

# Towards a new generation of multi-dimensional stellar evolution models: development of an implicit hydrodynamic code

M. Viallet<sup>1,2</sup>, I. Baraffe<sup>2,1</sup>, R. Walder<sup>1</sup>

<sup>1</sup> École Normale Supérieure, Lyon, CRAL (UMR CNRS 5574), Université de Lyon 1, France  
e-mail: [maxime.viallet@ens-lyon.fr](mailto:maxime.viallet@ens-lyon.fr)

<sup>2</sup> Physics and Astronomy, University of Exeter, Stocker Road, Exeter, UK EX4 4QL

Received; accepted

## ABSTRACT

This paper describes the first steps in the development of a new multi-dimensional time implicit code devoted to the study of hydrodynamical processes in stellar interiors. The code solves the hydrodynamical equations in spherical geometry and is based on the finite volume method. Radiation transport is taken into account within the diffusion approximation. Realistic equation of state and opacities are implemented, allowing study of a wide range of the problems characteristic of stellar interiors. We describe the numerical method and various standard tests performed to validate the method in detail. We present preliminary results devoted to describing stellar convection. We first performed a local simulation of convection in the surface layers of a A-type star model. This simulation tested the ability of the code to address stellar conditions and to validate our results, since they can be compared to similar previous simulations based on explicit codes. We then present a global simulation of turbulent convective motions in a cold giant envelope, covering 80% in radius of the stellar structure. Although our implicit scheme is unconditionally stable, we show that in practice there is a limitation on the time step that prevents the flow moving over several cells during a time step. Nevertheless, in the cold giant model we reach a hydro CFL number of 100. We also show that we are able to address flows with a wide range of Mach numbers ( $10^{-3} \lesssim M_s \lesssim 0.5$ ), which is impossible with an anelastic approach. Our first developments are meant to demonstrate that applying an implicit scheme to a stellar evolution context is perfectly thinkable and to provide useful guidelines for optimising the development of an implicit multi-dimensional hydrodynamical code.

**Key words.** Hydrodynamics - Convection - Methods: numerical - Stars: interiors

## 1. Introduction

With the advent of massively parallel computers and the development of advanced numerical techniques like adaptive mesh refinement and massive parallelisation, astrophysical fields like cosmology, star formation, or stellar/galactic environment studies have recently made remarkable steps forward. Despite this revolution in computational astrophysics, our physical understanding of stellar interiors and evolution still largely relies on one-dimensional calculations. Implicit 1D stellar evolution codes were widely developed during the past century, gaining more and more in sophistication through improvements in the input physics and implementation of an increasing number of complex physical processes such as time-dependent turbulent convection, rotation, or MHD processes. Their description, however, still mostly relies on simplified, phenomenological approaches, with a predictive power hampered by the use of several free parameters. These approaches have now reached their limits for understanding stellar structure and evolution. The development of multi-dimensional hydrodynamical simulations becomes crucial for progress in the field of stellar physics and for the enormous observational efforts aimed at producing data of unprecedented quality, as expected from the asteroseismological space projects COROT and Kepler.

Several efforts have been devoted to the development of 2D and 3D tools for studying some aspects and spe-

cific problems of stellar structure and evolution. Impressive progress and developments have been achieved in multi-dimensional hydrodynamics and MHD models of stellar atmospheres and photospheres (see e.g. Freytag et al. 1996; Stein & Nordlund 1998; Asplund et al. 2000; Bigot et al. 2006; Nordlund & Stein 2009). In stellar interiors, many studies of convection, rotation, and magnetic fields are performed with anelastic hydrodynamic solvers, which filter out sound waves and have the priceless advantage of not being restricted to the Courant-Friedrich-Lewy (CFL hereafter) time step limit, a severe limitation for explicit compressible hydrodynamic solvers (Brun & Toomre 2002; Guzik 2010, and references therein). Anelastic approaches, however, are restricted to the study of flows characterised by very low Mach numbers and small thermodynamic fluctuations from the background (see discussions in Almgren et al. 2006; Meakin & Arnett 2007a).

Other groups use 2D/3D explicit hydrodynamical simulations to follow the properties on short snapshots of turbulent convection, mixing processes, or nuclear burning in stellar interiors such as cores and burning shells of massive stars or during the core He flash in low-mass stars (Dearborn et al. 2006; Meakin & Arnett 2007b; Eggleton et al. 2008; Mocák et al. 2009; Arnett et al. 2009). However, owing to the intrinsic time-step limitation of explicit codes, simulating those processes on time scales that are significantly larger than the dynamical one and relevant to stellar evolution would be prohibitively expen-

sive. To overcome this problem, increasing efforts are now devoted to developing algorithms for low Mach-number, compressible stellar flows (Hujeirat & Thielemann 2009; Nonaka et al. 2010). However, many stellar regions and evolutionary phases are characterised by low-to-moderate Mach-number flows, which would ideally be described by fully time-implicit solvers.

We only know two multi-dimensional implicit codes applied to stellar physics problems, namely the 2D stellar evolution code ROTORC developed by Deupree (1990) and the 2D code VULCAN of Livne (1993). The work of Deupree is pioneering in the field of multi-dimensional implicit stellar evolution, with ROTORC applied to the study of stellar convection and rotation in stars of different masses. However, the numerical method involved in this code cannot compete with current, advanced numerical algorithms, because it is based on a non-conservative finite-difference scheme with the implicit solver as an extension to the 2D of the Henyey method. This method is widely used in 1D stellar evolution calculations but cumbersome (if not impossible) to parallelise, thus it significantly restricts the spatial resolution for multi-dimensional calculations. VULCAN is based on a two-step method consisting of an implicit Lagrangian step followed by an explicit remapping step on an arbitrary grid. The code has been essentially applied to studying core-collapse supernovae (see e.g. Livne et al. 2004; Burrows et al. 2007) and novae explosions (see e.g. Glasner & Livne 1995; Livne & Arnett 1995; Glasner et al. 1997; Livne 1999). Although VULCAN offers the possibility of solving advection implicitly, to our knowledge in these works the authors performed multi-dimensional computations by solving advection explicitly and radiation implicitly. Thus, we cannot gauge the capacity of this code to describe long-term stellar evolution problems, in general, and convective flows in various stellar interiors, in particular. Hujeirat (2005a,b) develops a multi-dimensional implicit MHD solver, but the applications are oriented more towards high-energy processes (e.g. accretion on a black hole). Finally, we mention the work of Toth et al. (1998) and Keppens et al. (1999), which describe the implementation and tests of various implicit methods for multi-dimensional computations of steady state and time-dependent problem in (magneto)hydrodynamic.

In this context, we started to develop a multi-dimensional time implicit code, guided by the motivation of improving the description of stellar interiors during different phases of evolution. The holy grail is to develop a tool that allows the 3D description of a complete star during thermal, nuclear, or accretion time scales that would be relevant to various stellar evolution phases. Our primary interests focus on the description of time-dependent convection in the envelope of pulsating stars like Cepheids, a long-standing problem in asteroseismology, or during the very early phases of stellar evolution (early pre-main sequence) following the hydrodynamical phase of a proto-star formation. Improving our description of these very early stages of evolution is crucial to a more accurate analysis of current observations and, in particular, to derivation of a key property for understanding star formation, namely the initial mass function (see e.g. discussion in Baraffe et al. 2002). For the above-mentioned problems, neither an anelastic nor a low-Mach approach is appropriate, since the convective flows are characterised by very low ( $M \ll 0.01$ ) to moderate ( $0.1 \lesssim M \lesssim 1$ ) Mach numbers from the deep interior

to the stellar surface. Also, a wide range of other astrophysically interesting problems can be addressed with such a tool, providing a wealth of applications on a long-term future and a revival of the field of stellar physics.

Our goal is exceedingly challenging, with a long way to go before achievement. This first paper describes our method and the first tests performed to describe convection in stellar interiors. Because implicit schemes are much more expensive in terms of CPU than explicit ones, part of the astrophysical, computational community casts doubts on the advantage and applicability of the former to any stellar physics problem. Our first developments are meant to demonstrate that the use of an implicit scheme in a stellar evolution context is perfectly thinkable. For this purpose, we present two examples of stellar convection for which we derive the exact CFL numbers and discuss the advantages and limits of an implicit approach. Even though the implicit method used throughout the paper has no stability limit on the time step (see Sect. 3.2.2), the time step cannot take arbitrarily high values for both technical and accuracy reasons. We show in this paper that both reasons are actually related and that the time step cannot grow much larger than the shortest time scale of the flow for crossing a grid cell. Technically this implies that the solution change is moderate between two time steps, insuring a safe convergence in the non-linear solver. In addition, this is also a good accuracy criterion, since advection over more than one grid cells is subject to strong numerical damping. Our first numerical experiments are meant to provide useful guidelines for developing an implicit multi-dimensional hydrodynamical code.

The paper is organised as follows. We describe in Sect. 2 our physical model and in Sect. 3 our numerical method. We present a test case in Sect. 4 and results for stellar models in Sect. 5. In Sect. 6, we conclude the paper and discuss future numerical advancements that are planned to optimise the code.

## 2. Equations

We solve the equations describing the evolution of density, momentum, and internal energy, taking into account external gravity and radiative diffusion:

$$\frac{\partial}{\partial t} \rho = -\nabla \cdot (\rho \mathbf{u}) \quad (1)$$

$$\frac{\partial}{\partial t} \rho e = -\nabla \cdot (\rho e \mathbf{u}) - P \nabla \cdot \mathbf{u} - \nabla \cdot \mathbf{F}_r + \frac{1}{2\rho\nu} \tau_{ij} \tau^{ij} \quad (2)$$

$$\frac{\partial}{\partial t} \rho \mathbf{u} = -\nabla \cdot (\rho \overline{\mathbf{u}\mathbf{u}} - \overline{\tau}) - \nabla P + \rho \mathbf{g}, \quad (3)$$

where  $\rho$  is the density,  $e$  the specific internal energy,  $\mathbf{u}$  the velocity,  $P$  the gas pressure,  $\mathbf{F}_r$  the radiative flux (see below),  $\mathbf{g}$  the gravitational acceleration, and  $\overline{\tau}$  the viscous stress tensor, whose components are given by

$$\tau_{ij} = 2\rho\nu \left( e_{ij} - \frac{1}{3} \delta_{ij} \nabla \cdot \mathbf{u} \right), \quad (4)$$

where  $e_{ij}$  is the strain rate tensor and  $\nu$  the viscosity coefficient. The expected value of the molecular viscosity in stellar interiors implies large Reynolds numbers characterising the flow (up to  $10^{12}$ ). It is therefore impossible to

model all scales of the flow, from the stellar scale down to the dissipation scale, on current generation of computers. As a result, any numerical simulations of stellar interiors should be interpreted in the large eddy simulation (LES) paradigm, in which only large-scale motions are resolved on the grid. In the standard LES, the effect of subgrid-scale motions (i.e. turbulence) is taken into account by introducing a so-called subgrid scale (SGS) model. A classical example is the effective viscosity coefficient proposed in Smagorinsky (1963). On the other hand, the MILES approach (for Monotone Integrated Large Eddy simulation, see e.g. Boris et al. 1992) relies solely on the numerical viscosity of the scheme, which results from discretisation, to mimic the subgrid scale dissipation. In our case, a standard LES approach should rely on an SGS model relevant for compressible flow in a stratified medium, which to our knowledge has not been designed yet. It is far from certain that the use of standard SGS models, mostly relevant for incompressible and homogenous flow, will improve our results. Therefore, we choose to follow the MILES strategy and all the computations of stellar convection presented in this paper are based on the advection-diffusion equations without explicit viscosity.

We solve the internal energy density equation. Some of the popular numerical codes that have been used to study stellar convection solve the entropy equation (e.g. ASH code, Pencil code), which is equivalent to the internal energy equation. A better approach to energy conservation would be to solve the total energy equation, as for instance in the COBOLD code (see Freytag and collaborators) and in the PROMPI code (see Meakin & Arnett 2007b). In the future we plan to implement a total energy equation solver, which might be more appropriate for problems like stellar pulsations.

For stellar calculations, we treat radiation transport in the diffusion limit approximation. This approximation is suitable for an optically thick region but becomes inaccurate in the photosphere and optically thin regions. In this framework, the radiative flux  $F_r$  writes as

$$\mathbf{F}_r = -k_{\text{rad}} \nabla T, \quad (5)$$

where the photon conductivity  $k_{\text{rad}}$  is given by

$$k_{\text{rad}} = \frac{4acT^3}{3\kappa\rho}, \quad (6)$$

with  $\kappa$  the Rosseland opacity of the gas, which is interpolated from the OPAL tables (Iglesias & Rogers 1996).

The equations are closed with the equation of state:

$$P = P(\rho, e), \quad T = T(\rho, e). \quad (7)$$

Our equation of state is tabulated and includes the partial ionisation of several chemical elements (determined by the Saha-Boltzmann equations) from hydrogen to silicon and takes Coulombian pressure into account. The chemical abundances are set to the solar values. For details see Aubert et al. (1996). We consider here stellar envelopes and therefore we do not take nuclear reactions into account, which concern the central region of the star.

For the geometry, we adopt spherical coordinates. We assume azimuthal symmetry so the only independent coordinates are  $r$  (the radius) and  $\theta$  (the colatitude). In general, the coordinates  $(r, \theta)$  span a domain  $[r_{\text{in}}, r_{\text{out}}] \times [\theta_1, \theta_2]$ .

We use periodic conditions in the tangential direction. For boundary conditions at the top and bottom of the domain we impose

1. non-penetrative conditions:  $u_r = 0$  at  $r = r_{\text{in}}, r_{\text{out}}$ ;
2. stress-free conditions:  $\frac{\partial}{\partial r} \left( \frac{u_\theta}{r} \right)$  at  $r = r_{\text{in}}, r_{\text{out}}$ ;
3. constant energy flux  $F_\star$  at the bottom of the domain  $r = r_{\text{in}}$ ;
4. an energy flux at the top of the domain given by

$$F_{\text{out}} = \sigma T_{\text{out}}^4, \quad (8)$$

where  $T_{\text{out}}$  is the temperature of the cells in the top layer of the domain. Finally, we would like to stress that we adopted 2D geometry to simplify the development of the code, but extension to 3D is planned in the future.

### 3. Numerical method

We present below our numerical method, describing separately the spatial (Sect. 3.1) and temporal (Sect. 3.2) discretisations.

#### 3.1. Spatial discretisation

##### 3.1.1. One-dimensional advection scheme

The equations are discretised on a staggered grid, using a finite volume approach. Cell interfaces location are denoted by  $x_i, i = 1 \dots N_x$  and cell centres by  $x_{i+1/2}, i = 1 \dots N_x - 1$ . In the staggered grid approach, scalar quantities (e.g. density, internal energy, pressure, temperature, etc) are located at cell centres, while velocity components are located at cell boundaries.

In the finite volume approach, the physical equations are integrated over a control volume to yield the evolution law for the physical quantities (mass, internal energy, momentum) in each cell. Fluxes are computed at cell interfaces, yielding a conservative scheme with respect to advection. For scalar quantities (density and internal energy) at  $x_{i+1/2}$ , the control volume is the cell  $[x_i, x_{i+1}]$  with volume  $V_{i+1/2}$  and surfaces  $S_i, S_{i+1}$ . For the velocity component at  $x_i$ , the control volume is the cell  $[x_{i-1/2}, x_{i+1/2}]$ , with volume  $\hat{V}_i$  and surfaces  $\hat{S}_{i-1/2}, \hat{S}_{i+1/2}$ .

We use the Van Leer method to compute the fluxes (see van Leer 1977; Stone & Norman 1992). To compute the flux at the control volume interfaces, it is necessary to perform interpolation since not all quantities are available at the considered interface. We define “barred” quantities (i.e.  $\bar{\rho}, \bar{u}$ ) as quantities that are interpolated at the interface by taking a simple average between the values available on each side of the interface, for instance  $\bar{u}_{i+1/2} = (u_i + u_{i+1})/2$ . For a stable and TVD advection, it is necessary to introduce some limiting process in the interpolation scheme of advected quantities. Here we proceed with an upwind limited interpolation similar to the MUSCL method. In each cell a linear reconstruction of the solution is performed. For instance, we describe below the procedure for cell  $i + 1/2$  and cell centred variable  $q$ . We first compute the limited slope  $\Delta_{i+1/2} q$  of the linear reconstruction in the cell by using the values of the neighbouring cells with a limiting process:

$$\Delta_{i+1}q = q_{i+3/2} - q_{i+1/2} \quad (9)$$

$$\Delta_i q = q_{i+1/2} - q_{i-1/2} \quad (10)$$

$$\Delta_{i+1/2}q = \phi\left(\frac{\Delta_{i+1}q}{\Delta_i q}\right)\Delta_i q, \quad (11)$$

where  $\phi$  is the slope limiter. In the following we use the Van Leer limiter:

$$\phi(r) = \frac{2r}{1+r} \quad \text{if } r > 0 \quad (12)$$

$$= 0, \text{ otherwise.} \quad (13)$$

With such a reconstruction scheme, we reconstruct at each interface right and left values  $q^R$  and  $q^L$ :

$$q_i^R = q_{i+1/2} - \Delta_{i+1/2}q \quad (14)$$

$$q_i^L = q_{i-1/2} + \Delta_{i-1/2}q. \quad (15)$$

Depending on the sign of the velocity, one of these quantities is used to compute the flux at the interface. For instance, at interface  $x_i$  one has

$$\tilde{q}_i = q_i^L \text{ if } u_i \geq 0 \quad (16)$$

$$= q_i^R \text{ if } u_i < 0. \quad (17)$$

This defines the “tilded” quantities in the equations shown below. For the interfaces located at  $i + 1/2$ , the same reconstruction is performed and the interface value is chosen depending on the sign of  $\bar{u}_{i+1/2}$ .

The pressure gradient at the interface is computed with a second-order, centred difference. The overall scheme is expected to be second order in space in the smooth regions of the flow. We present in appendix A an order study for the linear advection problem.

After integration over the corresponding control volume and discretisation, the 1D Euler equations become

$$\frac{d}{dt}\rho_{i+1/2}V_{i+1/2} = -(S_{i+1}\tilde{\rho}_{i+1}u_{i+1} - S_i\tilde{\rho}_i u_i) \quad (18)$$

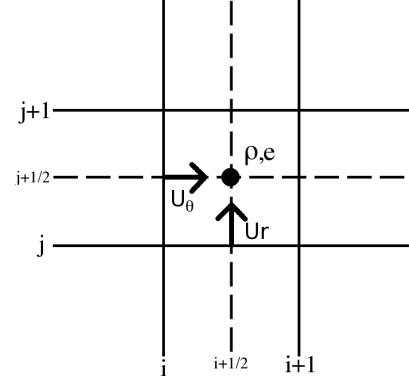
$$\begin{aligned} \frac{d}{dt}\rho_{i+1/2}e_{i+1/2}V_{i+1/2} = & -(S_{i+1}\tilde{\rho}e_{i+1}u_{i+1} - S_i\tilde{\rho}e_i u_i) \\ & - P_{i+1/2}(S_{i+1}u_{i+1} - S_i u_i) \end{aligned} \quad (19)$$

$$\begin{aligned} \frac{d}{dt}\tilde{\rho}_i u_i \hat{V}_i = & (\hat{S}_{i+1/2}\tilde{\rho}\bar{u}_{i+1/2}\bar{u}_{i+1/2} \\ & - \hat{S}_{i-1/2}\tilde{\rho}\bar{u}_{i-1/2}\bar{u}_{i-1/2}) \\ & - \frac{P_{i+1/2} - P_{i-1/2}}{\Delta x}\hat{V}_i. \end{aligned} \quad (20)$$

### 3.1.2. Gravity

Since we are solving the internal energy equation, gravity only enters the momentum equation. We assume that gravity remains *constant* and *spherically symmetric* so that the gravitational acceleration at radius  $r_i$  is given by

$$g_i = -\frac{GM_i}{r_i^2}, \quad (21)$$



**Fig. 1.** Staggered mesh in 2D

where  $G$  is the gravitational constant and  $M_i$  the mass contained inside the radius  $r_i$ . Our stellar models do not extend to the centre of the star so  $M_i$  is computed assuming  $M_1 = M_{\text{core}}$ , where  $M_{\text{core}}$  is determined from the original stellar model.

Under the assumption of spherical symmetry, the gravitational force enters the radial momentum equation only through the additional source term:

$$F_i^{\text{grav}} = \tilde{\rho}_i g_i \hat{V}_i. \quad (22)$$

The assumption of spherical symmetry could become invalid if the star developed significant non-radial oscillations. The convective flows, which are not symmetric, do not alter the mass distribution symmetry significantly. Implementation of a full solver for the Poisson equation is planned for the future.

### 3.1.3. Radiative diffusion

The radiative flux through the cell interface is computed with a second-order central difference

$$F_i^r = \frac{ac}{3} \frac{1}{\tilde{\rho}\tilde{\kappa}_i} \frac{T_{i+1/2}^4 - T_{i-1/2}^4}{\Delta x}, \quad (23)$$

where

$$\frac{1}{\tilde{\rho}\tilde{\kappa}_i} = \frac{1}{2} \left( \frac{1}{\rho_{i+1/2}\kappa_{i+1/2}} + \frac{1}{\rho_{i-1/2}\kappa_{i-1/2}} \right). \quad (24)$$

### 3.1.4. Extension to 2D and spherical geometry

Extension to 2D is done in the simplest possible way. Fluxes are computed along both directions following the 1D algorithm sketched above and added together in the update step (see below). Figure 1 shows the location of all quantities on our staggered 2D mesh. We avoid dimensional splitting (e.g. Strang splitting) that would increase the cost of the solver since several (implicit) substeps would have to be computed to advance the solution over one time step. It is clear in this case that the accuracy of the method as deduced from 1D analysis does not extend to multi-dimensional computation. One solution to improve the accuracy of the code would be to implement a genuinely multi-dimensional method (see e.g. Colella 1990; LeVeque 1997). Another possibility would be to increase the spatial order of the 1D reconstruction

scheme to obtain a better accuracy in multi-dimensional computations. This is left for future developments.

We describe below how spherical geometry is taken into account in our spatial discretisation: in our finite volume approach, the cell  $(i + 1/2, j + 1/2)$  spans the parameter space domain  $[r_i, r_{i+1}] \times [\theta_j, \theta_{j+1}]$ . The geometrical factors in the 2D version of cell-centred equations reads as

$$V_{i+1/2, j+1/2} = \frac{2\pi}{3}(r_{i+1}^3 - r_i^3)(\cos \theta_j - \cos \theta_{j+1}) \quad (25)$$

$$S_{i, j+1/2}^r = 2\pi r_i^2(\cos \theta_j - \cos \theta_{j+1}) \quad (26)$$

$$S_{i+1/2, j}^\theta = \pi \sin \theta_j(r_{i+1}^2 - r_i^2). \quad (27)$$

These geometrical factors account for the geometrical effect associated with spherical coordinates. There are similar expression for the radial momentum and the tangential momentum equations (see appendix B); however, due to the vector nature of the velocity, there are additional geometric source terms that appear in the momentum equations. In the radial momentum equation, the geometric source term reads as

$$S_{i, j+1/2}^{\text{geom}} = \hat{V}_{i, j+1/2} \frac{\bar{\rho}_{i, j+1/2} (\bar{u}_{i, j+1/2}^\theta)^2}{r_i}, \quad (28)$$

and the geometric term in the tangential momentum is

$$S_{i+1/2, j}^{\text{geom}} = -\hat{V}_{i+1/2, j} \frac{\bar{\rho}_{i+1/2, j} \bar{u}_{i+1/2, j}^r u_{i+1/2, j}^\theta}{r_{i+1/2}}. \quad (29)$$

The full semi-discretised formulae are detailed in the appendix B.

Spherical coordinates have severe drawbacks in the discretisation of a sphere. First, since  $r = 0$  is a singular point of the coordinates, the cell size reduces to zero toward the centre. They are therefore inappropriate for modelling the central regions of the star. Also, the cell size increases toward the surface, whereas finer resolution is needed in these regions (see Sect. 5.2). Calhoun et al. (2008) shows how to design structured grids that overcome these limitations.

### 3.2. Temporal method

#### 3.2.1. CFL time steps

When an explicit method is used to integrate a discrete set of equations in time, there is a constraint on the time step that results from the CFL condition. The CFL condition states that the physical domain of dependence of the solution should be included in the numerical domain of dependence, otherwise the numerical scheme cannot be convergent. For an explicit scheme, this yields a constraint on the time step that has to be smaller than the CFL time step to perform a stable computation (independently of any accuracy consideration).

Determining the stability limit for a given explicit scheme is not always trivial, if not impossible, especially in the non-linear and multi-dimensional cases. See, for instance, Hirsch (1990) for a careful derivation of the CFL time step for popular schemes. The CFL time steps that we define in this section are based on very simple arguments and should only be considered as a guideline for the maximum stable time step.

For the hydrodynamic equations, which are hyperbolic equations, the CFL condition states that the time step should not be longer than the crossing time of the fastest wave over a cell. In this case the maximum stable time step is equal to

$$\Delta t_{\text{hydro}} = \min \frac{\Delta x}{|u| + c_s}, \quad (30)$$

where  $|u| + c_s$  is the velocity of the fastest wave ( $u$  is the velocity of the background flow).

When time-dependent diffusion processes are considered, for instance in our case radiative diffusion, the corresponding CFL time step is related to the typical time scale for diffusion in a cell:

$$\Delta t_{\text{rad}} = \min \frac{\Delta x^2}{\chi}, \quad (31)$$

where  $\chi = k_{\text{rad}}/(\rho c_p)$  is the radiative diffusivity coefficient, and  $c_p$  the specific heat at constant pressure. For advection-diffusion problem, which are of mixed nature, the CFL time step is taken to be the minimum values of equations (30) and (31).

We also define here a useful time step that we call the *advective time scale*. It is defined by

$$\Delta t_{\text{adv}} = \min \frac{\Delta x}{|u|}. \quad (32)$$

Its definition is inspired by the hydro time step (30) where we have set the velocity of the wave to be simply  $u$  instead of  $u + c_s$ . This time step characterises the time needed for the flow to cross one cell. We stress that, for a code solving the standard compressible hydrodynamic equations with an explicit method, this time step is *not* a stability limit. We introduce this time scale since, as shown later, it corresponds to a natural limit for the time step; however, for anelastic codes, this time step represents the stability limit, since sound waves are filtered out in this approach.

With the definition of these three time steps, we can define the corresponding CFL numbers for a given numerical time step  $\Delta t$ :

$$\text{CFL}_{\text{hydro}} = \frac{\Delta t}{\Delta t_{\text{hydro}}}; \quad \text{CFL}_{\text{rad}} = \frac{\Delta t}{\Delta t_{\text{rad}}}; \quad \text{CFL}_{\text{adv}} = \frac{\Delta t}{\Delta t_{\text{adv}}}. \quad (33)$$

By adopting an implicit integration strategy, our goal is to release the stability limit on the time step. Therefore, to appreciate the performance of our solver, we carefully monitor the values of these three CFL numbers.

In 2D, the time steps defined above are computed independently in both directions and the CFL time step is defined as the minimum of these values.

#### 3.2.2. Implicit strategy

After the spatial discretisation method described in the previous section, one obtains a system of coupled ODEs (see Appendix B):

$$\frac{dU_{i,j}}{dt} = R(U_{k,l}), \quad (34)$$

where  $U_{k,l}$  is a vector gathering all the quantities appearing below time derivatives (i.e.  $\rho$ ,  $\rho e$ ,  $\rho u_r$ , and  $\rho u_\theta$ , multiplied

by the appropriate volumes), and  $R(U_{k,l})$  contains all fluxes and sources  $s$ :

$$\begin{aligned} R(U_{k,l}) = & -(S_{k+1,l}^r F_{k+1,l}^r - S_{k,l}^r F_{k,l}^r \\ & + S_{k,l+1}^\theta F_{k,l+1}^\theta - S_{k,l}^\theta F_{k,l}^\theta) \\ & + s_{k,l} V_{k,l}. \end{aligned} \quad (35)$$

To obtain an update formula we now apply an implicit temporal method to integrate (Eq. 34) in time:

$$U_{i,j}^{n+1} - U_{i,j}^n = \Delta t (\beta R^{n+1} + (1 - \beta) R^n) \quad (36)$$

where  $R^n$  (resp.  $R^{n+1}$ ) is the r.h.s. of equation (34) evaluated at time  $n$  (resp.  $n + 1$ ). The first-order accurate Backward Euler method corresponds to  $\beta = 1$ . The second-order accurate Crank-Nicholson corresponds to  $\beta = 1/2$ . Both the Backward Euler and the Crank-Nicholson methods are A-stable, which implies that they are unconditionally stable<sup>1</sup>. For stiff equations, a more relevant property is the L-stability of the scheme (see e.g. LeVeque 2007). The Backward Euler method is L-stable but the Crank-Nicholson method is not. This implies that the latter can have difficulties with stiff problems and produce spurious oscillations. Indeed, when computing stellar models with high values of the radiative CFL (see the A-type star model in Sect. 5.1), we have found that it is more stable to use a value of  $\beta$  that is slightly higher than  $1/2$ , for instance  $\beta = 0.55$ . In the future, we plan to implement the second-order backward differentiation formula, which is a second-order implicit multistep method having the A & L-stability property.

Equation (36) describes a system of non-linear equations that defines the solution at the new time step,  $U^{n+1}$ . We use  $\rho$ ,  $e$ ,  $u_r$ , and  $u_t$  as independent variables, so the quantities that appear in the time derivatives in our equations are thus *composite quantities* of the primitive variables.

### 3.2.3. Newton-Raphson procedure

To solve for the new time step  $\rho^{n+1}$ ,  $e^{n+1}$ ,  $u_r^{n+1}$ ,  $u_\theta^{n+1}$ , one has to solve the set of non-linear equations  $F(U^{n+1}) = 0$ , where the residual  $F$  is defined as

$$F(U^{n+1}) = U^{n+1} - U^n - \Delta t (\beta R^{n+1} + (1 - \beta) R^n). \quad (37)$$

Here, we proceed with a Newton-Raphson procedure. The procedure to find  $U^{n+1}$  is initialised by taking the current solution as an initial guess for the new solution:

$$U^{(0)} = U^n. \quad (38)$$

At each Newton-Raphson iteration we solve for the corrections  $\delta U^{(k)}$ :

$$J \times \delta U^{(k)} = -F(U^{(k)}), \quad (39)$$

where  $J$  is the Jacobian matrix of the residual vector  $F$ . The current iteration is then updated with

$$U^{(k+1)} = U^{(k)} + \lambda \delta U^{(k)}, \quad (40)$$

<sup>1</sup> Not all implicit methods are A-stable; e.g. the Adams-Moulton methods are implicit but only conditionally stable.

where  $0 < \lambda \leq 1$  is introduced to prevent divergence of the procedure. The classical Newton-Raphson update corresponds to  $\lambda = 1$ . A straightforward line-searching algorithm is applied to find the value of  $\lambda$  that ensures a decrease in the residual. However, we found that when solving the non-linear radiative diffusion term at very high radiative CFL numbers ( $\sim 10^6$ ), the first Newton-Raphson iteration always increases the residual. Different tests have shown that this seems unavoidable, and since the system would eventually converge within a few iterations, we always enforce  $\lambda = 1$  during the first iteration in this particular case.

We stop the iteration procedure when the relative correction drops below a certain level:

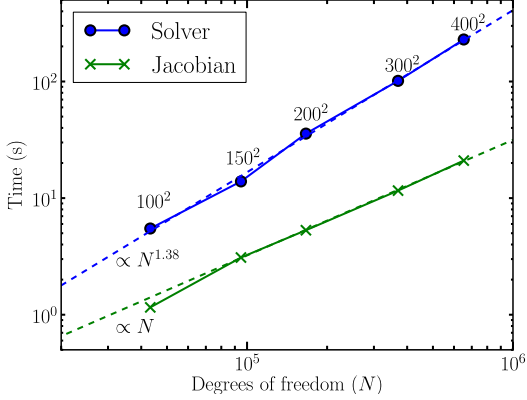
$$\left\| \frac{\delta U^k}{U_0} \right\|_\infty < \epsilon, \quad (41)$$

where  $U_0$  are typical values of the variables. For the density and internal energy we simply choose the corresponding values at iteration  $k$ , but for the velocities we choose  $\max(u, c_s)$ , where  $u$  is the value of the velocity at iteration  $k$ , and  $c_s$  is the sound speed. In our computations we set  $\epsilon = 10^{-6}$ .

### 3.2.4. Jacobian matrix computation

The Jacobian matrix elements are the partial derivatives of the numerical scheme equations (37) with respect to all numerical variables. We compute the partial derivatives by finite differencing. The location of non-zero elements of the Jacobian depends exactly on the stencil of the numerical scheme (i.e. which variables contribute to a given equation). It is therefore easy to determine the non-zero structure (also known as the sparsity pattern) of the Jacobian matrix. Any Jacobian computation technique should then take advantage of the non-zero pattern to compute the Jacobian matrix elements. We use here the *Coloured Finite Differencing* technique (CFD hereafter; see Curtis et al. 1974; Gebremedhin et al. 2005), which minimises the number of function evaluation needed to compute the Jacobian matrix. The CFD algorithm uses the non-zero pattern of the Jacobian to group independent columns (i.e. variables) of the Jacobian matrix, i.e. those that do not share a non-zero element on the same line, into “colours”. This yields a “column compressed” representation of the Jacobian. The number of colours  $n_g$ , which is also the number of columns of the compressed Jacobian matrix, is roughly equal to the maximum number of non-zero elements found in a row. This number depends *only* on the stencil of the scheme and is (ideally) *independent* of the dimension of the Jacobian matrix. To compute the matrix elements, the CFD algorithm perturbs all variables associated with the same colour and recomputes the non-linear equations. The matrix elements are then obtained by a straightforward finite difference against a reference value of the equations. The number of non-linear function evaluations is therefore equal to  $N(n_g + 1)$ , where  $N$  is the size of the matrix. As  $n_g$  is independent from  $N$ , the CFD algorithm is expected to scale as  $N$ .

In our code, we use the CFD algorithm that is implemented in the Trilinos library (see Heroux et al. 2003). The graph-colouring algorithm is the “greedy” algorithm proposed in Curtis et al. (1974). We find that for our matrices this algorithm typically produces a number of colour



**Fig. 2.** Time spent in computing the Jacobian matrix (crosses) and in solving one linear system with MUMPS (dots) for different matrix sizes on a Pentium III Xeon quad-core CPU at 3 GHz.

$n_G = 51 \pm 1$  for matrix sizes ranging from 2304 ( $20^2$  domain) to 652864 ( $400^2$  domain). Therefore, the number of colours is roughly independent of the matrix size, and the algorithm scales as  $N$ . This is illustrated in Fig. 2, where we plot the time needed to compute the Jacobian matrix for different spatial resolutions.

### 3.2.5. Linear solver

The Newton-Raphson procedure requires the solution of the linear equation (39). Equation (39) is a sparse linear problem that can be solved by either a direct method (i.e. LU decomposition) or an iterative method. Iterative solvers (e.g. GMRES, BiCGStab) perform usually faster than direct methods but they rely heavily on preconditioning to improve (or even obtain) convergence. On the other hand, direct solvers are more accurate and robust but they become very expensive as the size of the linear system increases. For convenience, and since our 2D approach allows it, we decided to use a direct solver for developing the code. We use in our code the MUMPS solver (Multifrontal Massively Parallel Solver, see Amestoy et al. 2001, 2006). MUMPS is able to handle general non-symmetric matrices; i.e., it is not restricted to tridiagonal or pentadiagonal matrices. MUMPS achieve robustness by detecting null pivots during the elimination phase and accuracy by applying a few (usually 2 or 3) steps of iterative refinement in the postprocessing phase.

We use the advantage of knowing the non-zero structure of the Jacobian matrix to perform the symbolic factorisation of the linear system only one time, at the beginning of a simulation. Later on, only the numerical factorisation is performed with a given r.h.s. vector to obtain the solution of the linear system. The cost of the direct solver is illustrated in Fig. 2 where the time needed to solve one linear system is shown for various spatial resolutions. The linear solver dominates the time needed to compute the Jacobian matrix. With a simple linear regression in the log-log space, we find that in our case (i.e. for the Jacobian matrices that result from our temporal and spatial discretisation) the time needed to solve one linear system scales as  $N^{1.38}$ . For 2D runs with resolution up to  $500^2$ , the number of variables is under one million. Linear systems are in

**Table 1.** Run parameters for the entropy bubble test.

Run	Final time	Resolution	time step	CFL
1	200	$50^2$	0.04	1
2	200	$50^2$	0.5	12.5

this case still tractable with direct methods. The memory requirement for the LU decomposition ranges from 10 Mb for 2304 degrees of freedom to 9.4 Gb for 652864 degrees of freedom. It is clear that larger systems (e.g. 3D problems) will require too much memory and computational time to be tractable. Therefore, we plan in the future to implement iterative solvers that are less expensive both in terms of memory and CPU time.

### 3.2.6. Time step control

In an implicit method, one has to introduce a time step control strategy, usually based on empirical grounds. We found that the following three strategies are useful, in the order of the most to the least basic:

**Convergence-based:** If the Newton-Raphson procedure converges within five iterations, increase the time step by a factor of 1.5. If NR converges in more than ten iterations, decrease the time step by a factor of 2.

**Relative change-based:** Compute relative change in the solution between time step  $n$  and  $n + 1$ . If it is lower than 10 %, increase the time step by a factor of 1.5. If it is larger than 20 %, decrease time step by a factor of 2.

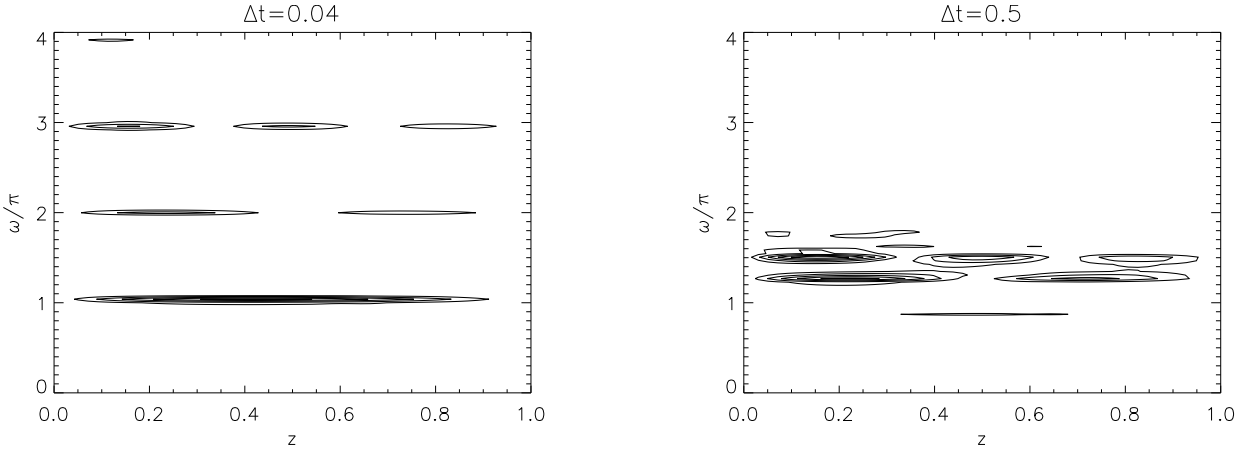
**Advection-based:** Set the time step so that  $\text{CFL}_{\text{adv}} = 1$ .

The last strategy is physically motivated since it ensures that the fluid does not move across more than one cell width during one time step. For the preliminary results presented in this paper, we have used the first, least stringent, and least physical, strategy. In the future we plan to compare these three strategies, both in terms of accuracy and computational time. Finally, in some cases, we also provide maximum/minimum values for the CFL number or the time step.

## 4. Test: oscillations of an entropy bubble

Basic tests (see appendix A) have been successfully done for code validation during its first stages of development, so we focus here on a test that is more relevant to the physical processes we would like to investigate. It was taken from Dintrans & Brandenburg 2004 (DB hereafter). We considered an isothermal stratified atmosphere with constant gravity perturbed by an entropy bubble. We computed the oscillations of this bubble around its equilibrium position, and we analysed the spectrum of internal gravity and sound waves driven by the bubble. The initial setup, parameters, and units normalisation are similar to DB and are therefore not reproduced here. The Brunt-Väisälä frequency of the atmosphere is  $N \sim 0.82$ . For this test, Eq. (1-3) were solved with the viscous terms included. The kinematic viscosity coefficient  $\nu$  was set to  $5 \times 10^{-4}$  in all runs. The thermal conductivity is set to zero, as we were interested in the adiabatic evolution of the system.





**Fig. 3.** Acoustic modes in the  $(z, \omega)$  plane for  $\Delta t = 0.04$  (left) and  $\Delta t = 0.5$  (right).

We used a Cartesian version of the code<sup>2</sup> and considered a domain  $(x, z) \in [-0.5, 0.5] \times [0, 1]$ . Periodic boundary conditions were used in the horizontal directions and non-penetrative, stress-free conditions were used at the bottom and top boundaries. All the run parameters are summarised in Table 1. For each run we set the time step to a constant value.

As in DB, we describe each wave mode by two integers  $l, n \geq 0$ , which are the number of nodes in the horizontal and vertical directions, respectively. In this test we looked at two types of waves:

1. Vertical sound waves that have frequencies  $\omega = (n+1)\pi$ . These waves have  $l = 0$  since they have no horizontal structure.
2. Internal gravity waves that have frequencies  $\omega < N$  and are characterised by  $l \neq 0$  (internal gravity modes cannot propagate in the vertical direction).

See DB for the derivation of these properties. These two kinds of waves occupy well-separated regions of the frequency spectrum. Vertical sound waves have vertical periods  $P \leq 2$  and will therefore be more sensitive to the time step. Internal gravity modes have longer periods  $P \gtrsim 7.3$  and will therefore be less sensitive to the time step. This test thus provides a good analysis of the influence of the time step on the accuracy of the numerical results.

We first analysed the vertical sound wave spectrum by using method 2 described in DB, for runs 1 & 2 (cf. Table 1). In this method, the wave spectrum is represented in the  $(z, \omega)$  plane. For any given time step  $\Delta t$ , the Nyquist theorem states that no periodic signal with frequency higher than  $\omega_N = \pi/\Delta t$  can be resolved. Also the temporal Fourier transform has a frequency resolution that is  $\Delta\omega = 2\pi/T$ , where  $T$  is the duration of the simulation. The results are shown in Fig. 3 for two different time steps. For  $\Delta t = 0.04$ , we recognise the signature of the  $n = 0, 1, 2$  modes. The  $n = 3$  mode is below the range of the isocontour levels. Furthermore, all the modes are located at the correct frequencies (within  $\Delta\omega \simeq 0.125$ ). For  $\Delta t = 0.5$ , one has  $\omega_N = 2\pi$  thus normally allowing only for modes  $n = 0$  and  $n = 1$ . In the map, one can recognise the signature of the  $n = 0, 1, 2$  modes. None of them is located at the correct

frequency, and it is surprising to find the  $n = 2$  mode whose eigenfrequency is above the Nyquist frequency. In this case it is clear that the time step is too large to accurately resolve the sound waves. To analyse the gravity modes, we used the method developed in DB: we projected the velocity field on the anelastic eigenvectors. With this method we obtained the individual mode amplitudes  $c_{ln}$  which are the coefficients in the eigenvector expansion. With such a decomposition it is straightforward to obtain the mode frequency. We checked that, for both runs, the gravity mode frequencies match the analytical predictions to the percent level.

This test highlights the importance of the choice of the time step to correctly describe the physical process of interest. In the present case, sound-wave amplitudes are much lower than the amplitudes of the gravity modes (the initial setup is almost in hydrostatic equilibrium), and therefore they do not play an important role in this problem. One can then safely use a higher time step (e.g.  $\Delta t = 0.5$ ) to study the internal gravity modes, as shown by the excellent agreement with predicted frequencies. Obviously, one could not do that if the initial setup was intended to be far from hydrostatic equilibrium, since sound waves would at least be as important as internal gravity modes.

## 5. Stellar models

We now test our code on realistic stellar conditions.

### 5.1. Local simulation: A-type star

We present in this section results of convection in A-type stars (see model I in Table 2), since comparison with previous models of such stars, calculated by different groups using explicit codes (see Sofia & Chan 1984; Freytag et al. 1996), are possible. These types of stars are weakly convective and possess two convectively unstable zone near their surface: around  $T \sim 40\,000$  K due to HeII ionisation and near  $T \sim 10\,000$  K due to HeI/H ionisation. Usually the HeI/H surface convective zone is strong, but very narrow, whereas the HeII convective zone is broader but much weaker. To model these convective regions it is therefore sufficient to focus on the surface layers of the star by performing *local* simulations.

<sup>2</sup> Cartesian coordinates are used in this particular case because of analytical solutions in this geometry.

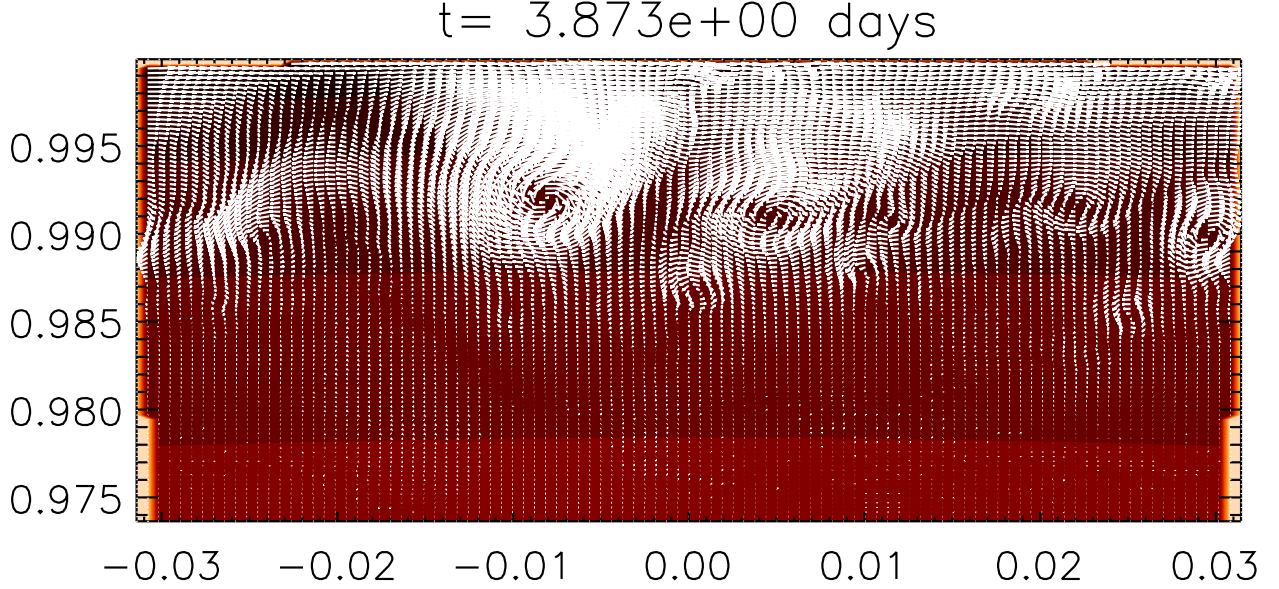
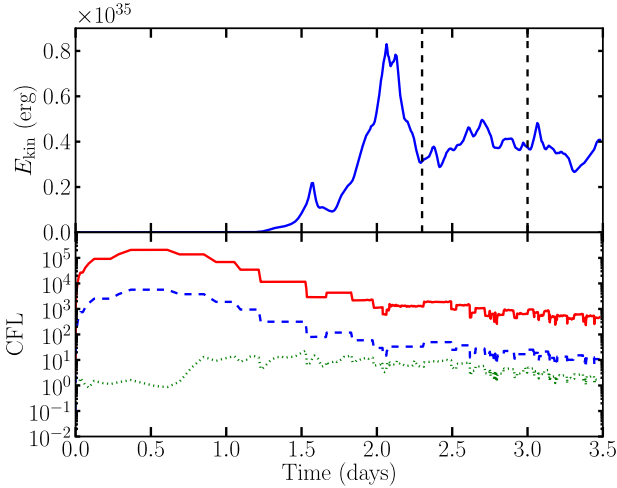


**Table 2.** Stellar parameters of the models considered in this paper.

Model	Mass ( $M_{\odot}$ )	R ( $R_{\odot}$ )	$T_{\text{eff}}$ (K)	$\log(g)$	Luminosity ( $L_{\odot}$ )	Comments
I	2	2.1	8 500	4.10	20	A-type star
II	5	58.8	4 500	1.6	1, 258	Cold giant

**Table 3.** Numerical parameters for the A-star run.

Resolution	Domain	$\tau_{\text{KH}}$	$\Delta t_{\text{hydro}}^{\text{CFL}}$	$\Delta t_{\text{rad}}^{\text{CFL}}$	Final time	Time steps	Wall Time	Mem.
$214 \times 512$	$[0.974, 1.] \times [88.2^{\circ}, 91.8^{\circ}]$	1.15 d	1.84 s	$3 \times 10^{-2}$ s	3.5 d	4032	2 w 5 d	4.8 Gb

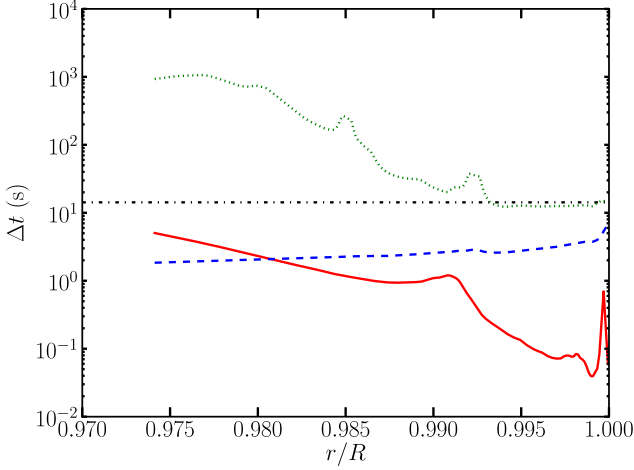

**Fig. 5.** Snapshot of our simulation of an A-type star. The axes are Cartesian coordinates normalised by the stellar radius and with the origin at the star centre.

**Fig. 4.** Evolution of the total kinetic energy (top panel) and CFL numbers (lower panel, solid: radiative; dashed: hydro; dotted: advection) for the A-type star run. The vertical dashed lines show the time interval on which the averaged fluxes shown in Fig. 7 are computed.

Our initial model starts at the photosphere, i.e. the location where  $\tau = 2/3$  and  $T_{\text{eff}} = 8500$  K. This particular choice inhibits the convection in the HeI/H ionisation zone, which is located at the photosphere, so we do not expect to

model this convective region correctly. A better modelling of this region should include a part of the stellar atmosphere in the computational domain, as is done for instance by Freytag et al. (1996). Our initial models are produced with a 1D stellar structure code (see Baraffe & El Eid 1991) where the mixing length theory (MLT hereafter) prescription for convection has been turned off. The resulting fully radiative models are very close to the MLT models (even near the surface) since convection is too weak, in terms of energy transport, to significantly modify the structure. The model presented here extends down to a depth where the temperature is  $T = 100\,000$  K, where the star is radiatively stable, so that the numerical domain includes a stable, radiative, buffer zone at the bottom of the domain. Since our simulation focusses on a very small fraction of the star, the geometry is Cartesian-like (but we use spherical geometry). In terms of physical dimensions, the physical domain has an average width of 90 000 km and a height of 37 500 km.

To construct the initial model for the multi-dimensional code, we simply copy the 1D structure along the tangential direction. However, the number of tangential grid points has to be carefully chosen in order to keep an acceptable aspect ratio of the cells. The number of grid points in the tangential direction therefore depends on the number of grid points in the radial direction through the radial extend of the cells.

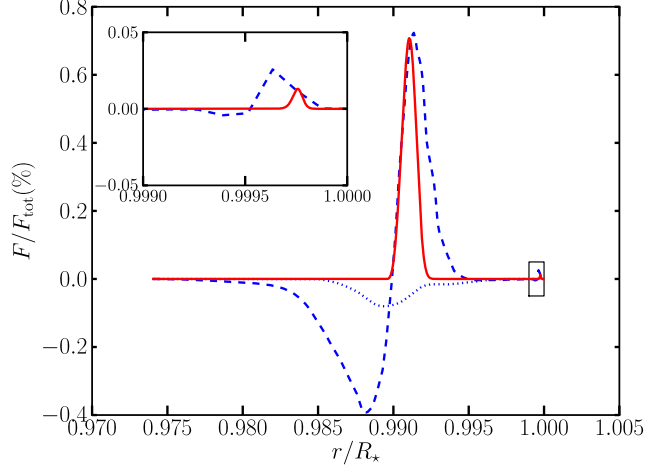
We discuss below the results of our run. Table 3 summarises important parameters.  $\tau_{\text{KH}}$  is the thermal time



**Fig. 6.** Radial profiles of the CFL time steps for the A-type star run (see text). From bottom to top: radiation (solid red), hydro (dash blue), and advection (dot green). The horizontal dash-dotted line indicates the time step of the snapshot under consideration.

scale defined as the ratio between the total internal energy in the computational domain divided by the luminosity of the model. The last column of Table 3 shows the memory requirement for the LU decomposition. Calculations were done on an AMD “Istanbul” processor (6 cores at 2.7 Ghz). Figure 4 shows the evolution as a function of time of the total kinetic energy in the domain (used to track the development and saturation of the convective instability) and the evolution of the different CFL numbers defined in Sect. 3.2.1. At the beginning, the model undergoes a relaxation toward hydrostatic equilibrium, and the CFL number rapidly reaches values  $> 10^5$ . Eventually, the velocity field that develops as a result of the relaxation process triggers the convective instability, which corresponds to an increase in the kinetic energy to a significant level. As the first vortices develop, the time step decreases since the Newton-Raphson procedure would not converge otherwise. After  $\sim 2.3$  days of simulated time, we reach a quasi-steady state that lasts until the end of the simulation at  $t \sim 3.5$  days. In this phase, the time step remains roughly constant with  $\Delta t = 18 \pm 2.6$  s, corresponding to  $\text{CFL}_{\text{hydro}} = 9.8 \pm 1.4$  and  $\text{CFL}_{\text{rad}} = 400 \pm 60$  (the second value indicated correspond to the standard mean deviation). In this quasi steady state, convection is fully developed, and we observe a complex vortex dynamics involving vortex pairing/merging and the formation of downdrafts. These phenomena are characteristic of turbulent convection (see snapshot in Fig. 5).

Figure 6 shows the radial profiles of the different CFL time steps defined in Sect. 3.2.1. The CFL time steps are first computed in each cell of the domain, and the radial profiles shown in Fig. 6 are then deduced by taking the *minimum* value along the tangential direction for each radius. We first note that the radiative CFL time step provides the most stringent constraint. Solving the radiative diffusion with an explicit method would require a time step lower than  $\Delta t = 3 \times 10^{-2}$  s and would make any calculation cumbersome. The hydro CFL time step increases with radius, so that the hydro limitation on the time step would come from the bottom of the numerical domain. This is ex-



**Fig. 7.** Radial profiles of averaged fluxes for the A-type star run (dashed line: enthalpy flux, dotted line: kinetic energy flux). The solid line is the convective flux as predicted by the mixing length theory with a mixing length parameter  $\alpha = 1.375$ . The box in the upper left corner is a zoom of the surface layer of the star.

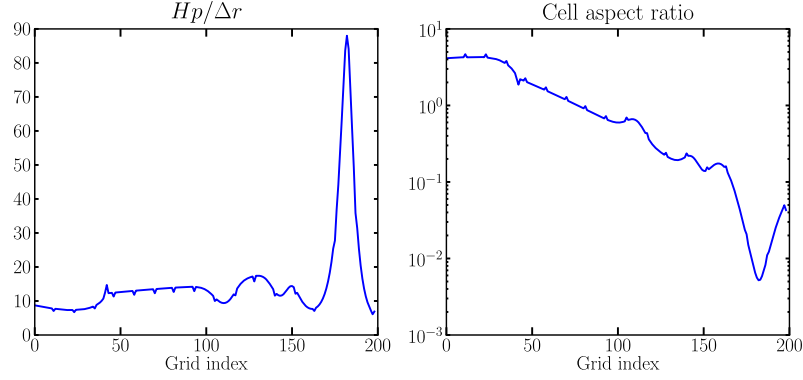
pected since we are using a uniform grid, and at the bottom of the envelope, the temperature, and hence the velocity of the sound waves, are the largest. The advective time scale shows the region where the flow velocities are the largest, i.e. near the surface. The time step of the actual simulation clearly corresponds to the lowest value of the advective time scale. If the time step was much longer than the advective time scale, the solution would change significantly since eddies would be moving across more than one grid cell. In this case, however, the Newton-Raphson procedure would have convergence problems since the initial guess (the solution at time step  $n$ ) would strongly depart from the correct solution. Since our time step strategy is based on limiting the number of iterations, the time step would be automatically decreased. Therefore, we do not expect our current implicit method to work with much larger time steps, and we identify the minimum value of the advective time scale as a natural (physical) limit for the time step.

We now analyse the properties of the fully developed convective state. We computed the average of the radiative, mass, kinetic, and enthalpy fluxes:

$$\begin{aligned} F_{\text{rad}} &= \langle \langle S^r (-k \frac{dT}{dr}) \rangle_{\theta} \rangle_t \\ F_{\text{m}} &= \langle \langle S^r \rho u_r \rangle_{\theta} \rangle_t \\ F_{\text{kin}} &= \langle \langle S^r u^r \frac{1}{2} \rho v^2 \rangle_{\theta} \rangle_t \\ F_{\text{enth}} &= \langle \langle S^r u^r (\rho e + P) \rangle_{\theta} - \langle \langle \rho e + P \rangle_{\theta} F_{\text{m}} \rangle_t. \end{aligned}$$

These quantities have the dimension of a flux multiplied by a surface (i.e. a luminosity), in order to take geometrical effects introduced by the spherical coordinates into account. We however use the term “flux” in the following to make the discussion clearer. The total flux is defined as the sum of the enthalpy, radiative, and kinetic energy fluxes.

Figure 7 shows the radial profile of the averaged fluxes defined above. The averaging was done on the interval 2.3



**Fig. 8.** Properties of the sampled grid for the giant model. Left panel: Pressure scale height to radial cell size ratio. Right panel: Cell aspect ratio when the radial mesh is extended in the tangential direction.

days - 3 days (represents 1408 time steps) but we checked that, by taking smaller windows (but still large enough to have several hundreds of time step), we would obtain similar results. Figure 7 shows the typical profile of the enthalpy flux in the HeII convective zone, as expected from general properties of convection: an inward/outward directed flux, with an outward part stronger than the inward flux. Also the kinetic energy flux is typical of what is expected from convection: downward, as a result of the acceleration of the downdrafts by gravity. The outward convective flux reaches a maximum amplitude of about 0.68 % of the stellar flux. Depending on the time interval on which the averaging was done, this maximum value ranges between 0.6 % and 0.77 % of the stellar flux. This corresponds to the prediction of MLT for an  $\alpha$  parameter in the range 1.35 – 1.4. As discussed above, we do not expect to have realistic results for the H/HeI top convective zone since our outer boundary condition at the photosphere inhibits convection in this region. However, despite the lack of resolution (only a few grid points resolve the ionisation region), we still find a non zero average convective flux with an amplitude of  $\sim 10^{-4}F_{\star}$ . For the aforementioned value of  $\alpha$ , tuned to match the HeII zone, the MLT predicts a convective flux in the HeI/H region, which has a location and an amplitude consistent with our results.

We now compare our results with the work of Sofia & Chan (1984) and Freytag et al. (1996), who have performed numerical models of a A-type star with the same parameters as ours (model A5 of Sofia & Chan paper, model AT85g41N3 of Freytag et al. paper). Sofia & Chan also considered models extending to the photosphere, but they did not find the small convective flux that appears near the surface in our simulation, most certainly because of the lack of resolution in their computations (their resolution is twice lower than our). Since only Freytag et al. (1996) resolve the HeI/H region properly, we restrict the comparison to the HeII convective zone. For this region, Sofia & Chan (1984) found a convective flux between 0.15 and 0.2 % of the stellar flux (depending on the aspect ratio of their model) and Freytag et al. (1996) found a convective flux about 0.12% of the stellar flux. This is significantly lower than the amplitude deduced from our simulations. This difference likely results from different input physics (metallicity, EOS, opacities).

Our results on A-type stars show that we could successfully describe 2D convective patterns in realistic stellar

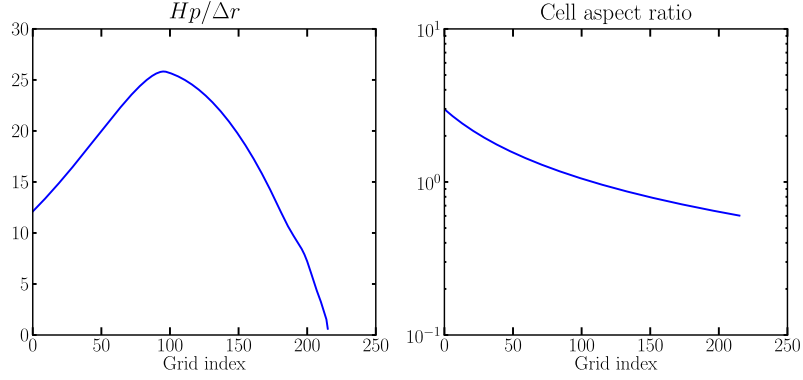
conditions with our fully implicit scheme. Thanks to this approach, the time step in our computation is not limited by sound waves or by radiative diffusion. However, we have shown that the time step is limited by the velocity of the fluid itself, which determines the rate of change in the solution. The hydro CFL cannot be much larger than 10 when convective motions are fully developed. On the other hand, the radiative CFL number is very high ( $\sim 400$ ). This is expected for the local models of “hot” stars that have very short radiative time scales in the low-density, photospheric region. The rather low value of the hydro CFL number and the high value of the radiative CFL number suggest that, for this type of local model, a mixed approach (explicit hydrodynamic + implicit radiation) could allow substantial speedup to be reached as compared to a fully explicit/implicit approach. However, our goal with the present simulation of A-type stars is not to compete with previous simulations performed with explicit codes, which are doing an excellent job for this type of star with very tiny atmospheric and subphotospheric convective zones. Also a realistic description of convection requires a 3D approach. The present study is meant to test the behaviour and the reliability of the implicit code to model convection under realistic conditions that have been already studied. Nevertheless, we stress that a fully implicit scheme remains very efficient at computing the initial development of convective instability, since the time step can grow to high values as long as velocities remain low.

## 5.2. Global simulation: cold giant star

The next step is to compute models involving a significant fraction of the star. We consider here a model with parameters corresponding to a cold giant star (model II in Table 2).

### 5.2.1. Initial model and grid design

For this model, MLT predicts that about 50 % of the stellar envelope is convective. The initial model for the multi-dimensional simulations is based on a 1D MLT model. Starting from a radiative model, as previously done for the A-type star, yields difficulties since a radiative structure departs strongly from the convective solution. The MLT model might also depart from the *real* convective state, but one can expect this difference to be less pronounced



**Fig. 9.** Properties of the uniform grid for the giant model with  $N_r = 217$ . Left panel: Pressure scale height to radial cell size ratio. Right panel: Cell aspect ratio when the radial mesh is extended in the tangential direction.

than with a fully radiative model. We model a significant fraction of the star (80 %) in order to have a large radiative zone below the convective zone. Our numerical domain therefore extends radially from  $0.2R_*$  to  $R_*$ , where  $R_*$  is the radius of the star as defined by the location of the photosphere. This yields a pressure contrast of roughly six orders of magnitude ( $\sim 14$  pressure scale heights) and a density contrast of almost five orders of magnitude throughout the whole domain. Throughout the convective zone the pressure contrast is  $\sim 400$  ( $\sim 6$  pressure scale heights) and the density contrast is  $\sim 50$ . Previous works that also consider an extended radial domain are the “star-in-a-box” simulations of Freytag et al. (2002), Woodward et al. (2003), and Dobler et al. (2006), where the full star is embedded in a Cartesian domain. Dobler et al. (2006) consider a very simplified stellar model, with a very low-density stratification (a factor of  $\sim 10$ ). The two other works consider more realistic stellar models of super giant and AGB stars and obtain convection over multiple pressure scale heights. We also mention Brun & Palacios (2009), which presents a numerical simulation of the inner  $\sim 50$  % of a red giant star with the anelastic code ASH. Their density stratification in their convective zone (which occupies the whole numerical domain) is about two orders of magnitude.

Our initial model extends up to the photosphere. The photospheric region is characterised by the ionisation regions of HeII and HeI/H, with “bumps” in the opacity profile. In addition, the surface region is characterised by the lowest values of the pressure scale height within the whole stellar structure. A non-uniform radial mesh would be necessary to solve the different scales present in the star. Our 1D stellar structure code uses such an adapted radial grid, which ensures a very good resolution of all features throughout the star. This is illustrated in the left panel of Fig. 8, where we have sampled the initial 1D model to create a radial mesh with a suitable number of grid points (200 in this case).

However, when considering multi-dimensional calculations, this grid presents a flaw: extension in the tangential direction yields intractable aspect ratio of the cells. This is illustrated in the right panel of Fig. 8, which shows the radial profile of the cell aspect ratio when the former radial mesh is simply extended in the tangential direction by putting 256 cells over a tangential angular width of  $\pi/2$ . Inspection of Fig. 8 shows that a constant number of cells per scale height translates into a large variation of the cell

aspect ratio. In the particularly demanding region near the photosphere the aspect ratio drops below  $10^{-2}$ , which is unacceptable for performing numerical simulations.

It is therefore obvious that to perform multi-dimensional simulations from the deep stellar interior to the surface, one has to make a compromise on the resolution when using a single grid. A solution to this problem would be to use a system of nested grids where the grids are refined in both the radial and tangential directions as moving near the surface of the star. Here, we choose to consider a computational grid with an uniform grid spacing in both  $\Delta r$  and  $\Delta \theta$  and apply a special treatment for the surface layers (see below). Due to spherical coordinates, the aspect ratio of the cells decreases with the radius. The radial step is chosen so that the cells in the middle of the domain are square. The right panel of Fig. 9 shows that our cells have an aspect ratio of 3 at the bottom of our domain and 0.6 at the top of the domain. As a result, the surface of the star is not resolved well, as shown by the number of grid points per pressure scale height (see left panel of Fig. 9).

### 5.2.2. Simulation setup

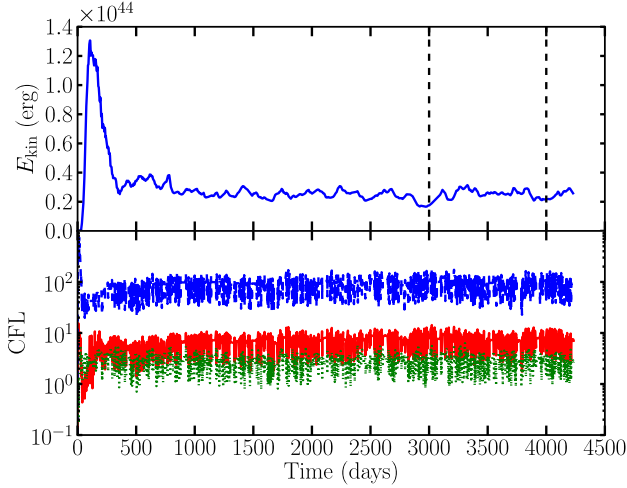
Since we cannot accurately describe the surface of the star, we choose to apply a heating function that drives an isothermal region in the outer, badly resolved, region of the star. To do so, we include in our energy equation a heating source term on the right-hand side:

$$Q = -\rho c_v (T - T_0) f(r) / \tau_{\text{heating}}, \quad (42)$$

where  $f(r)$  is a spatially varying function that is equal to 1 above a given radius  $r_c$  and zero elsewhere, with a smooth transition in between,  $\tau_{\text{heating}}$  is the typical cooling time scale and  $T_0$  is the forcing temperature. This method has been used for the “star-in-box” simulation presented in Dobler et al. (2006) (see also references therein). The parameter  $T_0$  is chosen to be the temperature of the gas at  $r_c$  in the initial model of the star. For the model presented in this section,  $r_c = 0.95R_*$ ,  $T_0 = 32\,750$  K, and  $\tau_{\text{heating}} = 10^4$  s. For comparison, the value of the radiative time scale (as defined in Eq. 31) decreases exponentially from  $\sim 3.5 \times 10^5$  s to  $\sim 3.5 \times 10^3$  s in this region so our value of  $\tau_{\text{heating}}$  insures that the source term acts on a short enough time scale to keep the outer region in a roughly isothermal state.

For the value of  $T_0$  used in our model, we do not have the HeI/H ionisation region near the surface. Also we prevent





**Fig. 10.** Evolution of the total kinetic energy (top panel) and CFL numbers for the giant star run (lower panel, solid red: radiative; dashed blue: hydro; dotted green: advection). The vertical dashed lines show the time interval on which the averaged fluxes shown in Fig. 12 were computed.

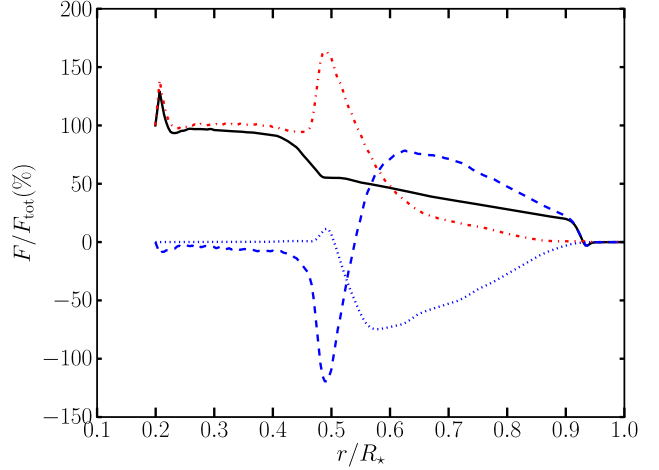
any convection motion in this region since the isothermal structure is stably stratified. There is essentially no radiative flux throughout this region, because the energy entering this region is evacuated by the source term instead.

This treatment of the surface layers is very basic, and is only used as a preliminary solution to produce our first results. We are currently investigating other more physically based methods to treat the surface layers.

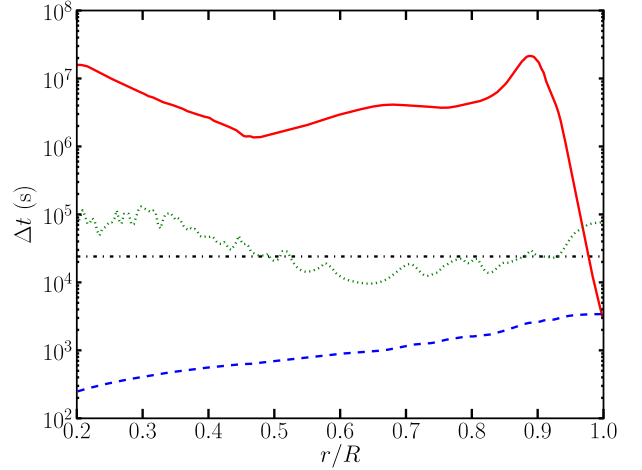
### 5.2.3. Results

We now present the results of the run (summarised in Table 4). Figure 10 shows the evolution of the kinetic energy and CFL numbers. As in the previous section, we see that, during the initial phase of relaxation/development of convection, we reach high CFL numbers, with the hydro CFL number reaching a maximum value around  $10^3$ . As the convective instability develops, we see a quick and strong rise in the kinetic energy in the domain, followed by a slower decrease as the model relaxes toward equilibrium. After 500 days, the kinetic energy tends toward a constant value, indicating that a quasi steady-state is reached. The time step remains roughly constant with a mean value of  $\Delta t = 2.5 \times 10^4 \pm 5 \times 10^3$  s, corresponding to  $\text{CFL}_{\text{hydro}} = 100 \pm 20$ , and  $\text{CFL}_{\text{rad}} = 7.6 \pm 1.5$ . The second value indicated corresponds to the standard mean deviation. Therefore in this kind of model, we see that the radiative time scale is much lower than the hydro time scale. Figure 11 shows a snapshot of the convective motions in the star.

We computed the averaged fluxes defined in Sect. 5.1 between  $t = 3000$  days and  $t = 4000$  days (corresponding to 3452 time steps). The resulting radial profiles are shown in Fig. 12. We see that thermal equilibrium has not been reached yet, since the total flux throughout the envelope is not constant. We see a significant upward enthalpy flux in the  $\sim 40\%$  outer radius of the star. Also, we see a very important downward kinetic energy flux in the convective zone. Around  $r \sim 0.5R_*$ , we observe strong bumps both in the radiative and in the enthalpy flux. This region cor-



**Fig. 12.** Radial profiles of averaged fluxes for the giant star run (dashed-dotted line: radiative flux; dashed line: enthalpy flux; dotted line: kinetic energy flux; solid line: total flux).



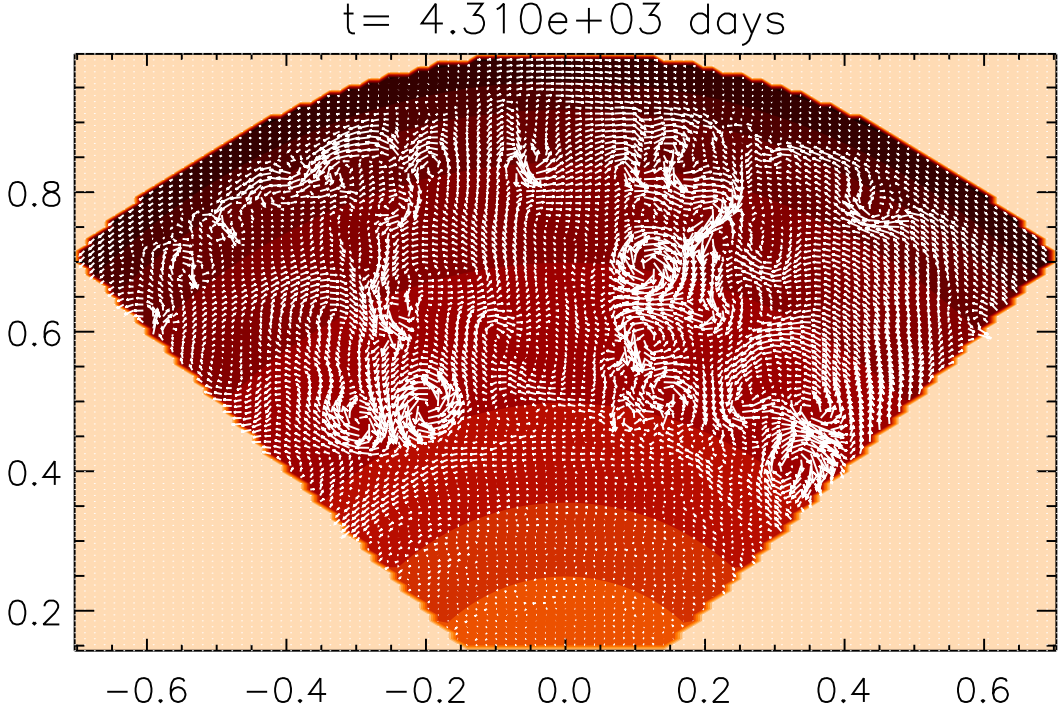
**Fig. 13.** Radial profiles of the CFL time steps for the giant star run (see text). From bottom to top: hydro (dashed blue), advection (dotted green), and radiation (solid red). The horizontal dash-dotted line indicates the time step of the snapshot under consideration.

responds to the interface between the convective zone and the radiative zone. When downdrafts cross the interface, the low efficiency of radiation prevent them from warming up, and as a result, they are subject to a very strong braking by buoyancy forces, which prevent them to significantly overshoot in the radiative zone. This is why the interface region is strongly perturbed whereas the inner radiative zone remains unperturbed. Finally, we checked the range of Mach numbers in the domain. The average Mach number of the flow extends from  $\sim 10^{-3}$  at the bottom of the envelope to  $\sim 0.15$  at the top of the convective zone. However, Mach numbers in the range  $0.30 - 0.40$  can be reached locally, in the downdrafts. We stress that such a range of Mach number cannot be addressed by anelastic codes, which are restricted to low Mach number flows ( $M_s \lesssim 10^{-2} - 10^{-1}$ ).

Figure 13 shows the radial profiles of the different CFL time steps (as in Fig. 6). As already mentioned above, we

**Table 4.** Numerical parameters for the cold giant run (same as Table 3).

Resolution	Domain	$\tau_{\text{KH}}$	$\Delta t_{\text{hydro}}^{\text{CFL}}$	$\Delta t_{\text{rad}}^{\text{CFL}}$	Final time	Time steps	Wall Time	Mem.
$217 \times 256$	$[0.2, 1.] \times [\pi/4, 3\pi/4]$	2200 yr	250 s	$3.2 \times 10^3$ s	4235 d	16216	8 w	2.2 Gb

**Fig. 11.** Snapshot of the cold giant star simulation. The axis are Cartesian coordinates normalised by the stellar radius and with the origin at the star centre.

see that the radiative CFL time step is much longer than the hydro time step; however, the strong decrease in the radiative time scale in the isothermal surface layer comes from the decrease in density in this region. We can expect that, if the surface regions were better resolved, the radiative time scale would drop to a lower value than the hydro time step, and we would recover locally a situation similar to the A-type star model. Here again the shortest hydro CFL time step is found at the bottom of the envelope, therefore we reach a higher hydro CFL number than for the A-type star model because our model extends deeper in the star. Finally, as discussed in the previous section, we see that the actual time step is roughly equal to the lowest value of the advective time scale, which is located in the convective zone. In this case, since the envelope extends deeper, our implicit scheme allows for reaching high values of the hydro CFL number by removing the limitation coming from the deeper layers of the star.

The results presented in this section show that it is meaningful to adopt a fully implicit approach for computing models of deep convective zones. Indeed, we show here an example of a situation where substantial CFL numbers could be reached by overcoming the time step limitation imposed by deep layers of the stellar envelope. This is possible since the flow velocity is low in those regions. In the surface convective region, however, we are still prone to the time step limitation imposed by the flow movement across grid cells.

Finally, the computation of such deep stellar envelopes also poses the problem of how to achieve the thermal relaxation of the convective envelope on an acceptable time scale. Indeed, the thermal time scale of our numerical domain is

$$\tau_{\text{KH}} = \frac{E_{\text{int}}}{L} \sim 2200 \text{ years}, \quad (43)$$

whereas the simulation only ran for 4235 days  $\sim 11.6$  years of stellar time. This is, however, an unavoidable difficulty that is common to all numerical approaches.

## 6. Conclusion

We have presented a new numerical code devoted to stellar interiors. The code solves the full set of the compressible Euler equations and radiative diffusion with a spatially and temporally accurate implicit scheme.

As a first step, we computed stellar convection following two approaches: a local approach for a relatively hot star (see Sect. 5.1) and a global approach for a cold giant star (see Sect. 5.2). In the first case, we were able to simulate the convective motions in the outer layers of a A-type star. For this type of model, we achieve a mean CFL number of 10 for hydro and of 400 for radiation. The rather low value of the hydro CFL number suggests that, in this particular case, a mixed approach (explicit hydro + implicit radiation) would result in a substantial speeding up of the calculation compared to a fully explicit or implicit approach. In the

second case, we modelled the turbulent convective motions that transport energy in the bulk of a cold giant star. Our domain includes both the outer convective region and the bottom, stable, radiative region. For this model, we could reach a mean CFL number of 100 for hydro and of 7 for radiation. Such values of the hydro CFL numbers are encouraging and suggest that we might gain computational time thanks to our implicit approach.

The aim of this paper is to show the *feasibility* of using a fully implicit method for the accurate computation of time-dependent flows. We stress that our goal is, however, not to demonstrate that the code is successful in the sense of efficient production. Implicit methods are computationally much more expensive than explicit methods because they need to solve a set of non-linear equations. The efficiency of an implicit code is essentially set by the efficiency of the method used to solve this difficult task. The general framework for solving the non-linear system is the Newton-Raphson method. For tests and development purposes, we chose here a very simple approach where the Jacobian matrix is computed at each iteration, and where the linear system is solved *exactly* inside the Newton-Raphson loop. This results in a robust, but expensive non linear solver. A more efficient class of non linear solvers include *inexact Newton-Krylov methods*, which are Newton-Raphson methods where an iterative method is used to solve the linear system, see e.g. see Knoll & Keyes 2004 for a review. These methods are Jacobian free methods and gain computational time by using an iterative method to *approximately* solve the linear system during the Newton-Raphson loop (hence the name “inexact”). These methods are very popular in computational physics and we plan to implement and test them in a future version of the code. Once the most efficient non-linear solvers currently available are implemented, we will be able to benchmark our code against explicit ones.

Concerning our numerical method, we found that the maximum time step that we can use without being faced with convergence issues in the Newton-Raphson procedure is closely related to the fluid velocity in the numerical domain. We introduced the *advective time scale*

$$t_{\text{adv}} = \frac{\Delta x}{|u|}, \quad (44)$$

and showed that for the computation of a time-dependent flow the time step cannot grow much above the minimum value of this time step. The definition of Eq. 44 should be adapted in the multi-dimensional case. In other words, our numerical method cannot handle fluid movements across several cells, which is intuitively and physically understandable. We stress that such a criteria is also suggested by accuracy concerns. This limitation on the time step stems from the fact that in order to advance the solution over one time step, one has to solve a system of non-linear equations with a Newton-Raphson procedure. For time step larger than the advective time scale, the fluid moves across several cells and the solution changes significantly between two time steps. This change is essentially the limiting factor for increasing the time step, since the Newton-Raphson procedure encounters problems when relative changes between two time steps are too large. We stress that one could improve this by finding a better initial guess for the Newton-Raphson procedure, instead of simply taking the solution of the last time step. In 1D implicit simulations, a time

extrapolation from preceding time steps has been used successfully to produce an initial guess for the Newton procedure. However, in multi-dimensional flows, this procedure does not necessarily gives a satisfactory guess of the solution. Indeed our experience did not show any improvement by using such an extrapolated guess.

Our first developments show that using a fully implicit code to describe stellar interiors is realistic, providing a powerful tool for studying a full stellar structure characterised by low-to-moderate Mach numbers. Our test performed on the envelope of a giant star indeed highlights the possibility to extend the spatial domain to a large fraction of the star (80% in radius), where Mach numbers cover values from less than 1% to a few tens of percent. More generally, this implicit tool will be efficient for computing relaxation toward steady state solutions and very low Mach number flows, and to avoid the CFL constraint due to a particular location of the numerical domain. In the later case, it is possible to have a moderate Mach number flow in another region of the domain (this region would then be responsible for the limitation of the time step), as illustrated by the red giant case mentioned above.

Our first developments also stress the numerical difficulty of modelling a full star with a grid-based code. Since the typical length scales (e.g. pressure scale height) vary significantly between the deep interior of the star and the surface, it is not possible to resolve both the interior and the surface layers on a single grid. We plan to overcome this difficulty by implementing a system of nested grids refined in both the radial and tangential directions. Among other future developments of the code, we plan to extend it to 3D and to implement iterative solvers, rather than a direct method, involving parallelisation. Among our first astrophysical applications, we quote (i) the very early stages of evolution of low-mass stars, bridging the gap between the hydrodynamical phase of star formation and the quasi-static pre-main sequence evolutionary phase, (ii) the study of convection in pulsating stars, and (iii) the effect of fast rotation on convection (in low-mass stars, massive stars). But clearly, a wide range of astrophysical problems can be studied with such a tool. Based on our experience, we hope to achieve these future developments and the first astrophysical applications within a reasonable period. We also hope that our efforts demonstrate the feasibility of applying an implicit multi-dimensional hydrodynamical code to stellar interiors and will motivate the computational astrophysics community to develop such a tool.

*Acknowledgements.* MV acknowledges support from a Newton International Fellowship during part of this work. The numerical simulations were achieved thanks to the resources of the PSMN (“Pôle Scientifique de Modélisation Numérique”) at the ENS de Lyon. The authors are indebted to Cédric Mulet-Marquis and Emmanuel Lévêque for their significant contribution to an earlier version of the code. The authors are particularly thankful to Bernd Freytag and Ewald Mueller for many valuable and useful discussions which considerably helped the elaboration of this work. We also thank Sacha Brun for discussion and advice. Part of this work was funded by the European Research Council under the European Communitys 7th Framework Programme (FP7/2007-2013 Grant Agreement no. 247060) and by the French “Programme National de Physique Stellaire” (PNPS). Finally, the authors thank the anonymous referee for his/her useful comments that helped to improve the paper.



## References

- Almgren, A. S., Bell, J. B., Rendleman, C. A., & Zingale, M. 2006, ApJ, 637, 922
- Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., & Koster, J. 2001, SIAM Journal on Matrix Analysis and Applications, 23, 15
- Amestoy, P. R., Guermouche, A., & Pralet, S. 2006, Parallel Computing, 32, 136
- Arnett, D., Meakin, C., & Young, P. A. 2009, ApJ, 690, 1715
- Asplund, M., Ludwig, H., Nordlund, Å., & Stein, R. F. 2000, A&A, 359, 669
- Aubert, O., Prantzos, N., & Baraffe, I. 1996, A&A, 312, 845
- Baraffe, I., Chabrier, G., Allard, F., & Hauschildt, P. H. 2002, A&A, 382, 563
- Baraffe, I. & El Eid, M. F. 1991, A&A, 245, 548
- Bigot, L., Kervella, P., Thévenin, F., & Ségransan, D. 2006, A&A, 446, 635
- Boris, J. P., Grinstein, F. F., Oran, E. S., & Kolbe, R. L. 1992, Fluid Dynamics Research, 10, 199
- Brun, A. S. & Palacios, A. 2009, ApJ, 702, 1078
- Brun, A. S. & Toomre, J. 2002, ApJ, 570, 865
- Burrows, A., Dessart, L., Ott, C. D., & Livne, E. 2007, Phys. Rep., 442, 23
- Calhoun, D. A., Helzel, C., & LeVeque, R. J. 2008, SIAM Rev., 50, 723
- Colella, P. 1990, J. Comput. Phys., 87, 171
- Curtis, A. R., Powell, M. J. D., & Reid, J. K. 1974, IMA Journal of Applied Mathematics, 13, 117
- Dearborn, D. S. P., Lattanzio, J. C., & Eggleton, P. P. 2006, ApJ, 639, 405
- Deupree, R. G. 1990, ApJ, 357, 175
- Dintrans, B. & Brandenburg, A. 2004, A&A, 421, 775
- Dobler, W., Stix, M., & Brandenburg, A. 2006, ApJ, 638, 336
- Eggleton, P. P., Dearborn, D. S. P., & Lattanzio, J. C. 2008, ApJ, 677, 581
- Freytag, B., Ludwig, H., & Steffen, M. 1996, A&A, 313, 497
- Freytag, B., Steffen, M., & Dorch, B. 2002, Astronomische Nachrichten, 323, 213
- Gebremedhin, A. H., Manne, F., & Pothen, A. 2005, SIAM Rev, 47, 629
- Glasner, S. A. & Livne, E. 1995, ApJ, 445, L149
- Glasner, S. A., Livne, E., & Truran, J. W. 1997, ApJ, 475, 754
- Guzik, J. A. 2010, ArXiv e-prints
- Heroux, M., Bartlett, R., Hoekstra, V. H. R., et al. 2003, An Overview of Trilinos, Tech. Rep. SAND2003-2927, Sandia National Laboratories
- Hirsch, C., ed. 1990, Numerical Computation Of Internal & External Flows: Computational Methods for Inviscid and Viscous Flows (New York, NY, USA: John Wiley & Sons, Inc.)
- Hujeirat, A. 2005a, A&A, 430, 893
- Hujeirat, A. 2005b, Computer Physics Communications, 168, 1
- Hujeirat, A. A. & Thielemann, F. 2009, MNRAS, 400, 903
- Iglesias, C. A. & Rogers, F. J. 1996, ApJ, 464, 943
- Keppens, R., Toth, G., Botchev, M., & van der Ploeg, A. 1999, International Journal for Numerical Methods in Fluids, 30, 335
- Knoll, D. A. & Keyes, D. E. 2004, Journal of Computational Physics, 193, 357
- LeVeque, R. J. 1997, Journal of Computational Physics, 131, 327
- LeVeque, R. J. 2007, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (SIAM)
- Livne, E. 1993, ApJ, 412, 634
- Livne, E. 1999, ApJ, 527, L97
- Livne, E. & Arnett, D. 1995, ApJ, 452, 62
- Livne, E., Burrows, A., Walder, R., Lichtenstadt, I., & Thompson, T. A. 2004, ApJ, 609, 277
- Meakin, C. A. & Arnett, D. 2007a, ApJ, 665, 690
- Meakin, C. A. & Arnett, D. 2007b, ApJ, 667, 448
- Mihalas, D. & Mihalas, B. W. 1999, Foundations of radiation hydrodynamics, ed. Dover
- Mocák, M., Müller, E., Weiss, A., & Kifonidis, K. 2009, A&A, 501, 659
- Nonaka, A., Almgren, A. S., Bell, J. B., et al. 2010, ApJS, 188, 358
- Nordlund, A. & Stein, R. F. 2009, in American Institute of Physics Conference Series, Vol. 1171, American Institute of Physics Conference Series, ed. I. Hubeny, J. M. Stone, K. MacGregor, & K. Werner, 242–259
- Smagorinsky, J. 1963, Monthly Weather Review, 91, 99
- Sofia, S. & Chan, K. L. 1984, ApJ, 282, 550
- Stein, R. F. & Nordlund, A. 1998, ApJ, 499, 914
- Stone, J. M. & Norman, M. L. 1992, ApJS, 80, 753
- Toth, G., Keppens, R., & Botchev, M. A. 1998, A&A, 332, 1159
- van Leer, B. 1977, Journal of Computational Physics, 23, 276
- Woodward, P. R., Porter, D. H., & Jacobs, M. 2003, in Astronomical Society of the Pacific Conference Series, Vol. 293, 3D Stellar Evolution, ed. S. Turcotte, S. C. Keller, & R. M. Cavallo, 45–

## Appendix A: Tests cases

## A.1. Linear test cases

## A.1.1. Advection equation

In this section we test our numerical scheme with the 1D scalar advection equation:

$$\frac{\partial q}{\partial t} + a \frac{\partial q}{\partial x} = 0. \quad (\text{A.1})$$

We use a smooth profile as an initial condition:

$$f(x) = \sin(x), \quad x \in [0, 2\pi]. \quad (\text{A.2})$$

The advection velocity  $a$  is set to 1. We advect the sin wave for one unit of time and the resulting numerical solution  $q_{i+1/2}$  is compared with the exact solution  $q_{i+1/2}^0 = f(x_{i+1/2} - 1)$  by computing the  $L_1$  and  $L_\infty$  errors:

$$\|q - q^0\|_1 = \Delta x \sum_i |q_{i+1/2} - q_{i+1/2}^0| \quad (\text{A.3})$$

$$\|q - q^0\|_\infty = \max_i |q_{i+1/2} - q_{i+1/2}^0|. \quad (\text{A.4})$$

For a given time step and mesh size, the CFL number is defined as  $a\Delta t/\Delta x$ . We use a uniform discretisation in space and perform a sequence of runs with different resolutions ranging from 50 to 800 grid points by successive increase by a factor of 2. This is done for three values of the time step in order to investigate a wide range of CFL:  $\Delta t = 10^{-1}, 10^{-2}, 10^{-3}$ . We use the Van Leer limiter and the Crank-Nicholson method, which are both second-order methods.

Such a grid of simulations allows to study the convergence properties of our code. The actual order  $r$  of the scheme between two consecutive resolutions is computed as

$$r = \log \frac{\|q - q^0\|^{(1)}}{\|q - q^0\|^{(2)}} / \log 2, \quad (\text{A.5})$$

where (2) corresponds to a resolution twice higher than (1). The results are shown in Table A.1, A.2, and A.3.

Before discussing the results, it is useful to derive a result that will help for their interpretation. Going back to Eq. (A.1), the simplest implicit scheme that can be designed combines the first order backward Euler method with the first-order upwind method:

$$q_i^{n+1} = q_i^n - \frac{a\Delta t}{\Delta x} (q_i^{n+1} - q_{i-1}^{n+1}). \quad (\text{A.6})$$

For this scheme, it is straightforward to derive the so-called “modified equation” (see e.g. Hirsch 1990):

**Table A.1.** Linear advection,  $\Delta t = 10^{-1}$ .

N	$L_1$ -error	Rate	$L_\infty$ -error	Rate	CFL	Wall Time (s)
50	1.380e-02	-	1.152e-02	-	7.80e-01	0.9
100	3.457e-03	2.00	2.587e-03	2.16	1.58e+00	1.8
200	3.138e-03	0.14	7.491e-04	1.79	3.17e+00	3.5
400	3.252e-03	-0.05	8.115e-04	-0.12	6.35e+00	7.7
800	3.308e-03	-0.02	8.269e-04	-0.03	1.27e+01	27.5

**Table A.2.** Linear advection,  $\Delta t = 10^{-2}$ .

N	$L_1$ -error	Rate	$L_\infty$ -error	Rate	CFL	Wall Time (s)
50	1.890e-02	-	1.292e-02	-	7.80e-02	3.8
100	4.481e-03	2.08	4.882e-03	1.40	1.58e-01	7.5
200	1.027e-03	2.13	1.830e-03	1.42	3.17e-01	15.3
400	2.094e-04	2.29	6.732e-04	1.44	6.35e-01	34.5
800	3.996e-05	2.39	1.936e-04	1.80	1.27e+00	134.2

**Table A.3.** Linear advection,  $\Delta t = 10^{-3}$ .

N	$L_1$ -error	Rate	$L_\infty$ -error	Rate	CFL	Wall Time (s)
50	1.895e-02	-	1.293e-02	-	7.80e-03	35.2
100	4.530e-03	2.06	4.892e-03	1.40	1.58e-02	68.5
200	1.074e-03	2.08	1.845e-03	1.41	3.17e-02	134.4
400	2.569e-04	2.06	6.918e-04	1.42	6.35e-02	273.9
800	6.102e-05	2.07	2.569e-04	1.43	1.27e-01	1016.7

$$\frac{\partial q}{\partial t} + a \frac{\partial q}{\partial x} = \frac{a}{2}(\Delta x + \Delta t) \frac{\partial^2 q}{\partial x^2} + \text{H.O.T.}, \quad (\text{A.7})$$

which shows that formula A.6 solves an advection-diffusion problem since the leading term of the truncature error is physically equivalent to a viscous term with a numerical viscosity coefficient  $D = \frac{a}{2}(\Delta x + \Delta t)$ . Usually, with explicit schemes in mind, it is supposed that the ratio  $\Delta x/\Delta t$  is kept constant (i.e. constant CFL number) so that  $D \propto \Delta x$  or  $D \propto \Delta t$ , which is equivalent to the statement that scheme A.6 is first order in time and in space. On the other hand, when an implicit scheme is used, we cannot make such an assumption for the ratio  $\Delta x/\Delta t$ , as the CFL number is in principle arbitrary. In our discussion we use the expression of this numerical viscosity coefficient to interpret our results.

It appears that for CFL values above 1 (e.g.  $\Delta t = 10^{-1}$ , see Tab. A.1), the numerical error remains roughly constant when the resolution is increased. This suggests that, in this case, the time truncation error dominates the spatial one. With the expression of the numerical viscosity derived above, we are in a regime where  $D = \frac{a}{2}\Delta t = cte$ , so when the resolution is increased the numerical dissipation does not decrease. Another way of interpreting this result is that the solution is advected over more than one cell width during one time step so that a strong numerical damping results. On the other hand, for low values of the CFL number (e.g.  $\Delta t = 10^{-3}$ , see Table A.3), the spatial truncation error dominates, and we recover the second-order convergence of our spatial scheme in the  $L_1$  norm. We do not see an accumulation effect of the round-off errors for very small time step. Again using the numerical dissipation derived above, one has for a first-order scheme  $D = \frac{a}{2}\Delta x$ . We expect that for our second-order scheme, a similar calculation would lead to  $D \propto \Delta x^2$ , in agreement with the observed tendency. A lower rate of convergence measured

in the  $L_\infty$  is characteristic of the action of the limiter on the smooth extrema of the solution. This is a well known drawback of TVD schemes: their order decreases also at smooth extrema of the solution. Finally for  $\Delta t = 10^{-2}$ , we are in the regime where  $\Delta x \sim \Delta t$ , so when the mesh size is decreased, the numerical dissipation decreases in a non trivial fashion. For instance, considering the first-order numerical dissipation derived above, decreasing the mesh size by a factor of two would result in a decrease in the numerical dissipation by a factor of  $\frac{1+\lambda}{1+\lambda/2}$ , where  $\lambda = \Delta x/\Delta t$ . As a result the scheme does not behave as having a specific order in this case.

To summarise these results, we stress that the error in the numerical solution of partial derivative equations depends on both the spatial and temporal discretisation. When the temporal (resp. the spatial) truncation errors dominates, i.e. for very large (resp. very small) CFL numbers, decreasing the spatial (resp. temporal) step does not lead to more accurate results. It is therefore essential to choose both the mesh size and the time step in order to optimise the accuracy and the computational cost. Finally, we stress that the advection equation is not a suitable problem for implicit methods, since the stability criterion for an explicit scheme, namely  $a\Delta t < \Delta x$ , is also an accuracy criterion. The optimum accuracy/computational cost is therefore obtained for CFL number of order unity. Any advection computation at high CFL numbers suffers from serious numerical damping and lack of accuracy.

#### A.1.2. Diffusion equation

In this section, we test our diffusion scheme with the simple linear diffusion equation:

$$\frac{\partial T}{\partial t} = \chi \frac{\partial^2 T}{\partial x^2}. \quad (\text{A.8})$$

**Table A.4.** Linear diffusion,  $\Delta t = 10^{-2}$ .

N	$L_1$ -error	Rate	$L_\infty$ -error	Rate	CFL	Wall Time (s)
50	2.072e-03	-	1.824e-03	-	1.50e+00	0.4
100	5.237e-04	1.98	3.659e-04	2.32	6.13e+00	0.6
200	8.663e-04	-0.73	8.220e-04	-1.17	2.48e+01	1.0
400	9.727e-04	-0.17	9.487e-04	-0.21	9.95e+01	2.0
800	9.999e-04	-0.04	9.802e-04	-0.05	3.99e+02	6.6

**Table A.5.** Linear diffusion,  $\Delta t = 10^{-3}$ .

N	$L_1$ -error	Rate	$L_\infty$ -error	Rate	CFL	Wall Time (s)
50	2.825e-03	-	2.812e-03	-	1.50e-01	2.6
100	6.874e-04	2.04	6.743e-04	2.06	6.13e-01	4.9
200	1.629e-04	2.08	1.586e-04	2.09	2.48e+00	9.2
400	3.419e-05	2.25	3.158e-05	2.33	9.95e+00	18.3
800	4.973e-06	2.78	3.406e-06	3.21	3.99e+01	64.4

**Table A.6.** Linear diffusion,  $\Delta t = 10^{-4}$ .

N	$L_1$ -error	Rate	$L_\infty$ -error	Rate	CFL	Wall Time (s)
50	2.833e-03	-	2.823e-03	-	1.50e-02	24.8
100	6.963e-04	2.02	6.847e-04	2.04	6.13e-02	47.9
200	1.717e-04	2.02	1.690e-04	2.02	2.48e-01	91.4
400	4.265e-05	2.01	4.193e-05	2.01	9.95e-01	183.7
800	1.059e-05	2.01	1.038e-05	2.01	3.99e+00	640.5

For an initial Dirac distribution of amplitude  $Q$  at  $x=0$ , the analytic solution of Eq. (A.8) reads (see e.g. Mihalas & Mihalas 1999) as

$$T(x, t) = \frac{Q}{\sqrt{4\pi\chi t}} \exp\left(-\frac{x^2}{4\chi t}\right). \quad (\text{A.9})$$

The problem is solved numerically on a uniform grid spanning the domain  $[-2, 2]$ . For a given mesh size and time step, the CFL number is defined as  $\chi\Delta t/\Delta x^2$ . We use our second-order spatial scheme for diffusion, together with the second-order Crank-Nicholson scheme for time stepping.

For simplicity, we consider here  $\chi = Q = 1$ . Since it is not possible to represent a Dirac distribution exactly on a discrete mesh, we adopt as an initial condition the solution of Eq. (A.9) at  $t=0.025$ . The solution is evolved until  $t=1$ , and we compare the numerical solution with the analytic solution, as done in the previous section for the linear advection problem. Again we perform a sequence of grid resolutions ranging from 50 to 800 grid points. This is done for three different time steps  $\Delta t = 10^{-2}$ ,  $10^{-3}$ , and  $10^{-4}$ , yielding a wide range of CFL numbers. The results are shown in Tables A.4, A.5 and A.6.

As in the previous section, when the time step is large ( $\Delta t = 10^{-2}$ , see Table A.4), the temporal truncation error dominates, and the numerical error does not decrease when the resolution is increased. On the other hand, for small time steps (e.g.  $\Delta t = 10^{-4}$ , see Table A.4), the spatial truncation error dominates, and we recover a second-order convergence. In this case, both the  $L_1$  and the  $L_\infty$  show second-order behaviour, since there is no limiting process in the construction of the interface fluxes. Again in the transition regime ( $\Delta t = 10^{-3}$ , see Table A.4), we do not observe a well-defined order. Such behaviour is likely for a similar reason to the advection case.

Here, an implicit scheme is more efficient than an explicit scheme since, for a given resolution, for instance  $N=800$ , the run with  $\Delta t = 10^{-2}$  has a solution that is

more accurate (by more than a factor of 10 in both norms) than the run with  $\Delta t = 10^{-3}$  (closer to the explicit time step), and has a computation time that is ten times shorter. In this case, the optimum ratio between accuracy and computational cost is obtained for CFL numbers larger than 1, because here the CFL condition is not an accuracy criterion.

## A.2. Non-linear test cases

### A.2.1. Sod test

A typical test for non-linear hydro is the 1D Sod shock-tube test. This is a Riemann problem with the following initial right and left states:  $\rho_L = 1$ ,  $p_L = 1$ ,  $\rho_R = 0.125$ ,  $p_R = 0.1$ ; the gas is initially at rest. The solution of the Sod problem is computed until  $t = 0.25$  on the domain  $[-0.5, 0.5]$ , where the interface between the initial states is located at  $x = 0$ . The number of grid points is 400. For simplicity and since the goal is to show how our code deals with shocks, we set the time step to keep the hydrodynamic CFL equal to 1.

The left panel of Fig. A.1 shows the result of the computation. One can easily identify the shock front, the contact discontinuity, and the rarefaction wave. The location and amplitude of these features are in excellent agreement with the analytical solution. However, one can see that spurious numerical effects affect the solution. Oscillations are produced at the location of the shock, despite the use of a limiter. An over/under shoot is also visible at the tail of the rarefaction wave. Solving the total energy equation and/or limiting the primitive variables (i.e. pressure) could improve these results. Otherwise, a possible cure is to introduce an artificial viscosity of the form:

$$\nu = C\Delta x c_s^2, \quad (\text{A.10})$$

where  $c_s$  is the sound speed,  $\Delta x$  the mesh size, and  $C$  a dimensionless parameter that has to be fine-tuned to obtain

acceptable results. The right panel of Fig. A.1 shows the result of the computation when  $C = 0.5$ . We stress that we do not encounter shocks in our current calculations of stellar models.

### A.2.2. Barenblatt test

To test our diffusion scheme in the non-linear regime, we performed various computations of the propagation of non-linear conduction wave, also known as the Barenblatt problem. We solve the equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial}{\partial x} \left( T^\beta \frac{\partial T}{\partial x} \right), \quad (\text{A.11})$$

which is a diffusion equation with a non-linear diffusivity law  $\chi = \alpha T^\beta$ .

As in the linear diffusion problem, we assume that an instantaneous pulse of energy is released at  $x = 0$ . Let  $Q$  be the amplitude of the temperature pulse released in the medium. Following Mihalas & Mihalas (1999), the analytical solution of the problem is

$$T(x, t) = \left( \frac{Q^2}{\alpha t} \right)^{1/\beta+2} f\left( \frac{x}{\alpha Q^\beta t} \right), \quad (\text{A.12})$$

with

$$f(\xi) = \left( \frac{\beta \xi_0^2}{2(\beta+2)} \right)^{1/\beta} \left[ 1 - \left( \frac{\xi}{\xi_0} \right)^2 \right]^{1/\beta} \quad \text{for } \xi < \xi_0 \quad (\text{A.13})$$

$$= 0, \quad \text{otherwise.} \quad (\text{A.14})$$

$\xi_0$  is defined by

$$\xi_0 = \left[ \frac{(\beta+2)^{\beta+1} 2^{1-\beta} \Gamma(\frac{1}{2} + 1/\beta)}{\beta \pi^{\beta/2} \Gamma(1/\beta)} \right]^{1/\beta+2}. \quad (\text{A.15})$$

The problem can be cast in dimensionless variables, defined in the following way:

$$\bar{t} = \alpha Q^\beta t \quad (\text{A.16})$$

$$\bar{x} = x/\xi_0 \quad (\text{A.17})$$

$$\bar{T} = T/Qf(0). \quad (\text{A.18})$$

In these variables, Eq. (A.11) becomes

$$\frac{\partial \bar{T}}{\partial \bar{t}} = \frac{\beta}{2(\beta+2)} \frac{\partial}{\partial \bar{x}} \left( \bar{T}^\beta \frac{\partial \bar{T}}{\partial \bar{x}} \right), \quad (\text{A.19})$$

and Eq. (A.12) now reads as

$$\bar{T}(\bar{x}, \bar{t}) = \bar{t}^{-1/\beta+2} \left[ 1 - \left( \frac{\bar{x}}{\bar{t}^{1/\beta+2}} \right)^2 \right]^{1/\beta}. \quad (\text{A.20})$$

The advantage is that, in the dimensionless form, the only parameter left is  $\beta$ . We performed runs for  $\beta = 1, 3, 5, 7$ , going from mildly to strongly non-linear regimes. The problem is solved in the dimensionless form for the domain  $[-1.5, 1.5]$ . Our spatial discretisation uses 100 grid points, and again the time step is set so that the CFL number is equal to 1. As we cannot represent a  $\delta$  pulse on a discrete grid accurately, the initial condition in each case is taken to be the analytical solution (A.20) at  $\bar{t} = 0.1$ . Figure A.2 shows the results at different times and demonstrates the ability of our code to resolve and track the heat front even for strongly non-linear cases.

## Appendix B: Formulae

We recall the geometrical factors for the scalar, cell centred, quantities:

$$\begin{aligned} V_{i+1/2,j+1/2} &= \frac{2\pi}{3} (r_{i+1}^3 - r_i^3) (\cos \theta_j - \cos \theta_{j+1}) \\ S_{i,j+1/2}^r &= 2\pi r_i^2 (\cos \theta_j - \cos \theta_{j+1}) \\ S_{i+1/2,j}^\theta &= \pi \sin \theta_j (r_{i+1}^2 - r_i^2). \end{aligned} \quad (\text{B.14})$$

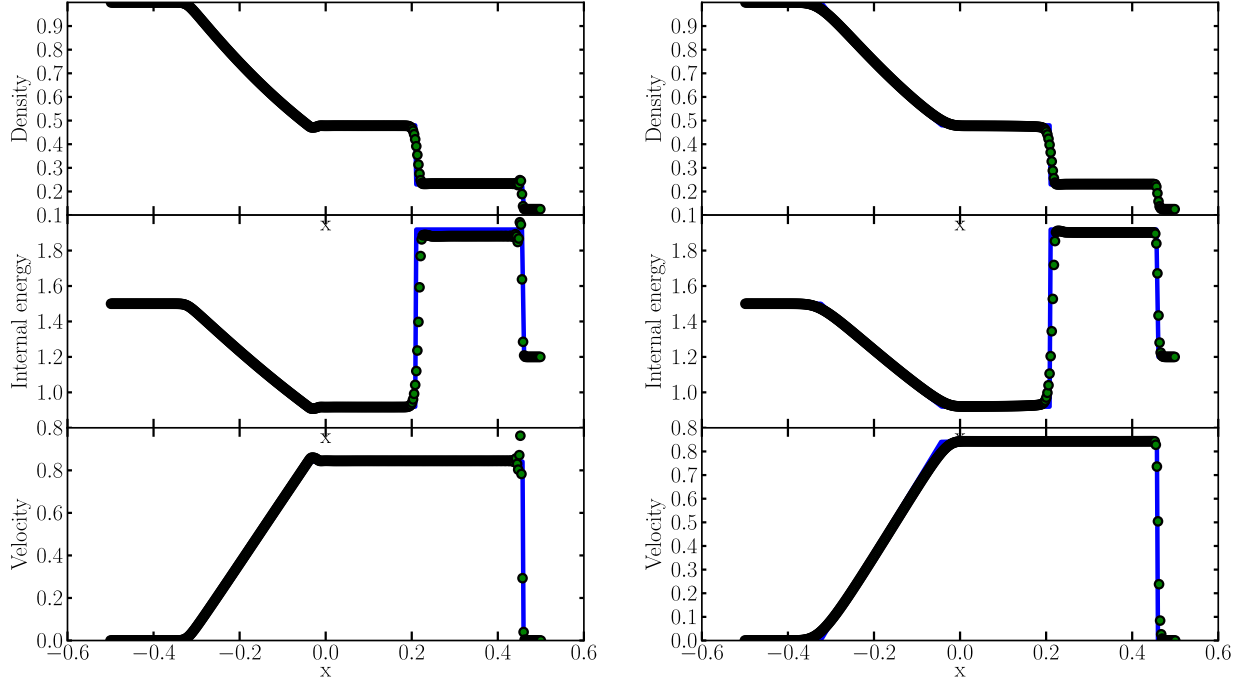
Geometrical factors for the radial velocity

$$\begin{aligned} \hat{V}_{i,j+1/2} &= \frac{2\pi}{3} (r_{i+1/2}^3 - r_{i-1/2}^3) (\cos \theta_j - \cos \theta_{j+1}) \\ \hat{S}_{i-1/2,j+1/2}^r &= 2\pi r_{i-1/2}^2 (\cos \theta_j - \cos \theta_{j+1}) \\ \hat{S}_{i,j}^\theta &= \pi \sin \theta_j (r_{i+1/2}^2 - r_{i-1/2}^2). \end{aligned} \quad (\text{B.15})$$

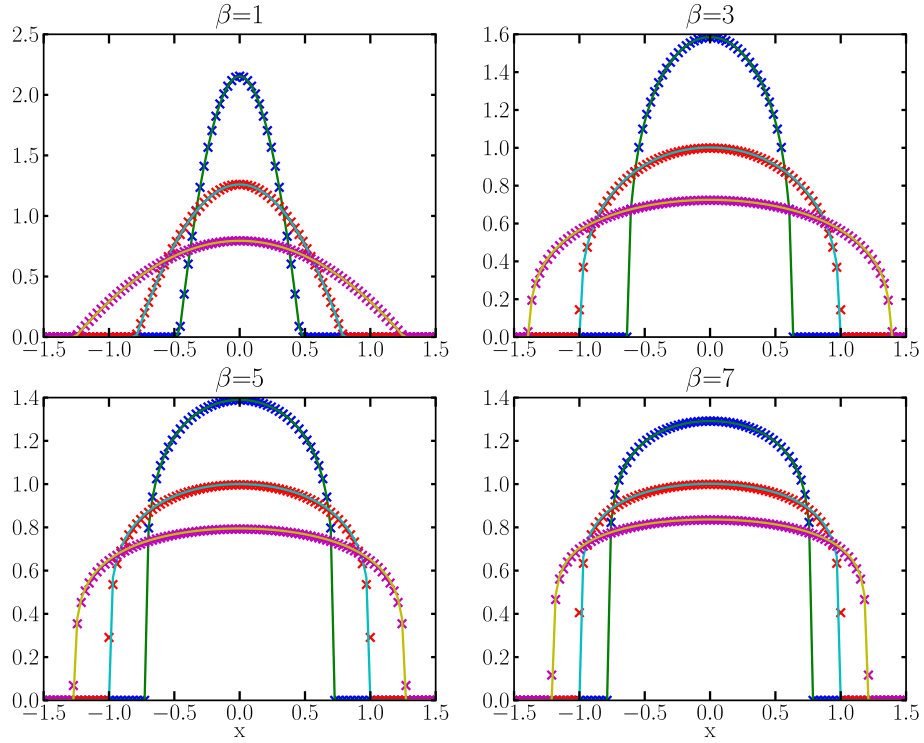
Geometrical factors for the tangential velocity

$$\begin{aligned} \check{V}_{i+1/2,j} &= \frac{2\pi}{3} (r_{i+1}^3 - r_i^3) (\cos \theta_{j-1/2} - \cos \theta_{j+1/2}) \\ \check{S}_{i,j}^r &= 2\pi r_i^2 (\cos \theta_{j-1/2} - \cos \theta_{j+1/2}) \\ \check{S}_{i+1/2,j-1/2}^\theta &= \pi \sin \theta_{j-1/2} (r_{i+1}^2 - r_i^2). \end{aligned} \quad (\text{B.16})$$

The complete set of ordinary differential equations resulting from our spatial discretisation in spherical coordinates are summarised in Tab. B.1.



**Fig. A.1.** Results of the Sod test at  $t = 0.25$  without artificial viscosity (left panel) and with  $C = 0.5$  (right panel). The dots are the numerical solution, the solid line is the analytical solution. The time step corresponds to  $\text{CFL}_{\text{hydro}} = 1$ .



**Fig. A.2.** Results of the Barenblatt problem for  $\beta = 1, 3, 5, 7$ . The solution is shown at  $\bar{t} = 0.1, 1, 2$  for  $\beta = 1$  and at  $\bar{t} = 0.1, 1, 5$  for  $\beta = 3, 5, 7$ , from upper to lower curves. The crosses are the numerical solutions, the solid lines are the analytical solutions given by Eq. (A.20).

$$\frac{d}{dt}\rho_{i+1/2,j+1/2}V_{i+1/2,j+1/2} = -(S_{i+1,j+1/2}^r\tilde{\rho}_{i+1,j+1/2}u_{i+1,j+1/2}^r - S_{i,j+1/2}^r\tilde{\rho}_{i,j+1/2}u_{i,j+1/2}^r \quad (\text{B.1})$$

$$+ S_{i+1/2,j+1}^\theta\tilde{\rho}_{i+1/2,j+1}u_{i+1/2,j+1}^\theta - S_{i,j+1/2}^\theta\tilde{\rho}_{i,j+1/2}u_{i,j+1/2}^\theta)$$

$$\frac{d}{dt}\rho_{i+1/2,j+1/2}e_{i+1/2,j+1/2}V_{i+1/2,j+1/2} = -(S_{i+1,j+1/2}^r\tilde{\rho}e_{i+1,j+1/2}u_{i+1,j+1/2}^r - S_{i,j+1/2}^r\tilde{\rho}e_{i,j+1/2}u_{i,j+1/2}^r \quad (\text{B.2})$$

$$+ S_{i+1/2,j+1}^\theta\tilde{\rho}e_{i+1/2,j+1}u_{i+1/2,j+1}^\theta - S_{i,j+1/2}^\theta\tilde{\rho}e_{i,j+1/2}u_{i,j+1/2}^\theta)$$

$$\begin{aligned} & -P_{i+1/2,j+1/2}(S_{i+1,j+1/2}^ru_{i+1,j+1/2}^r - S_{i,j+1/2}^ru_{i,j+1/2}^r + S_{i+1/2,j+1}^\theta u_{i+1/2,j+1}^\theta - S_{i,j+1/2}^\theta u_{i,j+1/2}^\theta) \\ & + \frac{ac}{3}\left(S_{i+1,j+1/2}^r\frac{1}{\bar{\rho}\bar{\kappa}_{i+1,j+1/2}}\frac{T_{i+3/2,j+1/2}^4 - T_{i+1/2,j+1/2}^4}{\Delta_{i+1}r} - S_{i,j+1/2}^r\frac{1}{\bar{\rho}\bar{\kappa}_{i,j+1/2}}\frac{T_{i+1/2,j+1/2}^4 - T_{i-1/2,j+1/2}^4}{\Delta_i r} \right. \\ & \left. + S_{i+1/2,j+1}^\theta\frac{1}{\bar{\rho}\bar{\kappa}_{i+1/2,j+1}}\frac{T_{i+1/2,j+3/2}^4 - T_{i+1/2,j+1/2}^4}{r_{i+1/2}\Delta_{j+1}\theta} - S_{i+1/2,j}^\theta\frac{1}{\bar{\rho}\bar{\kappa}_{i+1/2,j}}\frac{T_{i+1/2,j+1/2}^4 - T_{i+1/2,j-1/2}^4}{r_{i+1/2}\Delta_j\theta}\right) \end{aligned}$$

$$\frac{d}{dt}\bar{\rho}_{i,j+1/2}u_{i,j+1/2}^r\hat{V}_{i,j+1/2} = -(\hat{S}_{i+1/2,j+1/2}^r\widetilde{\rho u^r}_{i+1/2,j+1/2}\bar{u}_{i+1/2,j+1/2}^r \quad (\text{B.3})$$

$$- \hat{S}_{i-1/2,j+1/2}^r\widetilde{\rho u^r}_{i-1/2,j+1/2}\bar{u}_{i-1/2,j+1/2}^r$$

$$+ \hat{S}_{i,j+1}^\theta\widetilde{\rho u^r}_{i,j+1}\bar{u}_{i,j+1}^\theta - \hat{S}_{i,j}^\theta\widetilde{\rho u^r}_{i,j}\bar{u}_{i,j}^\theta)$$

$$- \frac{P_{i+1/2,j+1/2} - P_{i-1/2,j+1/2}}{\Delta_i r}\hat{V}_{i,j+1/2} + \bar{\rho}_{i,j+1/2}g_{i,j+1/2}\hat{V}_{i,j+1/2}$$

$$+ \hat{V}_{i,j+1/2}\frac{\bar{\rho}_{i,j+1/2}(\bar{u}_{i,j+1/2}^\theta)^2}{r_i}$$

$$\frac{d}{dt}\bar{\rho}_{i+1/2,j}u_{i+1/2,j}^\theta\check{V}_{i+1/2,j} = -(\check{S}_{i+1/2,j}^r\widetilde{\rho u^\theta}_{i+1/2,j}\bar{u}_{i+1/2,j}^r - \check{S}_{i,j}^r\widetilde{\rho u^\theta}_{i,j}\bar{u}_{i,j}^r \quad (\text{B.4})$$

$$+ \check{S}_{i+1/2,j+1/2}^\theta\widetilde{\rho u^\theta}_{i+1/2,j+1/2}\bar{u}_{i+1/2,j+1/2}^\theta$$

$$- \check{S}_{i+1/2,j-1/2}^\theta\widetilde{\rho u^\theta}_{i+1/2,j-1/2}\bar{u}_{i+1/2,j-1/2}^\theta)$$

$$- \frac{P_{i+1/2,j+1/2} - P_{i+1/2,j-1/2}}{r_{i+1/2}\Delta_j\theta}\check{V}_{i+1/2,j} - \check{V}_{i+1/2,j}\frac{\bar{\rho}_{i+1/2,j}\bar{u}_{i+1/2,j}^ru_{i+1/2,j}^\theta}{r_{i+1/2}} \quad (\text{B.5})$$

$$\frac{1}{\bar{\rho}\bar{\kappa}_{i,j+1/2}} = \frac{1}{2}\left(\frac{1}{\rho_{i-1/2,j+1/2}\kappa_{i-1/2,j+1/2}} + \frac{1}{\rho_{i+1/2,j+1/2}\kappa_{i+1/2,j+1/2}}\right) \quad (\text{B.6})$$

$$\frac{1}{\bar{\rho}\bar{\kappa}_{i+1/2,j}} = \frac{1}{2}\left(\frac{1}{\rho_{i+1/2,j-1/2}\kappa_{i+1/2,j-1/2}} + \frac{1}{\rho_{i-1/2,j+1/2}\kappa_{i+1/2,j+1/2}}\right) \quad (\text{B.7})$$

$$\bar{\rho}_{i,j+1/2} = (\rho_{i-1/2,j+1/2}V_{i-1/2,j+1/2} + \rho_{i+1/2,j+1/2}V_{i+1/2,j+1/2})/\hat{V}_{i,j+1/2} \quad (\text{B.8})$$

$$\bar{\rho}_{i+1/2,j} = (\rho_{i+1/2,j-1/2}V_{i+1/2,j-1/2} + \rho_{i+1/2,j+1/2}V_{i+1/2,j+1/2})/\check{V}_{i+1/2,j} \quad (\text{B.9})$$

$$\bar{u}_{i+1/2,j+1/2}^r = \frac{1}{2}(u_{i,j+1/2}^r + u_{i+1,j+1/2}^r) \quad \bar{u}_{i,j}^\theta = \frac{1}{2}(u_{i-1/2,j}^\theta + u_{i+1/2,j}^\theta) \quad (\text{B.10})$$

$$\bar{u}_{i,j}^r = \frac{1}{2}(u_{i,j-1/2}^r + u_{i,j+1/2}^r) \quad \bar{u}_{i+1/2,j+1/2}^\theta = \frac{1}{2}(u_{i+1/2,j}^\theta + u_{i+1/2,j+1}^\theta) \quad (\text{B.11})$$

$$\bar{u}_{i+1/2,j}^r = \frac{1}{2}(\bar{u}_{i,j}^r + \bar{u}_{i+1,j}^r) \quad \bar{u}_{i,j+1/2}^\theta = \frac{1}{2}(\bar{u}_{i,j}^\theta + \bar{u}_{i,j+1}^\theta) \quad (\text{B.12})$$

$$(\text{B.13})$$

**Table B.1.** Complete system of ordinary differential equations resulting from our finite volume discretisation on a staggered grid in spherical coordinates.