Research topics that I have worked on

Cosmological radio signal: Polarization, foregrounds, detection strategies and prospects with a single dish antenna.

Fortran 95+, C++, C, MATLAB, Healpix, Python, HDF, IDL

Fast independent component analysis, Mueller matrices

Supernova explosions: Radiative transfer and Nucleosynthesis.

Fortran 95+, MPI, OpenMP, LAPACK, MPACK, MKL, debugging tools (gdb), IDL, Torque/PBS

algorithms for linear system of equations through iterative methods, non-linear equation solution techniques for very large systems, many numerical methods

Large scale structure and modified gravity.

Fortran 95+ , MATLAB, PGPLOT

PDE solvers, Markov chain monte carlo, statistical moments

Circular polarization of the CMB : First stars explosions, magnetic field and relativistic plasma.

Fortran 95+, C++, C, MPI, OpenMP

tensor manipulations

Cosmic microwave background (CMB): Linear polarization and magnetic field.

Fortran 95+, C++, Healpix, Python, HDF

tensor manipulations, statistical interpretation of data

Application and scope of HPC in selected problems in astrophysics and cosmology: radiative transfer and radio astronomy

Soma King (De)
University of California, Davis
August 7, 2015

Radiative Transfer: A few definitions

1D problem: 3D geometry, with symmetry along x-y plane. For example, plane parallel (changes only along z axis), spherically symmetric (only radial dependence).

Specific intensity, I_ν : Energy passing per unit time, /projected area, /solid angle, /frequency interval. 6 dimensional quantity (3 spatial, 2 for photon's direction of propagation, 1 frequency).

Average intensity, J_ν : Specific intensity at a given point, integrated over all solid angles.

LTE- local thermal equilibrium, [R] stands for the rate operator and [n], the level population vector.

Interaction with matter and radiation -> **emission** (η_ν) and **absorption** (α_ν) coefficient, also called opacity. α_ν is inverse of the mean free path of photon. The line integral (over physical length) of α_ν is the **optical depth,** τ_ν

Source function, S_ν is the ratio of emission to absorption coefficient.

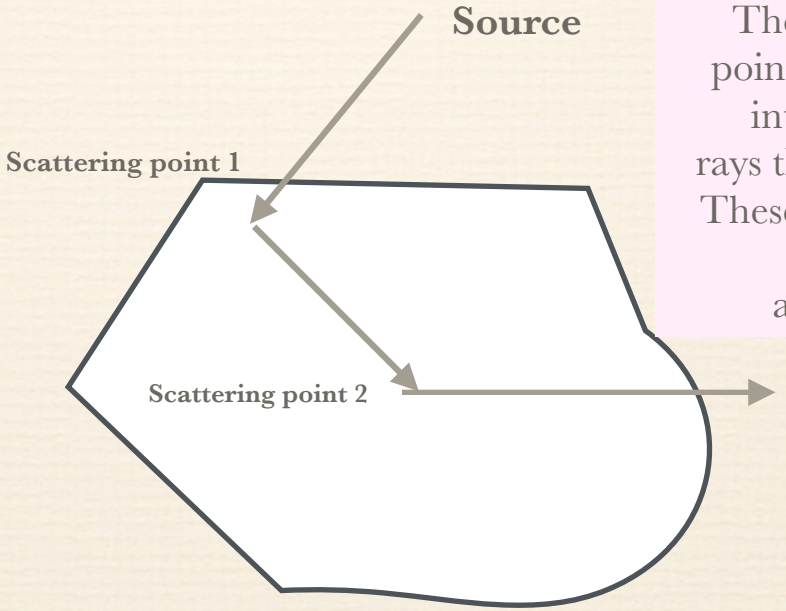
Why Radiative transfer, a HPC worthy problem?

Radiative transfer (hereafter RT) problems deals with the movement of photons by addressing the interaction between matter and radiation.

The purpose of RT models is to match the observed light curves (total light emitted as a function of time) and spectra (light emitted vs frequency), to learn more about the radiation emitting object (supernovae, stars and even planets) and its environment.

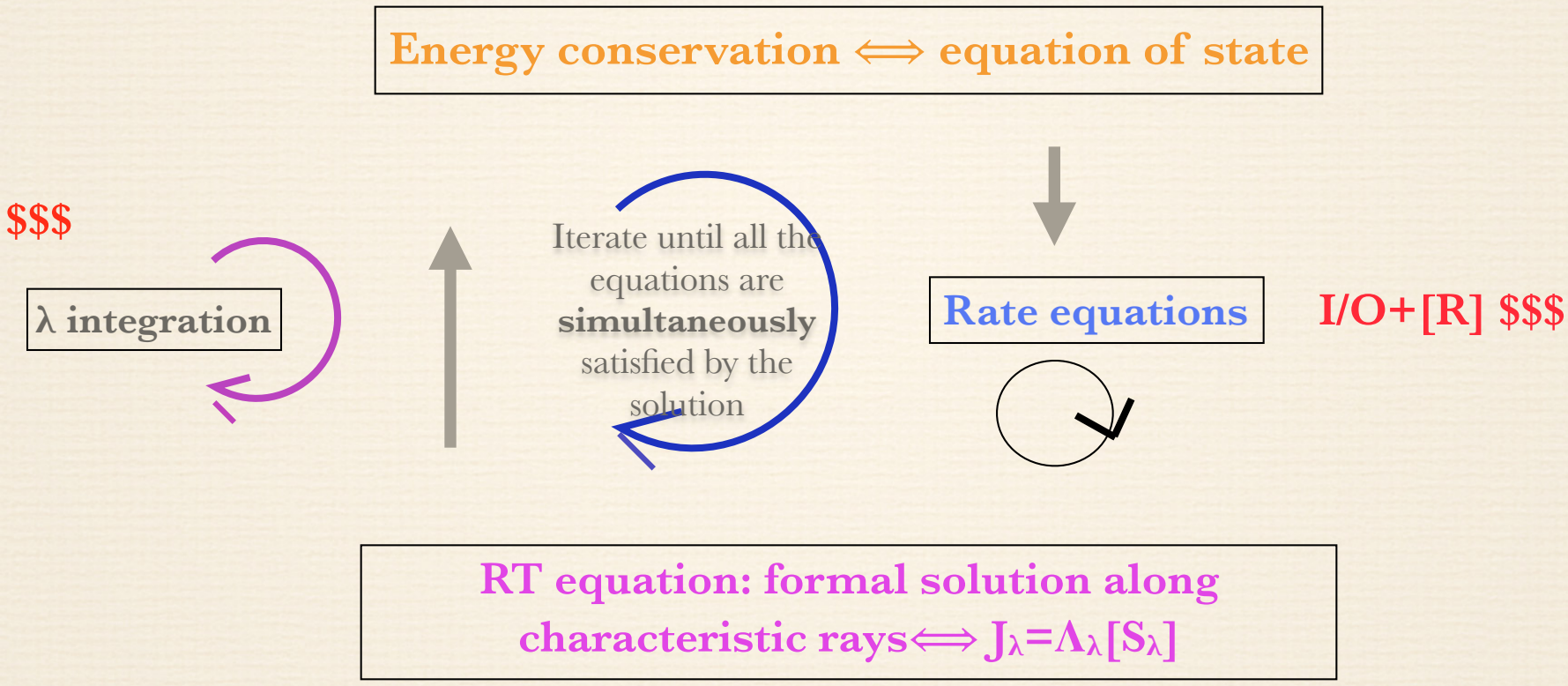
Along a given light ray,

$\mathbf{n}.\nabla I_\nu = \eta_\nu - \alpha_\nu I_\nu = \alpha_\nu (S_\nu - I_\nu)$, S_ν depends on the angle averaged I_ν , leads to a partial integro-differential equation. We will refer to the solution to this equation as the 'formal solution' in the future. It is a boundary value equation is spatial coordinates, initial value problem in frequency.



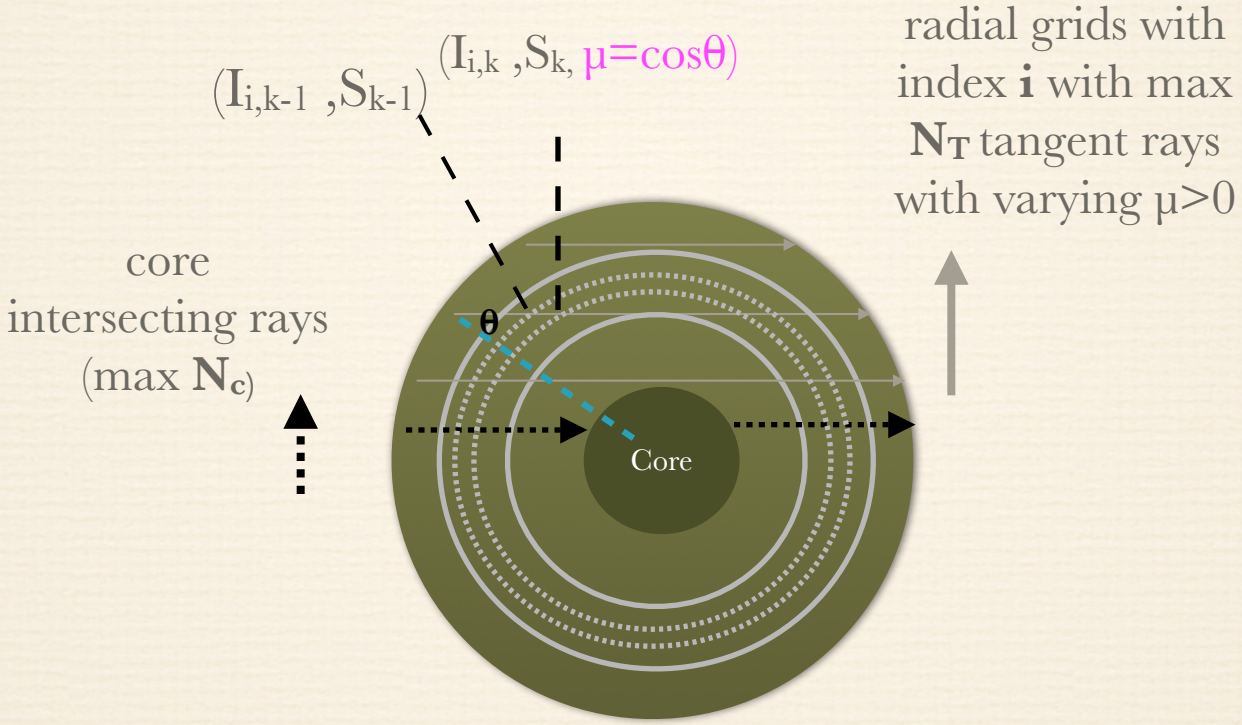
The average intensity at a point, x, depends on specific intensities due to all the rays that pass through point x. These specific intensities carry information from all other points x'≠ x

PHOENIX: A general purpose moving atmosphere RT code for general stellar objects



Pictorial description of RT

1D with single scattering per photon scenario. i is the index of a ray. (i,k) is the index of an intersection (with radial shells) of a ray with index i.



Algorithmic approach to handle RT

- The source function at a given frequency (from now on we drop ν subscripts) $J^{n+1}=\mathbf{\Lambda[S^n]}$, [] indicates that it contains information through integration of all solid angles. Other way of writing this is, $S^n=S[J^n]$
- $\mathbf{S^{n+1}=A+b\Lambda S^n}$, S^i is the source function after i number of iterations.
- At n->∞, S=A+b**Λ**S, at convergence
- ⇒ $\mathbf{S^n-S=A+b\Lambda^n(S^0-S)}$

⇢Linear convergence.

⇢$N_{iter} >> 1/\min(1-\lambda_{max})$

Typically it will take at least as many iterations as number of scatterings squared before a photon is absorbed at high optical depth.

An approximate operator is introduced,

such that $\mathbf{\Lambda}=\mathbf{\Lambda}^*+(\mathbf{\Lambda}-\mathbf{\Lambda}^*)$

Iteration scheme,

$(1-b\mathbf{\Lambda}^*)S^{n+1}=A+b(\mathbf{\Lambda}S^n-\mathbf{\Lambda}^*S^n)$

Most simple Choice of $\mathbf{\Lambda}^*$ =Diag(**Λ**)

⇢transport happens over at least one grid separation.

When the grid spacing is smaller, τ is smaller. $\mathbf{\Lambda}^*$ uses a **band diagonal** adoption of **Λ**.

⇢Speed up of convergence

Other ways?

Parallelization of RT

For a general 3D grid:
$\mathbf{\Lambda}$ is a square matrix of size =(total grid volume)^2.
For a box of 100x100x100 grid cells, $\mathbf{\Lambda}$ has 10^{12} number of elements.
Storage and calculation $\Rightarrow$impossible

Parameters for 1D grid

- Largest overhead in time come from computation of formal solution coefficients and the formal solution along each ray.
- Construction of $\mathbf{\Lambda}^*$ is not a signifiant contributor in time.
- Number of iterations needed to converge the whole system depend on the choice of $\mathbf{\Lambda}^*$

Parallelization Strategy:

- Distribution of rays~Number of intersection points/ number of processors
- Communication: Workers send I_ν data to the master and to have mean intensity, J_ν values evaluated by the master and broadcasted back to workers. Then begins the next iteration.
- Libraries used are LAPACK, MPACK and MKL
- Serial time ~10-20ms for each wavelength point

Size of the input data

Memory I/O requirements

- Number of individual energy levels ~10,000~10 MB
- Number of Non-LTE transitions: 100,00~150 MB
- Equation of state data storage~40 MB
- Auxiliary storage ~50 MB
- Excluding molecular species, total memory required ~200 MB
- Including molecular species, total memory required ~1 GB

Input data size:

Atomic +ionic+diatomic molecular lines >1 GB

Total molecular lines >10 GB

All line selection need to be made runtime which dynamically creates sub-lists which could be much smaller or just as long as the original lists.

Approach to Statistical equilibrium

- **Scenarios and cost**

Local thermal equilibrium,
level population=[n]=f(n_e,, T), $$ primarily I/O dominated
Non-local thermal equilibrium, [n]=g(n_e,, T, J_ν), $$ I/O and rate calculation

- **Nature of equation**:

After some simplification, $R_{du}(n_u,n_d) \Rightarrow R_{du}(n^{\mathbf{old}}_u,n^{\mathbf{old}}_d)$ or $R[n]=R^{old}$

⇢large set of linear equations.

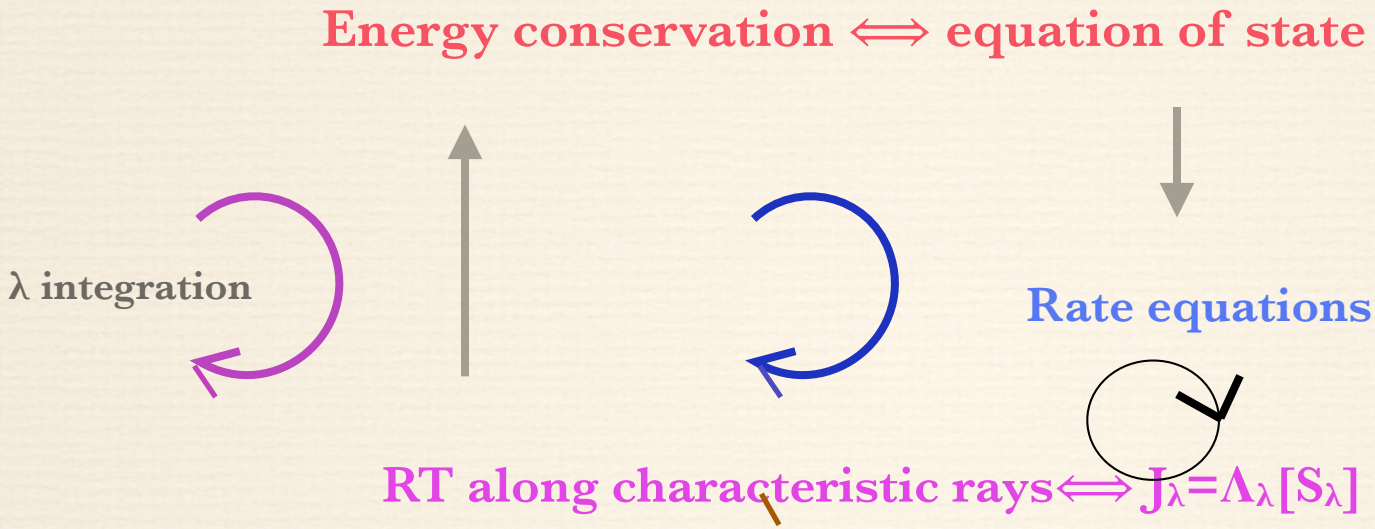
- **Strategy**

[R] is a sparse matrix, not all levels are strongly coupled to all other.
For each element/group, linear equation is solved
Inversion method: Gauss Seidel and Jacobi iteration
Libraries LAPACK, MPACK, MKL

- **Speed up**

Six times just LAPACK stuff.

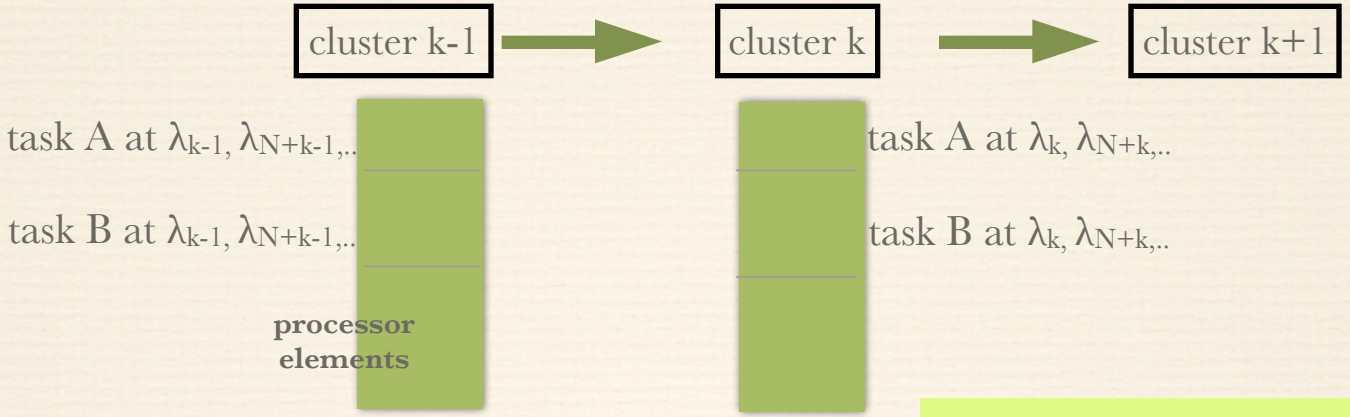
Summary of Parallel Implementation so far



1. Solves [n] by using Jacobi iterations or Gauss Seidel method. This [R] is a linear function of $\mathbf{\Lambda}$
2. Uses MPACK package
3. Typically 20 iterations are needed within one inversion and is ~6 times faster with MPACK

1. For each wavelength point, the formal solution need to be evaluated at all points.
2. For a given λ point, the characteristic rays are distributed among processor elements so each processor work on roughly equal number of points of intersections and each work on different rays.

Remainder of parallel Implementation- λ integration



Expected serial CPU time

- For each wavelength point, 10-100 millisecond.
- Number of wavelength points~ 1 million
- time needed ~few days for a moderate wavelength resolution (10,000 wavelength grid).
- For convergence of the temperature structure (~40 iterations of the entire RT+RE scheme) in order to match up the input luminosity, **many days** of CPU time needed.

Communication of RT results between clusters is needed due to first order discretization in λ.

Strategy: Cluster k, pre-process atomic data related work, receives RT results for λ_{k-1}, evaluates RT results for λ_k,passes RT result to cluster k+1, post-processes atomic and thermodynamic quantities

Total number of clusters saturates, optimal $N_{cluster}$~12
radial grid N=128 (256)
processor element/cluster~12
Number of grids per PE~6(21)

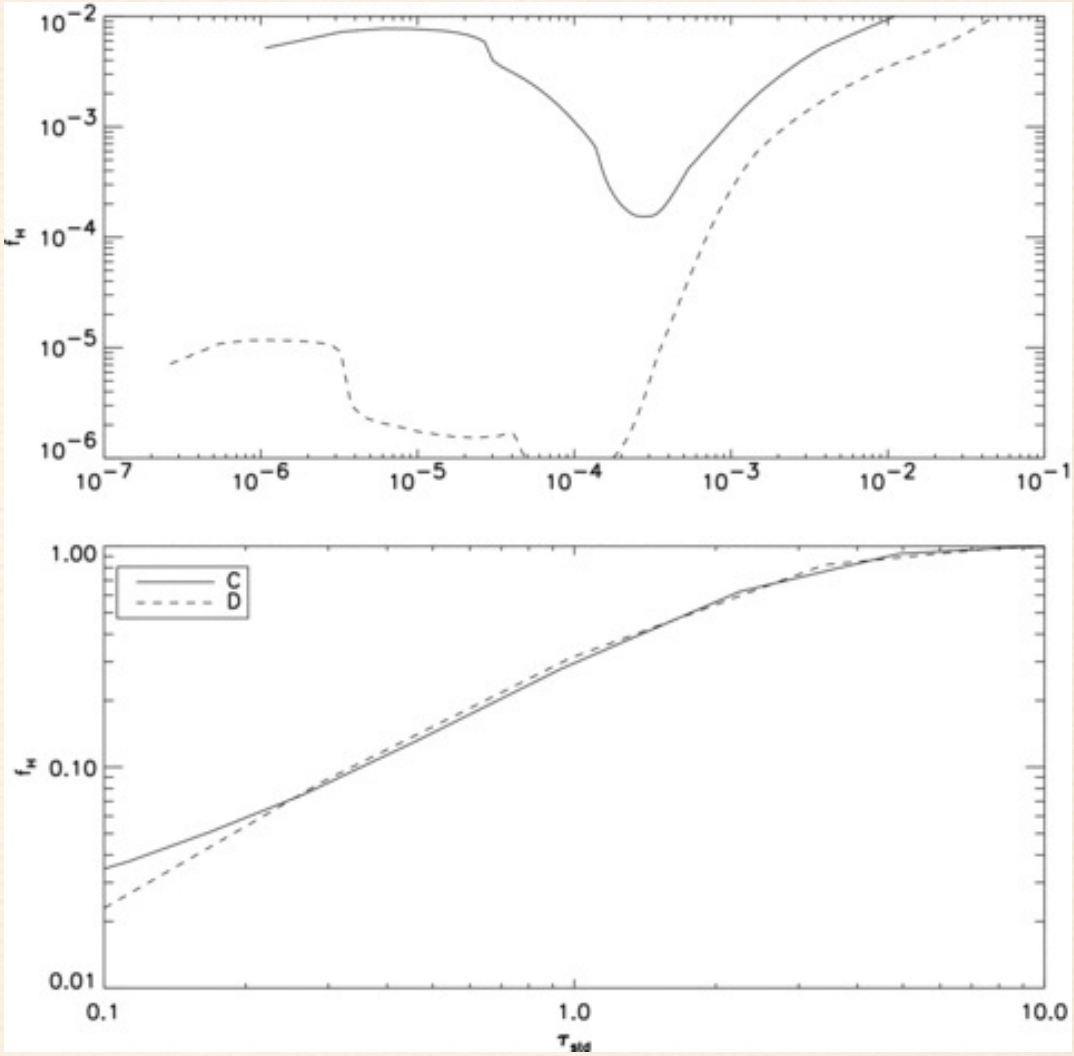
Work done at local cluster at University of Oklahoma.

Machine specs:
Type of machine: i486_64
Communication: Myrinet
MPI: Open MPI and MPICH
Compiler: Intel
other libraries LAPACK, QD

Task specs:
Number of nodes=8
Processors per node=4
Number of total processors= 32
Core rays= up to 50
Tangent rays= 128

Timeline specs:
Time step 0.5 day (for a light curve calculation).
Time for pure H (4 level +NLTE) +other metals(LTE) ~1 day
iter(RT, LTE)~ 20 (for each λ point)
iter(RT, NLTE) ~200
iter (Rate)~20
iter(LTE,Temp)~40

Comparison of the hydrogen ionization fraction found using the four-level (C) and 921-level (D) model atoms in a metal-rich environment.



Soma De et al. MNRAS 2010;407:658-668

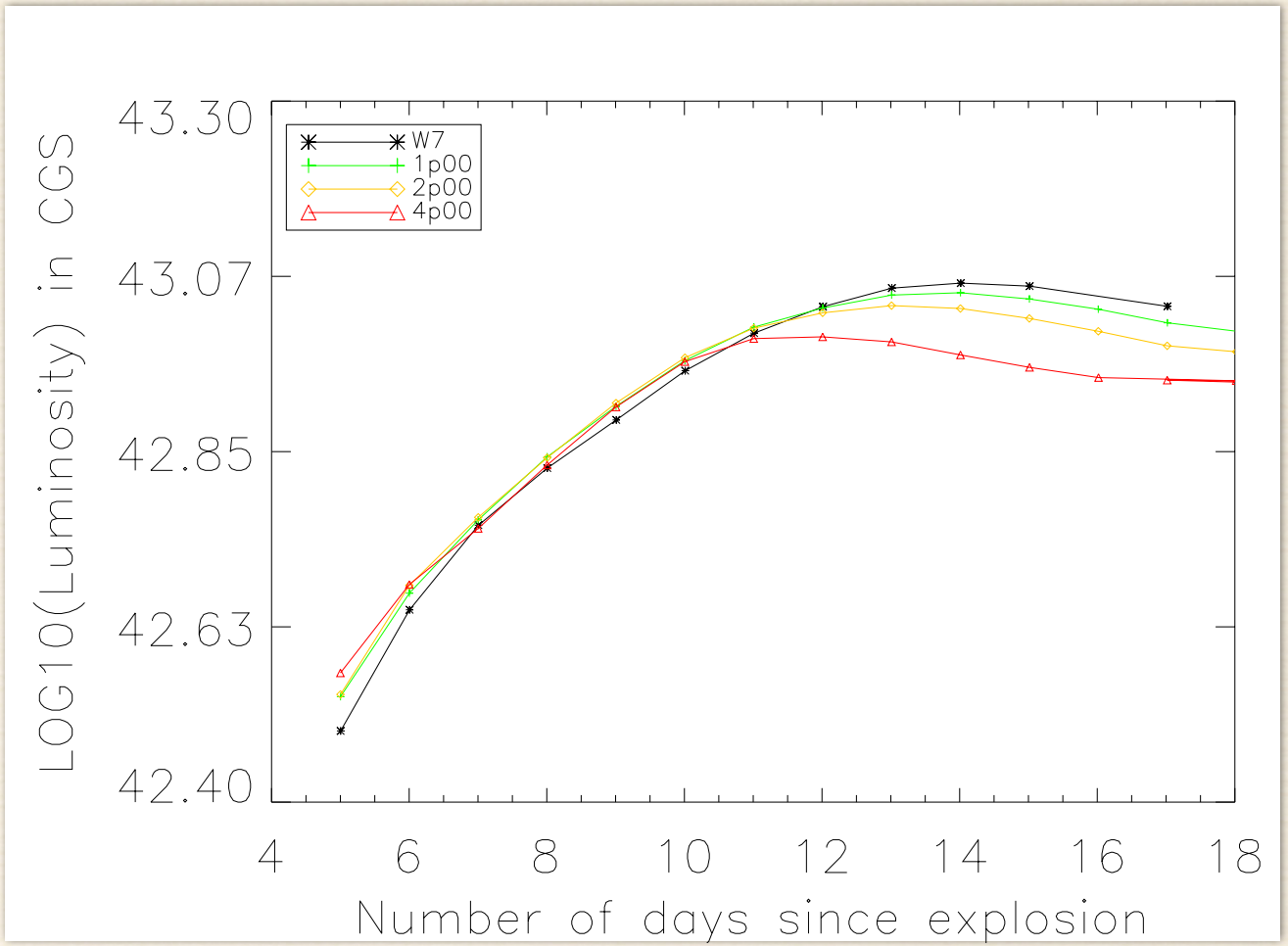
MONTHLY NOTICES *of the* Royal Astronomical Society

SN Ia Light Curves

Work done at ASU supercomputing center

Machine specs:
machine: i486_64
Communication: InfiniBand
I/O Lustre filesystem.
MPI: MVAPICH, with different tuning parameters
Compiler: Intel
libraries LAPACK, MKL, MPACK
Processors: Intel (Westmere, Harpertown, Sandy bridge)

Task specs:
Number of nodes=8 (scales up to ~12)
Number of total processors=64



Algorithm specs:
Operator Bandwidth=35
iter (RT)/ALI=20 (LTE for each λ point)
iter (Rate)/MPACK=20 (within Jacobi)
iter (Temp)/n-Newton Raphson=40

Wall time/iteration
LTE~ 20 -25 minutes
NLTE ~ 300-400 minutes
Total iterations (LTE) ~40
Time total ~9 hrs
Estimated time in Edison ~4 hrs (144 cores)
estimated ~60 hours for NLTE problem

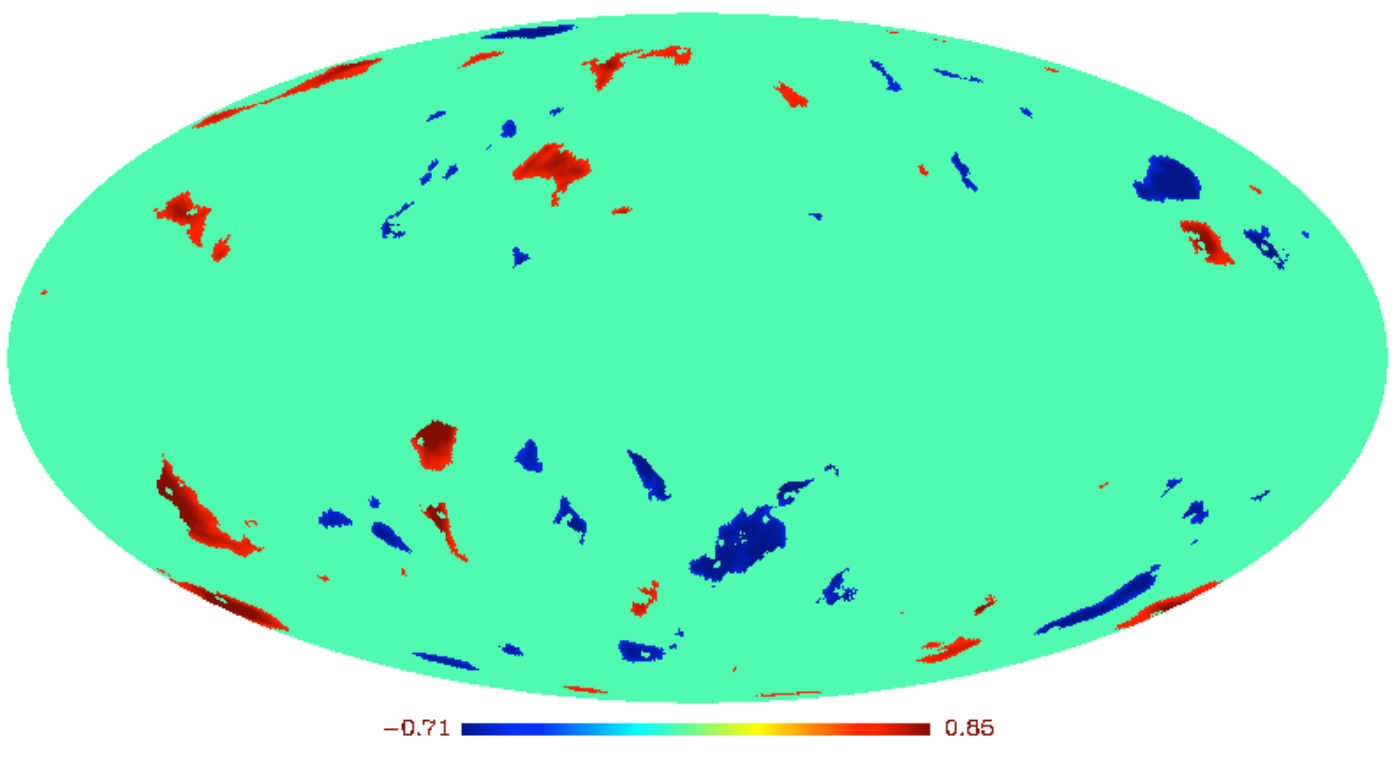
Summary

- Moving atmosphere, optimize for up to 144 processors.
- Static atmosphere, optimize to theoretical maximum.
- Estimated time needed for a simple problem in a machine like Edison ~ 4 hours, for an advanced problem ~60 hours.
- More strategies needed for full 3D

Cosmology, HPC and ongoing work

Cosmic temperature or local?

Cross correlation between galactic rotation measures data (information about the magnetic field and other plasma related stuff) and cosmic microwave background temperature data (after removing all foreground effects)



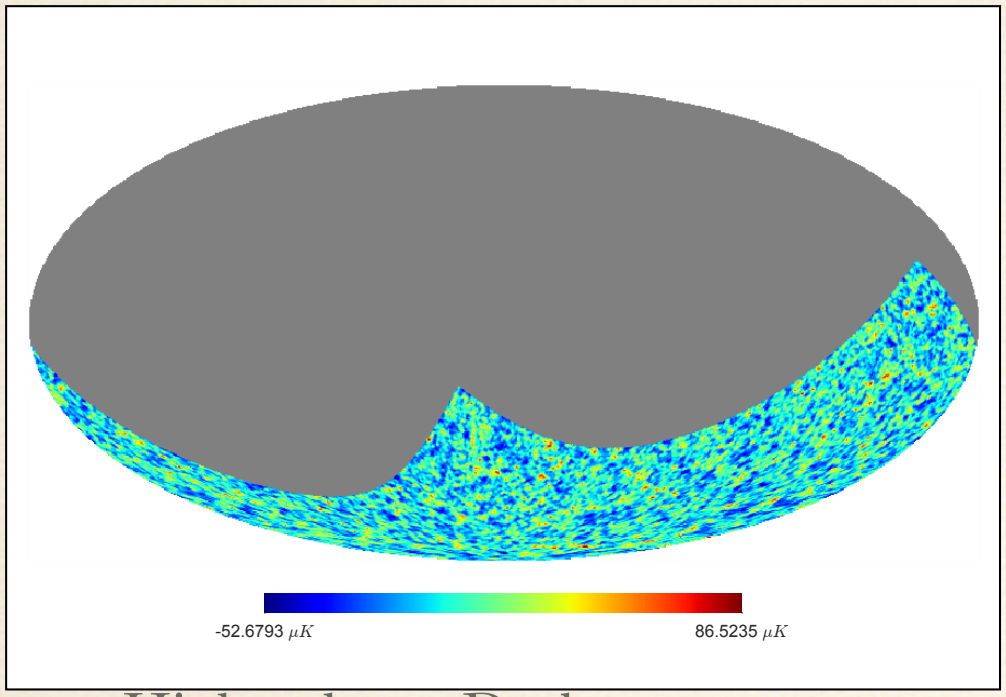
Data stored in HDF5 format and results are manipulated using Python , Healpix (used for high resolution discretization of functions on the surface of sphere) and IDL
Shared memory (OpenMP) implementation on certain functions of Healpix.

De, Pogosian and Vachaspati , 2015 in prep

Extraction of a tiny cosmological signal

- Cosmic 21 cm signal (a neutral hydrogen radiation) is very faint compared to the radiation that come from our galaxy. The galactic signal is at least 5 order of magnitude higher!
- FastICA (fast independent component analysis) was used to retrieve a cosmological signal when the signals mixed have a much larger magnitude and similar profile to the tiny signal of interest.
- The particular FastICA algorithm is based on fixed point iteration scheme on minimization or maximization of kurtosis.
- Gaussian components are left in residual

Input cosmology data

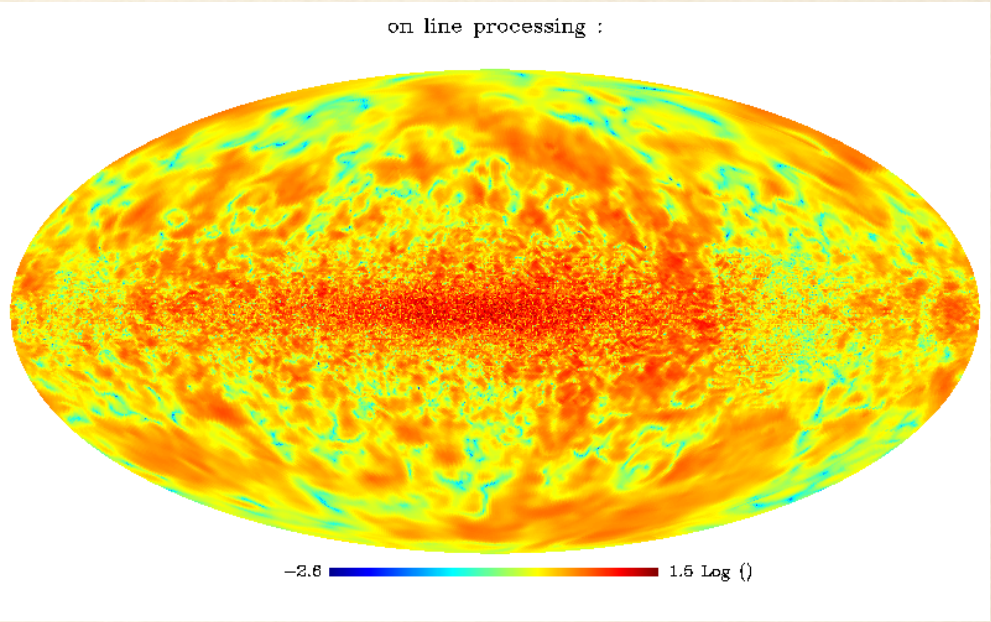


High volume Dark energy survey simulation data
Particle size 2048^3

This survey proposes to take 400 of ~GB sized images each night over 500 nights.

A neutral hydrogen density is then assigned using some prescription and smoothed

Foreground Simulation



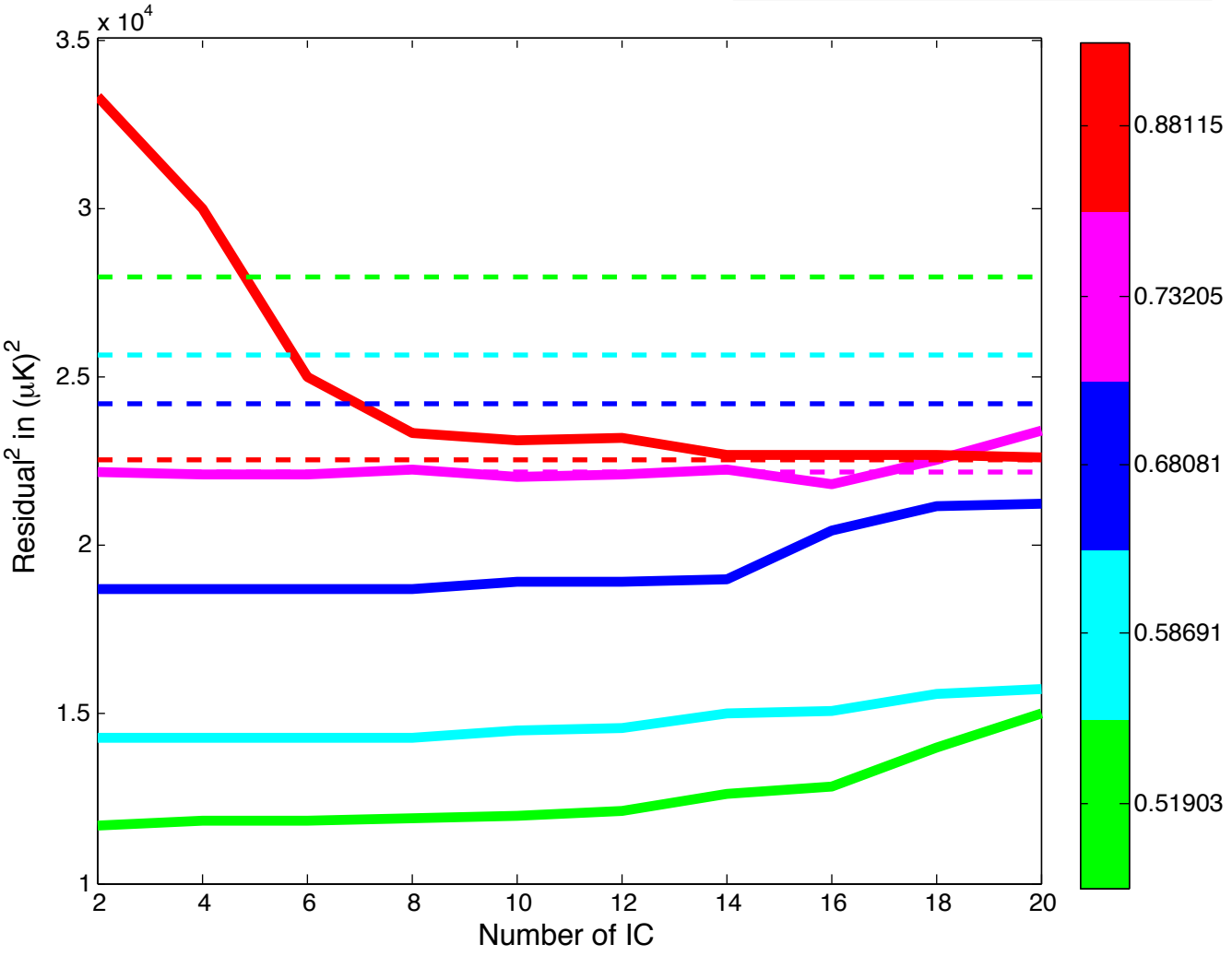
Simulation of polarized synchrotron emission made from using GALPROP code and HAMMURABI simulation (built on C, C++).

Results of FastICA and Future?

Built over MATLAB version of FastICA
The input data is Dark energy survey simulation smoothed over 0.5 degrees.

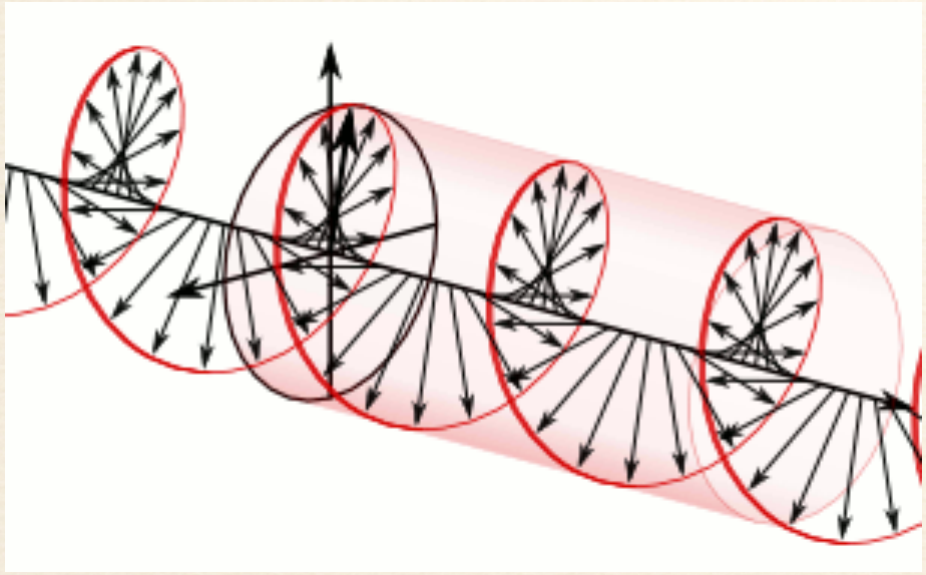
Parallelization: Currently, a better version of neutral hydrogen field is being worked on and use of FastICA in parallel (HiPerSat etc)

De and Refregier, 2015, in prep



Circular polarization: Simple analytics to HPC

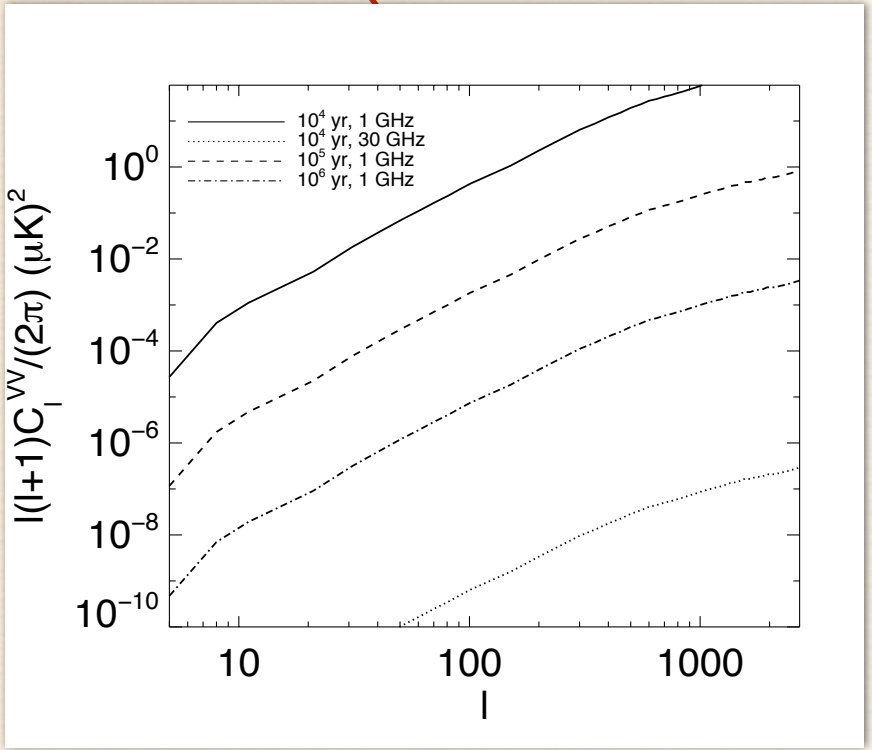
- Electromagnetic field has electric field vector perpendicular to direction of propagation.
- Sometimes, the electric vector points in one direction, which is fixed, but its magnitude can vary. This is linear polarization.
- Other times, the electric field vector can change direction while keeping its magnitude fixed. This is the case for circular polarization (CP).

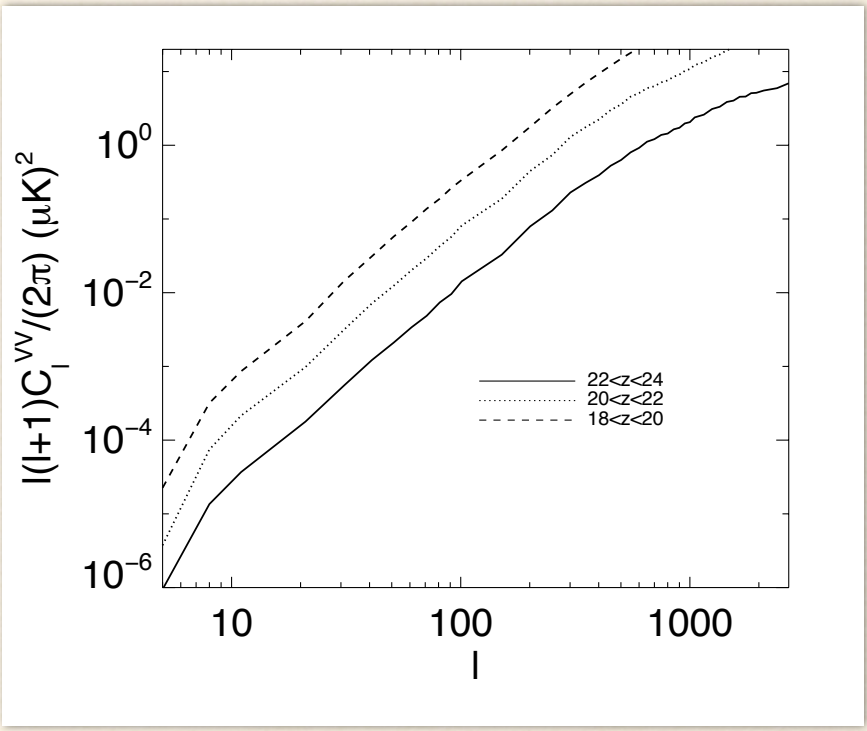


- In the context of CMB, or generally cosmology, already a lot has been done on linear polarization.
- We introduce a mechanism to generate CP in the CMB through the explosion of the First stars' explosions. These are like giant supernovae explosions creating energy of up to 10^{53} ergs. Generation of large magnetic field under these massive explosions and passage of the CMB photons through the shocked environment of the star, makes CMB pick up a circular polarization quality.

Circular polarization and HPC

$$C_l^{VV} \approx \frac{128}{15\pi} \sum_{l'l''l'''} \sum_{m'=-2}^{m'=2} \sum_{m''+m'''=m} \int_{r_*}^{0} dr \int \frac{k^2 dk}{(2l'+1)r^2} D_\alpha^2(r) P_\alpha\left(\frac{l''}{r}\right) P_{E_{lm}}(k,r)\, j_{l'''}^2(kr) I_{lm}^2,$$





Serial Time =4 days, m=0, m"=fixed, direction of B field fixed.

Corresponding full range m∈{-ℓ,ℓ}

Implementation is done in Fortran 95 and above and C, C++

De and Tashiro, Arxiv , 2014

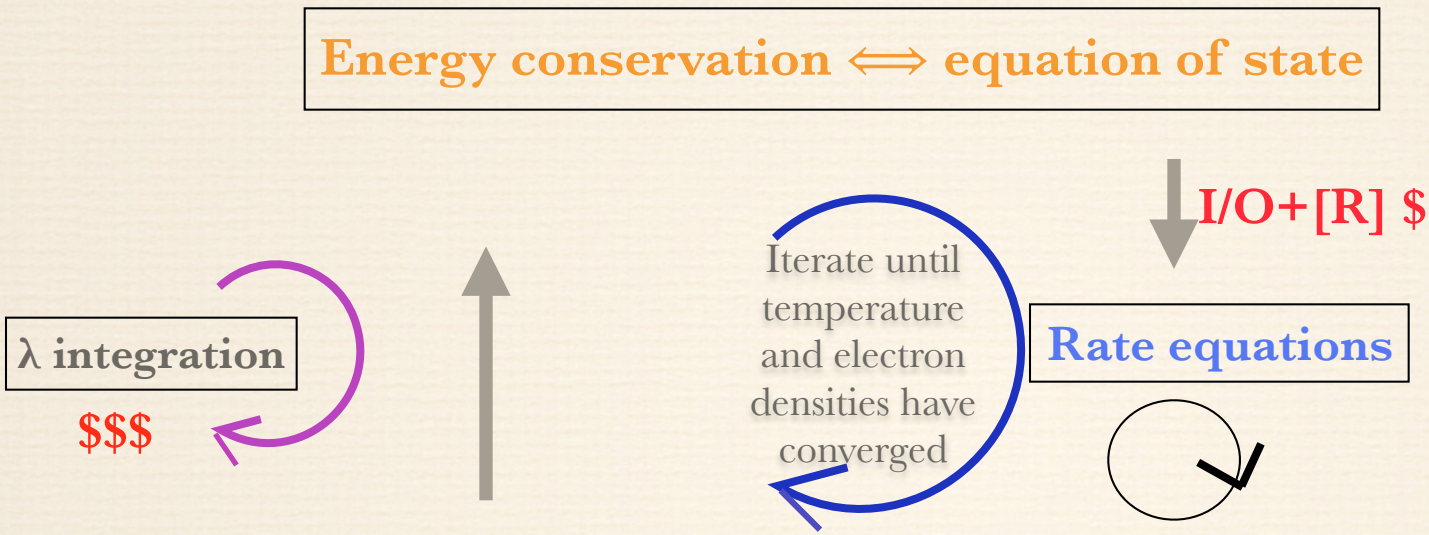
CP Parallelization and future?

- Not very difficult to parallelize at its current form, with precise care to load balancing.
- Now, magnetic field is held directionally constant among all lines of sight. There is a global direction of magnetic field. Many other simplifications for an analytic form have been used. When **B** is allowed to vary, a full calculation, pipelined with actual number of first star explosion and simulation is challenging and a very interesting problem.
- My own version of publicly available CP calculator (specific to certain systems and mechanisms) adaptable to different architectures in under construction.

PHOENIX

- A radiative transfer code that solves the full RT problem in moving atmospheres of general stellar objects, by consistently solving for the level populations of different species, rates (radiative and collisional) for associated transitions and temperature.
- The code relies on different blocks that handle
  - RT formal solution
  - Spectral line contributions
  - Non-LTE contributions
  - Wavelength integration and
  - Energy conservation
- The results from each block is communicated to others as necessary during the iterative approach to the solution which **simultaneously** satisfies RT equation, rate equation and energy conservation (for example, consistency with input value of luminosity)
- Implemented primarily in fortran and fortran implementation of MPI.

Solution overview

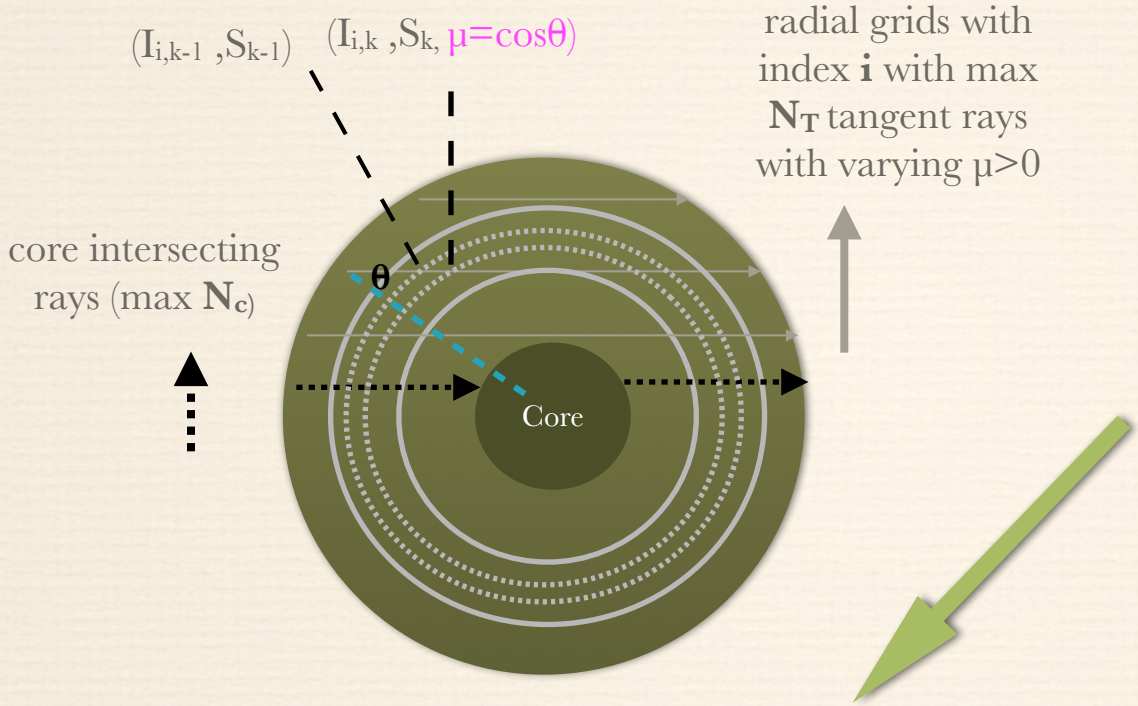


1. For given [n], n_e solve RT at a given λ, get J.
2. Update [R]
3. Solve rate equations for each group to get new [n]
4. Compute new n_e with updated [n]
5. Iterate until n_e,[n] converge, for a current value of J.

1. Pre-processing involves having emission and opacity coefficients evaluated which depend on level population of species. These appear in the $J=\Lambda S$ equation.
2. The r x μ plane is meshed with rays along which RT is solved.

Pictorial description of RT

1D with single scattering per photon scenario. i is the index of a ray. (i,k) is the index of an intersection (with radial shells) of a ray with index i.



Non-local or local thermal equilibrium, atomic data, lines, continuum, radiative and collisional rates, directly or indirectly, all depend on J_ν. Rate operator [**R**][n]=[function of J_ν and ν integrated over some frequencies]. This is the rate equation (hereafter RE) block

- The formal solution for each ray is evaluating a) $\mathbf{I_{i,k}}$ (i constant along a ray) with knowledge of $\mathbf{I_{i,k-1}}$ b) a polynomial interpolation (quadratic) of ($\mathbf{S_{k+1}}$,$\mathbf{S_k}$ and $\mathbf{S_{k-1}}$) with emission and opacity coefficients and c) radial depth.
- *The coefficients in the formal equation involves atomic physics, opacities, level population of the species and frequency. Some of these quantities themselves depend on angular average of* $\boldsymbol{I_{ij}}$
- $J=f(\omega_{ij}, I_{ij})$, ω_{ij} being the angular weights
- We write $\mathbf{J}=\mathbf{\Lambda}[S]$, and $\mathbf{\Lambda}$ with its coefficients depending on atomic physics and level population of species.
- Calculation and storage of $\mathbf{\Lambda}$⇢ $$$ ->highly impractical.

Approach to Statistical equilibrium

Local thermal equilibrium,
level population=$[n]=f(n_e, T)$
Non-local thermal equilibrium,
$[n]=g(n_e, T, J_\nu)$
$\Sigma_d\{n_d(R_{du}+C_{du})-n_u(R_{ud}+C_{ud})\}=0 \; \forall \; u \in \{1,N_{level}\}$, R, C are radiative and collisional rate matrices.

$\Sigma_d\{n_d(R_{du}(J_\nu)+C_{du})-n_u(R_{ud}(J_\nu)+C_{ud})\}=0$

Dropping ν subscripts,

$J^{n+1}=\Lambda^*S^{n+1}+(\Lambda-\Lambda^*)S^n=\Lambda^*S^{n+1}+\Delta J^n$

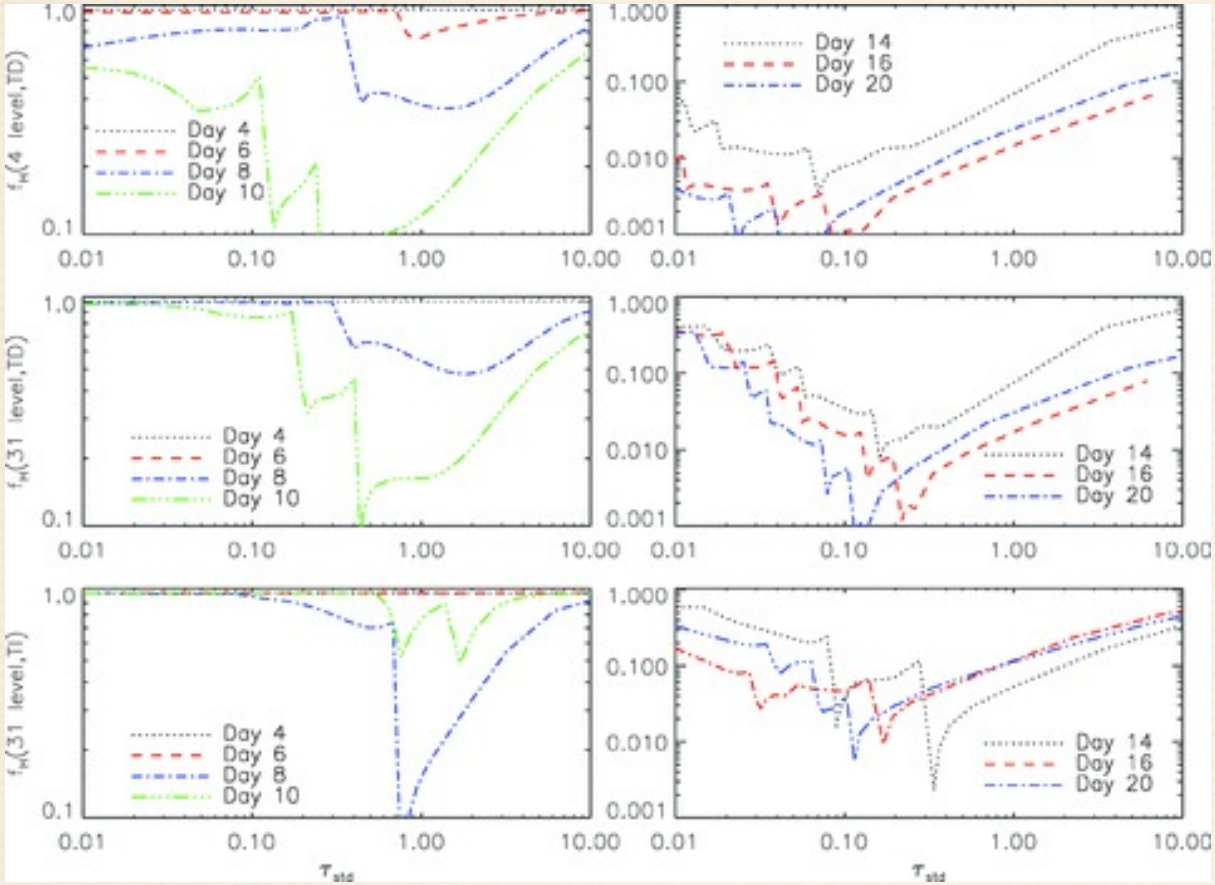
$\Rightarrow\Sigma_d\{n_d(R_{du}(\Lambda^*S^{n+1}+\Delta J^n)+C_{du})-n_u(R_{ud}(\Lambda^*S^{n+1}+\Delta J^n)+C_{ud})\}=0$

$\Rightarrow\Sigma_d\{n_d(R_{du}(\mathbf{n_u,n_d})+C_{du})-n_u(R_{ud}(\mathbf{n_u,n_d})+C_{ud})\}=0$

⇢ Non-linear in [n], so replace $R_{du}(n_u,n_d)\Rightarrow R_{du}(n^{\mathbf{old}}{}_u,n^{\mathbf{old}}{}_d)$

⇢ large set of linear equations. Not all levels are strongly coupled to all other, so for each element/group, linear equation is solved using LAPACK, liked with MKL libraries. Uses Gauss Seidel and Jacobi iteration and uses MPACK libraries. Six times speed up compared to just LAPACK stuff.

Science goals achieved



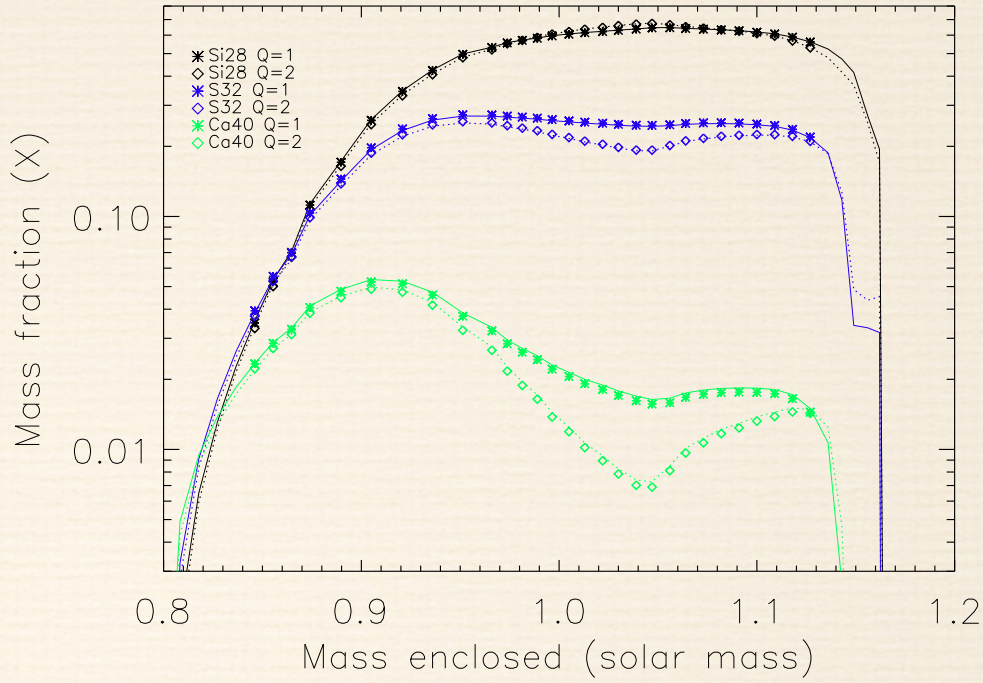
Ionization levels of different systems at different phases of expansion. The models include both time dependent and independent rate equations.

Soma De et al. MNRAS 2010;401:2081-2092

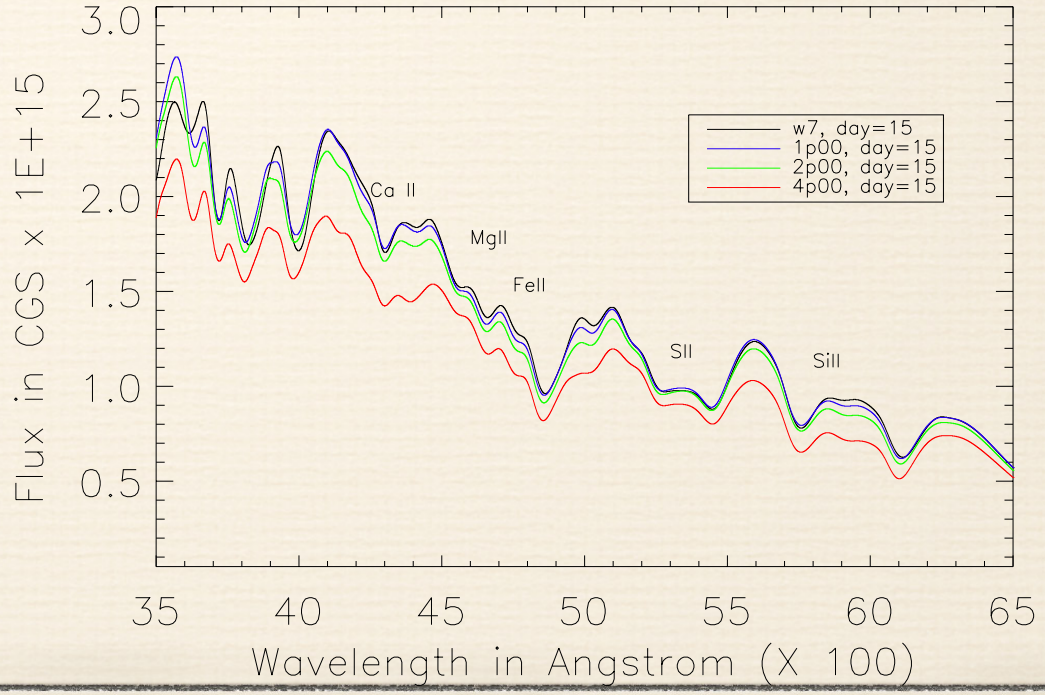
MONTHLY NOTICES
of the Royal Astronomical Society

Science goal achieved

Analytical estimates of element abundances show that the silicon groups elements freeze out while they are in **quasi-nuclear statistical equilibrium**.



Spectra of SN Ia for different nucleosynthesis models. **The abundance of elements were an input**. Then used RT code to generate the spectra (De et al, ApJ, 2013).



Double degenerate models with different angles of collision