Thermonuclear Driven Events

Michael Zingale
(Stony Brook)

- Intersection of advanced computing and nuclear astro—where should it head?
- What are the open science questions?
- What computing resources are required?

Limitations of Current Simulations

- SNe Ia, XRBs, and Nova can be broken down into multiple phases
  - Multiple codes are needed for each of piece of the puzzle
  - Variety is necessity
- Need to work smarter, not just harder
  - Full DNS of these events will always be out of reach (even at the exascale)
  - Algorithm development must be hand-in-hand with hardware improvements.

Status of SNe Ia

- Current successes
  - Chandra models explode now, although parameterized
  - WD mergers: work under restrictive conditions. General is unsettled
  - Sub-Ch: interesting, if anything, only a niche solution.
- What is the progenitor?
  - Observations don't pin down single system
  - Different codes needed for merger vs. convection in Chandra model.
  - Initial conditions determined here—subsequent evolution follows
- Deflagation → detonation transition?
  - Needed to match observations
  - Turbulence likely key, but models now are all parameterized.
- Why the width-luminosity relationship in the lightcurve?
  - Radiative transfer problem
  - Few codes up to the challenge of 3-d RT (none generally available)

Status of XRBs

- Current successes
  - 1-d models can model multiple bursts with extensive networks
  - 2-d shallow water simulations show importance of rotation
  - No 3-d. 2-d vertical simulations still maturing.
- How does convection change the nucleosynthesis?
  - Multi-d problem.
  - Are ashes brought to the surface?
- Is ignition localized to one point? How does it spread?
  - Lateral flame propagation is hard (resolution requirements put it out of reach for full star, 3-d). Sub-grid modeling?
- As with SNe Ia, multiple different codes are needed to answer the full spectrum.

Status of Novae

- Current successes
  - Long history of 1-d models
  - 2-d (and some 3-d) studies over the past decade focus on the role of convection in mixing. Some disagreements, but perhaps explained by algorithmic differences.
  - Not much focus on early evolution in multi-d
- How do we get the needed enrichment?
  - Dredge up?
  - Very difficult to model: C/O enrichment can depend on the preparation of the initial model (how sharp is the interface), and numerical diffusion across that interface—how do we know what's right?
- Modeling onset, accretion, common envelope evolution...
  - Not much multi-dimensional progress
  - Important in providing a full understanding of the event
  - Again, multiple different codes are needed

NP Needs for the Exascale

- Existing codes scale well to $O(10^5)$ processors using hybrid programming techniques (MPI + OpenMP)
- Need to use GPUs at the exascale
  - Portable standards are needed (like OpenACC). We don't want to have to write custom code kernels for each architecture.
  - Rate calculations (RHS, construction of Jacobian elements) a natural candidate for offloading
- Visualization
  - Transition to runtime visualization necessary as datasets exceed 100s of TB.
- Training
  - Summer schools to teach our students algoritms, parallel techniques, high performance computing

Collaborative Code Development

- Modern codes not built by a single investigator
  - Need mechanisms to support interdisciplinary code teams.
  - NP-funded people should not have to focus all their efforts on optimizing for exascale architectures.
  - Need computational scientists / applied mathematicians onboard and involved in the scientific process. Stop reinventing the wheel.
- Community codes
  - No code is perfect—they all make approximations and all have different strengths and weaknesses. Variety is key.
  - Many existing codes apply to problems of interest to NP.
  - Funding for support and maintainence of codes? Documentation lags.
    - Funding typically comes for science results, not writing code
    - Support is time consuming.
  - How do we reward "inactivity" during development period?
    - Code development can take years before first science results

Collaborative Code Development

- Community microphysics
  - Helps comparisons between groups
  - Expertise of code people differs from that of microphysics people
  - Threadsafe, OpenACC, etc. need to be thought about from the start.
- Community Radiation Transfer
  - Strongest link to observations
  - Even basic capability is valuable