A Jacobian-free Newton-Krylov method for time-implicit multidimensional hydrodynamics. Physics-based preconditioning for sound waves and thermal diffusion

M. Viallet, T. Goffrey, I. Baraffe, D. Folini, C. Geroux, M. V. Popov, J. Pratt, R. Walder

▶ **To cite this version:**

M. Viallet, T. Goffrey, I. Baraffe, D. Folini, C. Geroux, et al.. A Jacobian-free Newton-Krylov method for time-implicit multidimensional hydrodynamics. Physics-based preconditioning for sound waves and thermal diffusion. Astronomy and Astrophysics - A&A, 2016, 586, 10.1051/0004-6361/201527339 . insu-03710590

HAL Id: insu-03710590
https://insu.hal.science/insu-03710590

Submitted on 1 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A&A 586, A153 (2016)
DOI: 10.1051/0004-6361/201527339
© ESO 2016

Astronomy & Astrophysics

A Jacobian-free Newton-Krylov method for time-implicit multidimensional hydrodynamics

Physics-based preconditioning for sound waves and thermal diffusion

M. Viallet[1], T. Goffrey[2], I. Baraffe[2,3,1], D. Folini[3], C. Geroux[2], M. V. Popov[3], J. Pratt[2], and R. Walder[3]

[1] Max-Planck Institut für Astrophysik, Karl Schwarzschild Strasse 1, 85741 Garching, Germany
e-mail: mviallet@mpa-garching.mpg.de
[2] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QL, UK
[3] École Normale Supérieure, Lyon, CRAL, UMR CNRS 5574, Université de Lyon, Lyon France

Received 10 September 2015 / Accepted 7 December 2015

ABSTRACT

This work is a continuation of our efforts to develop an efficient implicit solver for multidimensional hydrodynamics for the purpose of studying important physical processes in stellar interiors, such as turbulent convection and overshooting. We present an implicit solver that results from the combination of a Jacobian-free Newton-Krylov method and a preconditioning technique tailored to the inviscid, compressible equations of stellar hydrodynamics. We assess the accuracy and performance of the solver for both 2D and 3D problems for Mach numbers down to 10^{-6}. Although our applications concern flows in stellar interiors, the method can be applied to general advection and/or diffusion-dominated flows. The method presented in this paper opens up new avenues in 3D modeling of realistic stellar interiors allowing the study of important problems in stellar structure and evolution.

Key words. hydrodynamics – methods: numerical – stars: interiors

1. Introduction

The transport of heat, chemical species, and angular momentum in stellar interiors is governed by three-dimensional, nonlinear (magneto-)hydrodynamical processes that develop over a wide range of temporal and spatial scales. The study of these processes with numerical simulations is a powerful way to improve our understanding of stellar structure and stellar evolution. Unfortunately, the integration of the compressible hydrodynamical equations with time explicit methods comes with a constraint on the time step resulting from the propagation of sound waves. This is the well-known Courant-Friedrich-Lewy (CFL) stability condition. We define $\mathrm{CFL}_{\mathrm{hydro}}$ as the ratio between the time step and the largest explicit time step allowed by the CFL condition:

$$\mathrm{CFL}_{\mathrm{hydro}} = \max \frac{(|\boldsymbol{u}| + c_{\mathrm{s}})\,\Delta t}{\Delta x}, \tag{1}$$

where Δt is the time step, Δx the mesh spacing, c_{s} the adiabatic sound speed, and $\boldsymbol{u}$ the flow velocity. Time-explicit methods require $\mathrm{CFL}_{\mathrm{hydro}} \lesssim 1$. This results in values of the time step that are smaller than the typical time scale of the relevant processes (e.g., the convective turnover time scale), making this approach computationally demanding. Nevertheless, an explicit time-integration method remains the method of choice for multidimensional hydrodynamics in the astrophysical community (see, e.g., Bazán et al. 2003; Meakin & Arnett 2007; Mocák et al. 2011; Herwig et al. 2014). A way to overcome this limitation is to rely on sound-proof models, which filter sound waves. Popular sound-proof models are the Boussinesq, the anelastic, or the pseudo-incompressible models (see, e.g., Glatzmaier 2013, for a review). The use of such models, however, comes at the cost of a restricted range of applications due to the underlying approximations. Ideally, one seeks a way to efficiently solve the hydrodynamical equations regardless of the wide range of physical conditions characterizing stellar interiors (e.g., density stratification and a wide range of Mach numbers).

The MUlti-dimensional Stellar Implicit Code (MUSIC) follows a different approach by solving implicitly the fully compressible hydrodynamical equations (Viallet et al. 2011, 2013). The challenge for an implicit solver lies in the necessity of solving a large nonlinear system at each time step. In Viallet et al. (2013), the best performance was obtained with Newton-Krylov methods, which combine the Newton-Raphson method with an iterative linear solver. It was shown that the iterative solver requires preconditioning in order to achieve fast convergence for large $\mathrm{CFL}_{\mathrm{hydro}}$. In fact, within the framework of Newton-Krylov methods, the preconditioner is the crucial ingredient of the implicit solver. One of the important performance bottlenecks that was identified by the authors, particularly when considering three-dimensional calculations, is the inefficiency of "black-box" algebraic preconditioning techniques such as incomplete LU (ILU) factorizations for large CFL number computations. Furthermore, the memory requirement for the storage of the Jacobian matrix and the ILU factorization increases significantly in 3D, restricting the range of problems that can be addressed with such a method.

In this paper, we present an implicit solver which aims at overcoming these limitations. This is achieved by combining a Jacobian-free Newton-Krylov (JFNK) method with a preconditioner that is tailored for our physical equations, as described in more detail in Sect. 2. In Sect. 3, we design semi-implicit schemes that treat sound waves and thermal diffusion implicitly;

in Sect. 4, we show how these semi-implicit schemes can be utilized to form an efficient preconditioner for the Newton-Krylov method. We present in Sect. 5 results that illustrate the performance of the solver for idealized test problems and for stellar interiors. We conclude in Sect. 6.

2. Numerical description

MUSIC solves the equations describing the evolution of density, momentum, and internal energy, taking external gravity and thermal diffusion into account:

$$\partial_t \rho = -\boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}), \tag{2}$$

$$\partial_t (\rho e) = -\boldsymbol{\nabla} \cdot (\rho e \boldsymbol{u}) - p \boldsymbol{\nabla} \cdot \boldsymbol{u} + \boldsymbol{\nabla} \cdot (\chi \boldsymbol{\nabla} T), \tag{3}$$

$$\partial_t (\rho \boldsymbol{u}) = -\boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} p + \rho \boldsymbol{g}, \tag{4}$$

where ρ is the density, e the specific internal energy, $\boldsymbol{u}$ the velocity, p the gas pressure, T the temperature, $\boldsymbol{g}$ the gravitational acceleration, and χ the thermal conductivity. The system of equations is closed with an equation of state (EOS). For stellar interiors, these equations describe radiation-hydrodynamics in the diffusion limit. This is appropriate when the plasma is optically thick. In this case, the thermal conductivity due to photons is given by

$$\chi = \frac{16 \sigma T^3}{3 \kappa \rho}, \tag{5}$$

where κ is the Rosseland mean opacity, and σ the Stefan-Boltzmann constant. Furthermore, the EOS includes the contribution of radiation to the internal energy and pressure. Optionally, MUSIC can solve the total energy equation in place of the internal energy equation:

$$\partial_t (\rho \epsilon_t) = -\boldsymbol{\nabla} \cdot (\rho \epsilon_t \boldsymbol{u} + p \boldsymbol{u}) + \rho \boldsymbol{u} \cdot \boldsymbol{g} + \boldsymbol{\nabla} \cdot (\chi \boldsymbol{\nabla} T), \tag{6}$$

where $\epsilon_t = e + \boldsymbol{u}^2/2$ is the specific total energy.

We follow the method of lines and perform the spatial discretization independently of the time discretization (see, e.g., LeVeque 2007). The spatial discretization is performed using a finite volume method with staggered velocity components located at cell interfaces. The numerical fluxes are calculated using an upwind, monotonicity preserving method of van Leer (1974). The resulting scheme is second-order in space and total variation diminishing.

The "conserved" variables, for which the conservation Eqs. (2)–(4) and (6) are solved, are represented as the column vector $U = (\rho, \rho e \rho \boldsymbol{u})$ when solving the internal energy equation, or $U = (\rho, \rho \epsilon_t, \rho \boldsymbol{u})$ when solving the total energy equation. In MUSIC, the unknowns are different from the conserved variables U, and are represented as the column vector $X = (\rho, e, \boldsymbol{u})$ when solving the internal energy equation or $X = (\rho, \epsilon_t, \boldsymbol{u})$ when solving the total energy equation.

The spatial discretization yields a system of ordinary differential equations:

$$\frac{\mathrm{d}U}{\mathrm{d}t} = R_U(X), \tag{7}$$

where R_U contains the flux differencing and source terms.

The time discretization is carried out using the second-order Crank-Nicolson method:

$$U(X^{n+1}) = U(X^n) + \frac{\Delta t}{2}(R_U(X^{n+1}) + R_U(X^n)). \tag{8}$$

We define the nonlinear residual function as

$$F_U(X) = \frac{U(X) - U(X^n)}{\Delta t} - \frac{1}{2}(R_U(X) + R_U(X^n)), \tag{9}$$

so that

$$F_U(X^{n+1}) = 0 \tag{10}$$

defines the solution at time step $n + 1$. Equation (10) is solved with a Newton-Raphson method. At each Newton iteration, a linear problem of the form

$$J \delta X = -F_U(X) \tag{11}$$

must be solved, where $J = \partial F_U / \partial X$ is the Jacobian matrix.

The use of a Krylov iterative method like GMRES (Saad & Schultz 1986) is a standard practice for solving Eq. (11) when the matrix is large. However, Viallet et al. (2013) find that, for $\mathrm{CFL}_{\mathrm{hydro}} \gtrsim 10$, the iterative method requires preconditioning to remain effective. ILU factorizations can perform an adequate job at modest CFL numbers ($\mathrm{CFL}_{\mathrm{hydro}} \lesssim 100$), but becomes inefficient at larger values of the CFL number. Furthermore, we find that such a preconditioning technique significantly increases the memory requirements.

In this work, we adopt an approach in which the Jacobian matrix is never explicitly formed. Jacobian-free Newton-Krylov methods are a popular choice for the resolution of large nonlinear system of equations, see Knoll & Keyes (2004) for a review. Since we do not form the Jacobian matrix, algebraic preconditioning techniques, such as ILU factorizations, have to be abandoned. Preconditioning the JFNK method, particularly when the CFL number is large, remains important for performance. One of the main goals of this paper is the design of an efficient preconditioner adapted specifically to the physics of stellar interiors.

From a physical point of view, at large CFL numbers ($\mathrm{CFL}_{\mathrm{hydro}} \gtrsim 100$) waves propagate over a large portion, if not all of the computational domain during a single time step. Effectively, it is as if information propagates at an infinite speed, as in parabolic problems. This changes the mathematical nature of the problem, i.e., from hyperbolic to parabolic, resulting in numerical stiffness. To be efficient, the numerical method has to take this property into consideration. Multigrid methods attempt to exploit this property by exchanging information between the large and small scales, see for example Kifonidis & Müller (2012). We adopt another approach which consists of using legacy methods, known as semi-implicit (SI) schemes, as preconditioners for the Krylov solver. This strategy is known as "physics-based preconditioning" (PBP), as the preconditioner is tailored for the physical problem, see, e.g., Mousseau et al. (2000), Knoll & Keyes (2004), Reisner et al. (2005), Park et al. (2009). SI schemes treat implicitly only the terms that are responsible for numerical stiffness. The scheme derived this way is numerically stable for $\mathrm{CFL}_{\mathrm{hydro}}$ greater than one, as the stiff physics is treated implicitly. However, the accuracy of the solution obtained by the SI scheme is usually quite poor due to the approximations involved in the derivation of the scheme (see Sect. 3). Good accuracy can be achieved by embedding such a scheme within a Newton-Krylov method as a preconditioner. This work closely follows Park et al. (2009), adapting their method to our numerical scheme and physical equations.

3. Semi-implicit schemes for gas dynamics

In this section, we derive SI schemes for the hydrodynamic Eqs. (2)–(4) and (6) which treat sound waves and thermal diffusion implicitly. The remaining terms (e.g., advection) are treated

explicitly. In this section, our only concern is to design schemes that are stable and inexpensive, rather than accurate. Later, we will use these schemes as preconditioners for a fully implicit and accurate method.

3.1. Equations for p, e, and $\boldsymbol{u}$

Our SI schemes are derived from the evolution equations for the primitive variables $V = (p, e, \boldsymbol{u})$. These are

$$\partial_t p + \boldsymbol{u}\cdot\boldsymbol{\nabla} p = -\Gamma_1 p \boldsymbol{\nabla}\cdot\boldsymbol{u} + (\Gamma_3 - 1)\boldsymbol{\nabla}\cdot(\chi\boldsymbol{\nabla} T), \tag{12}$$

$$\partial_t e + \boldsymbol{u}\cdot\boldsymbol{\nabla} e = -\frac{p}{\rho}\boldsymbol{\nabla}\cdot\boldsymbol{u} + \frac{1}{\rho}\boldsymbol{\nabla}\cdot(\chi\boldsymbol{\nabla} T), \tag{13}$$

$$\partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \boldsymbol{g}, \tag{14}$$

where Γ_1 and Γ_3 are the generalized adiabatic indices for a general equation of state. For a perfect gas without any internal degrees of freedom, these adiabatic indices reduce to $\Gamma_1 = \Gamma_3 = \gamma$, where γ is the usual adiabatic index. The detailed derivation of the pressure equation, Eq. (12), is given in Appendix A.1.

To simplify the notation and without loss of generality, we will consider the one-dimensional version of these equations:

$$\partial_t p + u\partial_x p = -\Gamma_1 p\partial_x u + (\Gamma_3 - 1)\partial_x(\chi\partial_x T), \tag{15}$$

$$\partial_t e + u\partial_x e = -\frac{p}{\rho}\partial_x u + \frac{1}{\rho}\partial_x(\chi\partial_x T), \tag{16}$$

$$\partial_t u + u\partial_x u = -\frac{1}{\rho}\partial_x p - g, \tag{17}$$

where we assumed that $\boldsymbol{g} = -g\boldsymbol{e}_x$, where $\boldsymbol{e}_x$ is a unity vector in the x-direction. Extension of the numerical scheme to higher dimensions is straightforward.

3.2. Transformation matrices

Having introduced the conserved variables U, the independent variables X, and the primitive variables V, we will need the transformation matrices $\partial V/\partial U$ and $\partial X/\partial V$, which are defined as:

$$\delta V = \frac{\partial V}{\partial U}\delta U, \tag{18}$$

$$\delta X = \frac{\partial X}{\partial V}\delta V. \tag{19}$$

These matrices are given below for both the case of the internal and total energy equations, and the details of their derivation is postponed to Appendix B.

3.2.1. Internal energy equation

When solving for the internal energy equation, $U = (\rho, \rho e, \rho u)$ and $X = (\rho, e, u)$, the transformation matrices take the following form:

$$\frac{\partial V}{\partial U} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e - \frac{e}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & \frac{1}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ -e/\rho & 1/\rho & 0 \\ -u/\rho & 0 & 1/\rho \end{pmatrix}, \tag{20}$$

and

$$\frac{\partial X}{\partial V} = \begin{pmatrix} \left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1} & -\left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1}\left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{21}$$

The required derivatives are those typically provided by EOS routines.

3.2.2. Total energy equation

When solving for the total energy equation, $U = (\rho, \rho\epsilon_t, \rho u)$ and $X = (\rho, \epsilon_t, u)$, the matrices take the form:

$$\frac{\partial V}{\partial U} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e - \frac{\epsilon_t - u^2}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & \frac{1}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & -\frac{u}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho \\ -(\epsilon_t - u^2)/\rho & 1/\rho & -u/\rho \\ -u/\rho & 0 & 1/\rho \end{pmatrix}, \tag{22}$$

and

$$\frac{\partial X}{\partial V} = \begin{pmatrix} \left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1} & -\left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1}\left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & u \\ 0 & 0 & 1 \end{pmatrix}. \tag{23}$$

3.3. SI scheme for sound waves

We first design a SI scheme that treats sound waves implicitly. In Sect. 3.3.1 we start with deriving the propagation equation for adiabatic acoustic fluctuations, which identifies the terms in the equations that need to be treated implicitly.

3.3.1. Propagation equation for acoustic fluctuations

In this section we neglect thermal diffusion and gravity. We linearize the 1D Eqs. (15) and (17) around a uniform background state:

$$\rho = \rho_0 + \rho', \tag{24}$$

$$p = p_0 + p', \tag{25}$$

$$u = 0 + u. \tag{26}$$

Keeping only linear terms in the perturbations, we obtain:

$$\partial_t p' = -\Gamma_1 p_0 \partial_x u, \tag{27}$$

$$\partial_t u = -\frac{1}{\rho_0}\partial_x p'. \tag{28}$$

Next, we take the derivatives of Eq. (27) with respect to t and Eq. (28) with respect to x. We substitute the result of the differentiation of Eq. (28) into the result of the differentiation of Eq. (27) to eliminate $\partial_{tx} u$ and obtain the wave equation that describes the adiabatic propagation of sound waves:

$$\partial_t^2 p' - a^2\partial_x^2 p' = 0, \tag{29}$$

where $a = \sqrt{\Gamma_1 p_0/\rho_0}$ is the adiabatic sound speed.

The terms on the r.h.s of Eqs. (27) and (28) are responsible for the propagation of sound waves. To overcome the corresponding CFL limit, we will treat them implicitly in the following section.

3.3.2. Pressure equation

To treat sound waves implicitly, Sect. 3.3.1 suggests that we treat the "$-\Gamma_1 p\partial_x u$" term in the pressure equation (Eq. (15)) implicitly, with a simple backward Euler method:

$$\frac{\delta p}{\Delta t} + \Gamma_1 p^n \partial_x u^{n+1} = -\left. u\partial_x p\right|^n + \left.(\Gamma_3 - 1)\partial_x(\chi\partial_x T)\right|^n. \tag{30}$$

Here $\delta p = p^{n+1} - p^n$, n being the temporal index. We use Picard linearization in order to keep the scheme linear[1]. All other terms in the equation are treated explicitly using the forward Euler method. Using Eq. (28), we approximate u^{n+1} with

$$u^{n+1} = u^n - \Delta t \frac{1}{\rho^n} \partial_x p^{n+1}, \tag{31}$$

which we substitute in Eq. (30) to obtain

$$\frac{\delta p}{\Delta t} - a^2 \Delta t \partial_x^2 p^{n+1} = -u \partial_x p\Big|^n - \Gamma_1 p^n \partial_x u^n + (\Gamma_3 - 1)\partial_x(\chi \partial_x T)\Big|^n, \tag{32}$$

where $a = \sqrt{\Gamma_1 p^n / \rho^n}$ is the adiabatic sound speed evaluated at time step n. The right hand side of Eq. (32) corresponds to the explicit discretization of the original equation, but on the left hand side a Laplacian operator illustrates the parabolic character of this equation.

3.3.3. Internal energy equation

We approach the internal energy equation, Eq. (16), in the same way as the pressure equation. Advection and thermal diffusion terms are discretized using an explicit scheme, and the compressional work is discretized using an implicit scheme. This produces:

$$\frac{\delta e}{\Delta t} + \frac{p^n}{\rho^n} \partial_x u^{n+1} = -u \partial_x e|^n + \frac{1}{\rho^n} \partial_x(\chi \partial_x T)|^n, \tag{33}$$

where $\delta e = e^{n+1} - e^n$. Again, we used Picard linearization to discretize the compressional work. We use again Eq. (31) to eliminate u^{n+1} in Eq. (33) to obtain

$$\frac{\delta e}{\Delta t} - \Delta t \frac{p^n}{(\rho^n)^2} \partial_x^2 p^{n+1} = -u \partial_x e|^n - \frac{p^n}{\rho^n} \partial_x u^n + \frac{1}{\rho^n} \partial_x(\chi \partial_x T)|^n. \tag{34}$$

The resulting equation is similar in form to the implicit version of the pressure equation, Eq. (32), as it contains the Laplacian of the pressure field.

3.3.4. Velocity equation

We discretize the pressure gradient in the velocity equation, Eq. (17), implicitly, using Picard linearization. All remaining terms are discretized explicitly. We obtain

$$\frac{\delta u}{\Delta t} + \frac{1}{\rho^n} \partial_x p^{n+1} = -u \partial_x u|^n - g, \tag{35}$$

where $\delta u = u^{n+1} - u^n$.

3.3.5. "δ-form" of the equations

By replacing $q^{n+1} = \delta q + q^n$ for all implicit terms in Eqs. (32), (34), (35), rather than only those terms involving time derivatives, one obtains the following system of equations:

$$\frac{\delta p}{\Delta t} - a^2 \Delta t \partial_x^2 \delta p = -\tilde{F}_p(p^n), \tag{36}$$

$$\frac{\delta e}{\Delta t} - \Delta t \frac{p^n}{(\rho^n)^2} \partial_x^2 \delta p = -\tilde{F}_e(e^n, p^n), \tag{37}$$

$$\frac{\delta u}{\Delta t} + \frac{1}{\rho^n} \partial_x \delta p = -\tilde{F}_u(u^n, p^n), \tag{38}$$

where we introduced the following residual functions:

$$\tilde{F}_p(p) = \frac{p - p^n}{\Delta t} - a^2 \Delta t \partial_x^2 p + u \partial_x p|^n + \Gamma_1 p^n \partial_x u^n - (\Gamma_3 - 1)\partial_x(\chi \partial_x T)|^n, \tag{39}$$

$$\tilde{F}_e(e, p) = \frac{e - e^n}{\Delta t} - \Delta t \frac{p^n}{(\rho^n)^2} \partial_x^2 p + u \partial_x e|^n + \frac{p^n}{\rho^n} \partial_x u^n - \frac{1}{\rho^n} \partial_x(\chi \partial_x T)|^n, \tag{40}$$

$$\tilde{F}_u(u, p) = \frac{u - u^n}{\Delta t} + \frac{1}{\rho^n} \partial_x p + u \partial_x u|^n + g. \tag{41}$$

The solution at time $n+1$ satisfies $\tilde{F}_p(p^{n+1}) = 0$, $\tilde{F}_e(e^{n+1}, p^{n+1}) = 0$, and $\tilde{F}_u(u^{n+1}, p^{n+1}) = 0$.

We write the system in a matrix form:

$$\tilde{J}_V \delta V = -\tilde{F}_V(V^n), \tag{42}$$

with $V = (p, e, u)$ and $\tilde{F}_V = (\tilde{F}_p, \tilde{F}_e, \tilde{F}_u)$. This formulation of the equations is known as the "δ−form". The block structure of $\tilde{J}_V$ is:

$$\tilde{J}_V = \begin{pmatrix} \tilde{J}_{p,p} & 0 & 0 \\ \tilde{J}_{e,p} & \tilde{J}_{e,e} & 0 \\ \tilde{J}_{u,p} & 0 & \tilde{J}_{u,u} \end{pmatrix}. \tag{43}$$

In this form, the system can be solved by operator splitting: the pressure Eq. (36) is first solved for δp, δe is deduced from the internal energy Eq. (37), and δu is deduced from the velocity Eq. (38). Note that in the energy equation, one can use the equality

$$\Delta t \partial_x^2 \delta p = \frac{1}{a^2}\Big(\frac{\delta p}{\Delta t} + \tilde{F}_p\Big), \tag{44}$$

which is obtained from the pressure equation, rather than writing the Laplacian of δp explicitly. In Sect. 3.5, we discuss how we solve numerically the parabolic equation for δp.

3.4. SI scheme for sound waves and thermal diffusion

When thermal diffusion is important, it can cause numerical stiffness. In this case, it also needs to be treated implicitly. This can be easily implemented in the framework of the previous section: all that is required is to treat thermal diffusion implicitly in the pressure and internal energy equations. Equation (32) now becomes:

$$\frac{\delta p}{\Delta t} - a^2 \Delta t \partial_x^2 p^{n+1} - (\Gamma_3 - 1)\partial_x(\chi^n \partial_x T^{n+1}) = -u \partial_x p\Big|^n - \Gamma_1 p^n \partial_x u^n, \tag{45}$$

[1] Picard linearization refers to the fact that we write $p^n \partial_x u^{n+1}$ instead of $p^{n+1} \partial_x u^{n+1}$ when applying the implicit discretization. This is an approximation, but it has the advantage of keeping the scheme linear in the new variables.

and Eq. (34) becomes:

$$\frac{\delta e}{\Delta t} - \Delta t \frac{p^n}{(\rho^n)^2}\partial_x^2 p^{n+1} - \frac{1}{\rho^n}\partial_x(\chi^n \partial_x T^{n+1}) = -u\partial_x e|^n - \frac{p^n}{\rho^n}\partial_x u^n. \quad (46)$$

In both Eqs. (45) and (46), we use Picard linearization to treat the diffusion term.

The new system for variables V in δ-form is

$$\frac{\delta p}{\Delta t} - a^2 \Delta t \partial_x^2 \delta p - (\Gamma_3 - 1)\partial_x(\chi^n \partial_x \delta T) = -\tilde{F}_p, \quad (47)$$

$$\frac{\delta e}{\Delta t} - \Delta t \frac{p^n}{(\rho^n)^2}\partial_x^2 \delta p - \frac{1}{\rho^n}\partial_x(\chi^n \partial_x \delta T) = -\tilde{F}_e, \quad (48)$$

$$\frac{\delta u}{\Delta t} + \frac{1}{\rho^n}\partial_x \delta p = -\tilde{F}_u, \quad (49)$$

where the residuals $\tilde{F}$ are unchanged, and given in Eqs. (39)–(41).

We use the linearized equation-of-state to express δT as

$$\delta T = \frac{1}{c_v}\delta e + \frac{\partial T}{\partial \rho}\Big|_e \delta\rho, \quad (50)$$

where $c_v = \partial e/\partial T|_\rho$ is the specific heat capacity at constant volume[2]. In general, the contribution due to density fluctuations is much smaller[3] and we neglect them:

$$\delta T \approx \frac{1}{c_v}\delta e. \quad (51)$$

This approximation is used to replace δT with δe in the previous system to obtain

$$\frac{\delta p}{\Delta t} - a^2 \Delta t \partial_x^2 \delta p - (\Gamma_3 - 1)\partial_x\Big(\frac{\chi^n}{c_v}\partial_x \delta e\Big) = -\tilde{F}_p, \quad (52)$$

$$\frac{\delta e}{\Delta t} - \Delta t \frac{p^n}{(\rho^n)^2}\partial_x^2 \delta p - \frac{1}{\rho^n}\partial_x\Big(\frac{\chi^n}{c_v}\partial_x \delta e\Big) = -\tilde{F}_e, \quad (53)$$

$$\frac{\delta u}{\Delta t} + \frac{1}{\rho^n}\partial_x \delta p = -\tilde{F}_u. \quad (54)$$

We now have a system of two coupled parabolic equations, as seen from the block structure of the matrix $\tilde{J}_V$:

$$\tilde{J}_V = \begin{pmatrix} \tilde{J}_{p,p} & \tilde{J}_{p,e} & 0 \\ \tilde{J}_{e,p} & \tilde{J}_{e,e} & 0 \\ \tilde{J}_{u,p} & 0 & \tilde{J}_{u,u} \end{pmatrix}. \quad (55)$$

The solution strategy of a system of equations in which thermal diffusion terms are treated implicitly is therefore more complicated than in the previous section: the two coupled parabolic Eqs. (52) and (53) have to be solved jointly for δe and δp, and finally δu is obtained from Eq. (54).

3.5. Numerical solution of the parabolic system

For both SI schemes presented previously, the numerical solution of a system of linear parabolic equations is required. This is accomplished in MUSIC using the Trilinos library (see Heroux et al. 2005). Specifically, MUSIC uses the iterative linear solver GMRES implemented in the package AztecOO to solve the parabolic system. The convergence of the linear solver is checked based on the criterion:

$$\|Px - b\|_2 < \eta' \|b\|_2, \quad (56)$$

where P is the system matrix, x the solution vector, b the r.h.s. of the linear system, and η' controls the accuracy of the solution. When setting $\eta' \leq 10^{-6}$, we find that the preconditioner has the same performances as when we use a direct solver to solve the parabolic system. However, in practice it is not necessary to solve the parabolic problem with such accuracy, as the preconditioner is only meant to provide an approximate solution of the problem. The results presented in this work were obtained by adopting a value $\eta' = 10^{-4}$. This value ensures an accuracy that is sufficient for the purpose of preconditioning. It is possible that the performance could be improved by adopting even larger values of η', as the decrease in the quality of the preconditioner could be mitigated by the decrease in its computational cost. This is left for future investigation. A multi-level preconditioner is applied to speed up convergence of the linear solver. We use the ML package of the Trilinos library to setup a multi-level preconditioner (Gee et al. 2006). Our preconditioner is based on the default parameters provided for the smoothed-aggregation setup in the ML package (parameters set "SA") with the following two modifications. First, instead of using the default method to estimate the eigenvalues of the matrix we use the 1-norm of the matrix. The default method of estimating the eigenvalues used a method based on a conjugate-gradient solution of the system, seeded by a random vector. This random vector caused simulations continued after restarting to differ from simulations without the restart, removing the ability to reproduce results. Second, we reduce the damping factor for the precondition from the default of 1.33 to 1.2, which in our case, results in fewer iterations for the parabolic solver to converge.

3.6. Time-stepping with SI schemes

The SI schemes designed in this section can be used as time-stepping methods to solve Eqs. (2)–(4) and (6). The time-marching algorithm is:

1. Given the solution at time step n, X^n, compute $F_U(X^n) = -R_U(X^n)$ (see Eq. (9));
2. Transform F_U into a residual for the primitive variables V:
$$\tilde{F}_V = \frac{\partial V}{\partial U} F_U; \quad (57)$$
3. Compute $\tilde{J}_V(V^n)$ corresponding to the desired SI scheme and solve
$$\tilde{J}_V \delta V = -\tilde{F}_V \quad (58)$$
for δV;
4. Transform δV into δX:
$$\delta X = \frac{\partial X}{\partial V}\delta V; \quad (59)$$
5. Set $X^{n+1} = X^n + \delta X$.

The scheme is linear (we used Picard linearization to deal with nonlinear terms) and only first-order in time (we used the forward/backward Euler methods). However the CFL limit is less restrictive as the terms associated with acoustic fluctuations were

[2] Here it is understood that partial derivatives are evaluated at time step n.

[3] It is zero for a perfect gas, for which $e = e(T)$.

discretized implicitly. Similarly, thermal diffusion does not imply any stability restriction on the time step if the second SI scheme is used. However, since the advective terms were discretized explicitly, a time step restriction based on the flow speed remains.

3.7. 2D isentropic vortex test

In this section, we test the SI scheme for sound waves. We use the isentropic vortex advection test originally proposed in Yee et al. (2000), and we adopt a setup similar to the one used by Kifonidis & Müller (2012) and Viallet et al. (2013) to test the accuracy of the SI scheme.

The initial state consists of an isentropic vortex (i.e., zero entropy perturbation) embedded in an uniform flow of norm $u_\infty = 1$. We use a Cartesian system of coordinates where the x-axis is taken in the direction of the flow. The vortex corresponds to the following perturbations in the state variables:

$$(\delta u, \delta v) = \frac{\beta}{2\pi} \mathrm{e}^{\frac{1-r^2}{2}} (-y, x), \tag{60}$$

$$\delta T = -\frac{(\gamma-1)\beta^2}{8\gamma\pi^2} \mathrm{e}^{1-r^2}, \tag{61}$$

where $r = \sqrt{x^2+y^2}$, $T = p/\rho$ (we set the gas constant $R = 1$ and work with dimensionless quantities), γ is the adiabatic index, and β the vortex strength. We use $\gamma = 1.4$ and $\beta = 0.75$, with initial conditions

$$\rho = (T_\infty + \delta T)^{\frac{1}{\gamma-1}}, \tag{62}$$

$$u = u_\infty + \delta u, \tag{63}$$

$$v = \delta v, \tag{64}$$

$$e = \frac{\rho^{\gamma-1}}{\gamma-1}, \tag{65}$$

where the subscript ∞ indicates the background value. The sound speed of the background is $c_\infty = \sqrt{\gamma T_\infty}$. The maximum velocity of the vortex is $v_{\max} = \max \|\delta \boldsymbol{u}\| = \beta/2\pi$. We define the vortex Mach number as $M_\mathrm{s} = v_{\max}/c_\infty = \beta/(2\pi\sqrt{\gamma T_\infty})$. By varying T_∞, we change the Mach number of the flow. We consider $T_\infty = 1, 10^2, 10^6, 10^{10}$, which corresponds to $M_\mathrm{s} = 10^{-1}$, 10^{-2}, 10^{-4}, 10^{-6} respectively.

The computations are performed on a 2D Cartesian domain $[-4, 4] \times [-4, 4]$. Initially, the vortex is centered on the origin. The vortex is advected until $t = 0.4$. The exact solution corresponds to the initial vortex profile being shifted by a distance equal to 0.4 in the x direction. To test the accuracy of the scheme, we compare the velocity in the direction of advection, u, with the expected analytic solution u^0. Kifonidis & Müller (2012) and Viallet et al. (2013) used the density field to monitor the error, but here the background density is changed when T_∞ is changed, which is not the case for the velocity field. We monitor three different norms of the error:

$$L_1\text{−error} : \|u - u^0\|_1 = \frac{1}{N_x N_y} \sum_{i,j} |u_{i,j} - u^0_{i,j}|, \tag{66}$$

$$L_2\text{−error} : \|u - u^0\|_2 = \sqrt{\frac{1}{N_x N_y} \sum_{i,j} (u_{i,j} - u^0_{i,j})^2}, \tag{67}$$

$$L_\infty\text{−error} : \|u - u^0\|_\infty = \max_{i,j} |u_{i,j} - u^0_{i,j}|, \tag{68}$$

where N_x, N_y are the grid dimensions, and the indices i, j range over the simulation grid.

We introduce the advective CFL number:

$$\mathrm{CFL}_\mathrm{adv} = \frac{u_\infty \Delta t}{\Delta x}, \tag{69}$$

where Δt is the time step and Δx the mesh spacing. For a vortex advected at u_∞, the advective CFL number provides a measure of the number of grid cells crossed per time step Δt.

To characterise the accuracy of the SI scheme, we perform a temporal convergence study. The resolution of the domain is set to 64^2 and we choose different time steps in order to cover a broad range of $\mathrm{CFL}_\mathrm{adv}$, between $\sim 10^{-2}$ and 3. This corresponds to $\mathrm{CFL}_\mathrm{hydro}$ as large as 4×10^5. Later, we will compare the result with a more accurate time integration method. We do not study convergence with spatial resolution here, as our spatial method remains the same in all schemes presented in this work, and is unchanged as compared to previous publications. A spatial resolution study is presented in Viallet et al. (2011).

We evolve the isentropic vortex varying both the Mach number and $\mathrm{CFL}_\mathrm{adv}$, and monitor the numerical errors. We expect two behavioral regimes. At low values of $\mathrm{CFL}_\mathrm{adv}$, the error should be approximately independent of the time step, as the spatial error dominates. At higher values of $\mathrm{CFL}_\mathrm{adv}$, the temporal error should dominate, and be proportional to Δt as the SI scheme is first-order in time. The results are presented in Fig. 1. The expected behavior is recovered, although the flat regime at low values of the time step is not clearly seen. Temporal truncation errors remain significant for small time steps, as a result of the approximations introduced when designing the SI scheme. The first-order character of the temporal discretization appears clearly for larger values of the time step. However, the most important conclusion is that the numerical error is independent of the Mach number. Effectively, we achieved our goal of designing a scheme that is independent of the stiffness of the background pressure field. We stress that this behavior is observed over a range of Mach numbers spanning five orders of magnitude.

Finally, although we successfully removed the stability constraint on the time step caused by sound waves, there is still a CFL-like condition based on the advective velocity. Such a stability limit is not evident from Fig. 1, as only a few models are computed for the largest time steps. Empirically, we determined that the SI scheme becomes unstable for $\mathrm{CFL}_\mathrm{adv} \gtrsim 0.2$.

4. Jacobian-free Newton-Krylov method and physics-based preconditioning

4.1. Newton-Krylov method

To solve the nonlinear system of equations, $F_U(X^{n+1}) = 0$, resulting from our fully implicit method we perform Newton-Raphson iterations. The Newton-Raphson procedure is initiated by taking an initial guess for the solution, typically $X^{(0)} = X^n$. At the kth Newton-Raphson iteration, the solution of a linear system is required:

$$J^{(k)} \delta X^{(k)} = -F_U(X^{(k)}), \tag{70}$$

where $\delta X^{(k)} = X^{(k+1)} - X^{(k)}$. The variable $X^{(k)}$ is the solution at iteration k, and

$$J^{(k)} = \frac{\partial F_U}{\partial X}(X^{(k)}) \tag{71}$$

is the Jacobian matrix at iteration k.

The components of δX and F_U can have considerably different numerical values as they represent different physical quantities in different units. For instance, densities can have typical

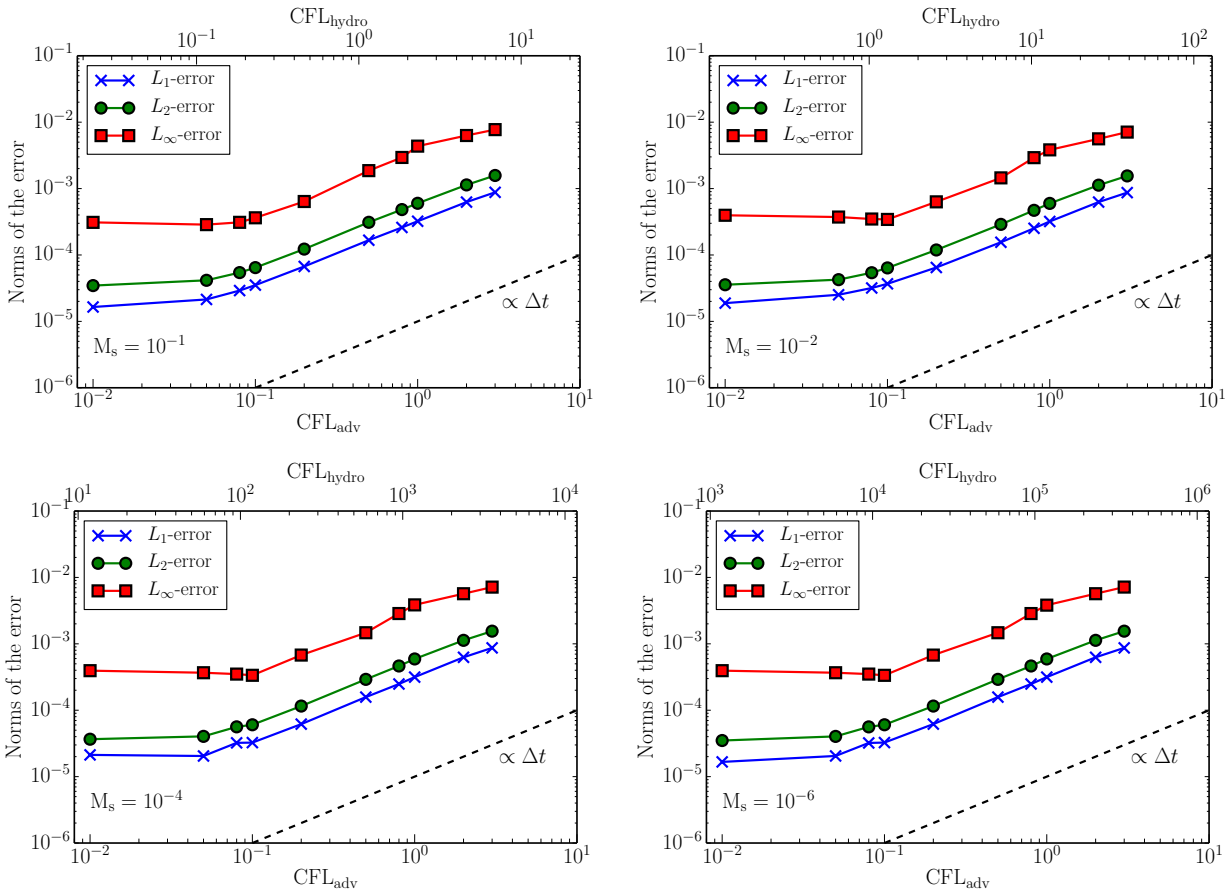

Fig. 1. Convergence tests of the SI scheme treating sound waves implicitly – advection of an isentropic vortex at different Mach numbers. The continuous lines show the norm of the errors measured in the velocity component parallel to the direction of advection.

values around 10^{-4} g/cc, whereas specific internal energies have values around 10^{14} erg/g. Also, due to the stratification of stellar interiors, some variables, such as the density, can vary by several orders of magnitude throughout the domain. Such a wide range of values can cause numerical difficulties due to round-off errors. Therefore, before the system (70) can be solved, it is necessary to scale it. We introduce two diagonal matrices L and R to scale Eq. (70):

$$\left(L^{-1}J^{(k)}R\right)\left(R^{-1}\delta X^{(k)}\right) = -L^{-1}F_U(X^{(k)}). \tag{72}$$

As L and R are diagonal matrices, we use the same symbol to represent their diagonal entries as a vector. The size of these vectors is equal to the number of variables multiplied by the number of cells. Each cell is treated in the same way, and the definitions of R and L only differ for different variables:

$$L_\rho = \rho^{(k)}, \qquad R_\rho = \rho^{(k)},$$
$$L_e = \rho^{(k)}e^{(k)}, \qquad R_e = e^{(k)},$$
$$L_u = \rho^{(k)}\max(|u^{(k)}|, \alpha_1 c_s^{(k)}), \qquad R_u = \max(|u^{(k)}|, \alpha_2 c_s^{(k)}),$$

where $c_s^{(k)}$ is the adiabatic sound speed computed from the solution at iteration k. R represents the typical value of the unknown vector $X^{(k)}$, and attempts to remove both the effects of units and stratification. We follow a similar idea for L and use the typical value of the conserved variables to scale the residual vector F_U. However, as velocities can be arbitrarily small, it is necessary to introduce a minimum velocity, here measured relative to the sound speed using the parameters α_1, α_2 in the definitions of L and R. The work described in Viallet et al. (2011) and Viallet et al. (2013) used $\alpha_1 = \alpha_2 = 1$. After testing, we found that $\alpha_1 = 10^{-5}$ and $\alpha_2 = 1$ gives good performance for a wide range of Mach numbers, typically $10^{-6} \lesssim M_s \lesssim 10^{-1}$, see discussion in Sect. 5.

The Newton-Raphson procedure is terminated when the relative corrections fall below a certain value ϵ:

$$\|R^{-1}\delta X^{(k)}\|_\infty < \epsilon. \tag{73}$$

In Viallet et al. (2013), it was shown that the nonlinear tolerance ϵ has to be chosen small enough so that the truncation errors of the scheme dominate the numerical error. We follow their recommendation and set $\epsilon = 10^{-6}$. Finally, if Eq. (73) is already fulfilled at the first iteration, we enforce a second Newton-Raphson iteration. For the sake of clarity, we drop from now on the superscript k of the outer nonlinear iteration of the Newton-Raphson procedure, and we do not carry the scaling matrices L and R in the notation in the rest of the paper.

We use the GMRES method to solve iteratively Eq. (72). We start from an initial guess δX_0, and we define the initial residual as $r_0 = -F_U(X) - J\delta X_0$. In practice, we choose $\delta X_0 = 0$ so that $r_0 = -F_U(X)$. At the pth iteration, the GMRES method seeks an approximation δX_p of the solution by solving a minimization problem in the pth Krylov space $\mathcal{K}_p$ of J:

$$\delta X_p \in \mathcal{K}_p(J) = \mathrm{span}\left(r_0, Jr_0, \ldots, J^{p-1}r_0\right). \tag{74}$$

The dimension of the search space increases at each iteration until convergence is achieved. The convergence of the linear solver is tested with the criterion

$$||J\delta X_p + F_U(X)||_2 < \eta||F_U(X)||_2, \tag{75}$$

where η is a parameter that determines the accuracy of the solution. Typical values of η that are used in this paper are $\eta = 10^{-2}$ and $\eta = 10^{-4}$, see discussion in Sect. 5.

4.2. Jacobian-free approach

To build successive Krylov spaces, the GMRES algorithm computes the action of the Jacobian matrix on a vector. This is the only use of the Jacobian operator, and we take advantage of the fact that this operation can be approximated by finite-differencing:

$$J(\boldsymbol{u})\boldsymbol{v} \approx \frac{F(\boldsymbol{u} + \delta\boldsymbol{v}) - F(\boldsymbol{u})}{\delta}, \tag{76}$$

where δ is a small number. We rely on the implementation of matrix-free operators available from the Trilinos package NOX. This package contains two preset options for calculating δ:

$$\delta = \lambda\left(\lambda + \frac{||\boldsymbol{u}||}{||\boldsymbol{v}||}\right), \tag{77}$$

and,

$$\delta = \lambda\left(\frac{10^{-12}}{\lambda} + \frac{|\boldsymbol{u}\cdot\boldsymbol{v}|}{\boldsymbol{v}\cdot\boldsymbol{v}}\right)\mathrm{sign}\,(\boldsymbol{u}\cdot\boldsymbol{v})\,. \tag{78}$$

In both cases, λ is a small parameter with a default value of 10^{-6}. The standard choice in MUSIC is to use Eq. (77), as it gives the best results (see discussion in Sect. 5).

In this Jacobian-free approach, the Jacobian matrix is not needed explicitly, lowering the memory cost of the scheme. Instead, computing the action of the Jacobian on a given vector requires one evaluation of the nonlinear residual in Eq. (76), assuming that $F(\boldsymbol{u})$ has been already computed and stored.

When J has a large condition number, the Krylov method fails to converge in an acceptable number of iterations (a few dozen) as the Krylov space is dominated by the direction of the eigenvector associated with the largest eigenvalue. In such cases, preconditioning is necessary. In this work, we use the SI schemes presented in Sect. 3 as preconditioners for the Krylov method. This is detailed in the next section.

4.3. Right-preconditioning of GMRES with SI schemes

Right-preconditioning of system (72) corresponds to solving the equivalent system:

$$(JM^{-1})\delta X' = -F_U(X), \tag{79}$$

$$M\delta X = \delta X', \tag{80}$$

where M is the preconditioning matrix. $\delta X'$ is an intermediate solution vector, which once known, is used to find the solution δX. If the preconditioning matrix is a good approximation of J, i.e., JM^{-1} has a low condition number, the Krylov space of JM^{-1} is better suited to construct an approximation of the solution

$$\delta X'_p \in \mathcal{K}_p(JM^{-1}) = \mathrm{span}\Big(r_0, JM^{-1}r_0, \ldots, (JM^{-1})^{p-1}r_0\Big). \tag{81}$$

Once a suitable solution $\delta X'$ has been found in the search space, based on the same convergence criterion as (75), a final linear system, Eq. (80), has to be solved to get the actual solution δX.

The key part of the right-preconditioning process is the application of JM^{-1} on a Krylov vector v, provided by GMRES. This operation is required at each iteration to build the successive Krylov spaces. In right-preconditioning, $JM^{-1}v$ is computed in two steps:

1. Solve $Mw = v$ for w;
2. Apply J to w.

The first step requires the inversion of a linear system; the second step requires the action of the Jacobian on the vector w and is approximated by a finite-difference formula (Jacobian-free approach).

The basic idea of physics-based preconditioning is to interpret the system $Mw = v$ in step 1 above as a system corresponding to a linear time-stepping scheme written in δ-form:

$$Mw = v \Leftrightarrow M\delta X = -G(X), \tag{82}$$

where (M, G) describes a numerical scheme that approximates the full nonlinear scheme (J, F). Another way to understand physics-based preconditioning is that Eq. (82) defines a mapping M from residuals to perturbations δX. Therefore, the Jacobian matrix is always applied to a δX to yield a residual vector F_U which is used to build Krylov spaces.

The SI schemes designed in Sect. 3 are good candidates for the scheme in Eq. (82). These schemes provide a good approximation of the solution (i.e., $M \sim J$), and most importantly they remove the numerical stiffness by solving the stiff physics (sounds waves and thermal diffusion) implicitly. Physics-based preconditioner therefore "injects" physical insight at the heart of the linear method, improving its convergence. However, the Krylov vector is a residual for variables U, and it needs to be transformed into a residual for variables V before a SI scheme can be used. Furthermore, the SI scheme provides δV, which needs to be transformed into δX before J can be applied. As in the time-stepping algorithm described in 3.6, we use the matrices derived in Sect. 3.2 to do these transformations. The complete algorithm to use the SI as a preconditioner is:

– Input: the GMRES method provides a vector $v \in \mathcal{K}_p$. v can be interpreted as a residual vector for the conservative variables U, which we denote F_U;

1. Transform F_U into a residual for the primitive variables V:
$$\tilde{F}_V = \frac{\partial V}{\partial U}F_U; \tag{83}$$
2. Apply the SI scheme to get δV:
$$\tilde{J}_V\delta V = -\tilde{F}_V; \tag{84}$$
3. Transform δV into δX:
$$\delta X = \frac{\partial X}{\partial V}\delta V; \tag{85}$$
4. Compute $J\delta X$ using the Jacobian-free method.

– Output: vector $J\delta X$, which is provided to GMRES to build successive Krylov spaces.

We note that the scheme is not fully matrix free: the SI scheme requires the resolution of a linear problem for which the matrix is explicitly formed and stored. However, thanks to the simplifications made in deriving the SI scheme, the matrix system is significantly smaller and more sparse than the Jacobian matrix. This keeps memory demand low.

In the remainder of the paper, the combination of the JFNK method with physics-based preconditioning presented in this section will be referred to as the JFNK+PBP method.

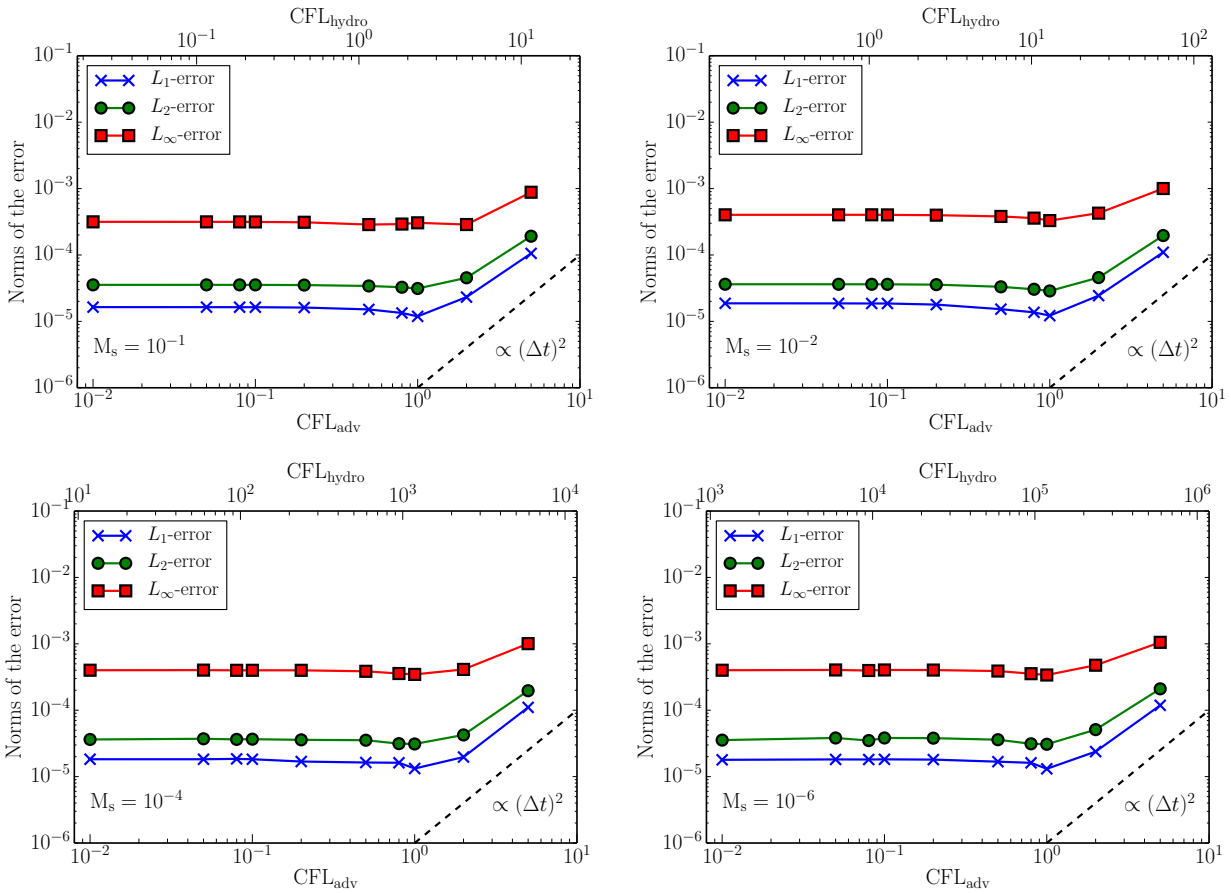

Fig. 2. Same as Fig. 1, but using the JFNK+PBP scheme to advect the isentropic vortex.

5. Results

In this section, we assess the performance of our JFNK+PBP method in both 2D and 3D. In Sect. 5.1, we test the accuracy and efficiency of the method using idealized tests that use an ideal-gas equation of state and a Cartesian geometry. In Sect. 5.2, we test the method to model stellar interiors in a spherical geometry. An important goal of this section is to demonstrate the good performance, robustness and accuracy of the solver for a wide range of Mach numbers, typically from $M_s = 10^{-1}$ down to $M_s = 10^{-6}$.

5.1. Ideal test cases

5.1.1. 2D isentropic vortex

We first investigate the accuracy of the solver by considering the 2D isentropic vortex problem that we used to test the SI scheme in Sect. 3.7. We perform the same set of runs with the JFNK+PBP method, and the computed errors are shown in Fig. 2. Comparing with the error of the semi-implicit scheme (see Fig. 1), the JFNK+PBP scheme achieves an overall reduction in the error. The range of time steps where spatial truncation errors dominate is larger, and we observe the second-order character of the temporal error at large time steps. The use of a SI scheme as a preconditioner does not impact the overall accuracy of the JFNK+PBP method. Figure 2 also shows that the results of the JFNK+PBP method are independent of the Mach number, and this desirable property of the SI scheme has been inherited by the nonlinear method.

Four free parameters enter the JFNK+PBP method: the choice of perturbation strategy (Eqs. (77) and (78)); the two scaling coefficients for the velocity components of the solution and residual vectors (parameters α_1 and α_2 in Sect. 4.1); the tolerance required for the solution of the Jacobian equation (parameter η in Eq. (75)). These free parameters were determined by testing.

We find that the accuracy of the solver for the higher end of the Mach number range being considered ($M_s > 10^{-4}$) is good regardless of the choice of perturbation strategy. However, for lower Mach numbers ($M_s \leq 10^{-4}$) we find that only Eq. (77), with $\lambda = 10^{-7}$, is able to yield accurate results. When using Eq. (78), the Jacobian operator is poorly approximated, regardless of the value of λ, resulting in a failure of the nonlinear method.

In the scaling of the linear system, we find that a value of $\alpha_1 = 10^{-5}$ gives the most consistent errors across Mach numbers for the range being considered, i.e., $10^{-6} \leq M_s \leq 10^{-1}$. However, this range can be adjusted by tuning α_1 to the problem at hand, with a higher value producing more accurate solutions for high Mach number flows. We find that $\alpha_2 = 1$ enables us to obtain accurate results for the range of Mach numbers being considered.

We find that a linear tolerance $\eta = 10^{-4}$ produces solutions with similar errors for the full range of Mach numbers considered in this work. A choice of $\eta = 10^{-2}$ produces similar results for the higher Mach numbers, but the quality of the solutions for low Mach numbers degrades seriously.

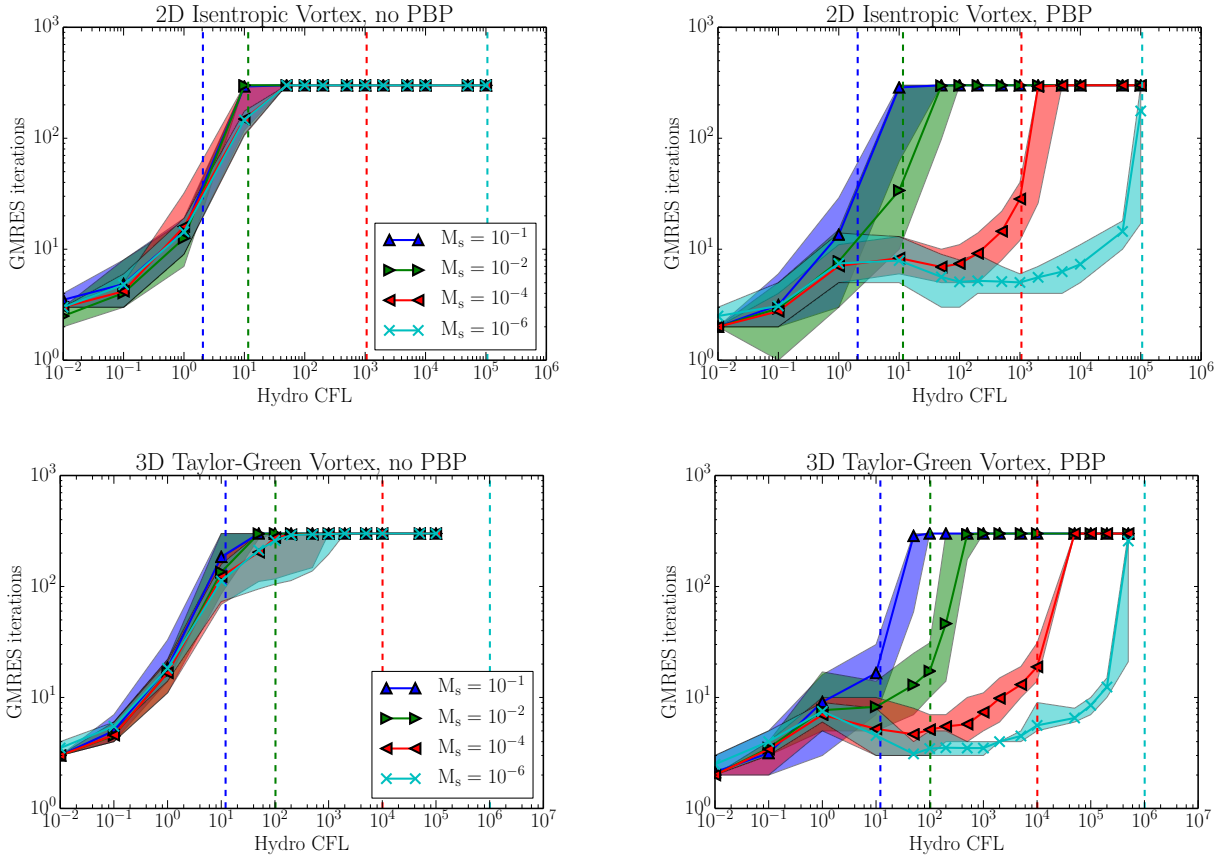

Fig. 3. Convergence of the GMRES solver without preconditioning (*left panels*) and with the physics-based preconditioner (*right panels*). The *upper panels* correspond to the 2D isentropic vortex, and the *lower panels* to the 3D Taylor-Green vortex. In both cases, the Mach numbers considered are $M_s = 10^{-1}$, 10^{-2}, 10^{-4}, 10^{-6} (*the right panels* assume the same legend as the *left ones*). The maximum allowed number of GMRES iterations was set to 300. The mean values of the number of iterations for convergence is plotted, with shaded areas showing maximum and minimum values. For each Mach number, the location of $\mathrm{CFL_{hydro}}$ corresponding to $\mathrm{CFL_{adv}} = 1$ is shown by a vertical dashed line.

Next, we assess the efficiency of the SI scheme as a preconditioner. This is done by considering the number of GMRES iterations necessary to reach convergence without preconditioning and with physics-based preconditioning, for different values of $\mathrm{CFL_{hydro}}$ and different Mach numbers (the linear tolerance is set to $\eta = 10^{-4}$). When solving the unpreconditioned linear system, we found that for $\alpha_1 = 10^{-5}$ the majority of linear problems, particularly for higher Mach numbers, fails to converge. Instead, we present for the unpreconditioned case the convergence behavior for $\alpha_1 = 1$, as a best case scenario. When solving the linear system with physics-based preconditioning, we use the optimal parameters described previously.

The results of convergence tests for the iterative method are shown in Fig. 3. For these tests, the simulations are run for 100 time steps. Without preconditioning, the different Mach number cases behave similarly: the number of GMRES iterations increases rapidly for $\mathrm{CFL_{hydro}} \gtrsim 1$, and above $\mathrm{CFL_{hydro}} \gtrsim 10$ no convergence is achieved despite the large number of iterations allowed. Such behavior is due to the stiffness of acoustic waves which increases with $\mathrm{CFL_{hydro}}$. Our physics-based preconditioner is tailored to treat this effect, and the improvement is demonstrated in the right panel of Fig. 3, as compared to the left panel without preconditioning. In each case, the increase in the number of GMRES iterations takes place at larger values of the time step. With physics-based preconditioning, the failure of the linear solver is now coming from the unstable behavior of the SI scheme for too large a $\mathrm{CFL_{adv}}$.

5.1.2. 3D Taylor-Green vortex

We consider the Taylor-Green vortex problem to test our physics-based preconditioner for an adiabatic (i.e., no thermal diffusion) flow in 3D. We consider a Cartesian domain $(x, y, z) \in [0, 2\pi L]^3$, where L is a lengthscale that sets the physical size of the domain. The initial conditions for the velocity field are

$$u_x(x, y, z) = u_0 \sin\frac{x}{L}\cos\frac{y}{L}\cos\frac{z}{L}, \tag{86}$$

$$u_y(x, y, z) = -u_0 \cos\frac{x}{L}\sin\frac{y}{L}\cos\frac{z}{L}, \tag{87}$$

$$u_z(x, y, z) = 0. \tag{88}$$

The domain has a uniform density of ρ_0. The initial pressure field is

$$p(x, y, z) = p_0 + \frac{1}{16}\rho_0 u_0^2\left(2 + \cos 2\frac{z}{L}\right)\left(\cos 2\frac{x}{L} + \cos 2\frac{y}{L}\right), \tag{89}$$

which ensures that

$$\partial_t(\boldsymbol{\nabla}\cdot\boldsymbol{u}) = 0 \text{ at } t = 0, \tag{90}$$

i.e., the initial conditions do not induce any acoustic modes. The initial amplitude of the vortex is measured in terms of the Mach

number $M_s = u_0/c_s$, where $c_s = \sqrt{\gamma p_0/\rho_0}$ is the adiabatic sound speed. The adiabatic index γ is taken as $7/5 = 1.4$. We take L as our unit of length, u_0 as our unit of velocity, and $\rho_0 L^3$ as our unit of mass. In this normalization, time is measured in units of L/u_0, and energy density in units of $\rho_0 u_0^2$. We change p_0 to vary the Mach number in the range $10^{-6} \leq M_s \leq 10^{-1}$. We consider a numerical domain with a resolution of 64^3. For this test case, we define the advective CFL number as

$$\mathrm{CFL}_{\mathrm{adv}} = \max \frac{|\boldsymbol{u}|\Delta t}{\Delta x}, \tag{91}$$

where $\boldsymbol{u}$ is the velocity, Δt the time step, Δx the mesh spacing.

Similarly to the 2D isentropic vortex, we first investigate the efficiency of the physics-based preconditioner in reducing the number of iterations required by the linear solver to converge to the desired accuracy. As condition (90) is never exactly fulfilled in the discretized problem, some acoustic fluctuations are produced at the first time step. To remove these transients, we evolve each case for 100 time steps at a fixed $\mathrm{CFL}_{\mathrm{hydro}} = 1$. We then compute another 100 time steps with different values of Δt, corresponding to different values of $\mathrm{CFL}_{\mathrm{hydro}}$. We monitor the number of iterations required for convergence, with and without physics-based preconditioning[4]. The results are shown in Fig. 3. The conclusions are very similar to the ones drawn from the 2D vortex test presented in the previous section: physics-based preconditioning allows for a fast convergence over a broad range of hydrodynamical CFL numbers. Here again, the convergence of the linear solver becomes difficult when $\mathrm{CFL}_{\mathrm{adv}} = 1$ is approached.

Next we use the Taylor Green vortex to benchmark the implicit JFNK+PBP method against the second-order accurate Adams-Bashforth explicit scheme. Starting at $t = 0$, we evolve the vortex using both the JFNK+PBP method and the Adams-Bashforth method. We run the tests for a fixed wall-clock time of six hours, and record the final time achieved by each method, varying the Mach number of the test case. In the explicit case, the time step is limited by stability to $\mathrm{CFL}_{\mathrm{hydro}} = 0.1$; for the implicit case, the time step is limited to $\mathrm{CFL}_{\mathrm{adv}} = 0.5$ for accuracy and for the stability of the underlying SI. The results are recorded in Table 1. The final times obtained with the explicit solver scale approximately with the Mach number, due to the scaling of the CFL time step with the background sound speed. The final times obtained with the JFNK+PBP method show less of a clear pattern, with performance peaking at a Mach number of $M_s = 10^{-4}$. Nevertheless, the JFNK+PBP method is already more than five times faster than the explicit solver for $M_s = 10^{-2}$. For lower Mach numbers, the speed-up is larger than two orders of magnitude. We observe a dramatic drop in performance at $M_s = 10^{-6}$ when using the criterion $\mathrm{CFL}_{\mathrm{adv}} = 0.5$ on the time step. From analyzing the performance of the scheme for this run (see Table 1), it appears that the physics-based preconditioner becomes less effective, resulting in a very large number of GMRES iterations and a substantial loss of performance. Such a loss of effectiveness of the preconditioner at a very low-Mach number close to $\mathrm{CFL}_{\mathrm{adv}} \sim 1$ can be already seen on Fig. 3, and seems to highlight the limit of what is currently feasible with the solver. We repeated the timing test for this Mach number with $\mathrm{CFL}_{\mathrm{Hydro}} = 5 \times 10^4$, which corresponds to $\mathrm{CFL}_{\mathrm{adv}} \sim 0.05$. This improves the final time by approximately an order of magnitude. It remains the least efficient case, but it is still roughly three orders of magnitude faster than the corresponding explicit calculation.

[4] The parameters of the solver $(\eta, \alpha_1, \alpha_2, \ldots)$ are adjusted as discussed for the 2D test case.

Finally, we monitor the decay of the Taylor-Green vortex for the range of Mach numbers explored here. We simulate for a fixed time of $t = 20$, a time at which most of the dissipation has occurred. We show in Fig. 4 the evolution of the decay rate of kinetic energy. The left panel shows a global view where the different curves are indistinguishable from each other. The right panel shows a zoom on the peak of the decay rate. The difference between the curves represents less than a percent. In Table 2 we record the maximum decay rate and the time at which it occurs. The purpose of Fig. 4 is twofold: firstly, it complements the performance results presented previously as it shows that the results are independent, at the percent level, of the Mach numbers; secondly, it provides confidence in using the code as an ILES tool to model turbulent flows over a wide range of Mach numbers. In the ILES framework, dissipation of kinetic energy is due to the truncation errors of the scheme, and it is not obvious that these behave similarly for different Mach numbers.

5.2. Stellar test cases

In this section, we examine how the JFNK+PBP method performs in realistic stellar models. We use the same models as in Viallet et al. (2013) of a 2D young Sun and a red giant in which convection is fully developed and has reached a quasi-steady state. Both models are first considered in a 2D spherically axisymmetric geometry. The red giant model is then considered in a full 3D spherical wedge geometry.

5.2.1. 2D stellar models

We compute 100 time steps of the red giant and young Sun models using the JFNK+PBP method. We limit $\mathrm{CFL}_{\mathrm{adv}}$ (defined as in Eq. (91)) to values of 0.5, 1, and 1.5. We compare the performance of the JFNK+PBP method with the best method identified in Viallet et al. (2013). The results are summarized in Table 3 for the red giant, and Table 4 for the young Sun. They show that the JFNK+PBP method is less efficient than the Broyden+ILU method. It is seen that the JFNK+PBP method is becoming less and less effective when $\mathrm{CFL}_{\mathrm{adv}}$ increases, as the physics-based preconditioner fails as the underlying SI becomes unstable. In practice, the JFNK+PBP method should not be used with $\mathrm{CFL}_{\mathrm{adv}}$ larger than one when computing an unsteady flow. This limitation is not very penalizing, as numerical accuracy is expected to decrease when $\mathrm{CFL}_{\mathrm{adv}} > 1$, meaning that larger time steps are not desirable anyway[5]. For both the red giant and young Sun models, the performance of the JFNK+PBP solver is the same for $\mathrm{CFL}_{\mathrm{adv}} = 0.5$ and $\mathrm{CFL}_{\mathrm{adv}} = 1$, as the increase in the time step is compensated by the increase in the number of GMRES iterations per Newton iteration. One should keep in mind that the performance of the JFNK+PBP solver presented here could probably be improved by fine tuning the parameters discussed in Sect. 5.1.1. For instance, the red giant and young Sun models differ in the average Mach number, with the red giant having a larger Mach number (~0.1) than the young Sun (~0.01). Although the performance of the JFNK+PBP method could be made closer to the Broyden method, we could expect the latter to remain the most efficient option for these cases.

[5] Based on the 2D vortex advection test, Viallet et al. (2013) showed that one could use $\mathrm{CFL}_{\mathrm{adv}} \sim 2$ without degrading the accuracy too much. However, this conclusion might not be adequate for an unsteady, turbulent flow, which could require smaller time step.

Table 1. Summary of the results for the Taylor-Green vortex tests.

Mach No.	Implicit						Explicit
	CFL_{hydro}	CFL_{adv}	$\frac{Newton}{\Delta t}$	$\frac{GMRES}{Newton}$	$\frac{Parabolic}{GMRES}$	Final time	Final time
10^{-1}	7.1e+00	0.5	3.8	16.6	2.4	17.8	34.8
10^{-2}	7.0e+01	0.5	3.1	15.4	2.5	20.9	3.62
10^{-3}	8.2e+02	0.5	2.7	16.2	2.5	31.6	0.300
10^{-4}	1.1e+04	0.5	2.0	16.1	2.8	55.1	3.65(−2)
10^{-5}	8.1e+04	0.5	2.0	22.2	2.9	30.9	3.64(−3)
10^{-6}	4.9e+05	0.49	4.5	291.5	3.0	0.6	3.64(−4)
10^{-6}	5.0e+04	0.05	2.0	7.0	2.8	4.5	3.64(−4)

Notes. The columns for the implicit case represent: the average hydrodynamical and advective CFL numbers, the average number of Newton iterations per time step, the average number of GMRES iterations per Newton iteration, the average number of parabolic iterations for the preconditioner per GMRES iteration.

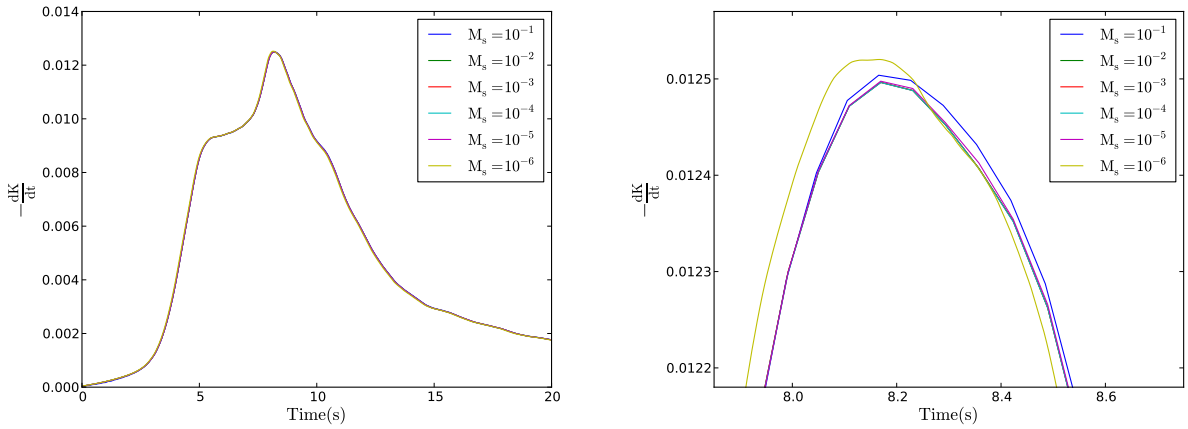

Fig. 4. Decay rate of the Taylor Green vortex for different Mach numbers. The *right panel* shows a zoom on the peak of the decay rate. Time is measured in units of L/u_0, the decay rate in units of $\rho_0 u_0^3/L$.

Table 2. Maximum decay rate measured during the decay of the Taylor-Green vortex for different Mach numbers.

Mach No.	Time of maximum	Value
10^{-1}	8.1656	1.2504(–2)
10^{-2}	8.1695	1.2496(–2)
10^{-3}	8.1695	1.2496(–2)
10^{-4}	8.1696	1.2496(–2)
10^{-5}	8.1695	1.2498(–2)
10^{-6}	8.1681	1.2520(–2)

Notes. Time is measured in units of L/u_0, the decay rate in units of $\rho_0 u_0^3/L$.

5.2.2. 3D red giant models

The efficient computation of 3D models is the main motivation for moving beyond the framework of quasi-Newton methods. We cannot, however, meaningfully compare the performance of the later method to that of the JFNK+PBP method for 3D stellar models, as done in the previous section. As shown previously, quasi-Newton methods, such as the Broyden method, perform well in 2D. However, their cost increases significantly in 3D. The reasons are twofold. Firstly, in 3D, the Jacobian matrix has a more complex structure than in 2D, due to the third dimension. This implies an increase in the cost for the construction and storage of the Jacobian matrix and its ILU factorization. As a result, for the same number of degrees of freedom (i.e., same matrix size), a 3D computation is inherently more expensive than a 2D computation. Secondly, in 3D, the typical size of a problem is much larger than in 2D, essentially due to the larger number of cells, but also due to the extra variable (the third velocity component). For instance, a 128^2 computation has $4 \times 128^2 = 65\,536$ degrees of freedom, whereas a 128^3 computation has $5 \times 128^3 = 10\,485\,760$ degrees of freedom. The computational costs (cpu time+memory) for some of the components of the quasi-Newton methods do not scale linearly with the problem size. Thus, this increase in degrees of freedom corresponds to a prohibitive increase in both cpu time and memory. For these reasons, we can only perform a comparison with an extremely low resolution, not necessarily relevant to the analysis of physical processes in stars. Since the JFNK+PBP is now the method implemented in MUSIC for the purpose of running 3D simulations, we want to illustrate the potential of this method.

We do so by performing computations of the red giant model for a grid size of 72×65^2 (roughly 1.5 million degrees of freedom), using both the Broyden and JFNK+PBP methods. For the reasons presented previously, this is the largest problem size that we could consider using the serial version of MUSIC on a single node of the supercomputer Zen at the University of Exeter. Each node has 12 cores and 24 Gb of RAM, and a full node was requested to benefit from the available memory. It is clear that the memory requirement of the ILU factorization restricts the range of accessible resolutions, even if domain decomposition is used to distribute the problem among several computer nodes.

Table 3. Comparison of the performance of the JFNK+PBP method presented in this paper with the Broyden methods presented in Viallet et al. (2013), for the 2D red giant test case.

Method	CFL_{hydro}	CFL_{adv}	CFL_{rad}	$\frac{\text{Newton}}{\Delta t}$	$\frac{\text{GMRES}}{\text{Newton}}$	$\frac{\text{Simulated time}}{\text{Wall time}}$
$CFL_{adv,max}$ = 0.5						
Broyden(10^{-2})+ILU(1)	17.7	0.46	4.8	7.3	4.0	227
JFNK(10^{-1})+ PBP	18.9	0.49	4.9	6.2	7.2	202
JFNK(10^{-2})+ PBP	18.9	0.49	4.9	4.6	18.4	161
JFNK(10^{-4})+PBP	18.9	0.49	4.9	4.6	37.8	80
$CFL_{adv,max}$ = 1						
Broyden(10^{-1})+ILU(1)	37.9	0.93	8.2	9.5	4.3	382
JFNK(10^{-1})+PBP	40.2	0.98	8.6	6.6	18.5	200
JFNK(10^{-2})+PBP	40.0	0.98	8.5	5.7	41.3	124
JFNK(10^{-4})+PBP	40.0	0.98	8.5	5.7	200.7	32
$CFL_{adv,max}$ = 1.5						
Broyden(10^{-1})+ILU(1)	45.2	1.10	9.6	13.2	4.9	383
JFNK(10^{-1})+PBP	55.2	1.42	11.0	8.9	37.8	121
JFNK(10^{-2})+PBP	55.8	1.43	11.0	7.8	79.3	73
JFNK(10^{-4})+PBP	55.7	1.43	11.0	7.7	252.0	33

Notes. The value of the linear tolerance η is given in parenthesis after the name of the method; ILU(k) refers to an incomplete LU factorization of order k; PBP refers to physics-based preconditioning for sound waves only.

Table 4. Similar to Table 3, for the 2D young Sun models.

Method	CFL_{hydro}	CFL_{adv}	CFL_{rad}	$\frac{\text{Newton}}{\Delta t}$	$\frac{\text{GMRES}}{\text{Newton}}$	$\frac{\text{Simulated time}}{\text{Wall time}}$
$CFL_{adv,max}$ = 0.5						
Broyden(10^{-1})+ILU(2)	235.5	0.50	2.5(−7)	6.6	12.3	74
JFNK(10^{-1})+PBP	235.0	0.50	2.5(−7)	6.7	10.4	39
JFNK(10^{-2})+PBP	227.4	0.50	2.4(−7)	5.2	15.8	35
JFNK(10^{-4})+PBP	235.0	0.50	2.5(−7)	5.2	19.9	28
$CFL_{adv,max}$ = 1						
Broyden(10^{-1})+ILU(2)	474.8	1.00	5.1(−7)	8.2	16.7	106
JFNK(10^{-1})+PBP	474.9	1.00	5.1(−7)	7.5	19.5	39
JFNK(10^{-2})+PBP	471.8	0.99	5.1(−7)	8.1	20.0	35
JFNK(10^{-4})+PBP	471.8	0.99	5.1(−7)	8.0	20.0	34
$CFL_{adv,max}$ = 1.5						
Broyden(10^{-2})+ILU(2)	681.3	1.50	7.3(−7)	9.9	19.8	121
JFNK(10^{-1})+PBP	662.8	1.45	7.1(−7)	15.6	19.9	26
JFNK(10^{-2})+PBP	656.9	1.44	7.1(−7)	15.9	20.0	25
JFNK(10^{-4})+PBP	668.0	1.46	7.2(−7)	16.0	20.0	25

Table 5. Similar to Table 3, for the 3D red giant models.

Method	CFL_{hydro}	CFL_{adv}	CFL_{rad}	$\frac{\text{Newton}}{\Delta t}$	$\frac{\text{GMRES}}{\text{Newton}}$	$\frac{\text{Simulated time}}{\text{Wall time}}$
$CFL_{adv,max}$ = 0.5						
JFNK(10^{-1})+PBP	107.6	0.50	0.74	10.8	7.8	188
JFNK(10^{-2})+PBP	107.6	0.50	0.74	7.3	13.3	183
JFNK(10^{-4})+PBP	107.6	0.50	0.74	5.1	25.6	147
Broyden(10^{-1}) w/o preconditioner	107.6	0.50	0.74	7.6	219.7	122
Broyden(10^{-1})+ILU(2)	107.6	0.50	0.74	6.8	8.7	109
$CFL_{adv,max}$ = 1						
Broyden(10^{-1})+ILU(2)	234.5	1.00	1.4	8.7	14.0	203
Broyden(10^{-1}) w/o preconditioner	234.6	1.00	1.4	10.4	268.1	180
JFNK(10^{-1})+PBP	227.4	0.97	1.4	21.5	15.7	109
$CFL_{adv,max}$ = 1.5						
Broyden(10^{-1})+ILU(2)	342.5	1.50	1.9	10.9	17.1	257
Broyden(10^{-1}) w/o preconditioner	342.5	1.50	1.9	12.7	286.0	205
JFNK(10^{-1})+PBP	227.0	0.97	1.4	25.0	22.5	67

Test runs are performed similarly to the previous section, and the results are summarized in Table 5. The JFNK+PBP method is more efficient for $\mathrm{CFL}_{\mathrm{adv}} = 0.5$, but the Broyden method remains more efficient for $\mathrm{CFL}_{\mathrm{adv}} = 1$ and $\mathrm{CFL}_{\mathrm{adv}} = 1.5$. Surprisingly enough, for this particular case the unpreconditioned Broyden method performs almost as well as the preconditioned version, showing that the ILU preconditioner becomes inefficient due to its cost. However, we stress that an unpreconditioned Broyden method is not a viable option for scientific applications. The preconditioned version will not be viable for larger problems, due to the increasing cost for computing and storing the ILU factorization. We expect a more substantial gain compared to the quasi-Newton methods when larger problems will be considered, but the serial tests performed here limit us to relatively small 3D problems. Such small problems appear already to be on the edge of the capabilities of quasi-Newton methods. The implementation of the JFNK+PBP method in MUSIC now allows us to perform 3D simulations with resolutions of 512^3 of a large fraction (~80% in radius) of a partly convective star, as an initial step toward the study of turbulent convection and overshooting under realistic stellar interior conditions and over relevant physical time scales (Pratt et al., in prep.)

Finally, as stellar models include radiative diffusion, we have the possibility of using the second version of the physics-based preconditioner, in which thermal diffusion is also treated implicitly. The stiffness of thermal diffusion is measured by the radiative CFL number, defined as:

$$\mathrm{CFL}_{\mathrm{rad}} = \max \frac{\chi \Delta t}{\Delta x^2}, \tag{92}$$

where Δt is the time step, Δx the mesh spacing, χ the thermal diffusivity. As for sound waves, solving thermal diffusion with a time explicit method requires $\mathrm{CFL}_{\mathrm{rad}} \lesssim 1$ for stability. Implicit methods allow for $\mathrm{CFL}_{\mathrm{rad}} \gg 1$, but preconditioning is necessary to improve the convergence of the iterative method. For the red giant model, however, our particular treatment of the surface implies that the radiative diffusion is not very stiff ($\mathrm{CFL}_{\mathrm{rad}} \sim 1$), and as such we do not see a substantial difference between the two versions of the physics-based preconditioner. Concrete examples of stellar cases where preconditioning of thermal diffusion is necessary will be presented elsewhere.

6. Conclusion

This work is a continuation of previous efforts devoted to developing an efficient, accurate fully implicit solver for multidimensional hydrodynamics. In Sect. 4 we presented a Jacobian-free Newton-Krylov method, which avoids the explicit construction of the Jacobian matrix. The use of iterative methods to solve the Jacobian equation requires preconditioning at large hydrodynamical CFL numbers. The main purpose of this paper was to present an efficient preconditioner that specifically targets the physical processes that are responsible for numerical stiffness, hence the name of "physics-based" preconditioners.

This strategy is very different from the more usual algebraic preconditioners (as, e.g., ILU factorization) which try to address the stiffness of the system by only looking at the Jacobian matrix structure and numerical values, without considering the underlying equations. In the context of stellar hydrodynamics, stiffness results from acoustic perturbations that propagate on a time scale much shorter than the fluid bulk motion and possibly from thermal diffusion. Therefore, the preconditioning step relies on a semi-implicit solver, which is inexpensive and rather inaccurate[6], that treats sounds waves (and thermal diffusion, if required) implicitly in order to overcome the associated CFL limit on the time step (see Sect. 3). Although we aim at using MUSIC to model stellar interiors, the JFNK+PBP method can be applied to general advection and/or diffusion-dominated problems. Although many approximations enter the derivation of our SI scheme, they do not restrict its range of applicability.

Section 5 presented the results of extensive tests assessing the performance and accuracy of the new method. A strong emphasis was put on exploring a wide range of Mach numbers, namely six orders of magnitude from $M_{\mathrm{s}} = 10^{-1}$ down to $M_{\mathrm{s}} = 10^{-6}$. The tests assessed the ability of the physics-based preconditioner to reduce the number of linear iterations required by the linear solver. Using the 3D Taylor-Green vortex test, we showed that this solver is computationally efficient, beating the Adams-Bashforth explicit scheme for $M_{\mathrm{s}} \lesssim 10^{-2}$. We emphasize that the method has several parameters that can be tuned to improve its performance. To achieve the best performance, these parameters should be tuned for the particular problem being considered. Therefore, we do not claim to have found the best set of parameters, but rather a set that gives very satisfying performances for the various tests performed in this paper. Furthermore, the performance does not come with any loss of accuracy: our method exhibits accuracy that is consistent with the second-order nature of its discretization, and most importantly, the numerical errors are independent of the Mach number, at least in the investigated range. However, it appears that $M_{\mathrm{s}} \sim 10^{-6}$ is on the edge of the abilities of the solver, as fine-tuning of some parameters and a reduction of the time step was necessary to pass the tests at such a low Mach number.

The JFNK+PBP method is now the work-horse of the MUSIC code, and is used to investigate long-standing problems in stellar hydrodynamics such as shear mixing and convective overshooting. Further developments are now devoted to the parallelization of the method, in order to take advantage of multi-cores/multi-nodes high-performance computers that are now routinely used in computational physics. Performance and scalability tests will be presented elsewhere.

Acknowledgements. M.V. would like to thank Dana Knoll and Ryosuke Park for useful discussions on physics-based preconditioning during several visits at LANL that motivated this work. M.V. also thanks Hannes Grimm-Strele and Philipp Edelmann for discussions related to the work presented in this paper. M.V. acknowledges support from a Newton International Fellowship and Alumni program from the Royal Society during earlier part of this work. Part of this work was funded by the Royal Society Wolfson Merit award WM090065, the Consolidated STFC grant ST/J001627/1STFC, the French "Programme National de Physique Stellaire" (PNPS) and "Programme National Hautes Énergies" (PNHE), and by the European Research Council through grants ERC-AdG No. 320478-TOFU and ERC-AdG No. 341157-COCO2CASA. The calculations for this paper were performed on the DiRAC Complexity machine, jointly funded by STFC and the Large Facilities Capital Fund of BIS, and the University of Exeter Supercomputer, a DiRAC Facility jointly funded by STFC, the Large Facilities Capital Fund of BIS, and the University of Exeter.

References

Bazán, G., Dearborn, D. S. P., Dossa, D. D., et al. 2003, in 3D Stellar Evolution, eds. S. Turcotte, S. C. Keller, & R. M. Cavallo, ASP Conf. Ser., 293, 1

[6] We applied Picard linearization to derive the scheme and used first-order time discretization methods.

Gee, M., Siefert, C., Hu, J., Tuminaro, R., & Sala, M. 2006, ML 5.0 Smoothed Aggregation User's Guide, Tech. Rep. Sandia National Laboratories
Glatzmaier, G. 2013, Introduction to Modeling Convection in Planets and Stars: Magnetic Field, Density Stratification, Rotation (Princeton University Press)
Heroux, M. A., Bartlett, R. A., Howle, V. E., et al. 2005, ACM Trans. Math. Softw., 31, 397
Herwig, F., Woodward, P. R., Lin, P.-H., Knox, M., & Fryer, C. 2014, ApJ, 792, L3
Kifonidis, K., & Müller, E. 2012, A&A, 544, A47
Knoll, D. A., & Keyes, D. E. 2004, J. Comput. Phys., 193, 357
LeVeque, R. J. 2007, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (SIAM)
Meakin, C. A., & Arnett, D. 2007, ApJ, 667, 448
Mocák, M., Siess, L., & Müller, E. 2011, A&A, 533, A53
Mousseau, V., Knoll, D., & Rider, W. 2000, J. Comput. Phys., 160, 743
Park, H., Nourgaliev, R. R., Martineau, R. C., & Knoll, D. A. 2009, J. Comp. Phys., 228, 9131
Reisner, J. M., Mousseau, A., Wyszogrodzki, A. A., & Knoll, D. A. 2005, Monthly Weather Rev., 133, 1003
Saad, Y., & Schultz, M. H. 1986, SIAM J. Sci. Stat. Comput., 7, 856
van Leer, B. 1974, J. Comput. Phys., 14, 361
Viallet, M., Baraffe, I., & Walder, R. 2011, A&A, 531, A86
Viallet, M., Baraffe, I., & Walder, R. 2013, A&A, 555, A81
Yee, H., Vinokur, M., & Djomehri, M. 2000, J. Comp. Phys., 162, 33

Appendix A: Evolution equation for pressure and acoustic fluctuations

A.1. Evolution equation for pressure

The linearized equation-of-state yields

$$\delta p = \left.\frac{\partial p}{\partial \rho}\right|_e \delta\rho + \left.\frac{\partial p}{\partial e}\right|_\rho \delta e. \tag{A.1}$$

We have the thermodynamic relationships

$$\left.\frac{\partial p}{\partial \rho}\right|_e = \frac{p}{\rho}(\Gamma_1 - \Gamma_3 + 1), \tag{A.2}$$

$$\left.\frac{\partial p}{\partial e}\right|_\rho = \rho(\Gamma_3 - 1), \tag{A.3}$$

where Γ_1 and Γ_3 are the first and third generalised adiabatic indices. We now substitute δ's with Lagrangian derivatives $D_t = \partial_t + \boldsymbol{u}\cdot\boldsymbol{\nabla}$ in Eq. (A.1):

$$D_t p = \frac{p}{\rho}(\Gamma_1 - \Gamma_3 + 1) D_t\rho + \rho(\Gamma_3 - 1) D_t e, \tag{A.4}$$

and we use the Lagrangian equations

$$D_t\rho = -\rho\boldsymbol{\nabla}\cdot\boldsymbol{u}, \tag{A.5}$$

$$\rho D_t e = -p\boldsymbol{\nabla}\cdot\boldsymbol{u} + \boldsymbol{\nabla}\cdot(\chi\boldsymbol{\nabla}T), \tag{A.6}$$

to obtain

$$\partial_t p + \boldsymbol{u}\cdot\boldsymbol{\nabla}p = -\Gamma_1 p\boldsymbol{\nabla}\cdot\boldsymbol{u} + (\Gamma_3 - 1)\boldsymbol{\nabla}\cdot(\chi\boldsymbol{\nabla}T). \tag{A.7}$$

Appendix B: Transformation matrices

B.1. Internal energy equation

In this case $U = (\rho, \rho e, \rho u)$, $X = (\rho, e, u)$, and $V = (p, e, u)$. The transformation matrix $\partial U/\partial X$ between variables U and X is such that $\delta U = (\partial U/\partial X)\delta X$. We have

$$\frac{\partial U}{\partial X} = \begin{pmatrix} 1 & 0 & 0 \\ e & \rho & 0 \\ u & 0 & \rho \end{pmatrix}. \tag{B.1}$$

The inverse transformation is:

$$\frac{\partial X}{\partial U} = \left(\frac{\partial U}{\partial X}\right)^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -e/\rho & 1/\rho & 0 \\ -u/\rho & 0 & 1/\rho \end{pmatrix}. \tag{B.2}$$

The transformation matrix $\partial V/\partial X$ is

$$\frac{\partial V}{\partial X} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e & \left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{B.3}$$

and its inverse $\partial X/\partial V$ is

$$\frac{\partial X}{\partial V} = \begin{pmatrix} \left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1} & -\left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1}\left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{B.4}$$

We have

$$\frac{\partial V}{\partial U} = \frac{\partial V}{\partial X}\times\frac{\partial X}{\partial U} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e & \left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ -e/\rho & 1/\rho & 0 \\ -u/\rho & 0 & 1/\rho \end{pmatrix} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e - \frac{e}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & \frac{1}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ -e/\rho & 1/\rho & 0 \\ -u/\rho & 0 & 1/\rho \end{pmatrix}. \tag{B.5}$$

B.2. Total energy equation

In this case $U = (\rho, \rho\epsilon_t, \rho u)$, $X = (\rho, \epsilon_t, u)$, and $V = (p, e, u)$. The transformation matrix $\partial U/\partial X$ is:

$$\frac{\partial U}{\partial X} = \begin{pmatrix} 1 & 0 & 0 \\ \epsilon_t & \rho & 0 \\ u & 0 & \rho \end{pmatrix}. \tag{B.6}$$

Its inverse is

$$\frac{\partial X}{\partial U} = \begin{pmatrix} 1 & 0 & 0 \\ -\epsilon_t/\rho & 1/\rho & 0 \\ -u/\rho & 0 & 1/\rho \end{pmatrix}. \tag{B.7}$$

The transformation matrix from (ρ, e, u) and (ρ, ϵ_t, u) is

$$\begin{pmatrix} \delta\rho \\ \delta\epsilon_t \\ \delta u \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & u \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \delta\rho \\ \delta e \\ \delta u \end{pmatrix}, \tag{B.8}$$

so that the transformation matrix $\partial V/\partial X$ is

$$\frac{\partial V}{\partial X} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e & \left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & u \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e & \left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -u \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e & \left.\frac{\partial p}{\partial e}\right|_\rho & -u\left(\left.\frac{\partial p}{\partial e}\right|_\rho\right) \\ 0 & 1 & -u \\ 0 & 0 & 1 \end{pmatrix}. \tag{B.9}$$

The inverse transformation is

$$\frac{\partial X}{\partial V} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & u \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e & \left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & u \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1} & -\left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1}\left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1} & -\left(\left.\frac{\partial p}{\partial \rho}\right|_e\right)^{-1}\left.\frac{\partial p}{\partial e}\right|_\rho & 0 \\ 0 & 1 & u \\ 0 & 0 & 1 \end{pmatrix}. \tag{B.10}$$

Finally, we have

$$\begin{aligned}\frac{\partial V}{\partial U} =& \frac{\partial V}{\partial X} \times \frac{\partial X}{\partial U} = \\ & \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e & \left.\frac{\partial p}{\partial e}\right|_\rho & -u\left(\left.\frac{\partial p}{\partial e}\right|_\rho\right) \\ 0 & 1 & -u \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\epsilon_t/\rho & 1/\rho & 0 \\ -u/\rho & 0 & 1/\rho \end{pmatrix} \\ =& \begin{pmatrix} \left.\frac{\partial p}{\partial \rho}\right|_e - \frac{\epsilon_t - u^2}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & \frac{1}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho & -\frac{u}{\rho}\left.\frac{\partial p}{\partial e}\right|_\rho \\ -(\epsilon_t - u^2)/\rho & 1/\rho & -u/\rho \\ -u/\rho & 0 & 1/\rho \end{pmatrix}. \end{aligned} \tag{B.11}$$