# *DIAPHANE*: a portable radiation transport library for astrophysical applications

Darren S. Reed[a,b], Tim Dykes[c], Rubén Cabezón[d], Claudio Gheller[e], Lucio Mayer[a]

[a]*Institute for Computational Science, University of Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland*
[b]*S3IT, University of Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland*
[c]*School of Creative Technologies, University of Portsmouth, Eldon Building, Winston Churchill Avenue, PO1 2DJ Portsmouth, United Kingdom*
[d]*Center for Scientific Computing, sciCORE, University of Basel Klingelbergstrasse 61, CH-4056 Basel, Switzerland*
[e]*Swiss National Supercomputing Centre, CSCS-ETHZ Via Trevano 131, CH-6900 Lugano, Switzerland*

## Abstract

One of the most computationally demanding aspects of the hydrodynamical modelling of Astrophysical phenomena is the transport of energy by radiation or relativistic particles. Physical processes involving energy transport are ubiquitous and of capital importance in many scenarios ranging from planet formation to cosmic structure evolution, including explosive events like core collapse supernova or gamma-ray bursts. Moreover, the ability to model and hence understand these processes has often been limited by the approximations and incompleteness in the treatment of radiation and relativistic particles. The DIAPHANE project has focused in developing a portable and scalable library that handles the transport of radiation and particles (in particular neutrinos) independently of the underlying hydrodynamic code. In this work, we present the computational framework and the functionalities of the first version of the DIAPHANE library, which has been successfully ported to three different smoothed-particle hydrodynamic codes, GADGET2, GASOLINE and SPHYNX. We also present validation of different modules solving the equations of radiation and neutrino transport using different numerical schemes.

*Email address:* `darren.reed@uzh.ch` (Darren S. Reed)

## 1. Introduction

The goal of the DIAPHANE library is to enable advances in computational modelling of complex astrophysical processes in which the transfer of energy by radiation or neutrinos is important. DIAPHANE has been supported by the PASC[1] initiative, which aims to promote the development of software that can fully utilize new and emerging supercomputing hardware technology. Astrophysics simulations are currently able to capture many of the most relevant hydrodynamical and gravitational processes involved in the formation and evolution of stars, planets, and galaxies. However, our advancement has been limited in a number of physical situations where energy transport by radiation or neutrinos is important. This project aims to improve the situation by providing a library of radiation and neutrino transport routines covering a range of physical regimes relevant for a wide range of astrophyics simulations, and also able to be used simultaneously within a single simulation. We hope that the modular and extensible nature of this library will facilitate community contributions in the form of additional physics routines and other improvements after its initial public release.

There are two primary reasons that including radiation and neutrino transport (RT/NT) in astrophysical simulations has been limited in scope. 1) The characteristic timescale for radiative and neutrino processes is very short compared to that of hydrodynamic processes. This is because the speed of light is much faster than the sound speed – light travels $\sim 10^4$ times faster than sound in the molecular clouds where stars are forming. This means that simulation timesteps must be very short in order to accurately model the problem, increasing the global computation time, in some cases, to scales that are unaffordable nowadays. And 2), RT/NT modelling is complex due to the many different processes involved at the atomic scale (e.g. emission, absorption, scattering). This complexity is increased by the fact that a vast majority of astrophysical scenarios are intrinsically three-dimensional (3D), which translates into solving a full Boltzmann transport equation in 6D. This situation raises 3D hydrodynamical simulations with full RT/NT modelling

---

[1]http://www.pasc-ch.org/

to problems that lie well within the sustained Exascale.

To skirt the computational complexity, RT/NT codes must implement approximate algorithms. Typically, approximated RT/NT algorithms work well only for a specific range of conditions, e.g. optically thin diffuse gas with a single radiative source. Nevertheless, most astrophysical phenomena include a large dynamic range and a corresponding large effective range in optical depth, so are not suited to be modeled by a single RT/NT approximate algorithm. The DIAPHANE project is unique at the moment in the international astrophysics community as it gathers different RT/NT approximations under a common portable framework, while so far, individual hydro codes have used only one or two radiative transfer solvers tailored to very specific problems, in very specific regimes, and adapted to very specific hydrodynamics implementation.

As a consequence, our understanding of the complex astrophysical Universe is hindered by the lack of radiation and neutrino transport (RT/NT) in our computational modeling of star and galaxy formation; planet formation; supernova explosions; and black hole formation. We note that both the short timescales and the complex physics algorithms are very similar for RT and NT; for this reason it is effective and convenient to model the transport of radiation and neutrinos within a single framework implemented as a single library.

Radiation transport is important for the dynamics of many astrophysical processes. Radiative feedback from stars and black holes plays a dominant role in regulating the rate of gas accretion into galaxies and star formation within galaxies. The accumulated regulation of star formation determines the observable galaxy properties such as stellar luminosity, color, and age. Because radiation can couple across a range of scales and conditions, multiple modes of RT modelling capabilities allow us to solve new classes of problems. For example, a planet-forming disk around a protostar requires modelling the star light from the protostar as well as diffusion of that radiation outward through the planet forming disk. Evolving the disk temperature accurately, in this case, is needed to follow the fragmentation of the disk which ultimately determines the number and masses of planets that form.

Neutrino transport is of capital importance for understanding the explosion mechanism of one of the most powerful events in the Universe: core collapse supernovas. During the last phases of the lives of massive stars $(M > 8 - 10 M_{sun})$, a fast collapse of the core drives the formation of a neutron star via a strong deleptonization. As a result, an incredibly large

quantity of neutrinos of all species is emitted. The interaction of these neutrinos, produced by the newly born neutron star, with the ensuing shockwave is critical to adequately understanding how the energy reservoir is tapped to energize and produce the explosions that we observe. Additionally, in the later phases of the explosion, as well as in different scenarios like neutron star mergers, the emission of highly neutronized matter by neutrino-driven winds is very important for characterizing the contribution of these scenarios to the r-process elements and understanding the origin of the nuclear abundances observed in the Galaxy . Therefore, the correct simulation of the neutrino-matter interactions in 3D geometrically distorted scenarios is the cornerstone of our understanding to such fundamental questions.

## 2. The *DIAPHANE* library

### *2.1. Overview of Library Modules for RT/NT physical processes*

**Common Utilities:** This module handles many fundamental operations, ranging from defining the data structure to calculating quantities such as the opacity or the optical depth of gas given its physical conditions; these can be used by other library modules or called directly from a hydro simulation code as needed.

**Flux-Limited Diffusion (FLD):** This is a relatively simple yet powerful method to approximate radiative diffusion. For this reason, FLD served as our 'prototype' module and was the first module to be ported to the library and validated for each of the three hydrodynamical codes included in the project. FLD is treated as thermal conduction as in [1] [2]. Therein, photons diffuse in the direction of the local energy gradient. FLD is most accurate when gas is optically thick, typical of dense star-forming or planet-forming gas. In this regime, the FLD energy transport rate matches the diffusion limit, governed by the random walk of photon scattering. See Sec. 2.3.1 for more details.

**Starrad:** This newly developed ray casting method tracks the direct propagation of energy from a heating source, such as a star, to the surrounding medium, by casting rays onto a spherical angular grid and numerically solving the radiative transfer equation considering the absorption of the medium along each ray. STARRAD is a new type of algorithm in the context of particle based codes. This technique of grid-based ray-tracing has been only used in grid-based codes to our knowledge (e.g. FLASH, ENZO). See Sec. 2.3.2 for more details.

**Advanced Spectral Leakage (ASL) [3]:** This is a novel approximate neutrino treatment that computes the neutrino cooling rates by interpolating local production and diffusion rates (relevant in optically thin and thick regimes, respectively) separately for discretized values of the neutrino energy. Neutrino trapped components are also modeled, based on equilibrium and timescale arguments. The better accuracy achieved by the spectral treatment allows a more reliable computation of neutrino heating rates in optically thin conditions. See Sec. 2.3.3 for more details.

**Isotropic Diffusion Source Approximation (IDSA) [4]:** IDSA is a more sophisticated scheme in which the distribution function of the neutrinos is decomposed into two components: trapped and streaming. Their separate evolution equations are coupled by a source term that converts trapped particles into streaming particles. The efficiency of the scheme, when compared with a more detailed solution of the Bolztmann equation, results from the freedom to use different approximations for each particle component.

**TRAPHIC (TRAnsport PHotons In Cones) [5]:** TRAPHIC is designed to model radiation from multiple sources efficiently. An example problem is to model the heating and ionization of gas in the inter-galactic medium during the stage of rapidly increasing galaxy formation in the first billion years of the Universe. The TRAPHIC module has been ported and used in post-processing mode between GADGET3 and GASOLINE in recent work on galaxy formation within our collaboration; see [6].

*2.2. Implementation details*

In the following we outline some details of the DIAPHANE library implementation.

*2.2.1. Library*

The library is currently hosted in a Bitbucket repository. The attached link[2] points to the repository wiki, and will also contain the public release of DIAPHANE in 2018 following a trial of validation and production runs conducted within the collaboration; v1.0 will consist of the first three modules (FLD, STARRAD, ASL) along with full documentation. The code will be published in the Computer Physics Communications (CPC) program library upon initial release.

The developmental focal points of the library are:

---

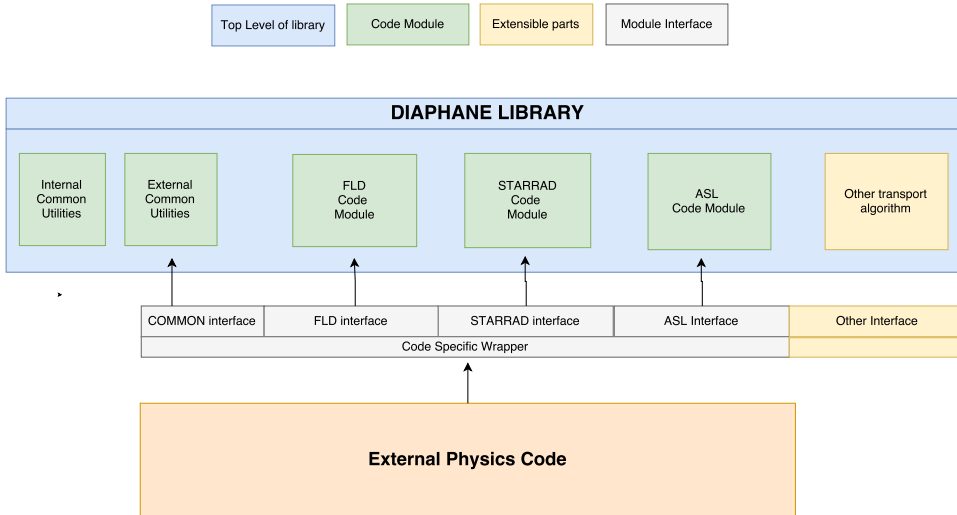[2]https://bitbucket.org/diaphane/diaphane-library

Figure 1: Overall structure of the Diaphane Library. Each module of the library contains an independent algorithm for radiation or neutrino transport. The "Code Specific Wrapper" is the common API, and acts as a direct interface to a simulation code. This wrapper, once called from a simulation code, then calls any module-specific interfaces (e.g., the FLD interface, the ASL interface, etc.), which contain any auxiliary functions needed to pass additional data to the modules that have been requested. The special internal and external "Common Utilties" layers contain routines that are used by multiple modules, and are accessed from the code specific wrapper via the special "common interface".

- *modular*: Because there are a multitude of RT/NT algorithms and approximations, each having advantages or disadvantages in terms of accuracy and efficiency for specific physical conditions, it is convenient to design DIAPHANE with a modular structure, so that adding new functionality is reasonably easy.

- *extensible*: Namely, we can add additional functions, support additional computational architectures, and couple with additional hydrodynamical codes – all with minimal effort by exploiting the already existing framework.

- *maintainable*: Well defined structure (Fig. 1), naming conventions, focus on code clarity, GIT-based version control.

- *portable*: Multi-platform, support for multiple compilers, with a view toward planned accelerator support.

- *robust*: Objective and efficient validation protocols.

The functional interactions between the library and a hydrodynamic code utilizing it are illustrated in Fig. 2. The core algorithmic layer of the library contains all the RT/NT physics modules, while the wrapper interface layer communicates with the main hydrodynamics code and with the module in the core physics layer. When the interface layer is called by the hydrodynamics code, it then calls each RT/NT physics module that has been specified as part of the simulation. The user will need to modify the wrapper layer to match the data structures and parallel methodology of their simulation code. The top-level "code specific wrapper" acts as a commoon API that is called from a hydrodynamics code; it will call a code and module-specific interface to any module to be used. This layer will contain any auxiliary functions needed by the specific module to interface with the hydrodynamics code. We provide examples of this layer for a number of simulation codes, currently GAGDET2 [7], ChaNGa/GASOLINE [8, 9], and SPHYNX [10], all three being smoothed particle hydrodynamics (SPH) codes. Our goal is to make it as simple as practical to port the library to a new simulation code. For routines that only require neighbor information, this is mainly a matter of passing a neighbor list and converting units. However, for other algorithms that require information across domain boundaries, the module-specific interface layer must be made aware of how to access information on other domains. Implementing this layer will require a reasonable working knowledge of the hydrodynamic code, but the examples already in the library will assist as a recipe for this process. Once the interface layer has been written for a hydrodynamic code to use a particular module, the process of building a new library module from an existing physics routine should be a relatively simple process, which we can provide assistance with. We hope that, over time, the community will add RT/NT modules to the library. We also note that although we focus on utilizing the library during live simulations, it can also be used for pure post-processing calculations, if desired.

In our effort to ensure portability we have tested that the library, implemented in C/C++, can interface effectively with both FORTRAN and C based codes. SPHYNX is a FORTRAN code while GADGET and GASOLINE are C codes. We also plan to soon support ChaNGa, a closely related offshoot
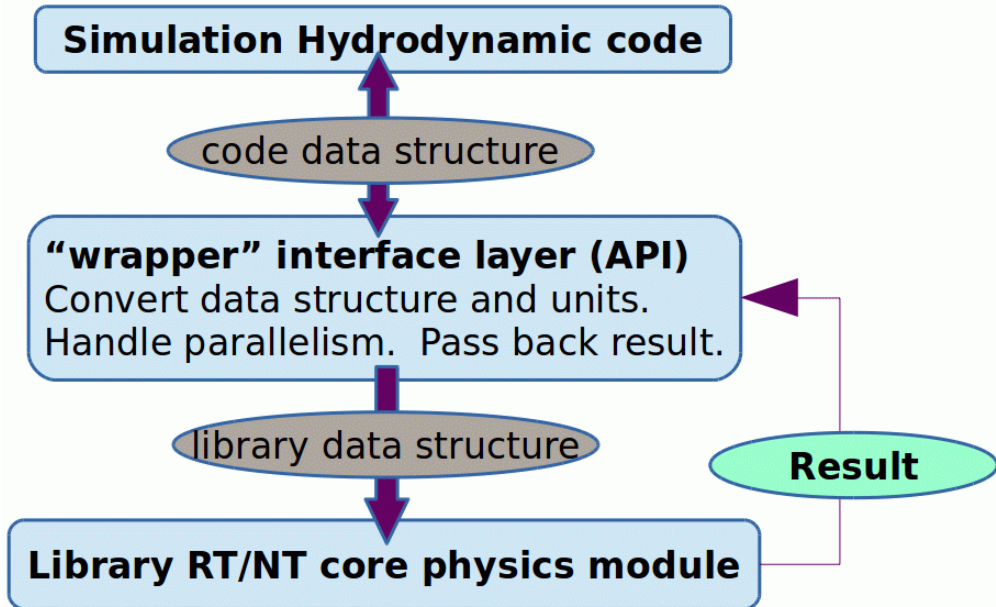
Figure 2: Functional schematic of the DIAPHANE Library showing the relation between the library and the simulation code. The wrapper layer contains the API to be called by a simulation code. This layer translates the data structures of the main simulation code into those used by the library functions. It has two levels. The top-level code specific wrapper interface layer (API) is called from the simulation code; the top-level wrapper handles translations that are common to all modules, and it calls a second level module-specific interface for each requested module. This module-specific interface layer contains auxiliary functions needed translate and pass any additional data that is needed by a specific module. The core physics module layer contains all of the physics of the library and utilizes data structures that are standard to the library.

of GASOLINE, written mainly in C++. For every interface and shared data structure or procedure, we use the iso_c_binding module to ensure compiler-independent functionality. This standard C-FORTRAN interface is necessary because, among other reasons, different compilers will align data structures in memory differently.

In the end, calls to any library module are made via a common API defined in the interface layer, which then triggers a call to each RT/NT module that has been specified. We can also provide a template upon request, to use the library coupled with a specific code, so that the required changes to the underlying hydrodynamical code are minimal. Nevertheless, in some cases the DIAPHANE library may require the hydrodynamical code to cal-

culate and carry some additional data that is not usually part of the main calculation. For example, an independent adaptive timestep length for radiation, discussed in the following paragraph, or the energy gradient, needed by FLD, is convenient to store as an additional per particle variable.

An important aspect regarding efficiency is that we also implemented the possibility of allowing DIAPHANE to have an independent timestep, since the radiation timescales are usually much shorter than the hydrodynamic timescale. This allows the radiation calculations to be decoupled from the hydrodynamical calculations. One benefit is that the hydrodynamic calculations can be done at a much lower frequency if the simulated scenario and conditions allow it. Another benefit is that the radiation calculations have the possibility to be carried out on separate hardware, opening the opportunity to exploit accelerators exclusively for the much more costly RT/NT calculations.

The first prototype module (FLD) of the library utilizes a relatively brute-force approach, looping through each hydrodynamic element in the wrapper layer and applying the library algorithms one fluid element at a time. This simple strategy means that only a small amount of information must be passed to the core library module, namely the properties of that element and its neighbors. This means that the first library version had no integrated parallelism; communication was instead handled by the parallel functionality of the main simulation code. This strategy was sufficient for the relatively simple FLD module. However, to implement the more complex STARRAD algorithm, it was necessary to provide means of accessing information across computational domains during the propagation of rays across the simulation volume. To accomplish this, we have utilized an abstraction layer for parallelism called the Machine Dependent Layer (MDL), which is inherited from one of the partipating codes GASOLINE. The library in its final form will rely on MDL for communication, the portability and flexibility of which will be crucial for its success. This paper focuses on the implementation of the modules and the general structure of the library while the parallel environment will be described in detail in a paper that will accompany the first public release of the code.

### 2.3. Algorithmic descriptions

We next describe the implementation of the three highlighted RT/NT modules that will be part of the first public release, with a focus on the relevant physics of the algorithms. This includes FLD as well as the two

most recently incorporated modules, STARRAD and ASL, which are currently undergoing testing and validation. The remaining modules of those that we have already mentioned in §2.1 are at earlier stages in development, and will be added later.

### 2.3.1. Flux-Limited Diffusion Module

FLD models radiation transport through gas using a photon diffusion approximation [11]. It has been adapted to the SPH method following [1, 2]. The FLD energy rate of particle $a$ is:

$$\dot{u}_a = \sum_b \frac{4m_b}{\rho_a \rho_b} \frac{k_a k_b}{k_a + k_b} \left(T_a - T_b\right) \frac{\mathbf{r}_{ab} \cdot \nabla W}{|\mathbf{r}_{ab}|^2} \tag{1}$$

summed over all neighbors $b$ of mass $m_b$, separated by the vector $\mathbf{r}_{ab}$; $T_a$ and $\rho_a$ are the temperature and density, respectively. FLD energy flows along the local energy gradient, determined from the kernel gradient $\nabla W$. The thermal conductivity term,

$$k_a = \frac{16\sigma}{\rho_a \kappa_a} \lambda_a T_a^3, \tag{2}$$

determines the effective diffusion coefficient; $\sigma$ is the Stefan-Boltzmann constant. The flux limiter ($\lambda$) sets the rate of energy flux in the energy equation. For optically thick gas, $\lambda$ matches that for the diffusion limit; for optically thin gas, $\lambda$ limits energy flux to the speed of light. Opacity, $\kappa_a(\rho, T)$, is interpolated from a table of Rosseland mean opacities. Finally, an FLD timestep constraint can be imposed such that

$$dt_{FLD} \leq \eta_{FLD} \frac{u_a}{\dot{u}_{aFLD}}, \tag{3}$$

and similar timestepping can be imposed for the other radiative modules.

### 2.3.2. STARRAD Module

The STARRAD module is based upon a ray casting technique to describe stellar irradiation of the surrounding environment by solving the radiative transfer equation currently considering only absorption:

$$I_{x1} = I_{x0} \exp(-\tau(x0, x1)) \tag{4}$$

I.e. intensity at position X1 is equal to the intensity at position X0 reduced by the exponent of the optical depth between positions X0 and X1, Where optical depth $\tau$ is:

$$\tau(x0, x1) = \int\limits_{x0}^{x1} \alpha(x) \ dx \tag{5}$$

bit and $\alpha$ (absorption coefficient) consists of the following, where $\kappa$ is opacity, and $\rho$ is the density at position x:

$$\alpha = \kappa\rho(x) \tag{6}$$

In order to calculate the direct radiative effect exerted by a heating source on its surroundings, a sphere of influence around the source is decomposed into pixels via HEALPix [12] subdivision (Fig. 3), and a ray allocated per pixel. Three tunable parameters control the resolution of spherical grid: the density of the ray distribution (synonymous with the resolution of the HEALPix distribution), the total length of each ray, and the length of each ray segment.

The algorithm is then structured in three key phases (Fig. 4):

1. **Optical depth**: Sample the volume for optical depth contributions at discrete points along each ray, stored in ray segments. For SPH this is accomplished by iterating once through the particles, calculating particle-ray intersections and spreading the particle optical depth across the affected ray segments.
2. **Ray integration**: Integrate optical depth contributions and solve the radiative transport equation along each ray, resulting in a radiation intensity field surrounding the source.
3. **Heating**: Apply the radiation intensity field to heat the medium by the appropriate amount of energy absorption, as determined by the optical depth calculation.

The STARRAD module is currently being tested within the GASOLINE code, and so is parallelized through use of the Machine Dependent Layer (MDL) library, which serves as an abstraction layer for parallelism and is used by GASOLINE). This approach is in line with the future direction of the DIAPHANE library (See Sect. 4).

The parallel STARRAD model evenly distributes ray groups across processing elements (PEs). Ray segments are fetched on demand from remote PEs to write optical depth contributions, and read the intensity field for medium heating. STARRAD makes use of fast caching mechanisms available in MDL to mitigate communication overhead. The HEALPix decomposition allows rays to be simply indexed

to maintain spatial coherency when distributed across processors. Rays are integrated locally on each PE to convert the accumulated optical depth into a radiative intensity field.

Referring to the planet forming disk example of Sect.1, in order to self-consistently model the evolution of gas surrounding a protostar, STARRAD can be used in a hybrid manner, similar to Klassen et al [13], where each time-step exploits multiple methods of calculating radiative transfer. Flux-limited diffusion, because it is well-suited for the optically thick environments of forming stellar systems, is the first module that we are coupling with STARRAD. This enables us to split the radiation field into direct and indirect components, allowing higher accuracy in the evolution of the gas properties [14]. In this case, we model direct heating of the gas by the protostar's light with STARRAD and subsequently use the FLD module to follow diffusion of that heat in the optically thick disk, allowing the temperatures throughout the disk to be modeled much more realistically.

### 2.3.3. ASL Module

The Advanced Spectral Leakage (ASL) method [3] is an approximate treatment for neutrinos and anti-neutrinos of all flavors, where we conjugate the usual positive aspects associated with leakage schemes (mainly, the reduced computational cost and flexibility), with an improved accuracy, obtained using a spectral approach (i.e., solving different leakage schemes, for different energy bins), modeling a neutrino trapped component in the optically thick region, and including a consistent absorption term obtained from the spectral cooling rates. ASL works in the following way:

1. compute the corresponding spectral emissivity, absorptivity or scattering rate at each fluid element position. The total emissivity and absorptivity are computed as the sum over all considered neutrino processes.
2. compute the local total mean free path of the neutrinos $\lambda_{\nu,tot}$ and the energy mean free path $\lambda_{\nu,en}$ as a geometrical average between the mean free path due to all absorption processes and that due only to highly inelastic processes (i.e. processes where the energies of the incoming and outgoing neutrinos is expected to differ considerably).
3. perform path integrals of both mean free paths to obtain the corresponding local optical depths $\tau_{\nu,tot}$ and $\tau_{\nu,en}$. According to their values we distinguish four different regimes: equilibrium-diffusive ($\tau_{\nu,tot} \gg 1$ and $\tau_{\nu,en} \gtrsim 1$), diffusive ($\tau_{\nu,tot} \gg 1$ but $\tau_{\nu,en} \lesssim 1$), semi-transparent ($\tau_{\nu,tot} \sim 1$), and free-streaming ($\tau_{\nu,tot} \lesssim 1$).
4. compute the trapped component variation rates of neutrino abundances and energies from the reconstructed neutrino distribution function adequately corrected by a cutoff at regimes other than equilibrium-diffusive.
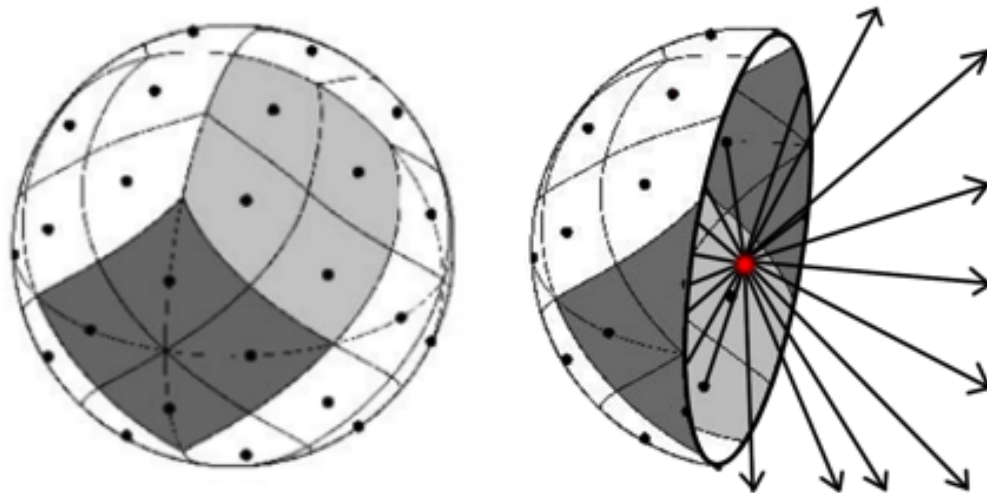
Figure 3: HEALPix decomposition of sphere around heating source



1.

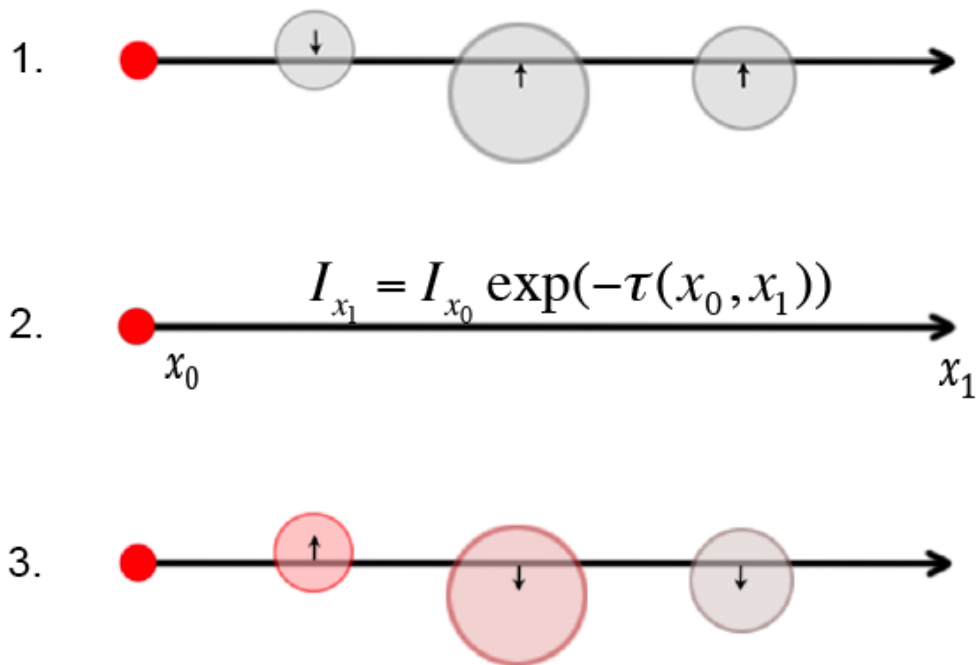2. $$I_{x_1} = I_{x_0} \exp(-\tau(x_0, x_1))$$
$x_0$ $x_1$

3.

Figure 4: Three phases of STARRAD algorithm; optical depth, ray integration, and heating, respectively. $\tau$ in the exponential refers to optical depth between points $X_0$ & $X_1$

5. evaluate the change of energy due to neutrino cooling and absorption, including and effective method to evaluate heating rates in optically thin regions based on the local spectral neutrino density.

In the end, ASL provides rates of change of specific energy, neutrino abundances and energies, and a neutrino pressure that can be added to the momentum equation in order to take into account momentum transfer from the neutrino field to the matter field.

ASL has been implemented in the 1D code AGILE [15], and the 3D codes FISH [16] (Cartesian mesh) and SPHYNX (Cartesian SPH). It is worth to note that the implementation of the (more detailed) IDSA neutrino treatment will easily follow from the current ASL implementation. The reason for this is that IDSA follows the same structure presented above, but the last two steps. Instead of steps 4. and 5., IDSA constructs a streaming particle background field from all sources of the previous time step/iteration and solves a diffusion equation that locally takes these streaming neutrinos into account to tailor the interpolation between the optically thick and thin regimes. Both methods, ASL and IDSA, have been extensively validated against BOLZTRAN [17, 18], that solves the Boltzmann transport equation in 1D without assuming any approximation [3, 4].

We are currently working in the development of the wrapper layer for this module, to be followed by a series of tests that can be compared with previous calculations performed in the core collapse scenario.

## 3. Validation

We now present results of our code validation tests. For each library module, we run a simple test problem using at least three different hydrodynamic codes. For FLD, diffusion through a periodic cube of gas provides a useful test for conditions relevant for planet forming disks around stars. The cube has uniform density $5 \times 10^{-13}$ g cm$^{-3}$ and length $6 \times 10^{15}$ cm. In Fig. 5 we show a projection of the particle distribution at four different times, from top-left to bottom-right respectively. Temperature is color-coded. As expected, the system evolves towards equilibrium, trying to homogenize the temperature profile via energy diffusion. A more detailed plot of the evolution of the temperature profile is shown in Fig. 6. There, we plot the results obtained by coupling DIAPHANE with three different SPH codes (GADGET, GASOLINE, and SPHYNX) at different stages of the evolution. We include a 1D semi-analytic simulation of diffusion as a verification of the FLD implementation. Aside from an initial offset during the early relaxation phase of the simulation, the 1D solution agrees well with the static 3D solution (run with GASOLINE with particle positions fixed in space). The effect of thermal pressure can be seen as a divergence between the static solutions and those that include

acceleration due to hydrodynamic forces. Fig. 7 confirms that the rates of energy change of simulation particles are in good agreement among different simulation codes.
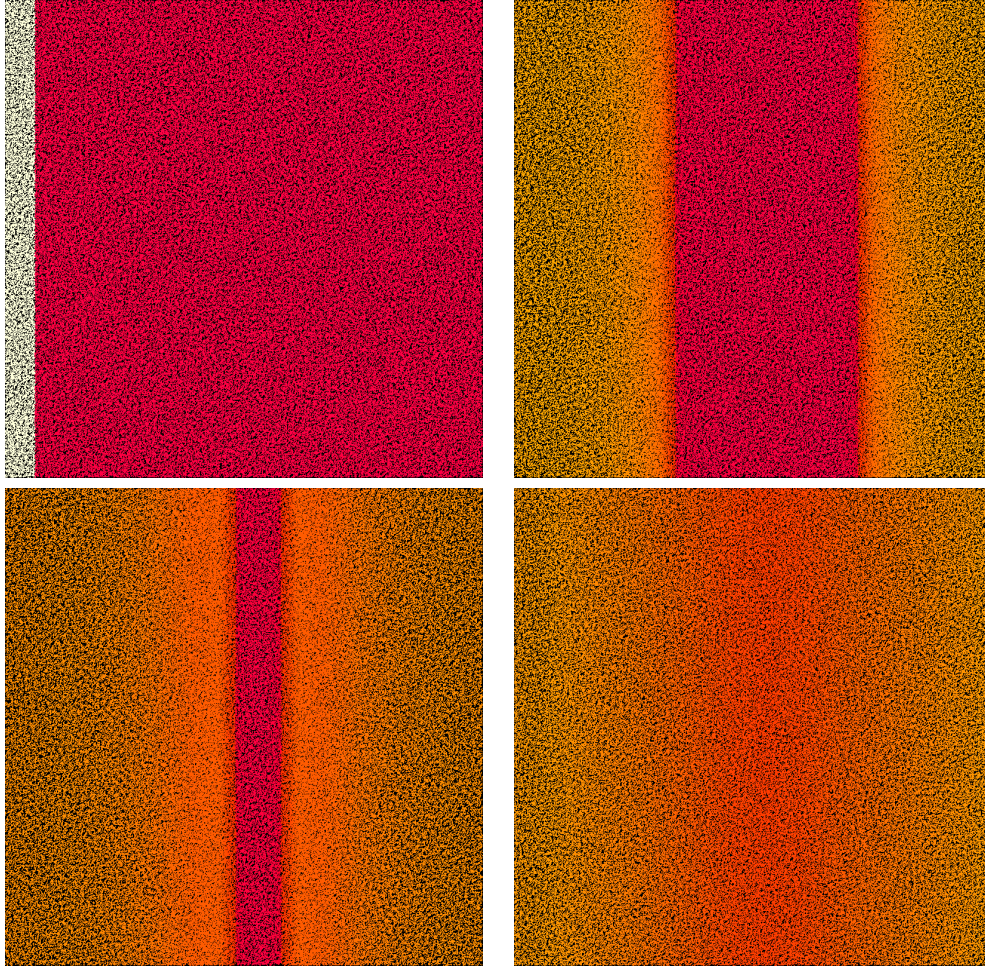


Figure 5: Image of cubical validation simulation for Flux-Limited Diffusion, colored by gas temperature. Initial conditions are a 100K periodic cube of cosmic gas with a 1000K layer. Time increases left to right and top to bottom; starting from upper left, the times shown are $t = 0$, 50, 100, and 300 $\times 10^8$s).

For STARRAD, we run a validation test consisting of a sun-like star enveloped in a gaseous disk. The disk is evolved statically, i.e. without hydrodynamical accelerations, from an initial disk shown in Fig. 8 (top/bottom left). Initial conditions are generated using an iterative procedure that enforces initial hydro-static equi-
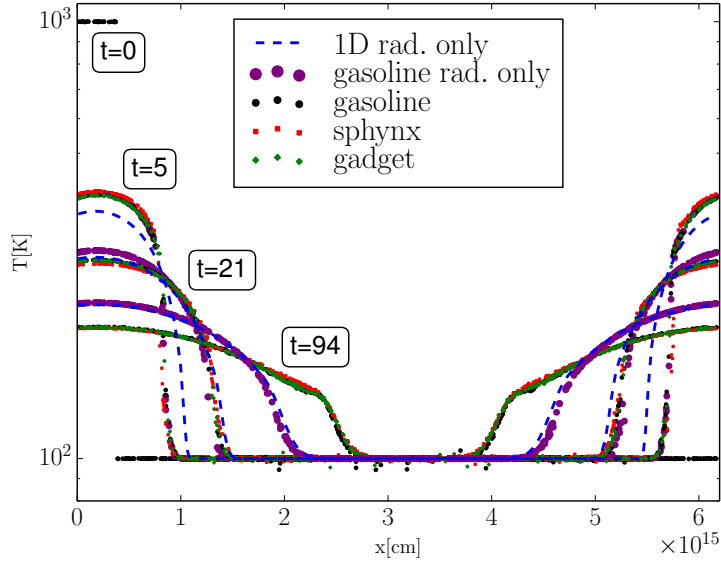
Figure 6: Cube test for Flux-Limited Diffusion. Initial conditions are a 100K periodic cube of cosmic gas with a 1000K layer. Plotted is a random 1 in 512 sampling of particles from each simulation. The three hydrodynamic codes agree with each other. The 1D semi-analytic solution does not include hydrodynamic forces. The 3D run radiation only case (i.e. static particle positions) agrees with the 1D case. The difference between the radiation-only simulation and the full hydrodynamic simulations at late times is due to thermal pressure. Time plotted is in units of $10^8$s.
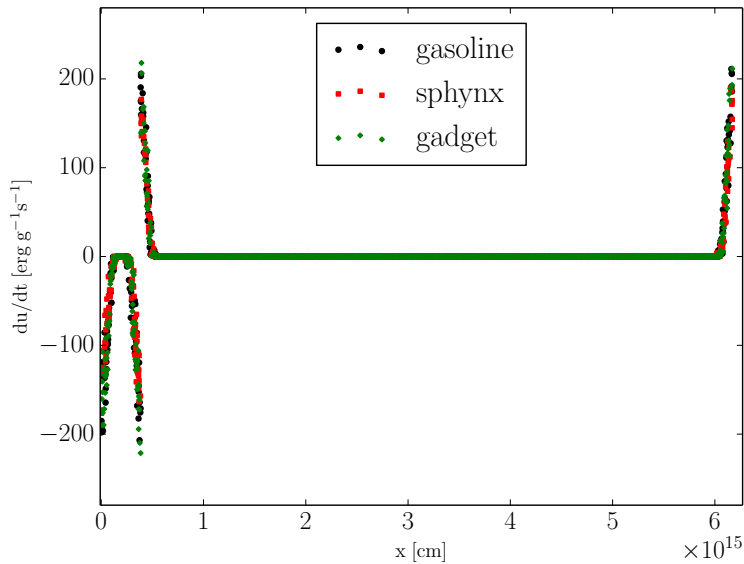
Figure 7: Cube test for Flux-Limited Diffusion heating/cooling rates verifying the independence of FLD rates with simulation code. Particles plotted are a random 1 in 128 sampling of the simulation. Shown at time $t = 0$, energy flows across the temperature discontinuity.
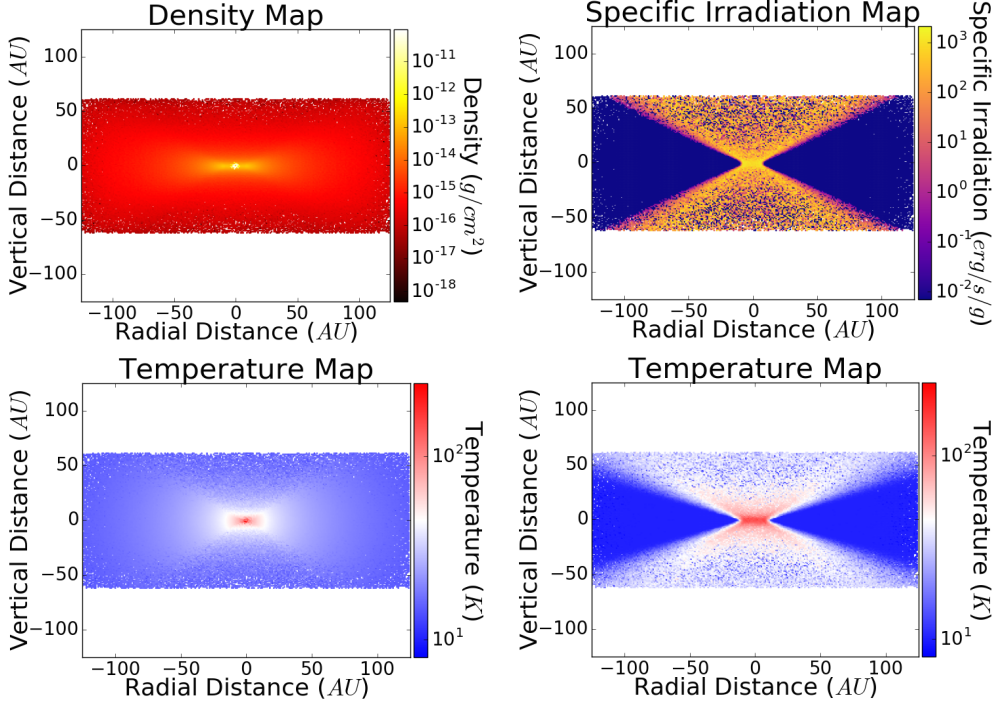
Figure 8: Color coded maps representing a STARRAD static irradiated disk test; initial density map (left, top), initial temperature map (left, bottom). Specific irradiation (energy per unit time per unit mass) during the first step (right, top), and temperature after 300 years evolution (right, bottom).

librium by taking into account all forces (pressure, gravitational and centrifugal, see Rogers & Wadsley 2011 [19]).

As STARRAD computes only heating, some form of cooling is required in order to evolve to thermal equilibrium. Whilst the FLD and STARRAD coupling process (see Sect 2.3.2) is on-going, we have paired STARRAD with a standard cooling module within GASOLINE [20] [21]. This cooling model adopts a local optical depth approach which recovers the optically thick and optically thin limits of the cooling rate (essentially equivalent to the diffusion limit employed by FLD for large optical depths). This takes advantage of the optical depth calculation performed by STARRAD, which will be a common module of the library also usable by other library modules.

Fig. 8 (right) shows the specific irradiation of the disk as calculated by STARRAD, i.e. the energy absorbed per unit time per unit mass, along with a temperature map after heating - the dense inner disk clearly shielding the outer disk in the midplane. The temperature profile in Fig.9 demonstrates that the STARRAD

module combined with cooling quickly reaches a steady-state to reasonable values for protoplanetary disks. The incline in temperature in the 6-10AU range results from high absorption reflecting the optical depth profile of the initial conditions. A plot of ray absorption in Fig. 10 confirms negligible absorption within a ˜6 AU radius, matching the density profile seen in Fig. 11. Beyond the 9 AU radius, fast cooling results in the subsequent temperature decline, this is expected to be more moderate when coupled with FLD.
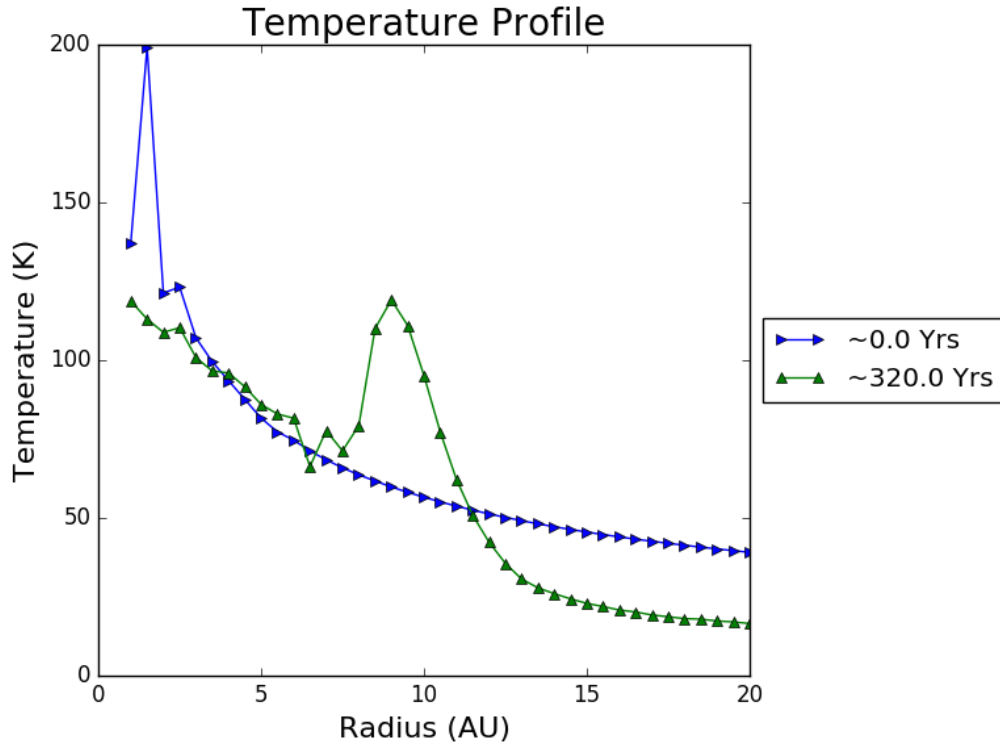


Figure 9: Temperature profile of the STARRAD static irradiated disk over ˜300 years of evolution, corresponding to ˜30 disk orbits at 10AU.

While the current tests utilize a static disk, to simplify validation of the radiative transport, STARRAD is also able to run with hydrodynamics and will be used in upcoming production runs; indeed in the current run the hydrodynamics are active (recomputing density/pressure at each step) however accelerations are turned off, hence a static disk. Note that the initial adjustment of the disk from time t=0 is due to initialization procedure (described above) applied to enforce hydrostatic equilibrium to the disk, whose inner region consists of few particles due to a hole
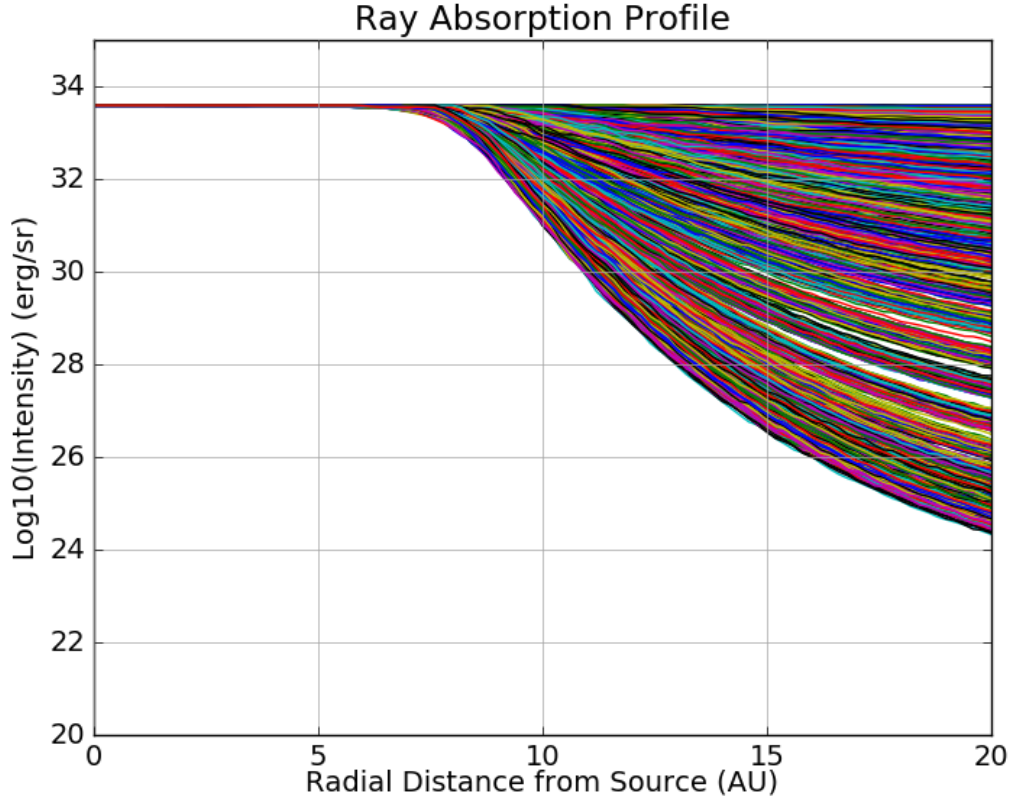
## Ray Absorption Profile



Figure 10: Absorption profiles for rays. The initial low density below 6AU provides negligible absorption, while above 6AU the energy (per unit solid angle) of each ray steadily declines as it is absorbed by the denser regions of the disk.

by construction (Fig 11, < 5AU). We have verified the accuracy of STARRAD in terms of energy conservation and heating rates; more rigorous verification of its accuracy is still underway. Full details of the STARRAD module will be the subject of a forthcoming dedicated paper [in preparation].

## 4. Conclusions and future work

We have presented the first results of coupling the DIAPHANE library to different hydrodynamics codes for simulating diffusion through a 3D-periodic uniform distribution of gas. The very good agreement among all codes is encouraging and proves that the porting of the library is successful, independently of the underlying hydrocode. Additional modules like STARRAD (a radiative transfer solver based
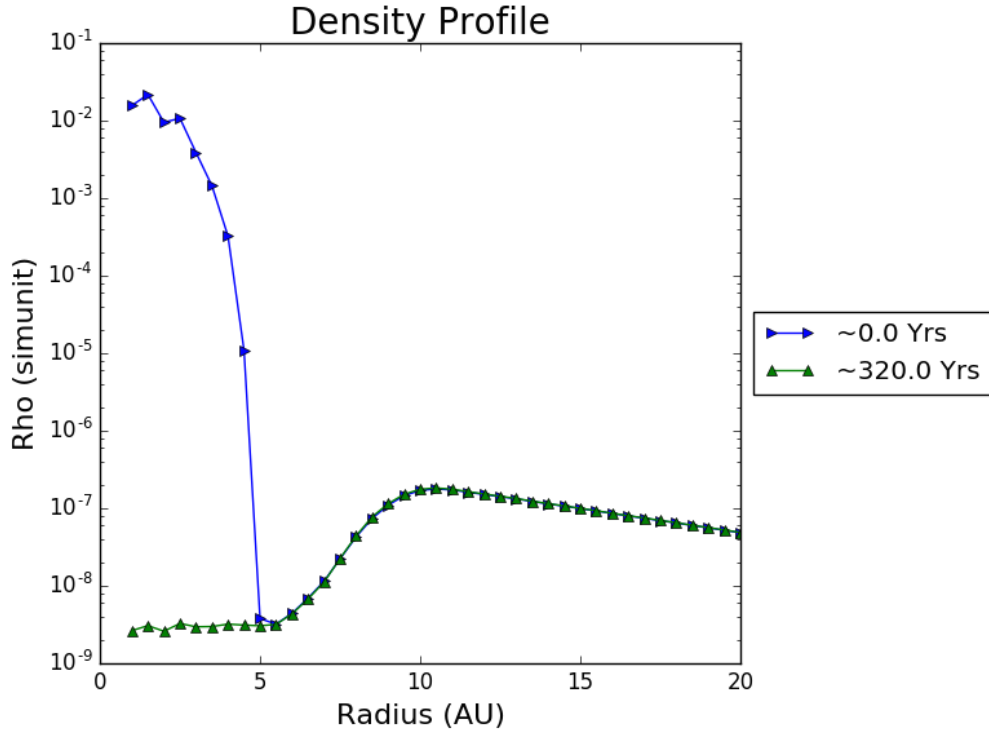
19

Figure 11: Density profile of the STARRAD initial conditions and irradiated disk after ˜300 years of evolution. This does not change significantly after an initial readjustment phase in the sparse inner region of the initial disk.

on ray casting) and ASL (a spectral leakage treatment for neutrino radiation) are currently being implemented and tested.

The DIAPHANE library allows multiple radiation and neutrino transport algorithms to be used by a generic hydrodynamic code by calling portable modules using a common API. This functionality has the potential for a transformative improvement in the modelling of complex astrophysical processes on scales spanning a vast range from kilometer scale black holes to star and galaxies and beyond.

The current implementation that calls the library modules using a fluid element basis will be improved to allow optimization by vectorization and accelerators such as GPUs or MIC processors. Such work is necessary to use the library on future Exascale machines. Having a GPU capable code, ChaNGa, already supported by

our collaboration will be conducive to future GPU porting work. ChaNGa[3] is the GASOLINE hydro code, which is part of the library, re-engineered to work with a CHARM++ parallel environment for better parallel performance owing to dynamic load balancing. Future versions of the library are planned to be capable of computing on a spatially local domain basis with a single call rather than just on one fluid element at a time. This would allow an accelerator to compute the RT/NT processes on its own set of particles independently of the main simulation code. Further optimization would be possible if the library handles operations like neighbor finding internally. The strategy for this development will be to extend our use of the Machine Dependent Layer (MDL), the parallel library already used by STARRAD. MDL has recently been updated to be efficient at hybrid parallel computing MPI and pthreads on node and GPU and was used for a two trillion particle cosmological gravity simulation [22]. Using the MDL is meant to simplify the work of porting the library to a new code (and onto new HPC hardware) by abstracting the parallel communication out of the library physics routines and into the interface layer. MDL is not limited to particle-based codes; a general MDL that interfaces also with grid-based codes such as RAMSES is at an advanced state of development, as part of an effort lead by Joachim Stadel.

We plan to extend our simulation support to grid-based codes, beginning with RAMSES. The development effort will be mainly focused on the common API layer and, even more, in its auxiliary wrapper functions. The prior development of an MDL version of RAMSES will help with this effort. In some simple algorithms such as FLD, we plan to provide both a grid-based and a particle-based module, depending on which type of code is running. However, some particle-based methods may not translate easily to grid-based methods. We also will allow the option of converting grid cells into pseudo-particles, so that a particle-based method can be applied.

A future goal is to incorporate adaptive coupling of the multiple modules. One method that we are exploring for this coupling is the Fuzzy Domain Method (FDM) [23]. This is a way of computing an approximate solution in intermediate regimes where no algorithm is ideal and can provide a better solution than simple interpolation. Conceptually, FDM is well suited to couple solutions formally analogous to those in moment-based methods, namely for which there is no underlying discretization of the radiation field using rays or domains but rather the radiation field is treated as continuum. Hence one can imagine providing two limiting solutions in the optically thin and optically thick regime, suited for certain physical problems, and use FDM to interpolate. FDM has already been utilized for

---

[3]http://www-hpcc.astro.washington.edu/tools/changa.html

neutrino transport in core-collapse supernova simulations with regard to neutrinos that are trapped (i.e. high optical depth) or stream outward (i.e. low optical depth) from the explosion [24]. Often intermediate regimes will have no accurate technique available. This is where the Fuzzy Domain Techniques hold promise beyond what a simple interpolation of two algorithms would yield. Although it is still under development, FDM will be an important feature of the library. We note that FDM is a novel approach that will enable multiple RT algorithms to operate within a running simulation in an automated and mathematically motivated way. The switching and coupling between different modules will be handled internally by the library, using for example, the local optical depth for the switching or coupling.

Future work, which will go in the second release of the library, will see the implementation of an M1 moment-based method (originating from the existing implementation in RAMSES; [25]) and the implementation of TRAPHIC. With these and the existing modules, we will cover all the main categories of RT approximations and schemes available in astrophysics. This means that when another team will want to implement a new scheme this will inevitably have commonalities with one of those already implemented, enabling to exploit the already existing API and auxiliary functions and those requiring little added work. Hence our strategy is by construction aimed at maximizing the output to the community at large.

Development of the neutrino transport modules is still ongoing. Nevertheless, the ASL scheme, as well as the IDSA method, have been already ported to different SPH codes and mesh codes, currently both in production stages. This knowledge is transferable to their inclusion in DIAPHANE and we are benefiting considerably from it.

## 5. Availability

The current version of the Diaphane library is in early use exclusively within the collaboration. A first public version is expected to be released as an open-source code in 2018, to be submitted for publication to *Computer Physics Communications* as a *Computer Programs in Physics* paper.

## 6. Acknowledgments

# References

[1] P. W. Cleary, J. J. Monaghan, Conduction modelling using smoothed particle hydrodynamics, J. Comput. Phys. 148 (1) (1999) 227–264. `doi:10.1006/jcph.1998.6118`.
URL `http://dx.doi.org/10.1006/jcph.1998.6118`

[2] L. Mayer, G. Lufkin, T. Quinn, J. Wadsley, Fragmentation of Gravitationally Unstable Gaseous Protoplanetary Disks with Radiative Transfer, ApJ661 (2007) L77–L80. `arXiv:astro-ph/0606361`, `doi:10.1086/518433`.

[3] A. Perego, R. M. Cabezón, R. Käppeli, An Advanced Leakage Scheme for Neutrino Treatment in Astrophysical Simulations, ApJS223 (2016) 22. `arXiv:1511.08519`, `doi:10.3847/0067-0049/223/2/22`.

[4] M. Liebendörfer, S. C. Whitehouse, T. Fischer, The Isotropic Diffusion Source Approximation for Supernova Neutrino Transport, ApJ698 (2009) 1174–1190. `arXiv:0711.2929`, `doi:10.1088/0004-637X/698/2/1174`.

[5] A. H. Pawlik, J. Schaye, Multifrequency, thermally coupled radiative transfer with TRAPHIC: method and tests, MNRAS412 (2011) 1943–1964. `arXiv:1008.1071`, `doi:10.1111/j.1365-2966.2010.18032.x`.

[6] V. Tamburello, A. Rahmati, L. Mayer, A. Cava, M. Dessauges-Zavadsky, D. Schaerer, Clumpy galaxies seen in H $\alpha$: inflated observed clump properties due to limited spatial resolution and sensitivity, MNRAS468 (2017) 4792–4800. `arXiv:1610.05304`, `doi:10.1093/mnras/stx784`.

[7] V. Springel, The cosmological simulation code GADGET-2, MNRAS364 (2005) 1105–1134. `arXiv:astro-ph/0505010`, `doi:10.1111/j.1365-2966.2005.09655.x`.

[8] P. Jetley, F. Gioachin, C. Mendes, L. V. Kale, T. R. Quinn, Massively parallel cosmological simulations with ChaNGa, in: Proceedings of IEEE International Parallel and Distributed Processing Symposium 2008, 2008.

[9] J. W. Wadsley, J. Stadel, T. Quinn, Gasoline: a flexible, parallel implementation of TreeSPH, New A9 (2004) 137–158. `arXiv:astro-ph/0303521`, `doi:10.1016/j.newast.2003.08.004`.

[10] R. M. Cabezón, D. García-Senz, J. Figueira, SPHYNX: an accurate density-based SPH method for astrophysical applications, A&A606 (2017) A78. `arXiv:1607.01698`, `doi:10.1051/0004-6361/201630208`.

[11] P. Bodenheimer, H. W. Yorke, M. Rozyczka, J. E. Tohline, The formation phase of the solar nebula, ApJ355 (1990) 651–660. `doi:10.1086/168798`.

[12] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, M. Bartelmann, HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere, ApJ622 (2005) 759–771. `arXiv:astro-ph/0409513`, `doi:10.1086/427976`.

[13] M. Klassen, R. Kuiper, R. E. Pudritz, T. Peters, R. Banerjee, L. Buntemeyer, A General Hybrid Radiation Transport Scheme for Star Formation Simulations on an Adaptive Grid, ApJ797 (2014) 4. `arXiv:1410.4259`, `doi:10.1088/0004-637X/797/1/4`.

[14] S. D. Murray, J. I. Castor, R. I. Klein, C. F. McKee, Accretion disk coronae in high-luminosity systems, ApJ435 (1994) 631–646. `arXiv:astro-ph/9405016`, `doi:10.1086/174842`.

[15] M. Liebendörfer, O. E. B. Messer, A. Mezzacappa, S. W. Bruenn, C. Y. Cardall, F.-K. Thielemann, A Finite Difference Representation of Neutrino Radiation Hydrodynamics in Spherically Symmetric General Relativistic Spacetime, ApJS150 (2004) 263–316. `arXiv:astro-ph/0207036`, `doi:10.1086/380191`.

[16] R. Käppeli, S. C. Whitehouse, S. Scheidegger, U.-L. Pen, M. Liebendörfer, FISH: A Three-dimensional Parallel Magnetohydrodynamics Code for Astrophysical Applications, ApJS195 (2011) 20. `arXiv:0910.2854`, `doi:10.1088/0067-0049/195/2/20`.

[17] A. Mezzacappa, S. W. Bruenn, A numerical method for solving the neutrino Boltzmann equation coupled to spherically symmetric stellar core collapse, ApJ405 (1993) 669–684. `doi:10.1086/172395`.

[18] A. Mezzacappa, S. W. Bruenn, Stellar core collapse - A Boltzmann treatment of neutrino-electron scattering, ApJ410 (1993) 740–760.

[19] P. D. Rogers, J. Wadsley, The importance of photosphere cooling in simulations of gravitational instability in the inner regions of protostellar discs, MNRAS414 (2011) 913–929. `doi:10.1111/j.1365-2966.2011.18523.x`.

[20] M. Galvagni, T. Hayfield, A. Boley, L. Mayer, R. Roškar, P. Saha, The collapse of protoplanetary clumps formed through disc instability: 3D simulations of the pre-dissociation phase, MNRAS427 (2012) 1725–1740. `arXiv:1209.2129`, `doi:10.1111/j.1365-2966.2012.22096.x`.

[21] J. Szulágyi, L. Mayer, T. Quinn, Circumplanetary discs around young giant planets: a comparison between core-accretion and disc instability, MNRAS464 (2017) 3158–3168. `arXiv:1610.01791`, `doi:10.1093/mnras/stw2617`.

[22] D. Potter, J. Stadel, R. Teyssier, PKDGRAV3: beyond trillion particle cosmological simulations for the next era of galaxy surveys, Computational Astrophysics and Cosmology 4 (2017) 2. `arXiv:1609.08621`, `doi:10.1186/s40668-017-0021-1`.

[23] M. J. Gander, J. Michaud, Fuzzy domain decomposition: a new perspective on heterogeneous dd methods, 2014.

[24] H. Berninger, M. Ohlberger, O. Sander, K. Smetana, Unsaturated subsurface flow with surface water and nonlinear in- and outflow conditions, Mathematical Models and Methods in Applied Sciences (M3AS) 24(5) (2014) 901–936. `arXiv:1301.2488`.

[25] J. Rosdahl, J. Blaizot, D. Aubert, T. Stranex, R. Teyssier, RAMSES-RT: radiation hydrodynamics in the cosmological context, MNRAS436 (2013) 2188–2231. `arXiv:1304.7126`, `doi:10.1093/mnras/stt1722`.