Computational improvements to line radiative transfer in Magritte

T. Ceulemans, F. De Ceuster, L. Decin, J. Yates (in prep.)

Thomas Ceulemans
KU Leuven
March 2023

0 My place within ATOMIUM

- Phd student under prof. Decin
- Working on line radiative transfer
- Developing MAGRITTE together with Frederik De Ceuster
- Phd goal: to make line radiative cooling computationally feasible in hydrodynamics simulations

0 Introducing Magritte

Magritte

- Open-source 3D NLTE line radiative transfer library
- Creates synthetic images of hydrodynamics simulations
- Written in c++, API in python
- Available at https://github.com/Magritte-code/Magritte

0 The computational challenges of line radiative transfer

1D Radiative transfer equation

$$\frac{\partial I}{\partial x}(\boldsymbol{x}, \nu, \hat{\boldsymbol{n}}) = \eta(\boldsymbol{x}, \nu) - \chi(\boldsymbol{x}, \nu) I(\boldsymbol{x}, \nu, \hat{\boldsymbol{n}}) \tag{1}$$

I monochromatic intensity, η emissivity, and χ opacity.

Applying radiative transfer on hydrodynamics simulations (even in post-processing), is computationally challenging.

- By definition, the equation is non-local
- Wildly different scales are involved
- Doppler shifts make it hard correctly treat the narrow line profiles

Computational improvements in MAGRITTE, up to **50 times** faster.

1 Outline

1. Computing line opacities/emissivities
2. Analytically handling doppler shifts
3. Other improvements
4. Combined effect of these improvements
5. Bonus slides: Comoving solver

1 Computing line opacities/emissivities

For a single line:

$$\chi_{ij}(x,\nu) = \frac{h\nu}{4\pi}\left(n_j(x)B_{ji} - n_i(x)B_{ij}\right)\phi_{ij}(x,\nu) \tag{2}$$

$$\eta_{ij}(x,\nu) = \frac{h\nu}{4\pi} n_i(x) A_{ij} \phi_{ij}(x,\nu) \tag{3}$$

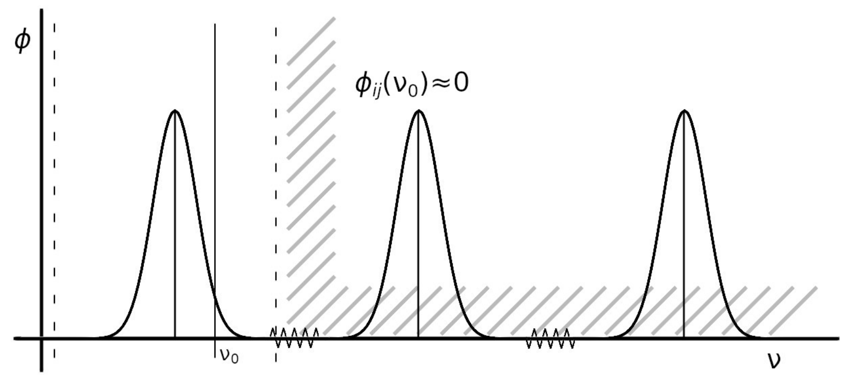
in which ϕ is the profile function (e.g. Gaussian).
For all lines together:

$$\chi(x,\nu) = \sum_{ij\in\text{lines}} \chi_{ij}(x,\nu) \tag{4}$$

$$\eta(x,\nu) = \sum_{ij\in\text{lines}} \eta_{ij}(x,\nu) \tag{5}$$

1 Computational inefficiencies when computing total opacity/emissivity

For any frequency, only a small fraction of the lines actually give a non-zero contribution. ($\phi_{ij}(\nu) \simeq 0$ far from the line center)



1 Solving this inefficiency

- Define maximal frequency range
- Ignore all lines outside this range

$$\nu_{ij} \in \left[\nu\left(1 - C\left(\frac{\delta\nu}{\nu}\right)_{\text{max}}\right), \nu\left(1 + C\left(\frac{\delta\nu}{\nu}\right)_{\text{max}}\right)\right] \tag{6}$$

in which C is a constant (by default 10) and $\left(\frac{\delta\nu}{\nu}\right)_{\text{max}}$ is the maximal relative line width at the point x in question.

Computation time improvement: $O(N_{\text{lines}}^2)$ to $O(N_{\text{lines}} ln(N_{\text{lines}}))$

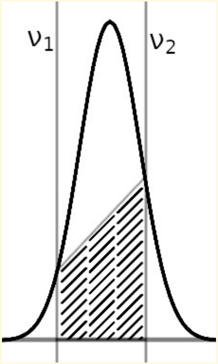
2 Outline

2 Importance of correctly handling doppler shifts

$$\Delta\tau = \int_{x_0}^{x_1} \chi(x,\nu)dx \tag{7}$$

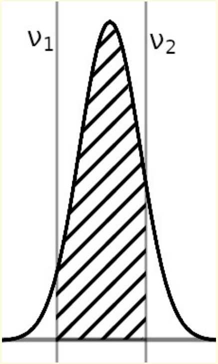
Trapezoidal rule

$$\Delta\tau = (x_1 - x_0)\,\frac{\chi(x_0,\nu) + \chi(x_1,\nu)}{2} \tag{8}$$



- Can fail to properly sample the line profile
- Previously, add extra points to interpolate linearly

2 Handling the doppler shift in a single line



$$\Delta\tau = \int_{x_0}^{x_1} \chi(x,\nu)dx \tag{9}$$

For a single line, separate out the line profile

$$\Delta\tau_{ij} = \int_{x}^{x+\Delta x} \underbrace{\overline{\chi_{ij}}(x)}_{\chi_{ij}/\phi_{ij}} \phi_{ij}(x,\nu)dx \tag{10}$$

Integrate the line profile, obtaining (similar to Sobolev approximation)

$$\Delta\tau_{ij}(\nu) = \Delta x \left(\frac{\overline{\chi_{ij}}(x_0) + \overline{\chi_{ij}}(x_1)}{2}\right)\left(\frac{\mathrm{Erf}(f(x_1)) - \mathrm{Erf}(f(x_0))}{2\Delta\nu_{ij}}\right) \tag{11}$$

2 Total optical depth

Total optical depth obtained by summing

$$\Delta\tau = \sum_{ij \in \text{lines}} \Delta\tau_{ij} \tag{12}$$

- Far lines do not contribute
- No interpolation points necessary
- Computation time improvement: $O(\nabla \cdot \bar{v})$ to $O(1)$

3 Outline

3 Other improvements to Magritte

- Change in internal datatypes (smaller, thus faster)
- Automated testing + versioning
- A new, fast re-meshing method (see next slide)
- Memory-sparse variant of Feautrier solver

3 Why re-mesh a grid

In De Ceuster 2020b, it has been proposed to re-mesh hydrodynamics model for radiative transfer.

- Hydrodynamics model contain many points
- By re-meshing, we reduce the amount of points
- Less points, thus faster computations
- Acceptable accuracy penalty

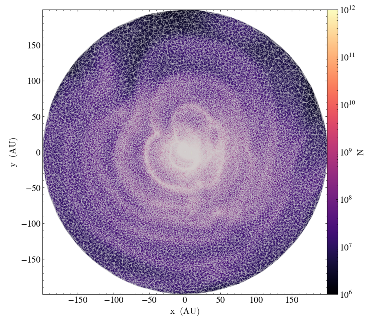


Figure: Slice of PHANTOM (Price et al 2018) binary wind model from Malfait et al. 2021

3 Re-meshed models

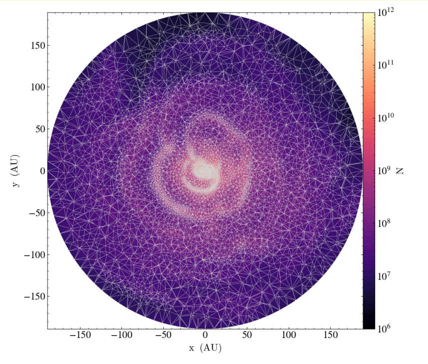


Figure: Re-meshed using GMSH

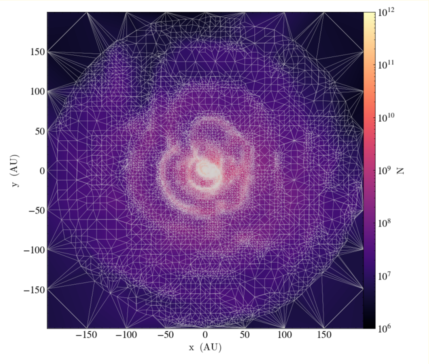
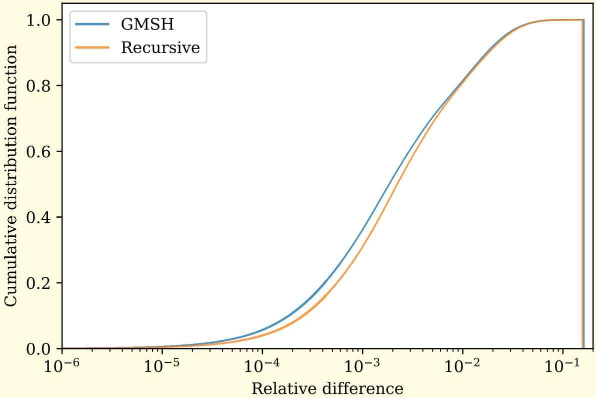


Figure: Re-meshed using recursive subdivision

Timings for re-meshing: GMSH 173s, Recursive 3s

3 Accuracy of re-meshing

Computed mean intensities



4 Outline

4 Timings (van Zadelhoff 1)

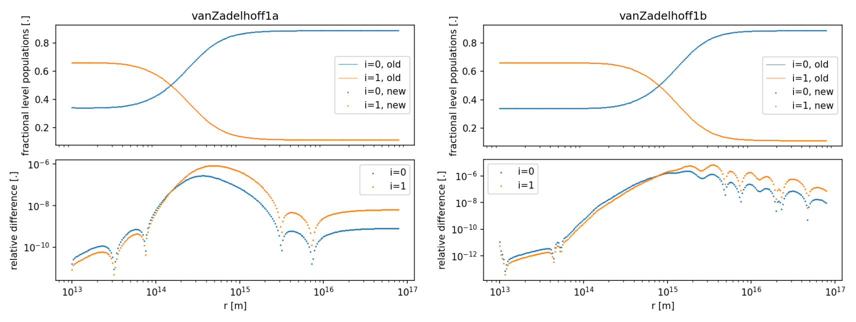
Van Zadelhoff NLTE benchmark models (Van Zadelhoff 2002).

Van Zadelhoff 1: Cloud without velocity gradient, single line.
Van Zadelhoff 2: Collapsing HCO+ cloud, 20 lines.

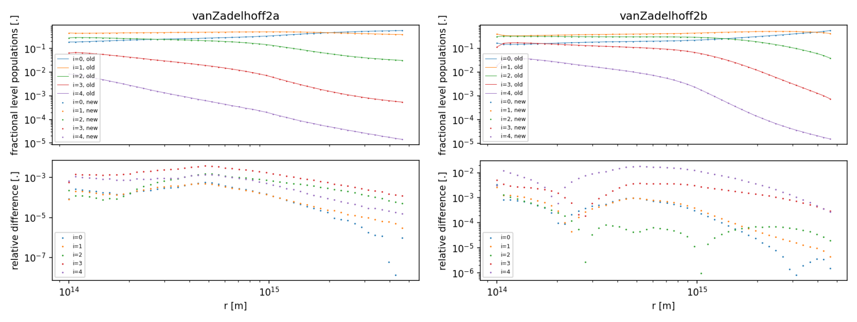
Time [s]	MAGRITTE 0.2.0	MAGRITTE 0.0.2
Van Zadelhoff 1a	8.1	19
Van Zadelhoff 1b	47	89
Van Zadelhoff 2a	12	598
Van Zadelhoff 2b	22	1169

Roughly 2 times faster in case of a single line.
Roughly **50 times** faster for 20 lines. $O(N_{\text{lines}}^2) \to O(N_{lines} \ln(N_{\text{lines}}))$

4 Accuracy (van Zadelhoff 1)



4 Accuracy (van Zadelhoff 2)



4 Conclusion

Significant speedups can be obtained with simple improvements.

Generally applicable improvements

- Efficiently ignoring far lines
- Analytically computing the optical depth

Magritte

Get started using MAGRITTE for synthetic line observations. Consult the extensive documentation at https://magritte.readthedocs.io/.

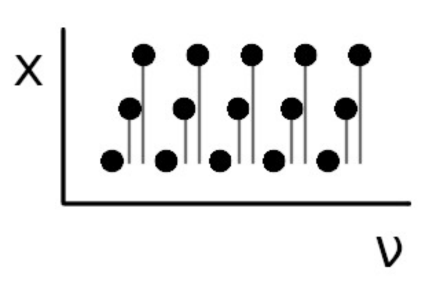
5 Outline

1. Computing line opacities/emissivities
2. Analytically handling doppler shifts
3. Other improvements
4. Combined effect of these improvements
5. Bonus slides: Comoving solver

5 Yet another bottleneck in NLTE ray-based line radiative transfer

In NLTE line radiative transfer, we want to compute the intensity around each line self-consistently at each point.

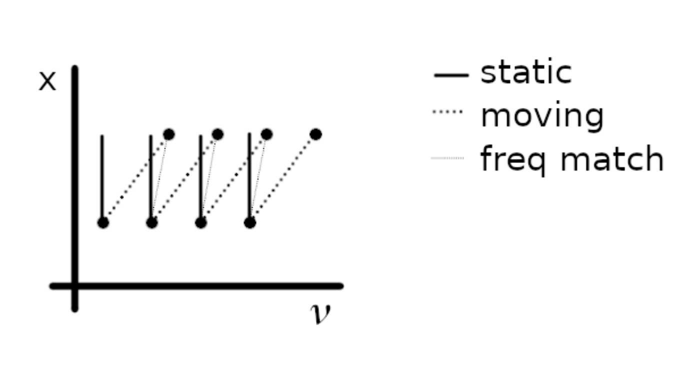
- Narrow line profile functions require a dense frequency sampling
- Doppler shifts misalign the frequency discretization
- These misalignements inhibit the reuse of computed intensity



5 A brief explanation on comoving frame RT

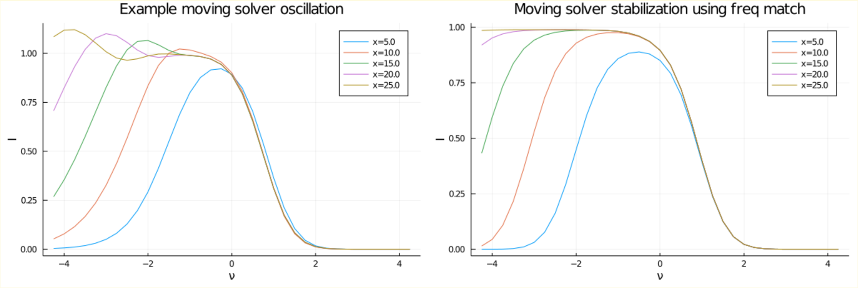
Similar to Baron et al. 2004

$$\frac{dI(x,\nu)}{dx} = \eta(x,\nu) - \chi(x,\nu)I(x,\nu) + \frac{d\nu}{dx}\frac{\partial I}{\partial \nu}(x,\nu) \qquad (13)$$



5 Frequency matching

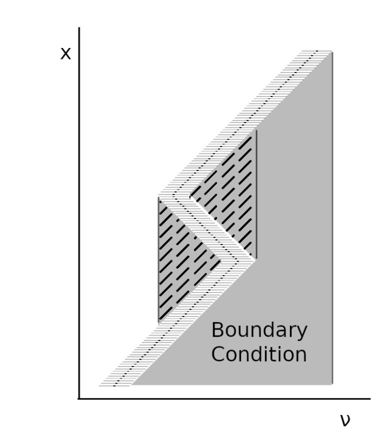
- Free choice of connecting frequency discretizations
- Numerical stability: do not extrapolate
- Thus connect with minimal frequency difference



5 Boundary conditions

Boundary conditions are required on the edge of the frequency discretization

- Non-local, taking into account previously encountered frequencies.
- Local, ignoring any previously computed frequency.



5 The comoving method applied to an actual model

Tested on a PHANTOM model of an outflow of a binary system (courtesy of J. Malfait).

Computation algorithm	time[s]
Feautrier	2510
Comoving (non-local bdy)	753
Comoving (local bdy)	667

Timings of a single NLTE iteration, using a single line, 54 directions.