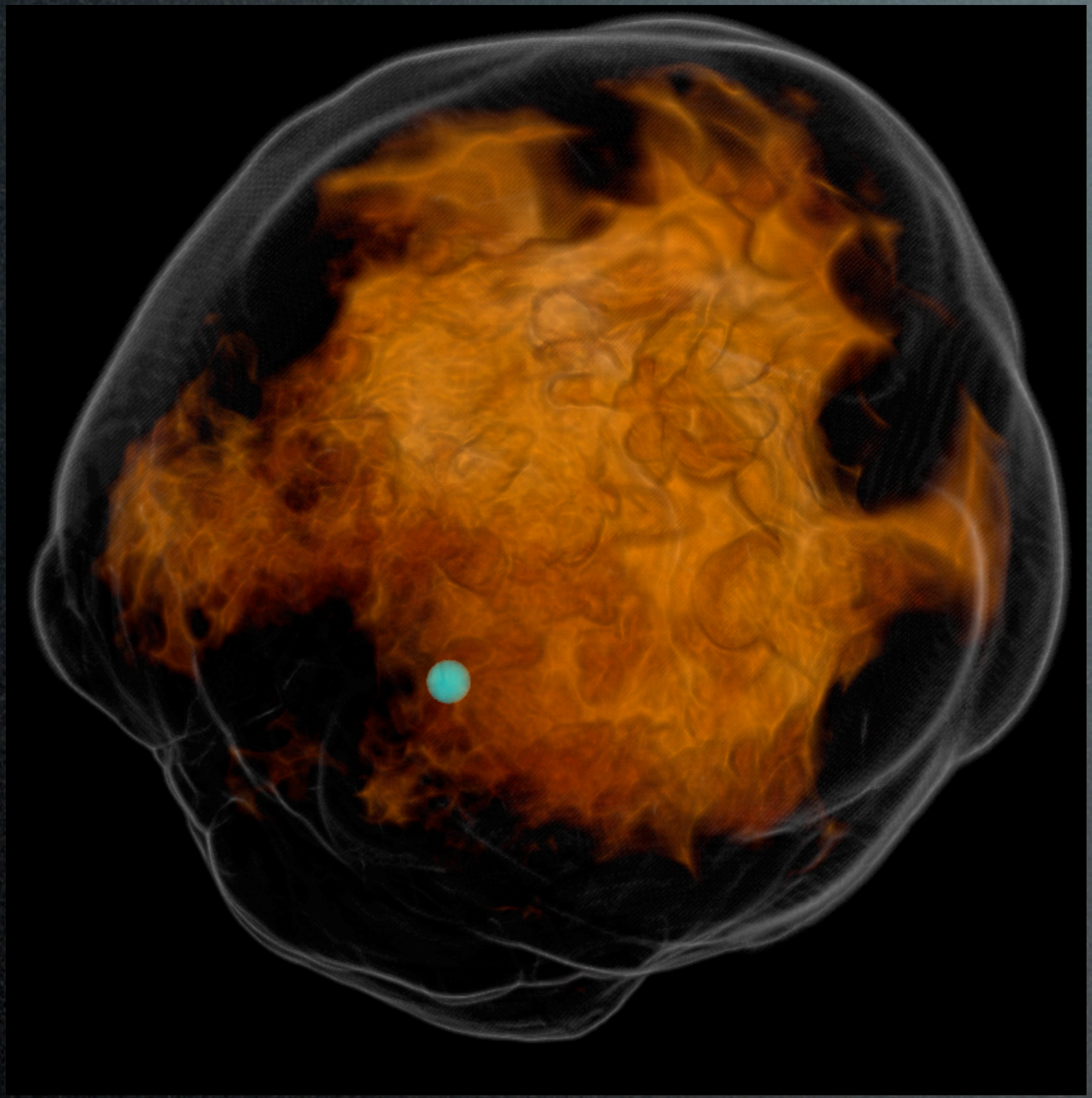
Discretionary Time

- Not just for code development and testing!
- Do science with it!
- Makes it easy for the Director to justify giving more time.

Give Back to Your Patrons

- Providing extreme-scale resources is not cheap!
- Funding is not easy to come by...
- Help you patrons justify their continued funding.
- Publish papers, obviously.
- Also, give back in other ways.

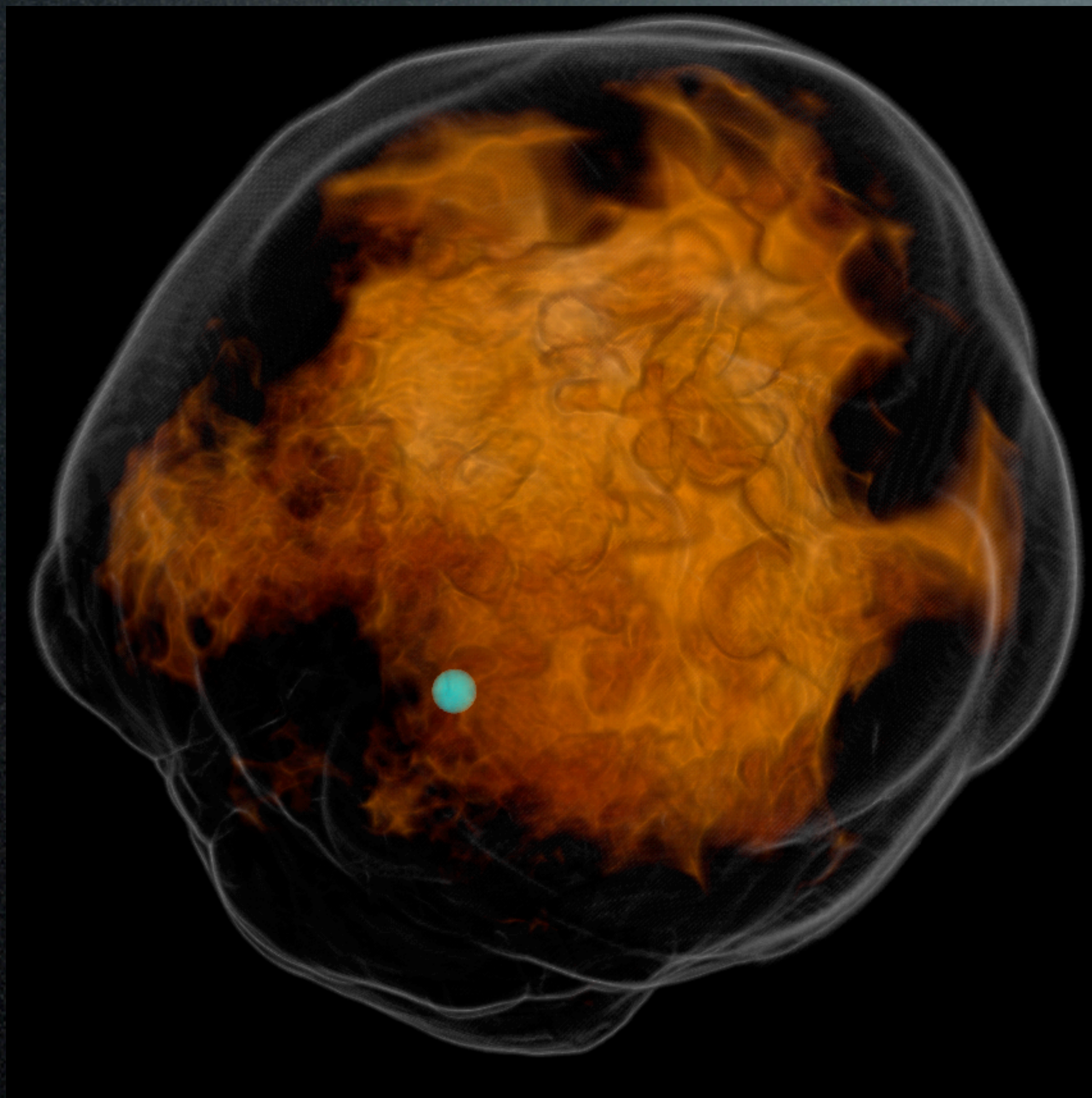
Era of 3D CCSN Simulation



SMC, 2012, ApJ in press

- Grand challenge for computational astrophysics.
- Compromises must be made at the peta-scale.
- 3D makes an enormous impact.
- Nature is (at least) 3-D, and so are SNe!

Era of 3D CCSN Simulation



SMC, 2012, ApJ in press

- Grand challenge for computational astrophysics.
- Compromises must be made at the peta-scale.
- 3D makes an <u>enormous</u> impact.
- Nature is (at least) 3-D, and so are SNe!

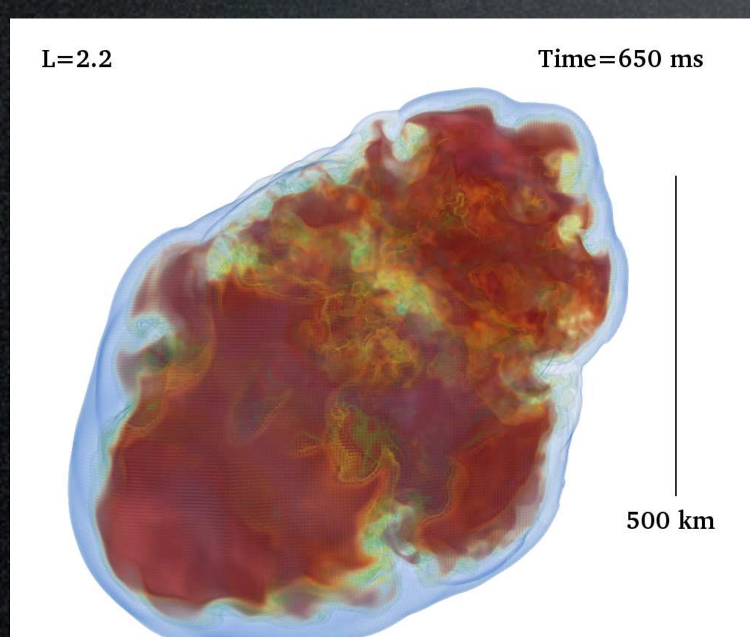
FLASH open-source
multi-physics
simulation framework,
flash.uchicago.edu

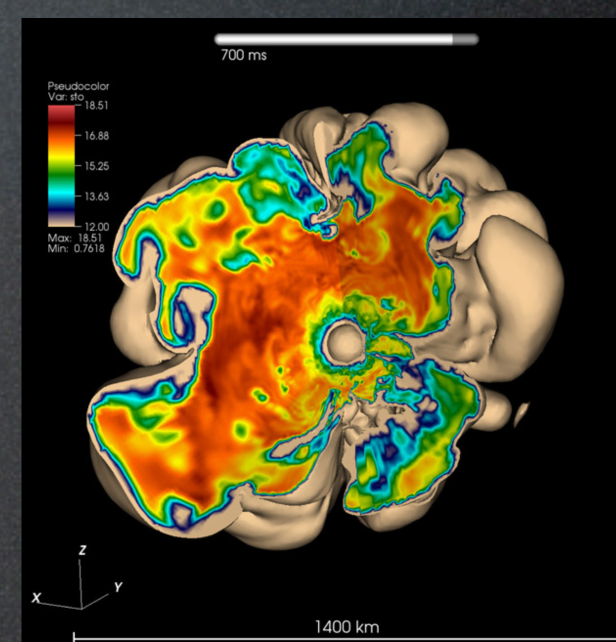
Thanks to Argonne
Leadership Computing
Facility!

Is 3D the Key to Robust Explosions?

- Going from 1D to 2D results in favorable conditions for explosions. Multidimensional effects important!
- Will a fundamentally-3D phenomenon aid explosions?
- Parametric sims from Princeton group show easier explosions in 3D v. 2D (Nordhaus et al. 2010, Dolence et al. 2013).
- Not corroborated by similar study from Garching group (Hanke et al. 2012): no significant difference 2D v. 3D.

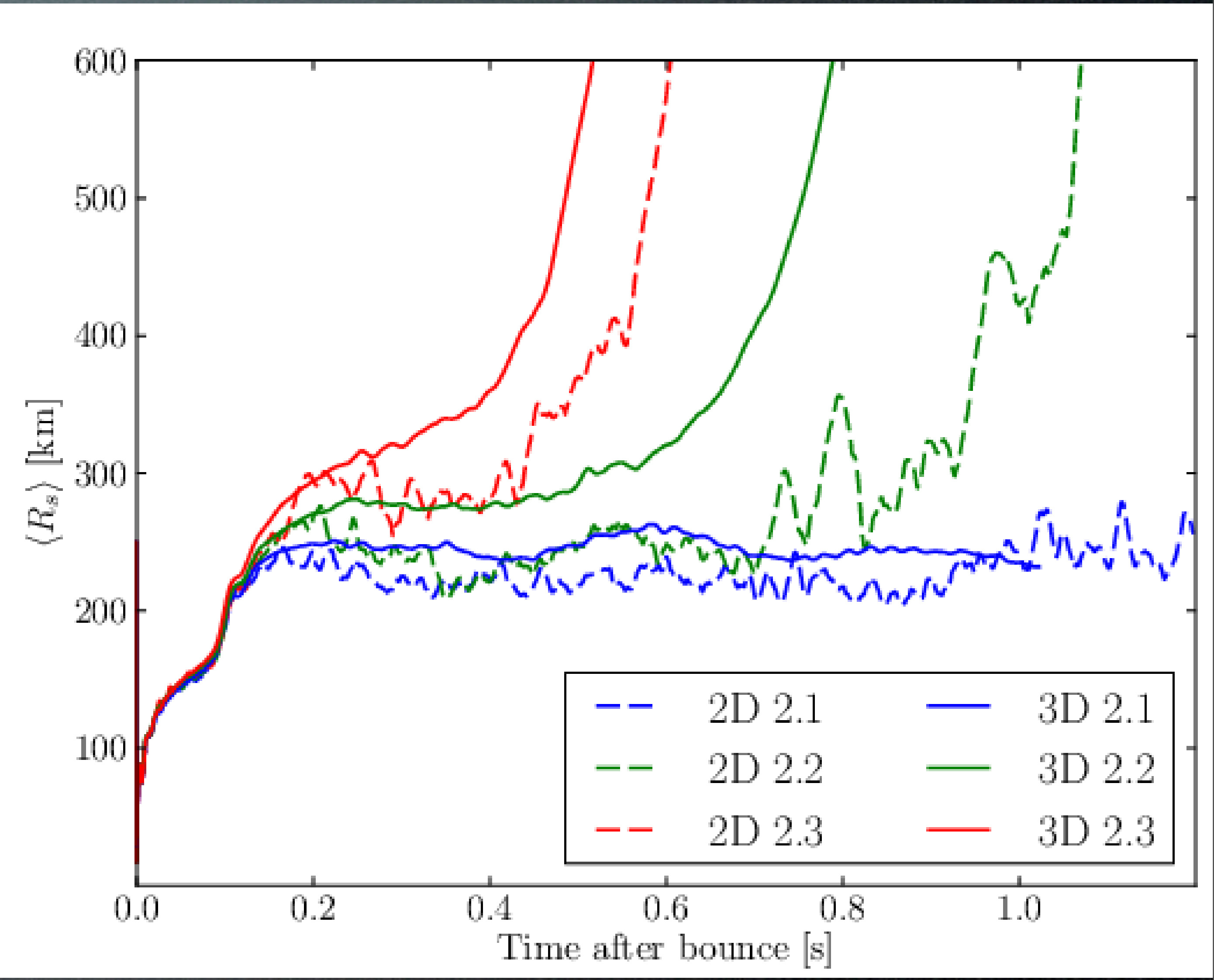
Dolence et al. (2013)





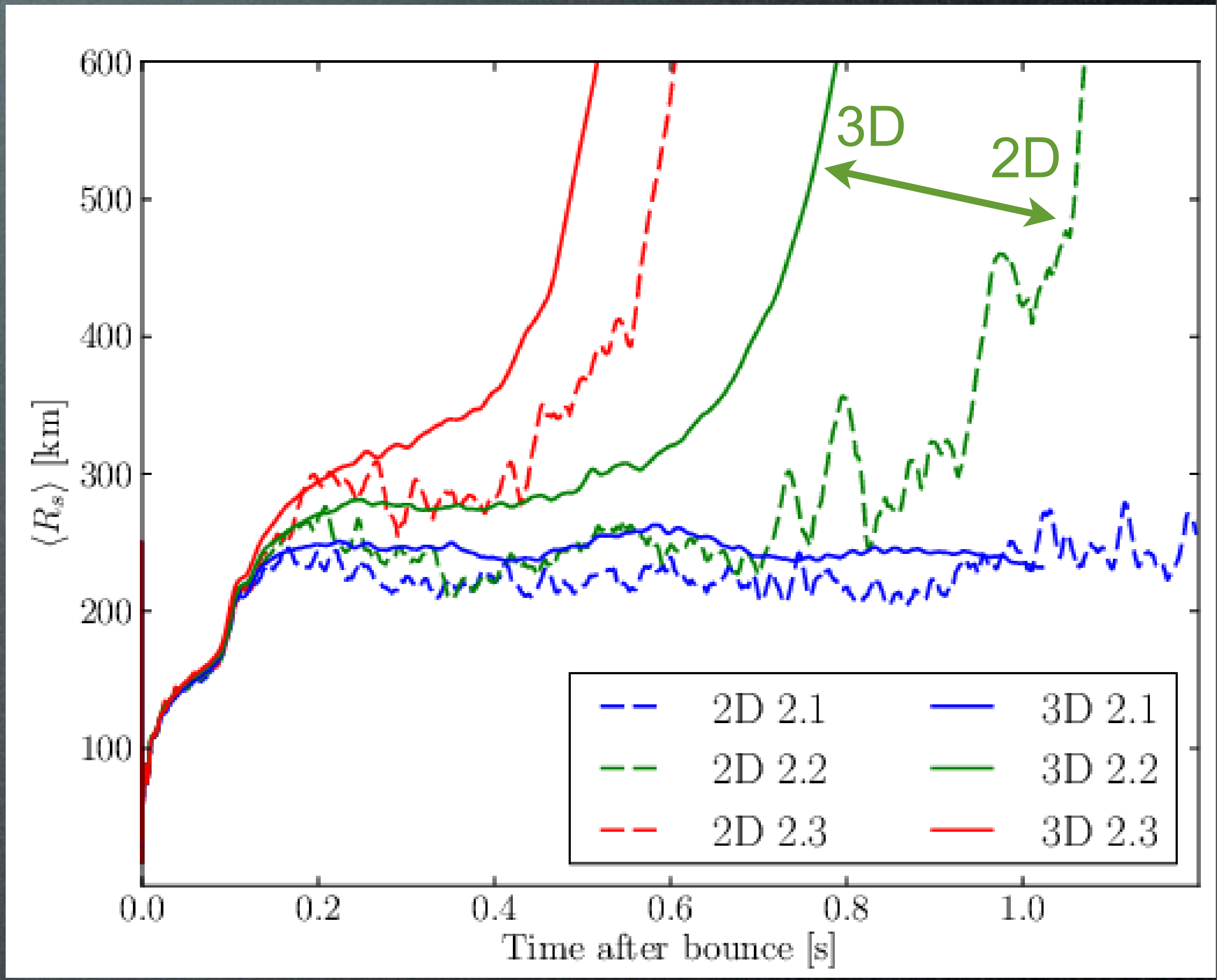
Hanke et al. (2012)

Simple Neutrino Physics Sims



Dolence et al. 2013
(also Nordhaus et al. 2010)

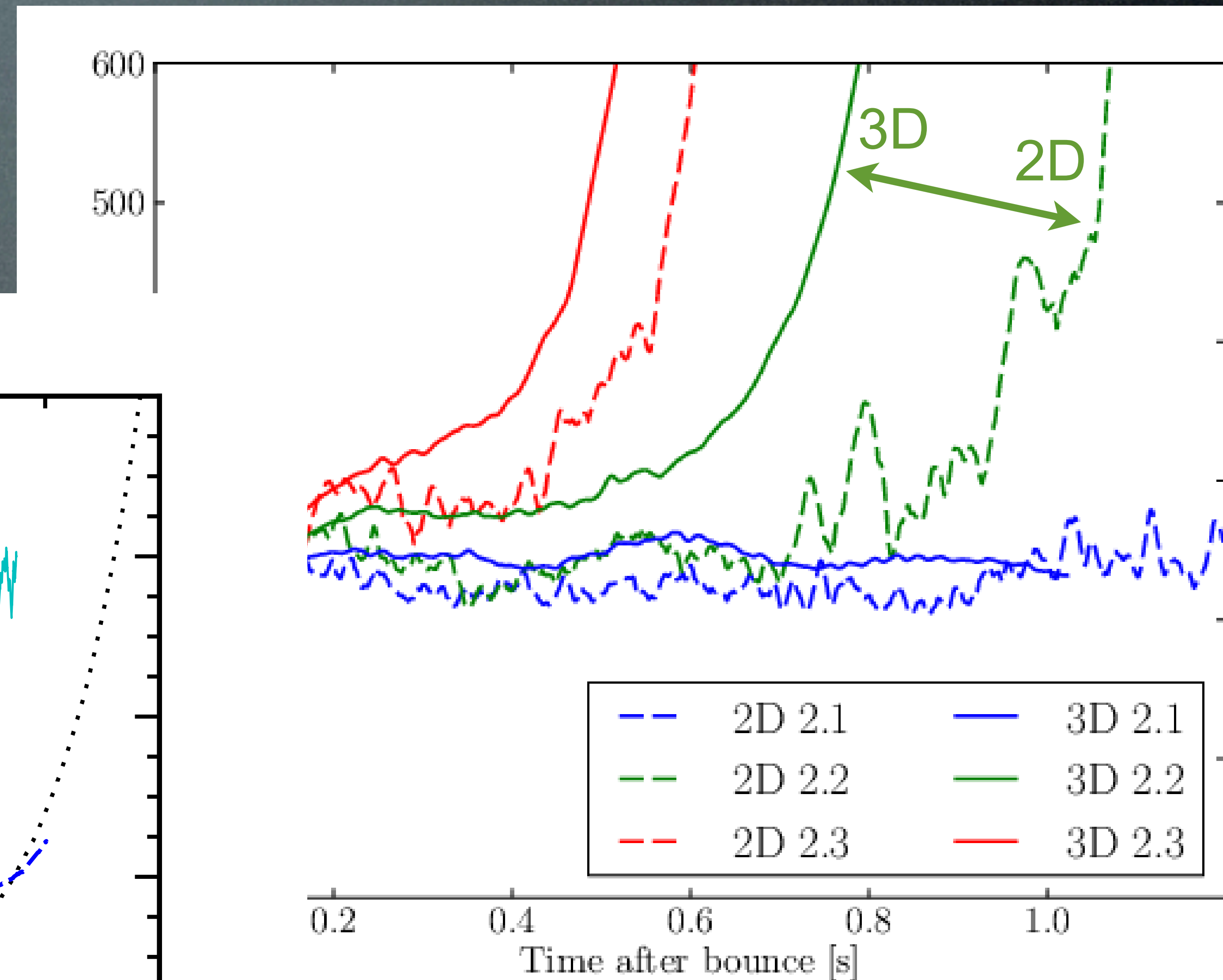
Simple Neutrino Physics Sims



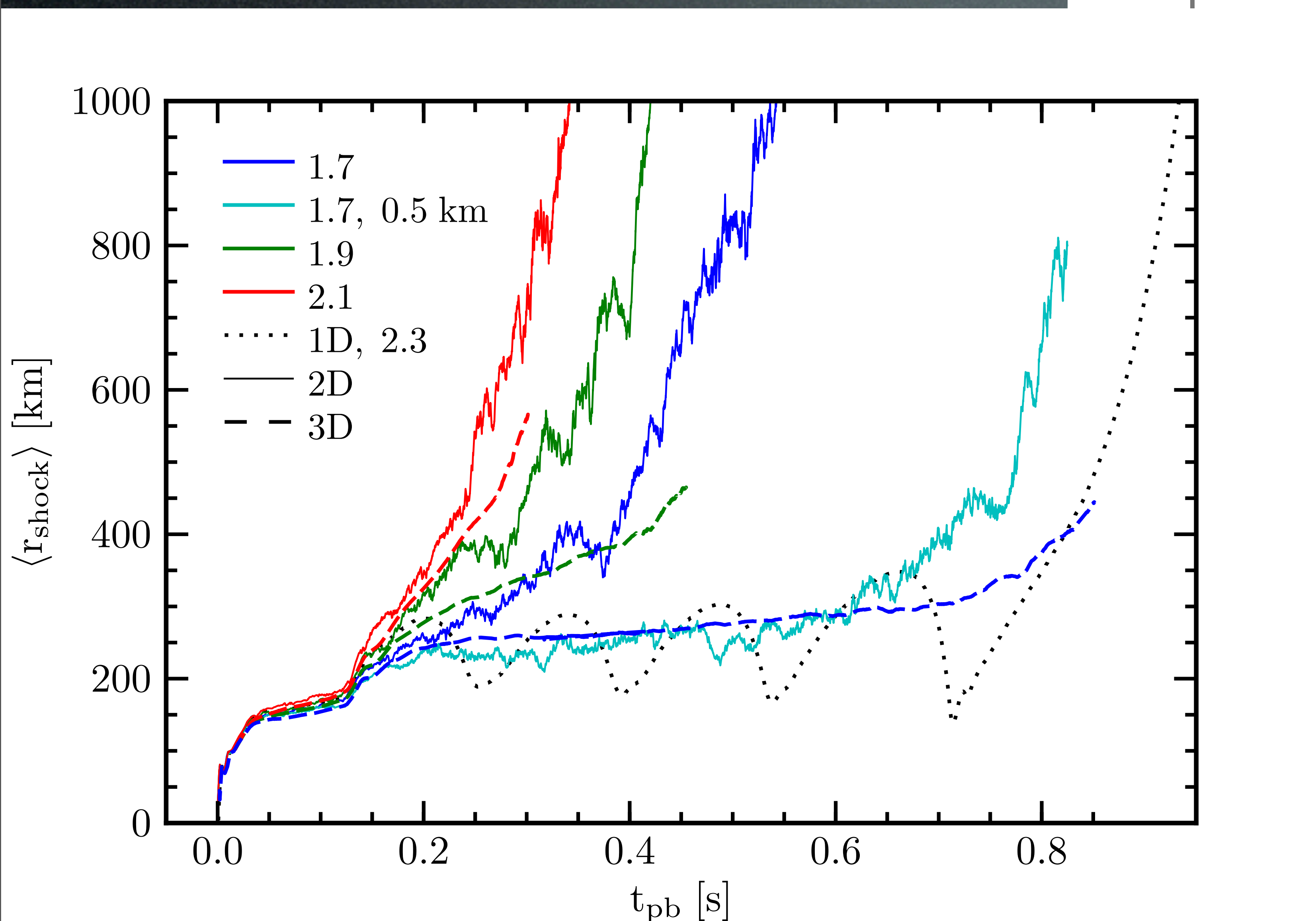
Dolence et al. 2013
(also Nordhaus et al. 2010)

3D explodes faster

Simple Neutrino Physics Sims



Dolence et al. 2013
(also Nordhaus et al. 2010)

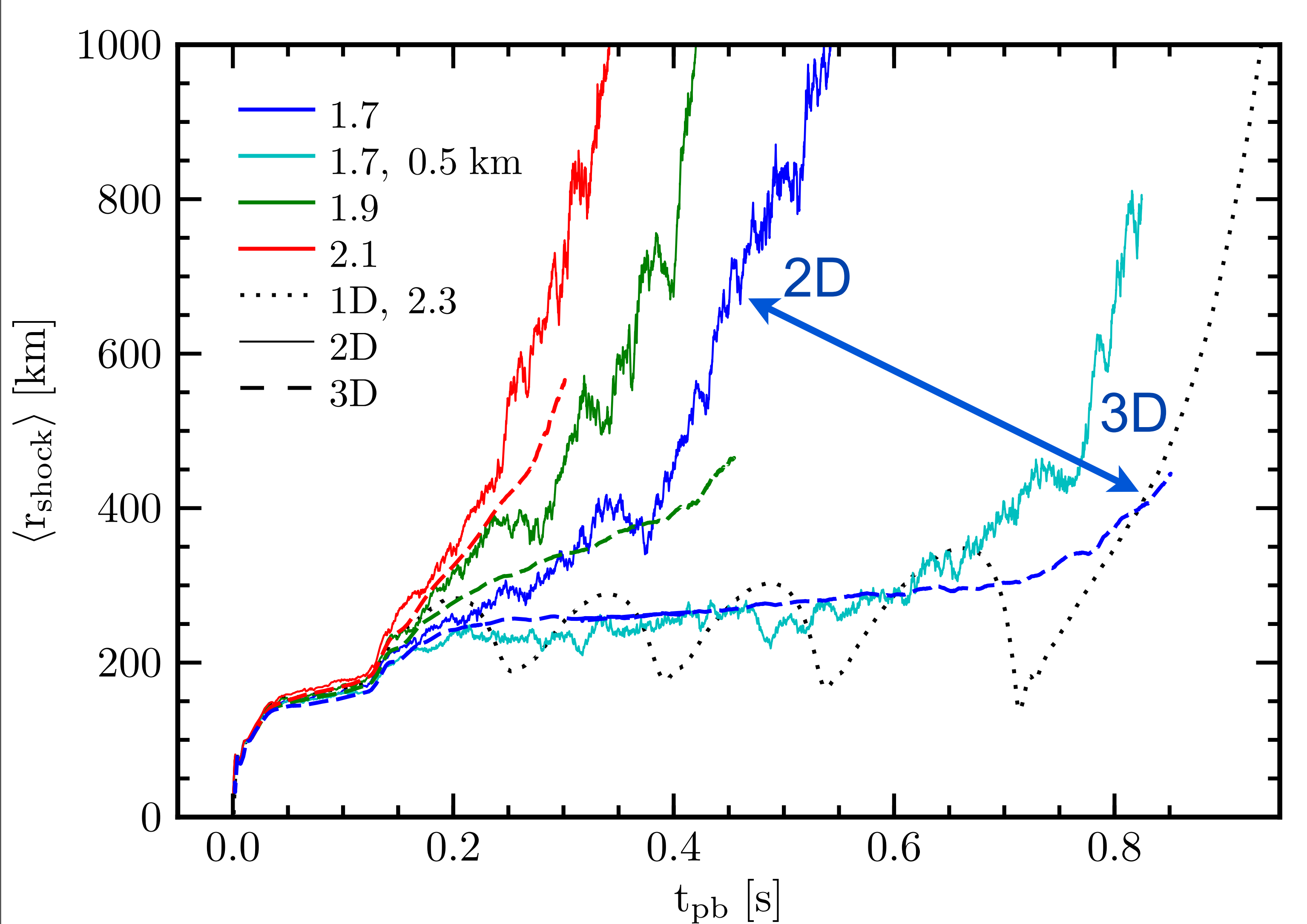


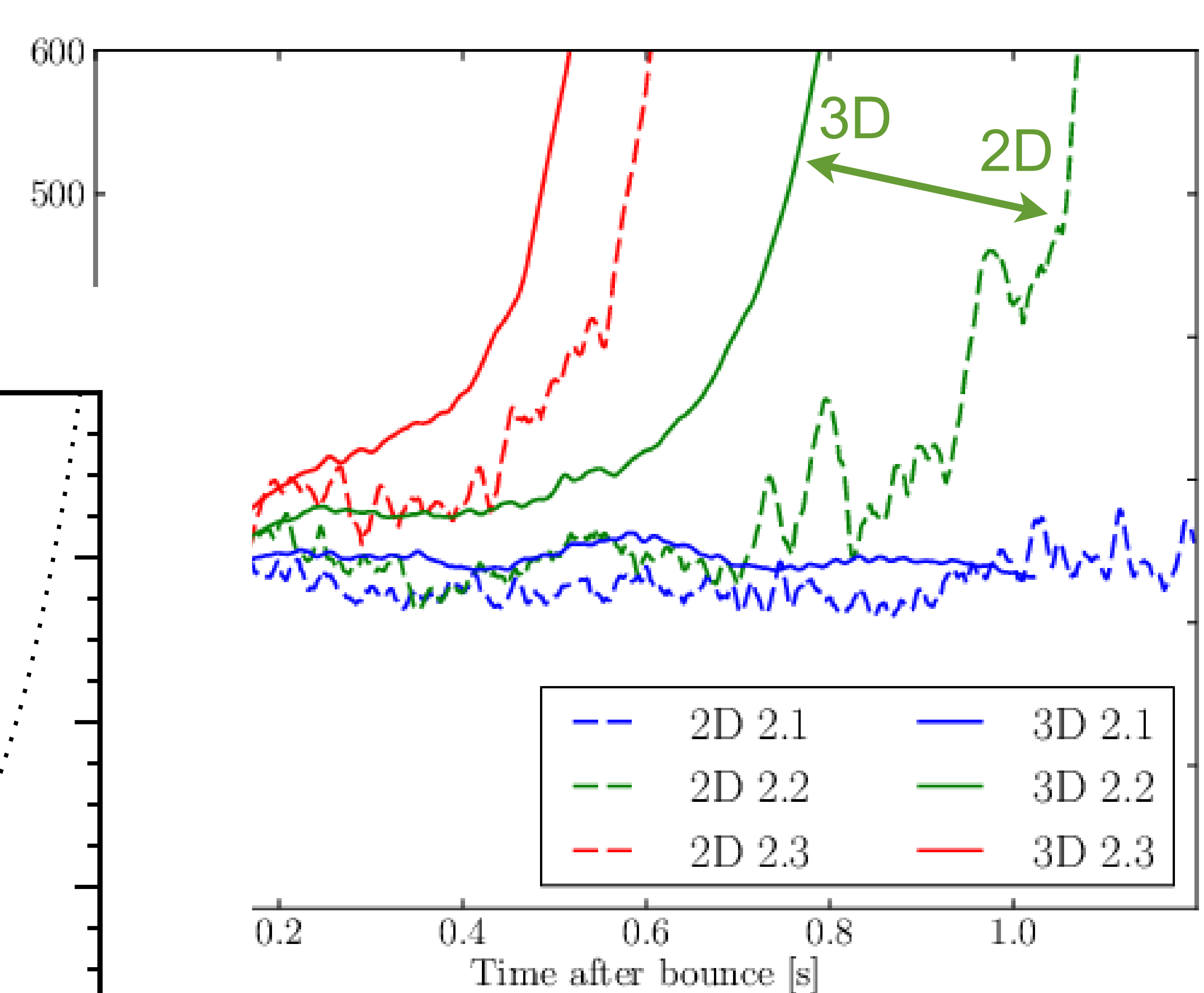
3D explodes faster

SMC, 2012, ApJ in press

Simple Neutrino Physics Sims

2D explodes faster!





Dolence et al. 2013
(also Nordhaus et al. 2010)

3D explodes faster

SMC, 2012, ApJ in press

3D Parametric Sims

SMC, 2012, ApJ in press

3D

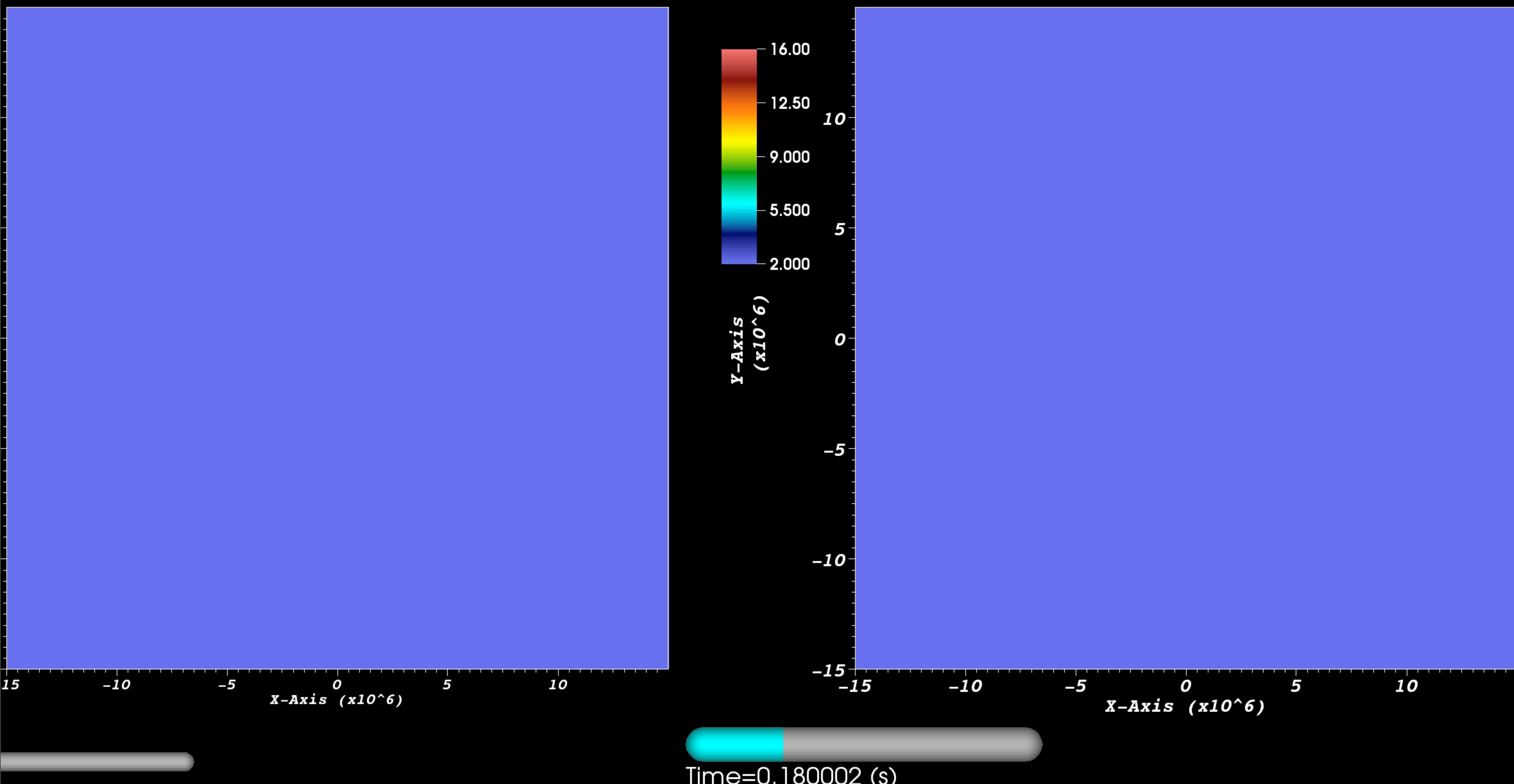
2D

3D Parametric Sims

SMC, 2012, ApJ in press

3D

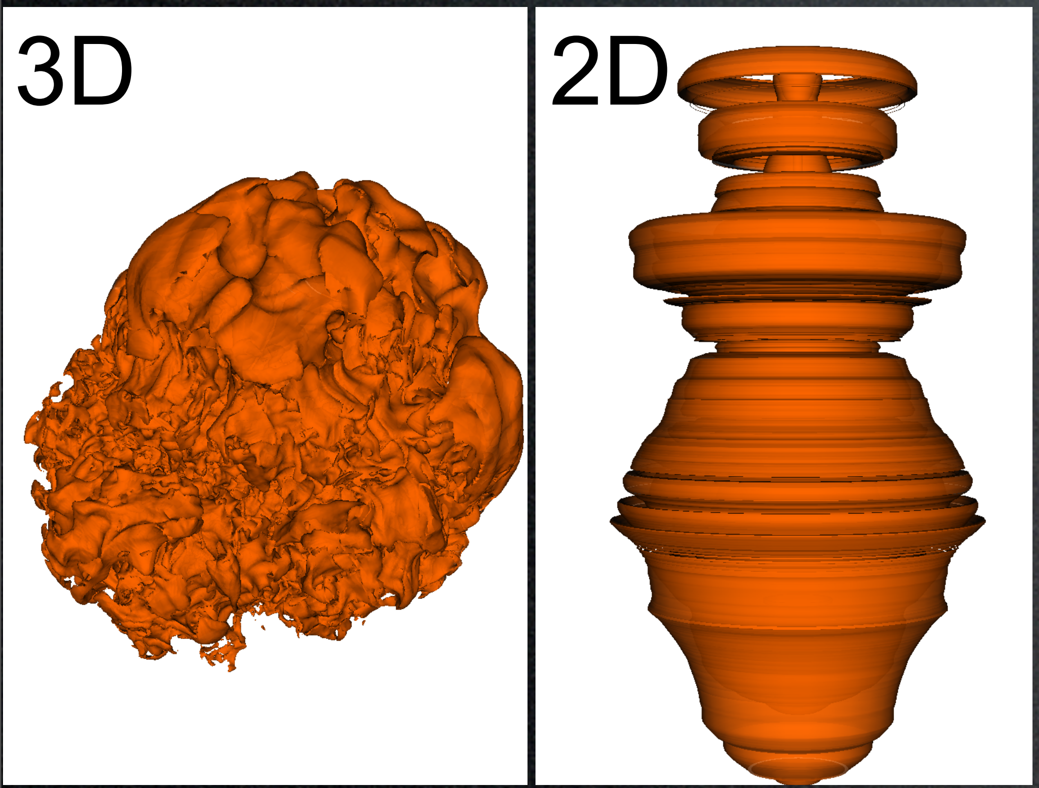
2D



Why does 2D explode?

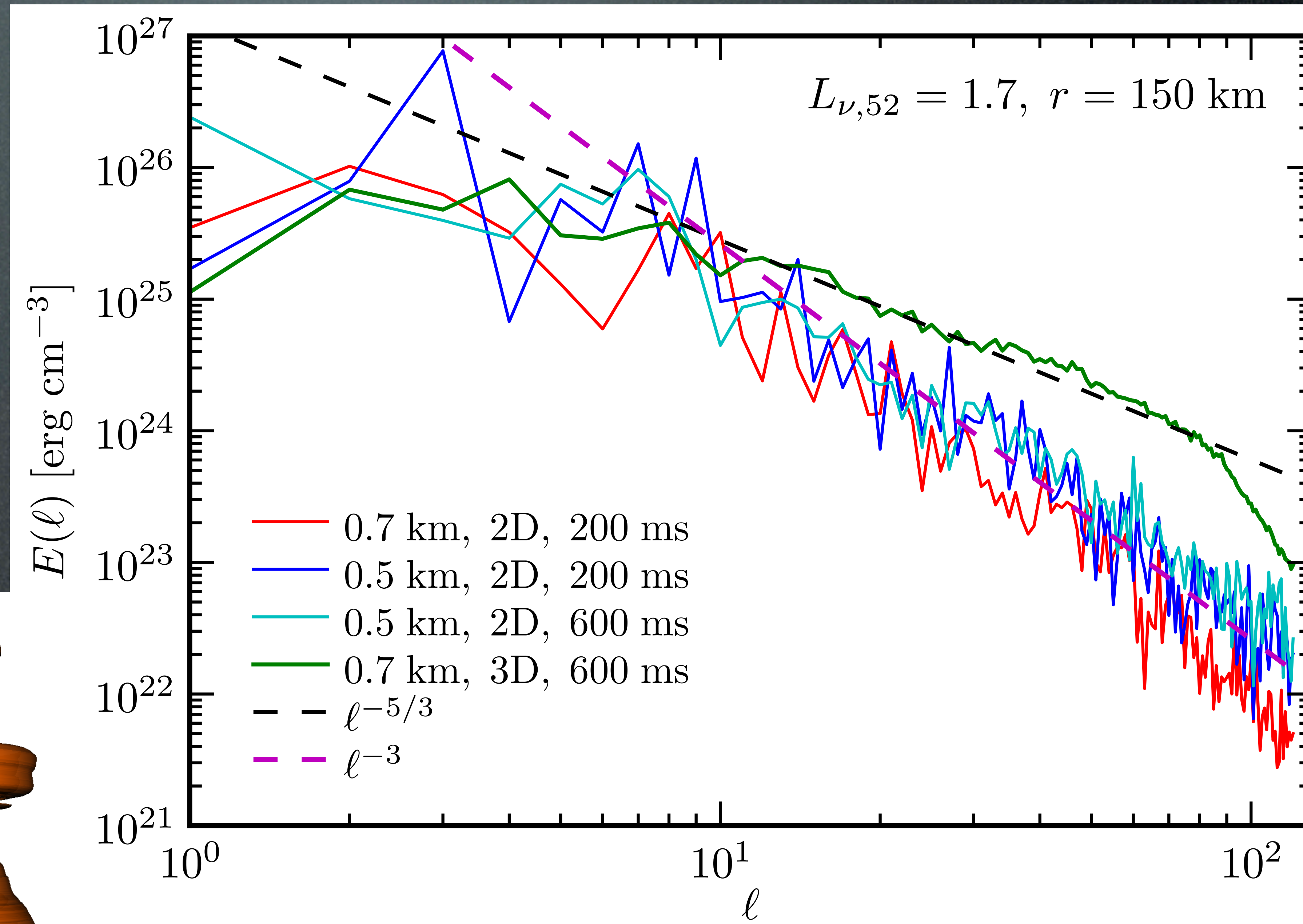
SMC, 2012, ApJ in press

Artificially larger, more buoyant convective plumes in 2D

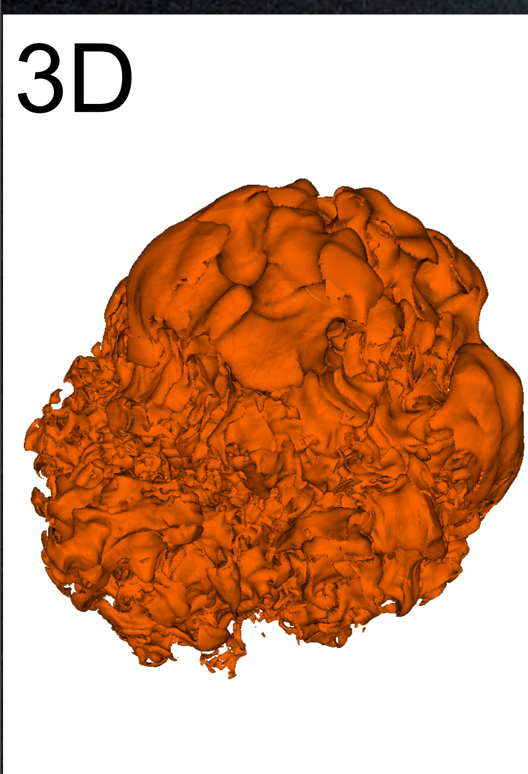


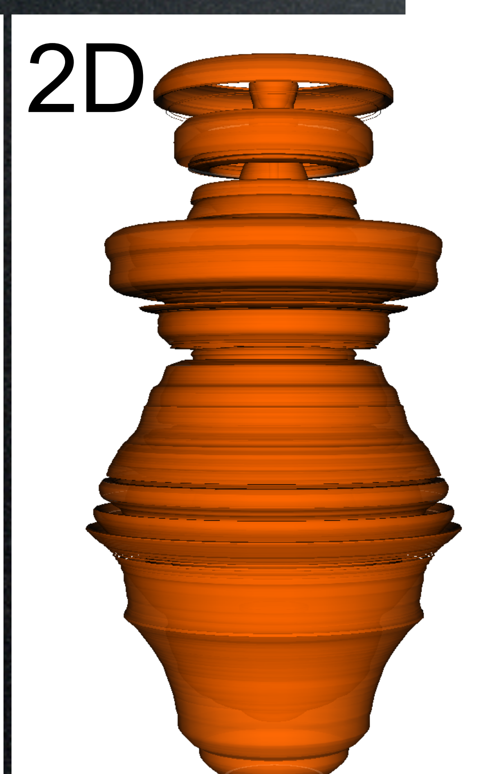
Why does 2D explode?

SMC, 2012, ApJ in press



Artificially larger, more buoyant convective plumes in 2D

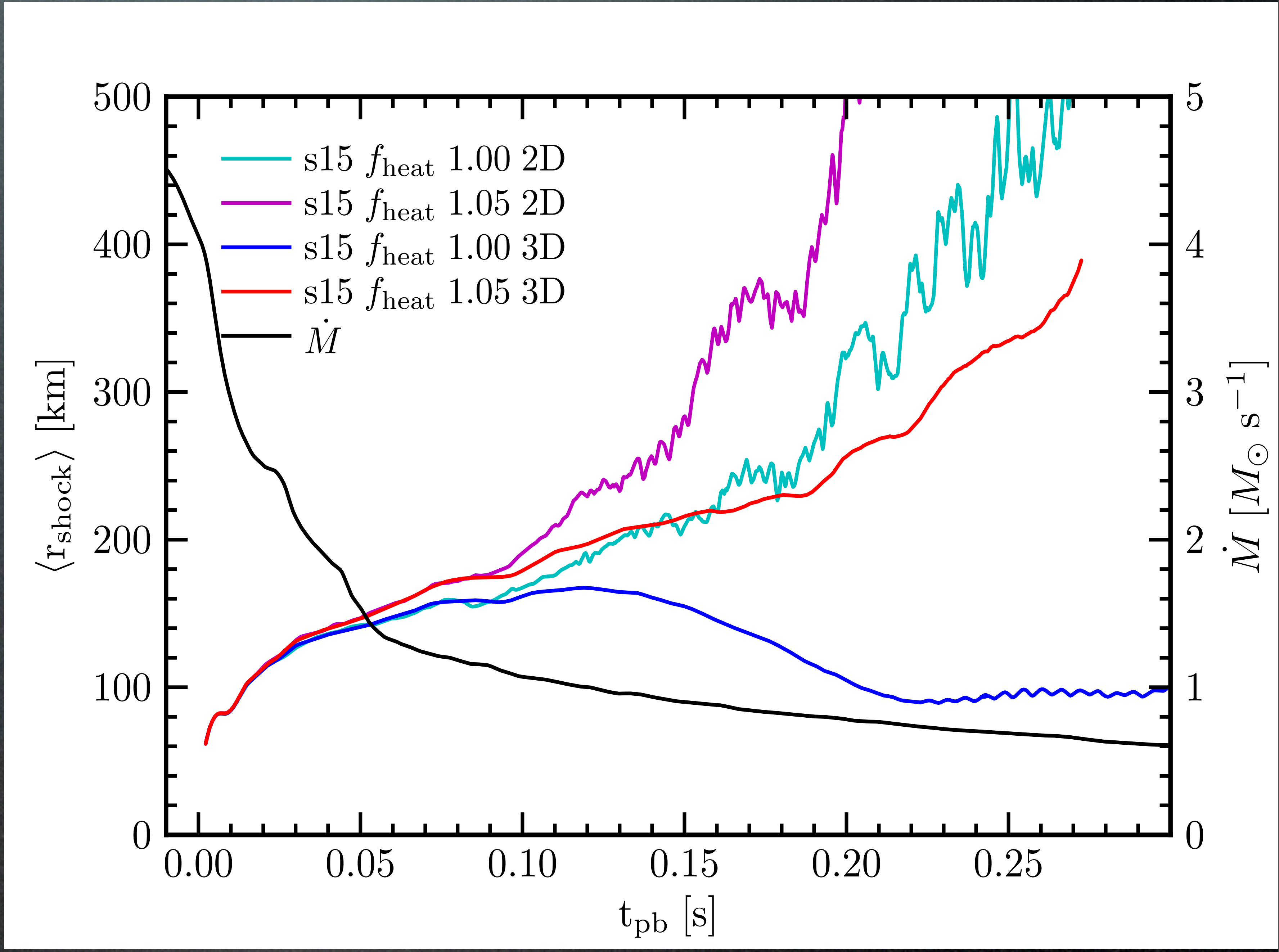




Inverse 2D turbulent cascade

Detailed Physics Sims

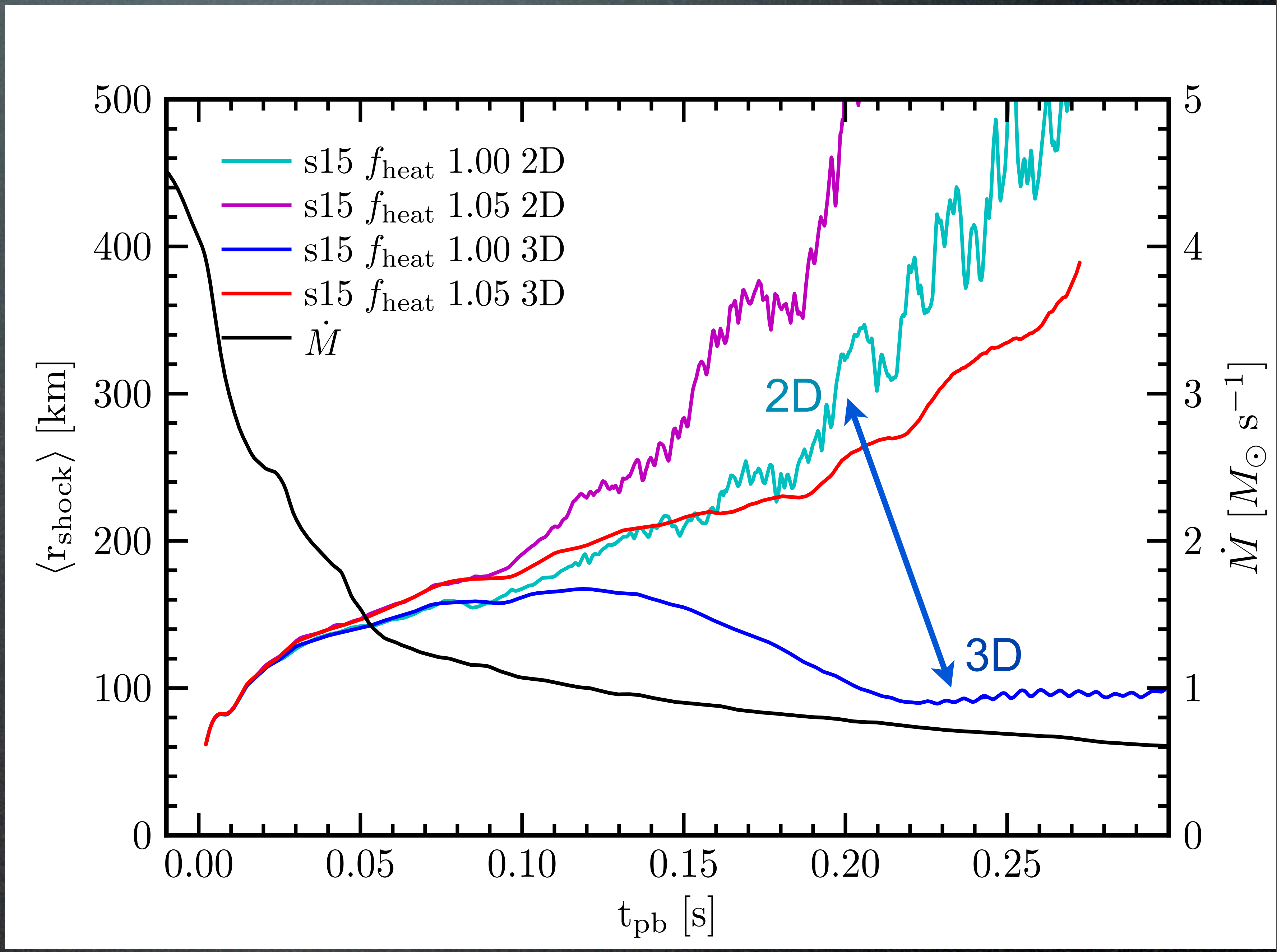
- More sophisticated 'leakage' treatment of neutrinos.
- 2D explodes faster.
- Without additional, artificial heating no explosion in 3D!
- 3D GR sims with same leakage scheme agree (Ott et al. 2013).



SMC & E. O'Connor, in prep.

Detailed Physics Sims

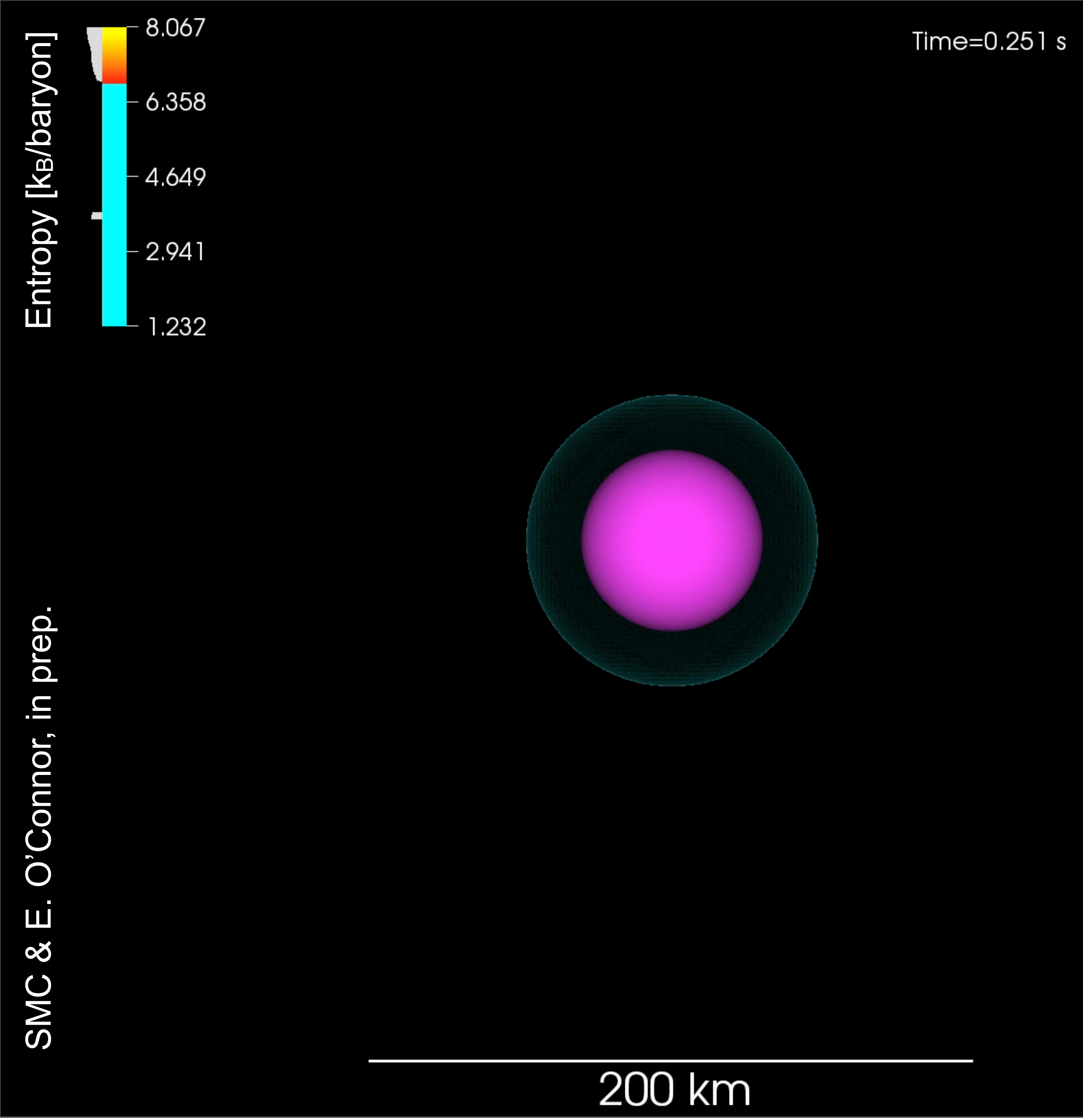
- More sophisticated 'leakage' treatment of neutrinos.
- 2D explodes faster.
- Without additional, artificial heating no explosion in 3D!
- 3D GR sims with same leakage scheme agree (Ott et al. 2013).



SMC & E. O'Connor, in prep.

Entropy [k_B/baryon]

SMC & E. O'Connor, in prep.



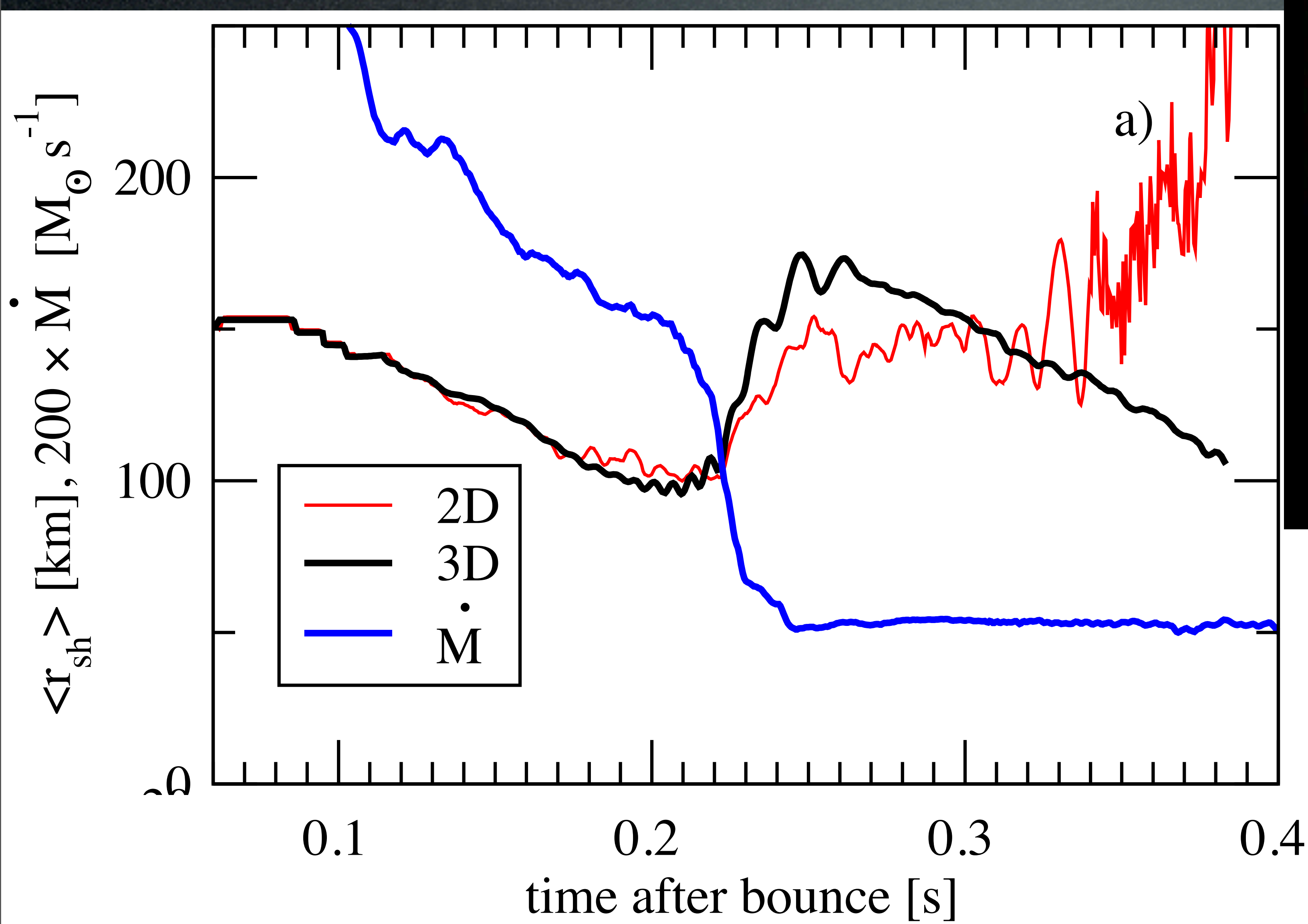
Strong
Convection
Early

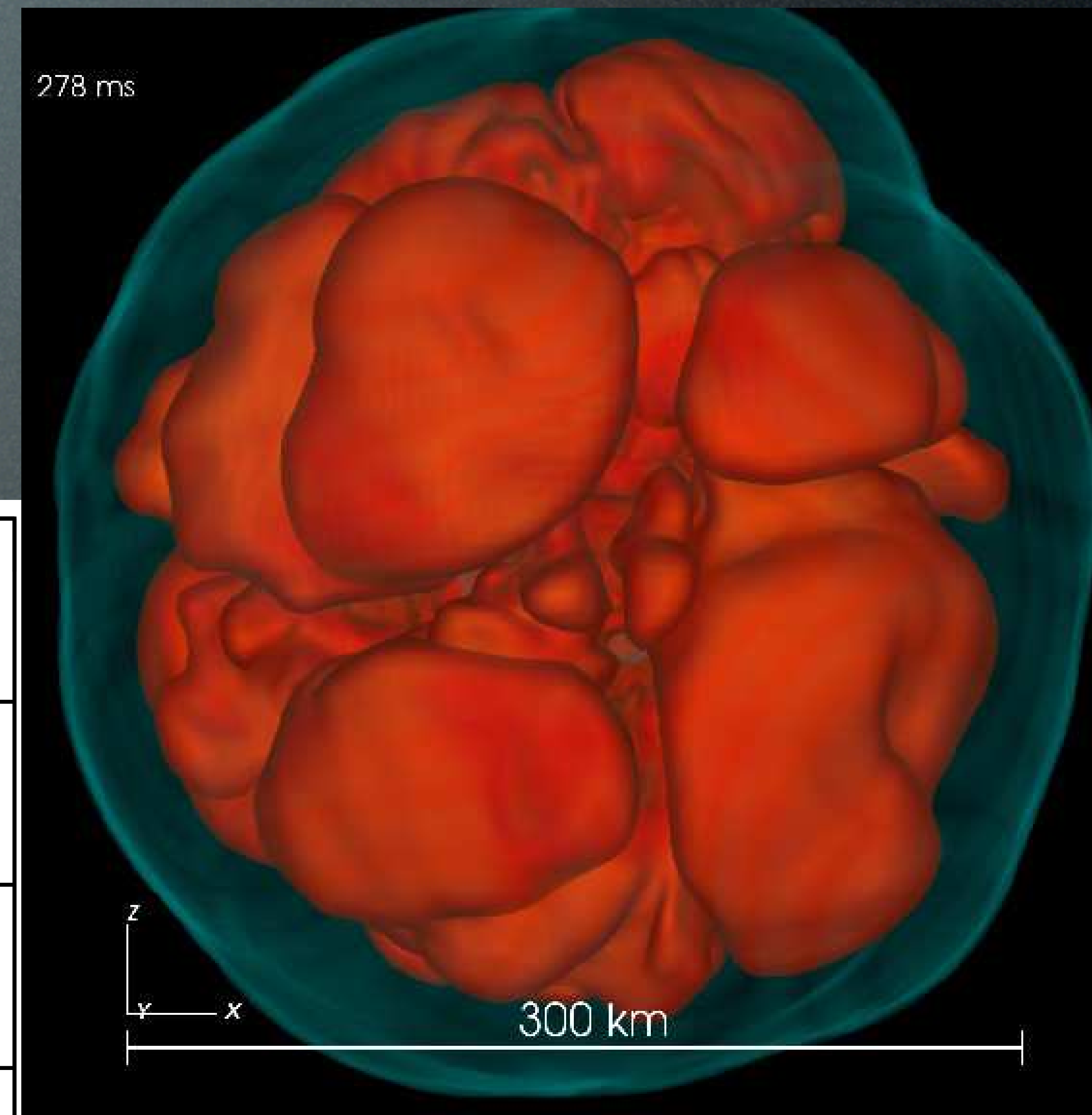
Sloshing
"SASI"

Spiral
"SASI"

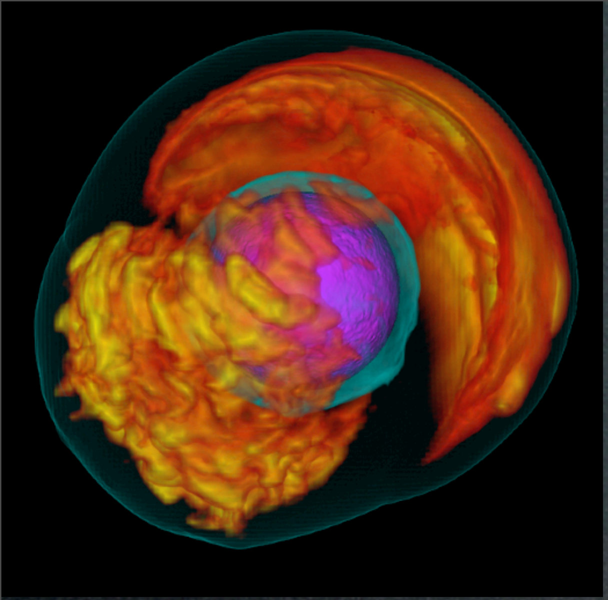
Full Transport Sims Agree

2D explodes, but
3D does not.

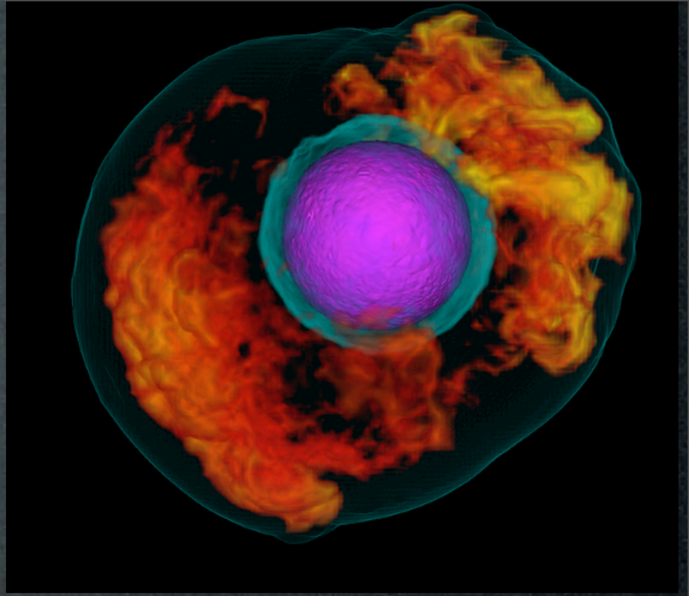




Hanke et al. 2013



Conclusions



- Results indicate explosions are artificially easy in 2D.
- 3D alone may NOT be the key to robust neutrino-driven explosions.
- But...Real CCSNe are 3D and explode all the time! CCSN mechanism must be studied in 3D.
- Results imply that we are missing key physics (or getting the physics wrong. Resolution?).
- Possible missing physics: rotation and magnetic fields, realistic 3D progenitor structure.

Some Things Learned

- For me, HPC is about doing *science.*
- "Better is the enemy of good (enough)."
- "Never let code optimization slow you down."
- human vs. computer time: It's all about wall clock time-to-solution.

Some Things Learned

- Be human: use tools!
- grep and adding print statements are not the only ways to debug
- look into, e.g., gdb, valgrind, hpctoolkit
- whenever possible, use a package manager
- use an IDE, such as, e.g., emacs or vi.

Some Things Learned

- (the most?) Important tool: version control!
- A must for code, but good for everything else too (especially papers!).
- Check-in early and often...like voting in Chicago.

Some Things Learned

- TEST YOUR CODE.
- Verification of implementation is crucial
- Validation important, but harder
- Have a test suite

Some Things Learned

- Back your data up!
- On your laptop/workstation, etc.
- and on the compute cluster
- use ALCF’s HPSS

Some Things Learned

- On using a new code or tool: RTFM.

Some Things Learned

- Seriously consider making your code open-source.
- Open-source is analogous to open-science.
- The benefits are demonstrable and many.

Thanks!