My computational astrophysics journey
(before/during/after UBC)

Alex Razoumov
alex.razoumov@westdri.ca
razoumov@gmail.com

PhD with Douglas in 1995-1999

- Thesis: “Calculating the inhomogeneous reionization of the Universe”
- My journey started a little before that
- Computational astrophysics: running **first-principles** numerical simulations to model astrophysical systems, i.e. solving the underlying equations of (magneto) hydrodynamics, gravity, radiative transfer, ...
    - not semi-analytical models
    - not using ML to make up data

Start of the journey

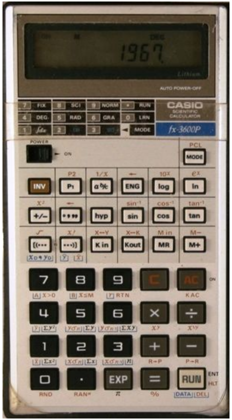
- Always enjoyed calculating things + interest in stars
- First application: spherical astronomy

$$\begin{aligned}\sin h &= \sin\delta\sin\phi + \cos\delta\cos\phi\cos(LST - RA)\\ \sin A\cos h &= -\cos\delta\sin(LST - RA)\end{aligned}$$

  - coolest thing: combined logarithmic trigonometric tables at $1'$ resolution, published around the turn of the 20th century $\Rightarrow$ no need for a calculator

- Went through the observational stage: built 1.5 telescopes, but did not really use them, especially with light pollution in the big city
- Computational: can build ever more sophisticated digital models of existing objects $\Rightarrow$ infinite possibilities

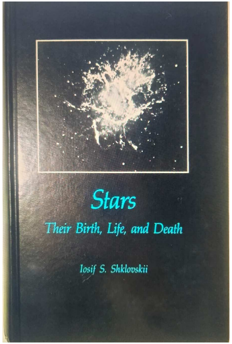
First missing element: hardware



- CASIO FX-3600P programmable calculator: my first serious tool
  - 38 steps of programmable memory
  - a digit key press counts as one step
  - Douglas only other person with the same model
- Mid-1980s behind the iron curtain: no computers in our school, cheapest PC clone ∼2-year salary
- Fall 1986: managed to secure 1 hour / week with on a Yamaha PC clone with BASIC

Second missing element: equations and methods

- Computational astrophysics is all about integrating the differential equations ... not really
- Just consider the forces acting on a body $\Rightarrow$ update acceleration and velocity $\Rightarrow$ trajectory over time
  - couple of examples
  - can apply the same to continuous fluids (equations of conservation)
- Fall of 1986: started building a star formation model
  - spherical symmetry, break cloud into shells, compute forces on each, compute accelerations and velocities
  - another inspiration: Moscow planetarium children's courses taught by university graduate students (Jeans length in grade 7)
  - no clue about the Courant condition $\Rightarrow$ quicky ran into numerical problems (no free lunch)
  - remained puzzled about the radiative losses
- In April 1987 submitted "Numerical modelling of molecular cloud fragmentation and collapse" technical paper



Moscow University: 1988-1994

- Choice between Faculty of physics (astronomy) and Faculty of Computational Mathematics and Cybernetics
  - awesome experience, rigorous math/physics/astronomy training, including PDE theory
  - however, no exposure to formal numerical techniques
  - arrested for walking on the observatory grounds at night $\Rightarrow$ confirmed the suspicions about my observational career prospects + overexposed a set of photographic plates
- 4th-year thesis: extracting energy from a rotating black hole
  - computing photons paths (null geodesics) in the Kerr metric
  - inject a photon into the ergosphere, let it split in two, one of them falls into the BH, the other can be captured at ∞
  - had trouble with accurate integration, in the end switched to processing some observational data

- 6th-year thesis: simulating accretion disks around compact objects
    - wrote my first proper hydro code: approximate Roe solver on a 2D grid
    - in the process learned Lagrangian and Eulerian methods, grid- and particle-based techniques, conservation schemes and Godunov's methods, Riemann solvers, compressible supersonic CFD
    - blew the power supply in my advisor's PC when I plugged it into a 220V outlet
    - spent few nights building an electric transformer from scrap wire
    - still was officially labelled fire hazard $\Rightarrow$ the rest of my thesis computing was done in an unheated roof shack on top of the astronomy institute in December 1993 - January 1994

UBC: early years (1994-1996)

- Did not care about the topic, but wanted to compute
- Started looking into illumination in LMXBs with Jason Auman problem
  - ☞ connected to the previous problem
  - a "perturbed" stellar structure code?
  - a large range of scales: thermal time, rotation time, convection time (hours to months)
  - played with SPH, but ultimately decided to write an implicit hydrodynamics code, hoping to get away with larger timesteps

- My first introduction to large-scale numerical linear algebra
- The need to compute illumination from the compact object brought me to numerical radiative transfer

Computational astrophysics branches

1. Trivial: N-body (simplest possible computational problem)
2. One step up in difficulty: compressible fluid dynamics (well-established field, still many opportunities)
3. Another step: MHD
4. More difficult: radiative transfer and RHD
5. Standalone: numerical relativity, broad range from working in a given metric (dipped my toes in this twice) to solving the Einstein's field equations (0 experience)

$\sim$1996: started thinking how to implement numerical RT in a hydro code

May-June 1996: one month at CITA

$\sim$1997: conversations with Douglas $\Rightarrow$ things fall into place for doing a thesis on reionization

Numerical radiative transfer – why is it so hard?

1. 6D + time
2. Light travels at the speed of light
3. At large τ tight coupling with matter
   - static: incorrect computation of sources and sinks $\Rightarrow$ wrong speed of fronts, e.g. I-fronts
   - scattering (static or moving) $\Rightarrow$ slow convergence
   - moving: aberration, advection, Doppler shift, etc.
4. Problems with parallel decomposition (more on this later)

Solutions circa 1999

- Symmetries to reduce dimensionality
- Invest into better spatial/angular discretization
  - long vs. short characteristics
  - moment methods, Eddington factors with some closure scheme
  - Monte Carlo? (but only if you are lazy)
  - 5D/6D grid-based Boltzmann solvers
  - coupled RHD Riemann solvers for simple standard problems, e.g. spherical radiation-driven explosion, and stitch your solution out of these
- Post-process without hydro
- Best in coupled RHD: write all terms as expansions in $O(v/c)$ and throw away smaller terms
  - pioneered for stellar atmospheres and winds by D. Mihalas in (1+1+1)D
  - repeated for some simpler geometries by other authors
  - ideally need to do the same in (3+2+1)D

$\Rightarrow$ "Calculating the inhomogeneous reionization of the Universe"

static RT solver on a 128^3 Cartesian grid, 3 frequency bands, coupled with a custom time-dependent chemistry solver

1st postdoc: continue to work on RT in galaxy formation context

1999-2003 with Michael Norman at UIUC and UCSD

(1) working to improve my RT solver

(2) putting it into Enzo (block-structured AMR galaxy formation code) ✘ not proper RHD

- Started to work with MPI-parallel codes
- Quickly discovered that parallelization/decomposition is different for local (hydro) vs. long-range (RT) physics
- RT speed was a big issue holding the entire code back (more on this later)

2nd postdoc: neutrino transport in core-collapse supernovae

2003-2006 postoc with Anthony Mezzacappa (and the Terrascale Supernova Initiative) at ORNL

- Neutrino transport ($\rho \gtrsim 10^{14}$g/cm^3) in GenASiS (General Astrophysical Simulation System)
  - cell-based AMR, F90/95, 6D coupled special relativistic MRHD with self-gravity and full Boltzmann transport
  - distributed MPI code with some PGAS features for storing neutrino–matter interaction coefficients
  - PETSc for linear solvers
- Responsible for:
  - neutrino emission and absorption
  - neutrino-matter scattering
  - neutrino-neutrino pair interaction (and neutrino flavour oscillations)
  - mutli-dimensional neutrino tables implemented with Global Arrays
  - Riemann hydro solver for the nuclear equation of state
- Wonderful team of 5-6 people, but this was really a monumental project requiring hundreds of man-years
- Hude code: the graph of functions calls (their names, arguments, and arrows) was several dozen pages long

Toward five-dimensional core-collapse supernova simulations

C Y Cardall[1,2], A O Razoumov[1,2,3], E Endeve[1,2,3], E J Lentz[1,2,3], and A Mezzacappa[1]

[1] Physics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6354, USA
[2] Department of Physics and Astronomy, University of Tennessee, Knoxville, TN 37996-1200, USA
[3] Joint Institute for Heavy Ion Research, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6374, USA

Abstract. The computational difficulty of six-dimensional neutrino radiation hydrodynamics has spawned a variety of approximations, provoking a long history of uncertainty in the core-collapse supernova explosion mechanism. Under the auspices of the Terascale Supernova Initiative, we are honoring the physical complexity of supernovae by meeting the computational challenge head-on, undertaking the development of a new adaptive mesh refinement code for self-gravitating, six-dimensional neutrino radiation magnetohydrodynamics. This code—called *GenASiS,* for *Gen*eral *A*strophysical *Si*mulation *S*ystem—is designed for modularity and extensibility of the physics. Presently in use or under development are capabilities for Newtonian self-gravity, Newtonian and special relativistic magnetohydrodynamics (with 'realistic' equation of state), and special relativistic energy- and angle-dependent neutrino transport—including full treatment of the energy and angle dependence of scattering and pair interactions.

First side project: 2D semi-analytic models of γ-ray bursts

- Initially developed as a toy test problem for GenASiS, took a life of its own
- Several astrophysics scenarios: core collapse with rotation, merger of two neutron stars, other mergers
- In all cases formation of a rotationally supported, short-lived, very dense disk
- Static hydro
- Full neutrino physics (relying on experimentally measured coefficients $\Rightarrow$ no analytic solution)
- Models were built as a function of 2 parameters: total disk mass and the accretion rate

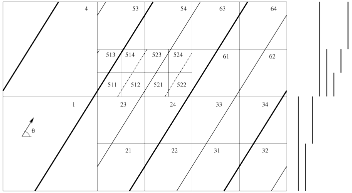
Second side project: fully-threaded transport engine (FTTE)

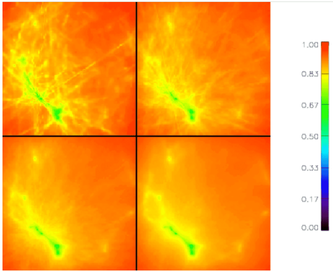
2005-2006 full solution to the standalone RTE, focusing on performance

- Two independent solvers: "diffuse" (continuous sources) and point-source – both implemented in object-oriented F90

(1) Diffuse solver:

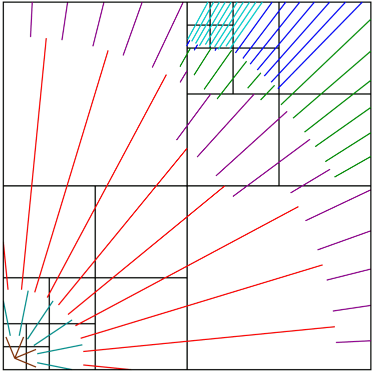
- 3D multi-resolution Cartesian mesh, cell-based AMR, long characteristics starting from the domain boundary and going recursively into higher-resolution cells
- all geometry pre-computed ahead of time, once per angle per 2D layer per resolution level
- all ray elements linked via F90 pointers from its host cell, and vice versa
- during actual transfer follow the interconnected data structures of cells and ray segments precomputed for a particular (θ,ϕ) $\Rightarrow$ fully "threaded"
- zero angular diffusion, HEALPix for angular discretization, fully conservative
- state-of-the-art implementation: benchmarked on scattering in stellar atmospheres, benchmarked against Intel's ray tracing packages (on much simplified physics) $\Rightarrow$ very competitive performance





(2) Point-source solver:

- uses the same multi-resolution Cartesian mesh
- rays start from points, each splitting hierarchically into four rays as a function of distance and local grid resolution, to satisfy the *min number of ray segments per cell*
- HEALPix for angular discretization, fully conservative
- sources grouped into hierarchical trees $\Rightarrow$ can scale up to tens of thousands of sources on a single CPU
- parallelization part is difficult, as it depends on the underlying problem: standalone RT with a very large grid and a few sources, very large number of sources, coupled CFD + RT



- Used both in a series of papers with Jesper Sommer-Larsen and Peter Laursen (Copenhagen)
  - $f_{\rm esc}$ of ionizing photons in early galaxies
  - Lyα radiative transport
- Vastly better than any Monte Carlo RT (faster, more accurate)
- Also forked it to do [my own 3D visualization] (all coded in F90)

- Numerical 3D radiative transfer is largely solved ... but the solution needs to be tailored to each specific problem
- F90 implementation (2005-2006) of the “duffuse” part is at `https://github.com/razoumov/radiativeTransfer.git`
  - assumes very specific memory storage format for nested cells $\Rightarrow$ likely cannot be used directly in someone else’s code
- General 3D RHD is still an unsolved problem
  1. discretized problem size: likely 6D (and not 5D) for coupled equations
  2. source/sink “slider”
  3. all those pesky $O(v/c)$ terms
  4. parallelization

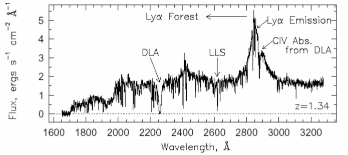
3rd postdoc: free to work on any project

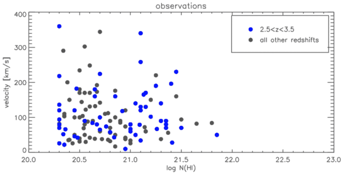
2006-2009 at the ICA at Saint Mary's University in Halifax

1. Radiative transport in early galaxies with J. Sommer-Larsen and P. Laursen
2. Damped Lyα absorbers (DLAs): trying to explain large observed gas velocity dispersion
3. Galactic disks with star formation and feedback (related to the previous problem)

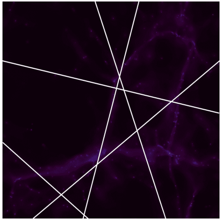
DLAs

- Lyα forest: $N(HI) < 10^{17}\mathrm{cm}^{-2}$, $\delta\rho/\rho \sim 10$
- Lyman-limit systems: $10^{17}\mathrm{cm}^{-2} < N(HI) < 2\times10^{20}\mathrm{cm}^{-2}$, $\delta\rho/\rho \sim 100$
- Damped Lyα absorbers (DLAs): $N(HI) > 2\times10^{20}\mathrm{cm}^{-2}$, $\delta\rho/\rho \gg 100$
  - probe neutral interstellar medium in forming proto-galaxies
  - dominant reservoir of neutral gas at high redshifts

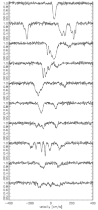


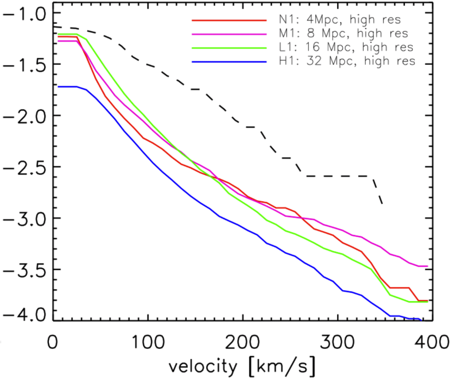


1. run a big simulation
2. draw $\sim 10^6$ random lines of sight
3. calculate line profiles in the velocity space
4. add S/N=20 for each 1 km/s pixel
5. line widths from 90% optical depth
6. multiple components within 400 km/s considered a single line



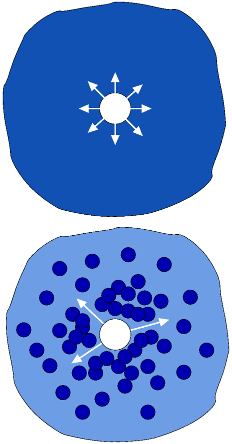
$\Rightarrow$





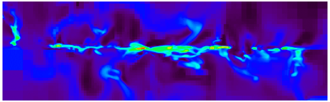
Models fail to explain large observed gas velocity dispersion ...

Injection of 1.8×10^{50}erg/yr/$10^{12}\,M_\odot$ into cloud kinetic energy would provide the right velocity dispersion

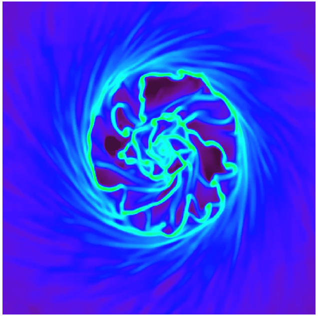


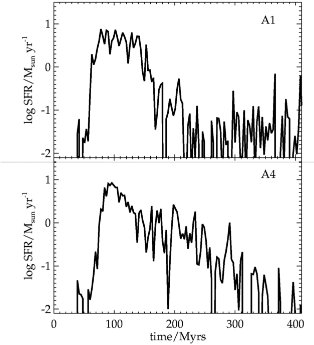
Problem: at low numerical resolution energy spread over too large a mass
$\Rightarrow$ temperature too low, cooling probably overestimated

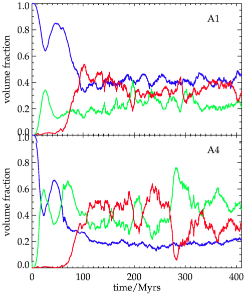
How to grow hot bubbles in the interstellar medium



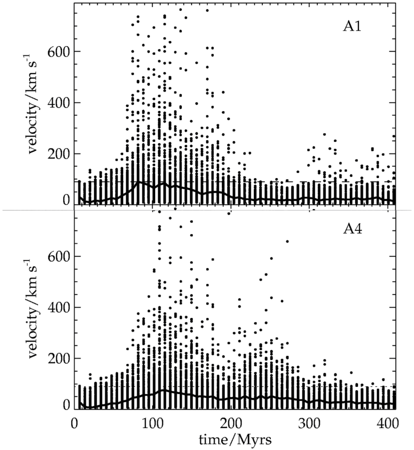
1. Heating must dominate over cooling
2. In hot regions pressure must overcome self-gravity
3. Single SN explosion must heat gas to $\sim 10^{7.5}$ K
4. Need high density contrast to preserve some of ongoing star formation
5. Need to use a sufficiently high density star formation threshold to decouple SF rates from the exact prescription ($n_{\mathrm{H}} \geq 6500\,\mathrm{cm}^{-3}$)



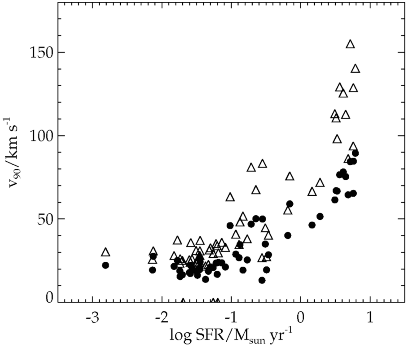




Volume fraction of cold, warm, hot phases in the galactic midplane



Most HI absorption comes from within ∼1kpc above/below the disk



Good velocity dispersion, but somewhat inconsistent with very low observed star formation rates in DLAs

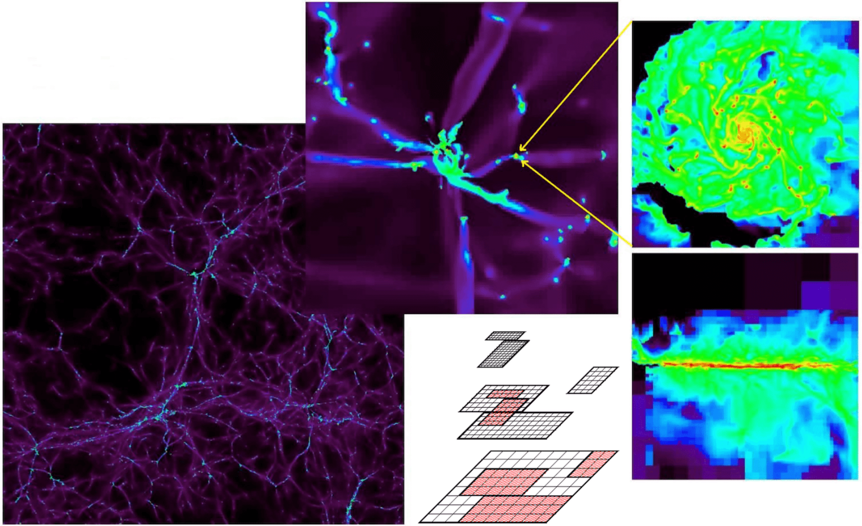
Range of scales

Cosmological simulations

- 50 pc spatial resolution ⇒ no reasonable outflows
- subgrid multiphase models: good for self-regulation of SF/feedback but do not separate phases dynamically

ISM models in isolated galaxies

- what spatial resolution is needed to resolve multiphase ISM in which thermal feedback would launch realistic winds – probably 10 pc or better
- need to resolve feedback from individual SNe



Dynamic range 10 Mpc - 10 pc

2009-2014

- HPC Consultant with SHARCNET, based at UOIT (now Ontario Tech University) near Toronto
- Working on programming projects, HPC problem tickets, general support in multiple domains, and training in HPC
- Organized biweekly in-person scientific computing seminars in the Faculty of Science

2014-present

- Working for WestGrid / Compute Canada / SFU – really the same evolving position
  - initially hired as a visualization specialist, quickly expanded to cover broader training and support
- Now with the Research Computing Group at SFU
  1. leading a 2-person training team at SFU (Marie-Hélène Burle and myself), providing research computing (RC) and HPC training and support in Western Canada
  2. responsible for sci-vis support across the country, leading the National Visualization Team since 2014
- Teaching ∼90 events per year
  - biweekly webinars (since 2015)
  - weekly online courses (since 2023)
  - week-long in-person summer/winter/etc schools (since 2017)
  - various national series and invited schools, local and one-off workshops, bootcamps, hackathons, etc.
- All our training is free to academic researchers in Canada
- Teaching ∼20 full-day courses in RC and HPC
  - sliced and offered in different formats
  - all levels from beginner's command line to large-scale parallel programming and niche topics
  - constantly developing new materials
  - limited bandwidth $\Rightarrow$ do not teach domain-specific computing, typically cycling through courses throughout an academic year, always struggling between depth and the number of topics

Courses

Each title amounts to ∼1 day of materials; more details at `https://training.westdri.ca`

Remote computing basics

- Bash command line
- Introduction to HPC

Programming tools

- Basics of Python
- Basics of R
- Introductory Julia
- Scientific Python

Parallel coding

- Parallel programming in Chapel
- Parallel computing in Julia
- HPC Python
- Speeding up computations with parallel R

(used to teach MPI/OpenMP/GPUs, but no bandwidth)

Virtualization

- Intro to the Alliance cloud and VMs
- Intro to Apptainer containers

Machine learning

- Deep learning with PyTorch
- Deep learning with JAX and Flax

Scientific visualization

- 3D sci-vis with ParaView
- 3D sci-vis with VisIt
- 3D visualization for the humanities
- Remote and large-scale rendering
- Many other shorter visualization topics: in-situ vis, topological data analysis, photorealistic rendering, custom ParaView filters and plugins, many 1D/2D plotting packages, 3D multi-resolution vis with YT, etc.

RDM topics

- Version control with Git
- Large data version control with Datalad, git-annex, DVC
- Managing files and formats: scientific formats and data libraries, parallel I/O, overlays, etc.

- Subscribe to our weekly emails (Sep-May) `https://bit.ly/rcweeklylist`
- Send us topic suggestions
- Doing this as part of RC/HPC training across the Alliance Federation

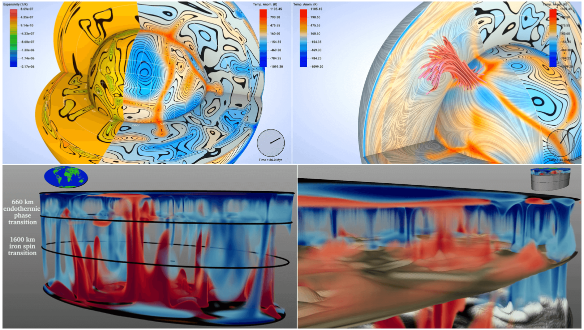
Biweekly webinars

- Focusing on more advanced, well-defined, shorter topics typically not covered in our regular training
  - we use these as test vehicles for new training topics
  - valuable service to researchers, since most have no knowledge of these tools / techniques
  - often covering topics that do not have good documentation $\Rightarrow$ quite research heavy, often requiring hundreds of hours of preparation
  - sometimes quite a scary experience, as these webinars can attract world experts in niche topics
- For upcoming webinars and registration links see `https://training.westdri.ca/blog`
  - fall 2024 scheduled will be published by the end of August
- $\sim$ 150 webinars archived at `https://training.westdri.ca`, both chronologically and by topic
- Currently delivered by a team of two $\Rightarrow$ looking for collaborations

Visualization support and training

Details at `https://ccvis.netlify.app`

- Visualization webinars is just one aspect `https://bit.ly/vispages`
- National Visualization Team since 2014: 5-6 HPC analysts interested in sci-vis
- Vis. queue in the national ticket system
- Centered around ParaView and VTK
  - modern standard for 3D sci-vis
  - open-source, general-purpose, multi-platform
  - scalable to tens of thousands of cores, TBs of data
  - scriptable, remote client-server and batch rendering, in-situ (Catalyst), Cinema science



- Very familiar with many other sci-vis tools as well (VisIt, YT, VMD, ...)
- National visualization contests since 2016 `https://ccvis.netlify.app/contests`
  - provide 1-2 datasets, give some tasks
  - in 2021 chaired the international IEEE Sci-Vis Contest
  - zero budget for prizes, no comms support of any kind, do everything ourselves

Summary

- Shockingly, computational astrophysics was not taught properly at all institutions I worked, even places like the ICA@SMU
  - pretty much the same can be said about teaching numerical methods to physicists
  - would be very happy to develop / teach beginner's courses
- Writing simulation code is easy; making sure it works as projected takes orders of magnitude longer
- If you publish (numerical simulations), your problem is probably too simple
- Standalone 3D radiative transfer is a solved problem: get in touch
  `alex.razoumov@westdri.ca`
- Coupled RHD remains a challenge: again, get in touch