Submitted to J. Computational and Applied Mathematics

A Fast and Accurate Algorithm for Computing Radial Transonic Flows

Hans De Sterck[1], Scott Rostrup[1] and Feng Tian[2]

[1] Department of Applied Mathematics, University of Waterloo, Ontario, Canada
[2] NASA Postdoctoral Program and High Altitude Observatory, National Center for Atmospheric Research, Boulder, Colorado, USA

Abstract

An efficient algorithm is described for calculating stationary one-dimensional transonic outflow solutions of the compressible Euler equations with gravity and heat source terms. The stationary equations are solved directly by exploiting their dynamical system form. Transonic expansions are the stable manifolds of saddle-point-type critical points, and can be obtained efficiently and accurately by adaptive integration outward from the critical points. The particular transonic solution and critical point that match the inflow boundary conditions are obtained by a two-by-two Newton iteration which allows the critical point to vary within the manifold of possible critical points. The proposed Newton Critical Point (NCP) method typically converges in a small number of Newton steps, and the adaptively calculated solution trajectories are highly accurate. A sample application area for this method is the calculation of transonic hydrodynamic escape flows from extrasolar planets and the early Earth. The method is also illustrated for an example flow problem that models accretion onto a black hole with a shock.

Key words: compressible gas dynamics, Euler equations, transonic flow, dynamical systems, critical points, Newton method

1 Introduction

In this paper we introduce a fast and accurate algorithm for calculating stationary one-dimensional (1D) transonic outflow solutions of the compressible Euler equations with gravity and heat source terms. The Euler equations for a compressible perfect gas flow with radial velocity and spherical symmetry are given by

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\, r^2\\ \rho\, u\, r^2\\ (\frac{p}{\gamma-1}+\frac{\rho u^2}{2})\, r^2\end{bmatrix}+\frac{\partial}{\partial r}\begin{bmatrix}\rho\, u\, r^2\\ \rho\, u^2\, r^2+p\, r^2\\ (\frac{\gamma p}{\gamma-1}+\frac{\rho u^2}{2})\, u\, r^2\end{bmatrix}=\begin{bmatrix}0\\ f_{ext}+2\, p\, r\\ f_{ext}\, u+q_{heat}\, r^2\end{bmatrix}. \tag{1}$$

Here, ρ is the fluid mass density, p is the pressure, u is the radial velocity, r is the radial coordinate, t is time, and γ is the adiabatic constant. Throughout this paper the value $\gamma = 7/5$ for di-atomic gases is used, except where noted. The right hand side of the equation contains source terms with radial external force f_{ext} and heat source q_{heat}. See the nomenclature below for an overview of the main variables and parameters used in this paper.

Nomenclature

dimensionless flow variables		*dynamical system variables*	
ρ	mass density	s	parameter along trajectory
p, e	pressure, energy	$\mathbf{V}(s)$	state vector
u, u_r, u_θ	velocity components	$\mathbf{G}(\mathbf{V})$	right hand side function
c	sound speed	$\vec{x}_i$	eigenvector
T	temperature	λ_i	eigenvalue
S	entropy	d	integration constant
r	radial coordinate		
t	time	*NCP algorithm description*	
F	mass flux	$\mathbf{B}$	values at boundary
G	gravitational constant	$\mathbf{C}$	values at critical point
M	planet mass	$\mathbf{F}(\mathbf{C})$	nonlinear mapping
f_{ext}, g	external force, gravity field	k	Newton iteration number
q_{heat}	heat source	$\mathbf{B}^{(k)}$	$\mathbf{B}$ in iteration k
γ	adiabatic constant	$B_i^{(k)}$	ith component of $\mathbf{B}^{(k)}$
r_a, r_b	left and right domain boundary	$\mathbf{C}^{(k)}$	$\mathbf{C}$ in iteration k
$\mathcal{M}$	(normal) Mach number	$J\|_{\mathbf{C}^{(\mathbf{k})}}$	Jacobian of $\mathbf{F}(\mathbf{C})$ at $\mathbf{C}^{(k)}$
μ	$u_\theta\, r$	J_{ij}	i, j matrix element of J
subscripts and superscripts		$\mathbf{B}^j$	variation of $\mathbf{B}^{(k)}$ in direction j
crit	value at critical point	B_i^j	ith component of $\mathbf{B}^j$
$*$	boundary condition value	ϵ_{Newton}	Newton convergence tolerance
0	scaling factor	ϵ_{ODE}	ODE integration error tolerance
various		δ_{Crit}	distance from critical point
I	identity matrix	δ_{Jac}	δ for Jacobian calculation

As illustrated in Fig. 1, Eq. (1) allows for stationary transonic outflow solutions, where the radial flow velocity is initially subsonic, but then passes through a critical point where $u = c$ (with c the sound speed of the gas), and subsequently takes on supersonic values beyond the critical point. (Note that this flow solution was obtained using the numerical method that is the

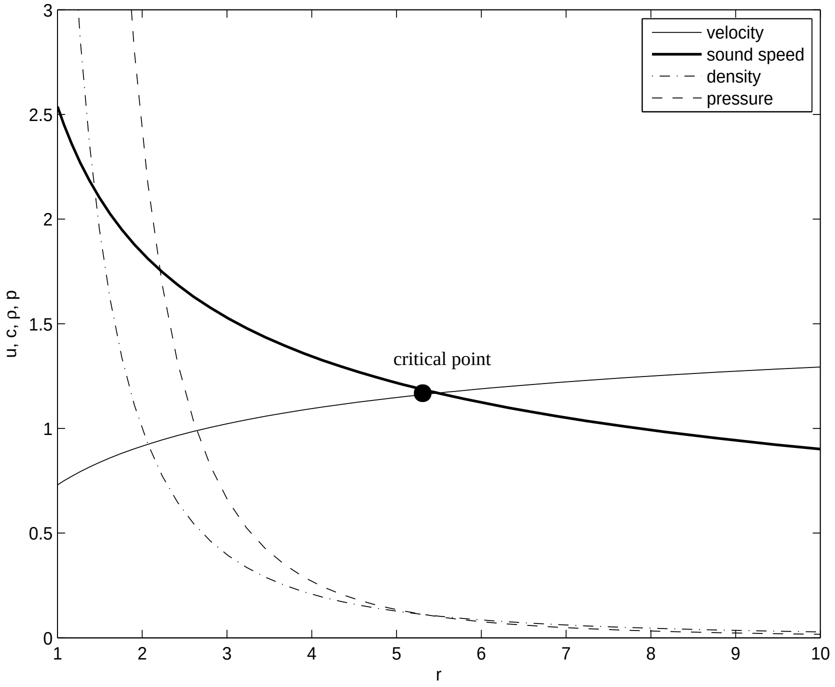

Fig. 1. Radial transonic flow solution for $f_{ext} = -\rho GM$, $GM = 15$ and $q_{heat} = 0$, and with boundary conditions $\rho = 5$ and $p = 23$ at the inflow boundary $r_a = 1$.

subject of this paper, as described in detail in Section 3.) This type of flows has applications in models of hydrodynamic escape from extrasolar planets, Venus and the early Earth [1–4], and in the context of solar and stellar wind modeling [5–7]. Eq. (1) may, for instance, model transonic outflow of hydrogen gas from close-in extrasolar planets [1]. In this case, $f_{ext} = -\rho GM$, where G is the gravitational constant and M is the planet mass. The flow solution in Fig. 1 was obtained for $GM = 15$, without heating ($q_{heat} = 0$), and with boundary conditions $\rho = 5$ and $p = 23$ at the inflow boundary $r_a = 1$. In our planetary atmosphere calculations, one of the major quantities of interest to be obtained by numerical simulation is the radial outflow flux $F = \rho\, u\, r^2$, and we want to determine how this radial mass flux varies as a function of the heating profile and the density and pressure imposed at the inflow boundary.

Note that all equations and numerical solutions in this paper are expressed in terms of dimensionless variables. The dimensionless variables and equations are derived from their dimensional counterparts in a standard way. Denoting dimensional variables with a bar, and scaling factors with a 0 subscript, we can write, for the mass density for example,

$$\rho = \bar{\rho}/\rho_0, \tag{2}$$

with ρ the dimensionless density, $\bar{\rho}$ the dimensional density, and ρ_0 the scaling factor. Scaling factors for four quantities can be chosen independently, and then the others follow from Eq. (1). We normally choose r_0, ρ_0 and T_0 such that the resulting normalized length, mass and temperature are $O(1)$ (see, for example, Fig. 1). Note that the choice of r_0, ρ_0 and T_0 depends on the specific application problem to be modeled. The fourth scaling factor we choose is for the constant $\bar{R}$ in the ideal gas law

$$\bar{p} = \bar{\rho}\,\bar{R}\,\bar{T}, \tag{3}$$

where $\bar{R} = \bar{k}/\bar{m}$, with $\bar{k}$ Boltzmann's constant and $\bar{m}$ the mass of a gas molecule. We choose $R_0 = \bar{k}/\bar{m}$, such that the normalized $R = \bar{R}/R_0 = 1$, and the ideal gas law becomes

$$p = \rho\,T \tag{4}$$

in normalized variables. Some of the other normalization factors are then given by

$$p_0 = R_0\,\rho_0\,T_0, \quad u_0 = c_0 = \sqrt{R_0\,T_0}, \quad (GM)_0 = \rho_0\,r_0^3, \tag{5}$$

and the others follow similarly.

Existing numerical approximation methods for stationary transonic flows typically require a surprisingly large computational effort. This is an issue even for one-dimensional (1D) flows: while the computational effort for stationary 1D transonic flow simulation may be modest in absolute terms, it still takes an amount of work that is disproportionate to the relatively small number of discrete unknowns to be calculated. This is due to the various difficulties that are hidden in Eq. (1), including strong nonlinearities, stiffness at the critical point, and difficulties with the number of boundary conditions. The same types of computational difficulties actually also arise for the multi-dimensional full form of Eq. (1) [8].

Most traditional methods for computing stationary transonic solutions start from the time-dependent form of the equations (Eq. (1)), rather than its stationary counterpart,

$$\frac{\partial}{\partial r}\begin{bmatrix} \rho\,u\,r^2 \\ \rho\,u^2\,r^2 + p\,r^2 \\ (\frac{\gamma p}{\gamma-1} + \frac{\rho u^2}{2})\,u\,r^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\rho\,GM + 2\,p\,r \\ -\rho\,u\,GM + q_{heat}\,r^2 \end{bmatrix}. \tag{6}$$

Traditional methods employ a time-marching strategy: some initial condition is advanced in time until a stationary solution is reached. An advantage of time-marching methods is that the solutions obtained are physically stable solutions. Eq. (1) is of hyperbolic type, with wave speeds $\lambda_1 = u - c, \lambda_2 = u, \lambda_3 = u + c$ [8]. The sound speed c, the temperature T, and the entropy S of a perfect gas are given by

$$c^2 = \frac{\gamma p}{\rho}, \tag{7}$$

$$T = \frac{p}{\rho} = \frac{c^2}{\gamma}, \tag{8}$$

$$S = \frac{p}{\rho^\gamma}, \tag{9}$$

in our choice of units.

One intuitive reason why traditional time-marching methods converge slowly, is that near critical points $u \sim c$ and thus $\lambda_1 = u - c \sim 0$, such that some error components are propagated out of the system with very slow wave speeds [9]. As a consequence, standard explicit finite volume methods [8] for Eq. (1) may require many thousands of timesteps to converge, also depending on the grid size and the resulting Courant-Friedrichs-Lewy (CFL) condition. Several methods have been introduced that accelerate convergence, including local timestepping and implicit timestepping. In particular, important advances for solving stationary Euler problems have been made through the technique of local preconditioning (see [9,10], and references therein). In this approach, convergence of the time-dependent equations is improved by pre-multiplying the time-dependent term in Eq. (1) by a judiciously chosen local preconditioning matrix P. The preconditioning matrix is chosen such that some of the stiffness related to vanishing eigenvalues of the Jacobian matrix A is removed.

In addition to slow convergence, a second major concern with existing methods for stationary Euler flow calculations is that the resulting flow profiles may not be accurate, in the sense that the error is large in low Mach-number regions, compared to the error in other regions of the flow [9]. The planetary outflow solutions targeted in our work are also especially challenging in terms of accuracy. For example, Fig. 2 shows a transonic solution of Eq. (1) for parameters that are representative for the extrasolar planet case [1]. Over a very small range close to the inner boundary, density and pressure drop by six and four orders of magnitude, respectively. The method we propose in this paper allows, in a straightforward way, to employ adaptive refinement in this region that is based on rigorous error estimation. This is in contrast with most standard solution techiques for Eq. (1), which may not produce accurate results in such areas with steep gradients, and in low Mach-number regions. Indeed, standard solvers for the hyperbolic problems Eqs. (1) and (6) were developed to handle discontinuities that may occur in transient behaviour [8], and these specialized discretizations, often of upwind type, may not be accurate for some steady flow regimes. It turns out, in fact, that the above mentioned preconditioning methods also remedy some of the accuracy issues [9]. Nevertheless, most standard methods for the type of transonic outflows we target in this paper, tend to be inaccurate in the region close to the inflow boundary, where density and pressure gradients are very large (Fig. 2). It has to be noted, however, that the mass outflow flux for planetary outflow problems is normally predicted well by standard methods, even though flow profiles may be inac-

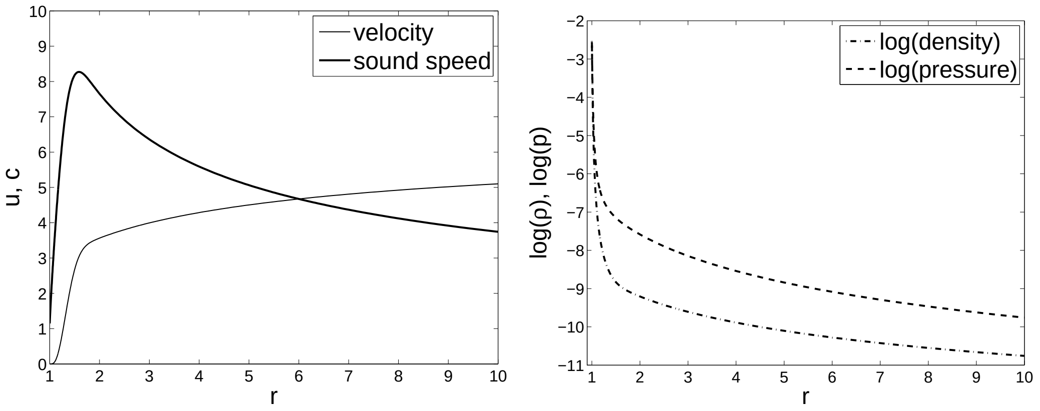

Fig. 2. More realistic radial transonic flow solution for $f_{ext} = -\rho GM$ with $GM \approx 2.6 \times 10^2$, $q_{heat} = A_0 \exp(-10\,(r-1.1)^2)$, and $A_0 \approx 4.4 \times 10^{-6}$. Density and pressure values at the inflow boundary $r_a = 1$ are $\rho \approx 2.9 \times 10^{-3}$ and $p \approx 2.7 \times 10^{-3}$. Note the decimal logarithm scale on the density and pressure plot.

curate near the lower boundary [1].
Driven by this generally unsatisfactory state of affairs in terms of convergence speed and flow profile accuracy, and by our need for doing large numbers of transonic flow simulations for planetary atmosphere research [1,4], we have developed a new approach for calculating radial transonic flow profiles that is based on directly solving the stationary problem Eq. (6), rather than using time-relaxation of the hyperbolic system (1). An important reason why the stationary route is normally not considered, is related to difficulties with the number of boundary conditions. For example, the problem in Fig. 1 has only two boundary conditions (at the inflow boundary), and it is not clear how that allows for solving the three equations in three unknowns of stationary problem (6). In contrast, if this problem is solved by hyperbolic time-relaxation using Eq. (1), the boundary conditions can be handled in a standard way: a subsonic inner boundary with two inflow conditions and one outflow condition, and a supersonic outer boundary with three outflow conditions. However, aided by insight derived from a dynamical system representation of the stationary equations, it is possible to find a solution for the boundary condition problem for the stationary equation. This paper describes how, starting from a dynamical system formulation of stationary equation (6) and using a few simple ideas, a method can be obtained that allows fast and accurate calculation of difficult stationary flow problems.

This paper is organized as follows. In the next section, we illustrate the first important component of our algorithm, namely calculation of transonic flow solutions by adaptive integration outward from the critical point. For simplicity, this idea is presented in the context of the isothermal Euler equations. In Section 3, the second important component of our approach is described, namely the use of a Newton method which allows the critical point to vary within the manifold of possible critical points, thus obtaining the particular transonic solution and critical point that match the inflow boundary conditions. The resulting algorithm is described, and its performance is illustrated using example flows with various heating profiles. In Section 4, it is shown how the proposed method can be extended to handle transonic flows with shocks, in the context of an example application of accretion onto a black hole. Finally, conclusions are drawn in Section 5.

2 A Critical Point method for isothermal Euler flows

In this section, we illustrate the first essential component of our algorithm for calculating radial transonic solutions of the Euler equations [5]. This ingredient of the algorithm is best illustrated for the simplified case of the isothermal Euler equations.

The isothermal limit of Eq. (1) is given by

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\, r^2 \\ \rho\, u\, r^2\end{bmatrix} + \frac{\partial}{\partial r}\begin{bmatrix}\rho\, u\, r^2 \\ \rho\, u^2\, r^2 + \rho\, c^2\, r^2\end{bmatrix} = \begin{bmatrix}0 \\ -\rho\, GM + 2\,\rho\, c^2\, r\end{bmatrix}, \tag{10}$$

and can be obtained from the full Euler equations by assuming constant temperature, such that the energy equation drops out and $\gamma = 1$. The sound speed c is now a constant to be specified. The isothermal radial transonic solution on an interval $[r_a, r_b]$ that is analogous to the solution in Fig. 1, now requires one boundary condition at the inflow boundary (namely, $\rho(r_a)$), and no boundary conditions are required at the outflow boundary. We emphasize again the remarkable fact that a single boundary condition (together with the condition that the flow makes a transition from subsonic to supersonic flow) uniquely specifies a stationary solution to system (10), which has two equations. The reason for this anomaly is clarified below.

A decoupled equation for $u(r)$ can be isolated from the stationary form of Eq. (10), resulting in

$$\frac{du}{dr} = \frac{2\, u\, c^2\, \left(r - \frac{GM}{2c^2}\right)}{r^2\, (u^2 - c^2)}. \tag{11}$$

This equation can be integrated easily as

$$\left(\frac{u}{c}\right)^2 - \log u^2 = 4\, \log r + 4\, \frac{GM}{2\, r\, c^2} + d, \tag{12}$$

with d an integration constant. Fig. 3 shows solutions for various values of the integration constant d, for the case $GM = 2$ and $c = 1$. Note that there is only one transonic outflow solution: the transonic conditions $u(r_a) < c$ and $u(r_b) > c$ fully specify the unique transonic outflow solution of Eq. (11), and no further boundary information is required. In contrast, specifying any of the

other solution curves requires a boundary condition. Also, it has to be noted that for the simplified case of the isothermal Euler equations, the transonic solution sought is given implicitly by Eq. (12), while for the full Euler equations (1) with heating profile q_{heat}, there is generally no analytical solution. For the isothermal case, an explicit expression for $u(r)$ for the transonic solution branch can be derived from Eq. (12) in terms of the Lambert W function [11], which is defined implicitly by $z = W(z)\,\exp(W(z))$.
We now turn to the dynamical system form of Eq. (11). Throughout this paper, investigating the dynamical system form of the stationary equations provides insight into the nature of the transonic solutions, and contributes to the formulation of efficient algorithms for finding stationary transonic solutions. Eq. (11) can be written as a dynamical system as follows:

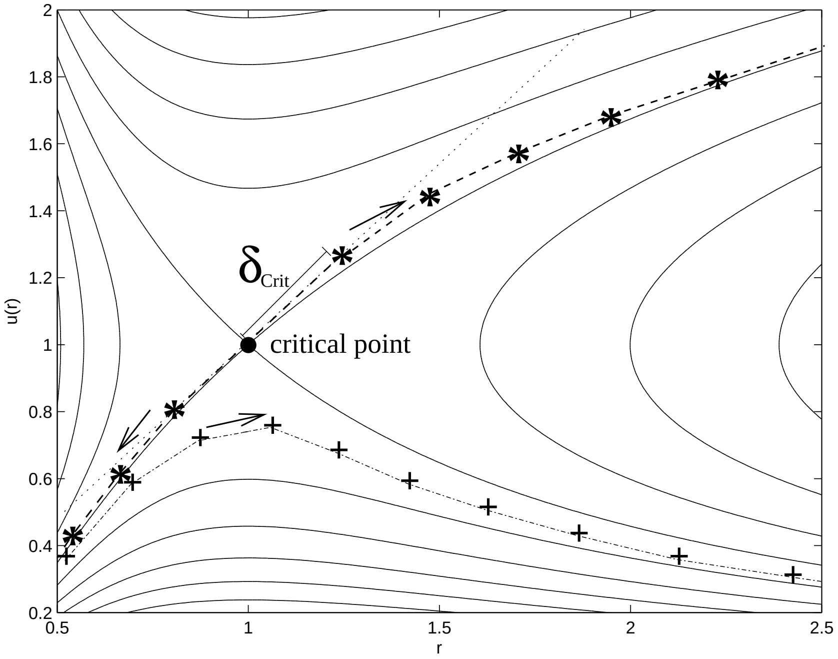

Fig. 3. Solution curves $u(r)$ for the isothermal Euler eqations with $GM = 2$ and $c = 1$. Numerical integration from the left boundary (dash-dotted curve) does not allow to approximate the transonic solution, but integration in two directions outward from the critical point (dashed curves) leads to an accurate numerical approximation of the transonic curve. The arrows indicate the direction of numerical integration. The two starting points for the numerical integration are chosen a distance δ_{Crit} away from the critical point.

$$\begin{aligned} \frac{du(s)}{ds} &= -2\,u(s)\,c^2\left(r(s) - \frac{GM}{2c^2}\right), \\ \frac{dr(s)}{ds} &= -r(s)^2\,(u(s)^2 - c^2), \end{aligned} \tag{13}$$

or, in matrix form,

$$\frac{d\mathbf{V}}{ds} = \mathbf{G}(\mathbf{V}), \tag{14}$$

with the state vector given by $\mathbf{V} = [\,u(s) \quad r(s)\,]^T$. The solutions $u(s)$, $r(s)$ of this dynamical system – which are also called solution trajectories — are parametrizations of the solution curves of Fig. 3. The critical point (or equilibrium point) of this dynamical system is attained when the right hand side of Eq. (14) vanishes, which leads to the conditions

$$\begin{aligned} r_{crit} &= \frac{GM}{2c^2}, \\ u_{crit} &= c. \end{aligned} \tag{15}$$

For our choice of the parameters GM and c, the critical point is the point $(1, 1)$ in the (r, u) phase plane of Fig. 3. Note that du/dr in Eq. (11) is undefined at the critical point. The equilibrium point is of saddle-point type. The transonic solution sought is called a stable manifold of dynamical system (14), because all solutions of system (14) with initial conditions on the transonic curve asymptotically tend to the equilibrium point.

Fig. 3 also provides some intuition as to what kind of numerical approach would be appropriate for approximating the transonic curve numerically, if an analytical solution were not known. The dash-dotted curve shows that numerical integration of Eq. (11) from the left boundary would not be succesful, even if the velocity on the transonic solution curve at the left boundary were known, because rounding and truncation errors would make the numerical approximation deviate from the transonic branch, regardless how close the initial condition is chosen to the transonic solution. On the other hand, the dashed curves suggest that integration outward from the critical point may be a more viable idea. Using a local linearization of Eq. (14) about the critical point, the directions tangent to the stable and unstable manifolds can be obtained. With initial points chosen at distance δ_{Crit} from the critical point in the direction tangent to the stable manifold, Eq. (11) can be integrated numerically in two directions away from the critical point. The trajectories are attracted toward the stable manifold as the integration progresses, resulting in an accurate numerical approximation of the transonic solution. The accuracy of the approximation can be increased by reducing δ_{Crit} and the integration step size. This method for calculating stable manifolds of dynamical systems is actually well-known and used often in the numerical study of dynamical systems [12]. In the case of the isothermal Euler equations, linearization of dynamical system (14) results in the following Jacobian matrix at the critical point,

$$\left.\frac{\partial \mathbf{G}}{\partial \mathbf{V}}\right|_{\mathbf{V_{crit}}} = \begin{bmatrix} 0 & 2\,c^3 \\ \frac{(GM)^2}{2\,c^3} & 0 \end{bmatrix}. \tag{16}$$

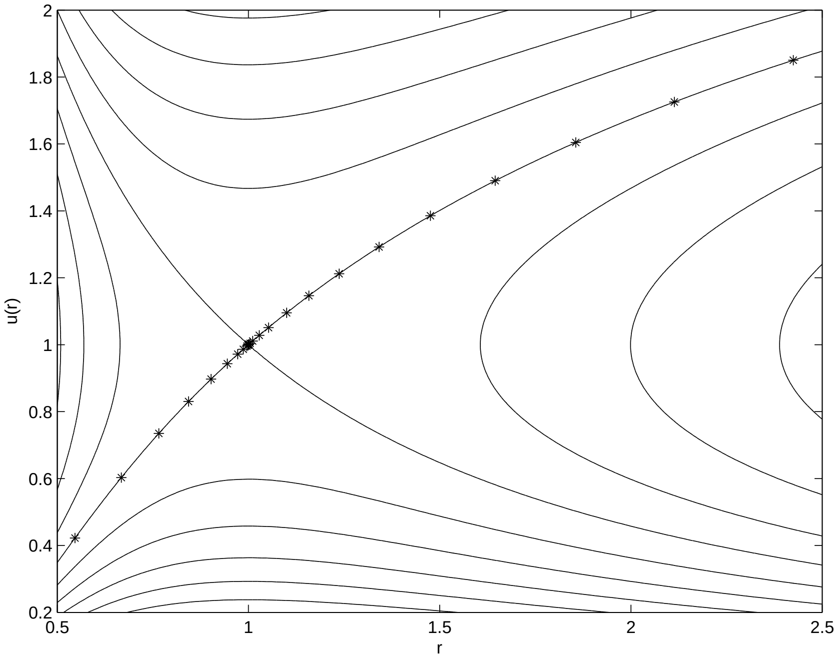

Fig. 4. Numerical approximation of a transonic outflow solution for the isothermal Euler equations using the approach sketched in Fig. 3.

Its eigenvalues are $\lambda_{1,2} = \pm GM$, and its eigenvectors $\vec{x}_1 = [\,1 \quad GM/(2c^3)\,]^T$ and $\vec{x}_2 = [\,1 \quad -GM/(2c^3)\,]^T$ give the directions tangent to the stable and unstable manifolds. Using these results and the approach outlined above, the transonic solution can be approximated accurately and efficiently, as illustrated in Fig. 4. We have used an adaptive fourth-fifth order accurate Runge-Kutta ordinary differential equation (ODE) integrator for the result shown, namely the RK45 Fehlberg method as described, for example, in [13]. The same ODE integrator is used throughout the remainder of this paper.

The reason why this method, which originates in dynamical systems research, works without change for our problem, is that the critical point for the isothermal Euler equations is known in advance, as all quantities in the right hand side of Eq. (15) are known constants. For the full Euler equations, however, the location of the critical point is not known *a priori*. In the next section the second important component of our algorithm is described, which allows the use of the above described integration technique even when the location of the critical point in the desired transonic solution is not known ahead of time.

In this paper, we focus primarily on the transonic outflow solution in Fig. 3, which makes a transition from subsonic to supersonic flow at the critical point. The solution curves in Fig. 3 that do not pass through the critical point, like the so-called ‘breeze’ outflow solutions that remain subsonic over the whole domain, can be integrated simply using forward numerical integration, or using shooting methods with boundaries that are fixed in space, depending on where the boundary conditions are posed. The critical point does not pose a difficulty in this case, and a dynamical systems formulation is not needed. In

the extrasolar planet outflow application, the transonic solution is normally the physically relevant solution, because it is the only solution that can match the usually very low stellar wind pressure [1]. This is similar to the case of the solar wind, which matches the very low interstellar medium pressure [5]. For planetary outflow, there are also cases with a fully subsonic solution, which are obtained when the stellar wind pressure is relatively strong (for example, when planets orbit very close to their star). However, the subsonic cases do not require the methods described in this paper.
Note that Fig. 3 does not show the trajectories with negative velocities that are also solutions of Eq. (12) by symmetry. The second transonic solution branch of Fig. 3 (or, in particular, its negative-velocity counterpart) also has important applications, for example in accretion processes in astrophysics. The numerical method described in this paper can also be applied for efficient numerical calculation of the second transonic solution branch, simply by choosing the other of the two eigenvectors of the dynamical system in the procedure above, and choosing the boundary condition on the appropriate side of the domain. This is illustrated in an example describing accretion onto a black hole in Section 4.

3 A Newton Critical Point method for Euler flows

In this section, the second important component of our algorithm is described, namely a Newton method for locating the critical point of the transonic solution. We proceed as in the previous section by writing the Euler equations (1) with $f_{ext} = -\rho\, GM$ as a dynamical system as follows:

$$\begin{aligned}
\frac{dF}{ds} &= 0, \\
\frac{du}{ds} &= 2\,u\,c^2\,(r - \frac{GM}{2c^2}) - (\gamma-1)\,q_{heat}\frac{r^4\,u}{F}, \\
\frac{dr}{ds} &= r^2\,(u^2 - c^2), \\
\frac{dT}{ds} &= (\gamma-1)\,T\,(GM - 2\,u^2\,r) - (\gamma-1)\,q_{heat}\frac{r^4}{F}\,(T-u^2).
\end{aligned} \tag{17}$$

Note that the radial mass flux F is a constant of motion in the dynamical system. Some straightforward algebra shows that the critical point of this system is not uniquely defined. In fact, there is a two-parameter family of critical points, which can be parametrized, for instance, by the flux F_{crit} and the radius r_{crit} at the critical point. Using this parametrization, the critical temperature and velocity are given by

$$T_{crit} = \frac{GM}{2\,\gamma\, r_{crit}} + (\gamma - 1)\,\frac{q_{heat}\, r_{crit}^3}{2\,\gamma\, F_{crit}}, \tag{18}$$

$$u_{crit} = \sqrt{\gamma\, T_{crit}}. \tag{19}$$

Let us now return to the problem of Fig. 1, namely a transonic outflow solution with boundary conditions $\rho = 5$ and $p = 23$ at the inflow boundary $r_a = 1$. If one seeks this solution, the integration method described in Sec. 2 cannot be used directly because the location of the critical point is not known. We propose the following approach. For given values F_{crit} and r_{crit} at the critical point, define $\mathbf{C} = [F_{crit} \;\; r_{crit}]^T$, and denote the density and pressure vector of the associated critical flow solution at the inflow boundary r_a by $\mathbf{B} = [\rho(r_a) \;\; p(r_a)]^T$. We can say that the critical point vector $\mathbf{C}$ is mapped onto the inflow boundary vector $\mathbf{B}$ by a mapping $\mathbf{F}$:

$$\mathbf{B} = \mathbf{F}(\mathbf{C}). \tag{20}$$

This mapping is provided by the transonic solution of Eq. (1) that passes through the critical point $\mathbf{C}$. Note that $\mathbf{F}$ is a nonlinear mapping. Denote the target density and pressure at the inflow boundary by $\mathbf{B}^* = [\rho^*(r_a) \;\; p^*(r_a)]^T$, with, for instance, $\mathbf{B}^* = [5 \;\; 23]^T$ for the example of Fig. 1. The problem of finding the critical point that corresponds to boundary value vector $\mathbf{B}^*$ can then be formulated as:

$$\text{find } \mathbf{C} \quad \text{s.t.} \quad \mathbf{B}^* = \mathbf{F}(\mathbf{C}). \tag{21}$$

Since $\mathbf{F}$ is a nonlinear function, this equation cannot be solved directly, but we can resort to a Newton approach to approximate $\mathbf{C}$ iteratively. Taylor series expansion directly leads to the Newton formula

$$\mathbf{B}^* = \mathbf{F}(\mathbf{C^{(k)}}) + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{C}}\right|_{\mathbf{C^{(k)}}} \left(\mathbf{C^{(k+1)}} - \mathbf{C^{(k)}}\right), \tag{22}$$

which can be rewritten as

$$\mathbf{C^{(k+1)}} = \mathbf{C^{(k)}} + \left(\left.\frac{\partial \mathbf{F}}{\partial \mathbf{C}}\right|_{\mathbf{C^{(k)}}}\right)^{-1} \left(\mathbf{B}^* - \mathbf{F}(\mathbf{C^{(k)}})\right), \tag{23}$$

or

$$\mathbf{C^{(k+1)}} = \mathbf{C^{(k)}} + \left(J|_{\mathbf{C^{(k)}}}\right)^{-1} \left(\mathbf{B}^* - \mathbf{B^{(k)}}\right), \tag{24}$$

with J denoting the Jacobian of function $\mathbf{F}$, and $\mathbf{B^{(k)}} = \mathbf{F}(\mathbf{C^{(k)}})$. Realizing that the mapping $\mathbf{F}$ can be approximated numerically using the integration technique outward from the critical point that was discussed in Sec. 2, this leads to a practical algorithm for determining the critical point $\mathbf{C}$ and the associated transonic solution that match boundary condition $\mathbf{B}^*$. The Newton Critical Point (NCP) method can then be described as follows:

Algorithm: Newton Critical Point (NCP) Method
In: $\mathbf{B}^*, \mathbf{C}^{(\mathbf{0})}; \epsilon_{Newton}, \epsilon_{ODE}, \delta_{Crit}, \delta_{Jac}$

```
k=0
do
  $\mathbf{B}^{(\mathbf{k})} = \mathbf{F}(\mathbf{C}^{(\mathbf{k})}; \epsilon_{ODE}, \delta_{Crit})$
  if $\|\mathbf{B}^{(\mathbf{k})} - \mathbf{B}^*\|_2 \leq \epsilon_{Newton}$ break
  $\mathbf{B}^{\mathbf{1}} = \mathbf{F}(\mathbf{C}^{(\mathbf{k})} + \delta_{Jac}\ [1\ 0]^T; \epsilon_{ODE}, \delta_{Crit})$
  $\mathbf{B}^{\mathbf{2}} = \mathbf{F}(\mathbf{C}^{(\mathbf{k})} + \delta_{Jac}\ [0\ 1]^T; \epsilon_{ODE}, \delta_{Crit})$
  $[J_{ij}] = (B_i^j - B_i^{(k)})/\delta_{Jac} \quad (i = 1, 2 \text{ and } j = 1, 2)$
  $\mathbf{C}^{(\mathbf{k+1})} = \mathbf{C}^{(\mathbf{k})} + J^{-1}\ (\mathbf{B}^* - \mathbf{B}^{(\mathbf{k})})$
  k=k+1
enddo
```

Inputs to this algorithm are the target boundary value vector $\mathbf{B}^*$, an initial guess for the critical point vector $\mathbf{C}^{(\mathbf{0})}$, and four numerical tolerance parameters: ϵ_{Newton} is the error tolerance for the Newton method, ϵ_{ODE} is the maximal error per step for the adaptive ODE integrator, δ_{Crit} is the distance between the critical point and the first integration point (see Fig. 3), and δ_{Jac} is the increment size for the numerical Jacobian calculation. Note that in the description of the algorithm, $B_i^{(k)}$ is the ith component of $\mathbf{B}^{(\mathbf{k})}$, and B_i^j is the ith component of $\mathbf{B}^{\mathbf{j}}$ ($i = 1, 2$ and $j = 1, 2$).

Fig. 5(a) shows how the NCP algorithm is applied for the calculation of the transonic flow of Fig. 1, starting from initial guess $\mathbf{C}^{(\mathbf{0})} = [F_{crit}\ r_{crit}]^T = [1\ \ 2]^T$. Parameters used were $\epsilon_{Newton} = 10^{-6}$, $\epsilon_{ODE} = 10^{-7}$, $\delta_{Crit} = 10^{-2}$, and $\delta_{Jac} = 10^{-7}$. Table 1 illustrates the convergence behaviour of the method.

Table 1
Newton method convergence for the NCP method applied to the problem of Fig. 1.

Newton step k	error $\|\mathbf{B}^{(\mathbf{k})} - \mathbf{B}^*\|_2$
1	4.41106268600662
2	2.28831581534917
3	1.43924405447424
4	0.10259052732943
5	0.00125578478131
6	0.00000037420499

Convergence of the method is fast: only six Newton steps were needed to

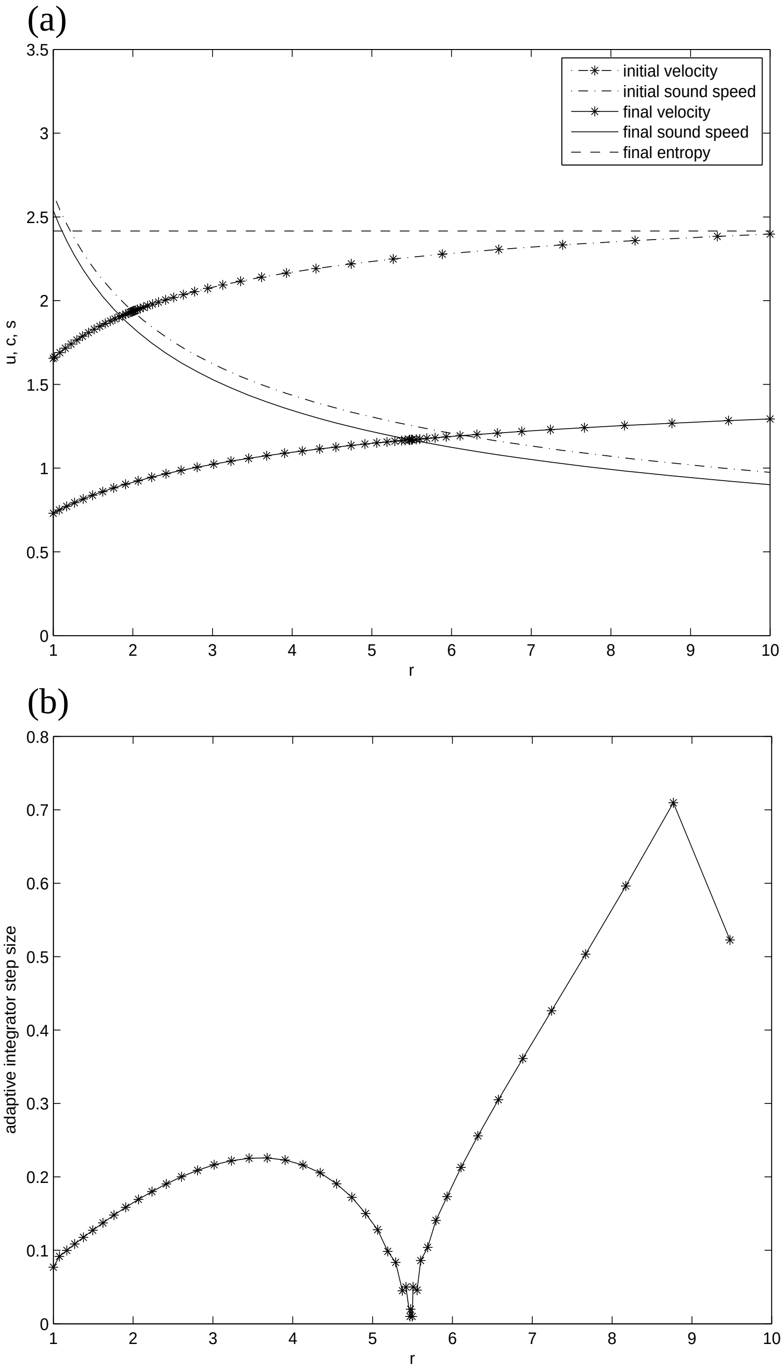

Fig. 5. (a) Initial and final critical points and trajectories for calculation of the transonic flow of Fig. 1 using the NCP algorithm. The algorithm proceeds by varying the critical point on the two-parameter manifold of critical points by use of a Newton method until the solution matches the specified boundary values. (b) Grid size values used by the adaptive integrator for the final solution obtained by the NCP algorithm. Smaller step sizes are selected close to the critical point, and near the inflow boundary where steep gradients are encountered.

obtain convergence for the result of Fig. 5. This is significant: only six explicit integration sweeps over half of the computational domain downward from the critical point are needed as the Newton process proceeds, together with six inversions of a two-by-two linear system, and a final full integration sweep for the final solution. This is to be compared with the work required for time-relaxation methods. Explicit relaxation methods require up to several thousands of full integration sweeps. Relaxation methods based on implicit discretization of time-dependent Eq. (1) may also require only a limited number of Newton-like iterations, but each step requires the inversion of a large Jacobian matrix that contains all the discrete degree of freedoms, while in our approach the Newton system to be solved is a small two-by-two system. It is thus clear that the NCP method offers an efficient alternative to these existing methods.

Fig. 5(b) shows how the adaptive RK45 integration method effectively varies the grid for the numerical integration. Fig. 5(a) also plots the entropy of the final solution. As $q_{heat} = 0$ for this test problem, the entropy should remain constant. The plot confirms that our method is highly accurate: the entropy remains constant with high precision. This is an important advantage of our method. Even for a simple test problem as the one in Fig. 5(a), most existing methods produce significant entropy errors near the inflow boundary where density and pressure have steep gradients. Our method does not suffer from such inaccuracies due to the adaptive integrator that limits the error per step to a given input tolerance.

It can be noted that the NCP algorithm is akin to a nonlinear shooting method, with the essential difference that the location from which the shooting integration is performed is updated in every step of the shooting process, as the location of the critical point is one of the unknowns to be determined by the shooting procedure.

We can make the following remarks about the adaptive integrator that calculates the solution trajectories downward from the critical point in order to approximate $\mathbf{B} = \mathbf{F}(\mathbf{C})$. As it turns out, it is not necessary to calculate the trajectories using dynamical system (17), but it is sufficiently accurate and easier to directly use the expressions for du/dr and dT/dr that correspond to the equations in system (17). The direction tangent to the stable manifold can be found as in Section 2, by linearizing the dynamical system about the critical point.

Without the heating term ($q_{heat} = 0$), dynamical system (17) simplifies to

$$
\begin{aligned}
&\frac{dF}{ds} = 0,\\
&\frac{dS}{ds} = 0,\\
&\frac{du}{ds} = 2\,u\,c^2\,(r - \frac{GM}{2c^2})\\
&\frac{dr}{ds} = r^2\,(u^2 - c^2).
\end{aligned}
\tag{25}
$$

Note that we have replaced the equation for dT/ds with an equation for dS/ds, which is very simple in this case. Linearization of the u, r sub-system about the critical point produces Jacobian

$$
\left.\frac{\partial \mathbf{G}}{\partial \mathbf{V}}\right|_{\mathbf{V_{crit}}} = \begin{bmatrix} GM\,(-\gamma+1) & 2\,c^3\,(-2\,\gamma+3) \\ \frac{(GM)^2}{4\,c^3}\,(\gamma+1) & GM\,(\gamma-1) \end{bmatrix}, \tag{26}
$$

with eigenvalues $\lambda_{1,2} = \pm GM\,\sqrt{(-3\,\gamma+5)/2}$. The eigenvectors tangent to the stable and unstable manifolds are given by $\vec{x}_1 = [\,2\,c^3\,(-2\,\gamma+3) \quad (\gamma - 1 + \sqrt{(-3\,\gamma+5)/2})\,GM\,]^T$ and $\vec{x}_1 = [\,2\,c^3\,(-2\,\gamma{+}3) \quad (\gamma{-}1{-}\sqrt{(-3\,\gamma+5)/2})\,GM\,]^T$. When $q_{heat} \neq 0$, the linearization does not produce a Jacobian matrix that leads to further insight. The analytical expression for the Jacobian can actually be quite complicated, because in general the heating function q_{heat} may depend on the radius r or the local state of the gas (for instance the density ρ). For this reason, we decided to calculate the linearization Jacobian about the critical point and its eigenvectors numerically in our implementation of the NCP method, with numerical Jacobian increment size δ_{Jac} the same as for the numerical calculation of the Jacobian of the function $\mathbf{F}$.

Fig. 6 shows some sample solutions obtained by the NCP algorithm for various heating profiles, namely an inverse-distance-squared heating profile, a step profile, and a Gaussian profile. Parameters were as above, with 5, 13 and 5 Newton steps required for convergence, respectively. It can be seen that the adaptive integrator increases the point density where necessary, for instance near the critical points, near the inflow boundary, and near the edges of the step heating profile in Fig. 6(b).

Note, finally, that the NCP method can be extended easily for flows with shocks. For example, the transonic outflow solution of Fig. 1, after the critical point, can be connected to a subsonic branch by a stationary shock wave (termination shock). This requires an additional boundary condition, for instance on the pressure, at the outflow boundary. The transonic solution part can be calculated first, and the location of the shock can then be matched to the value of the outflow boundary condition by a Newton procedure, just like in the NCP method described above. This is further illustrated in the next

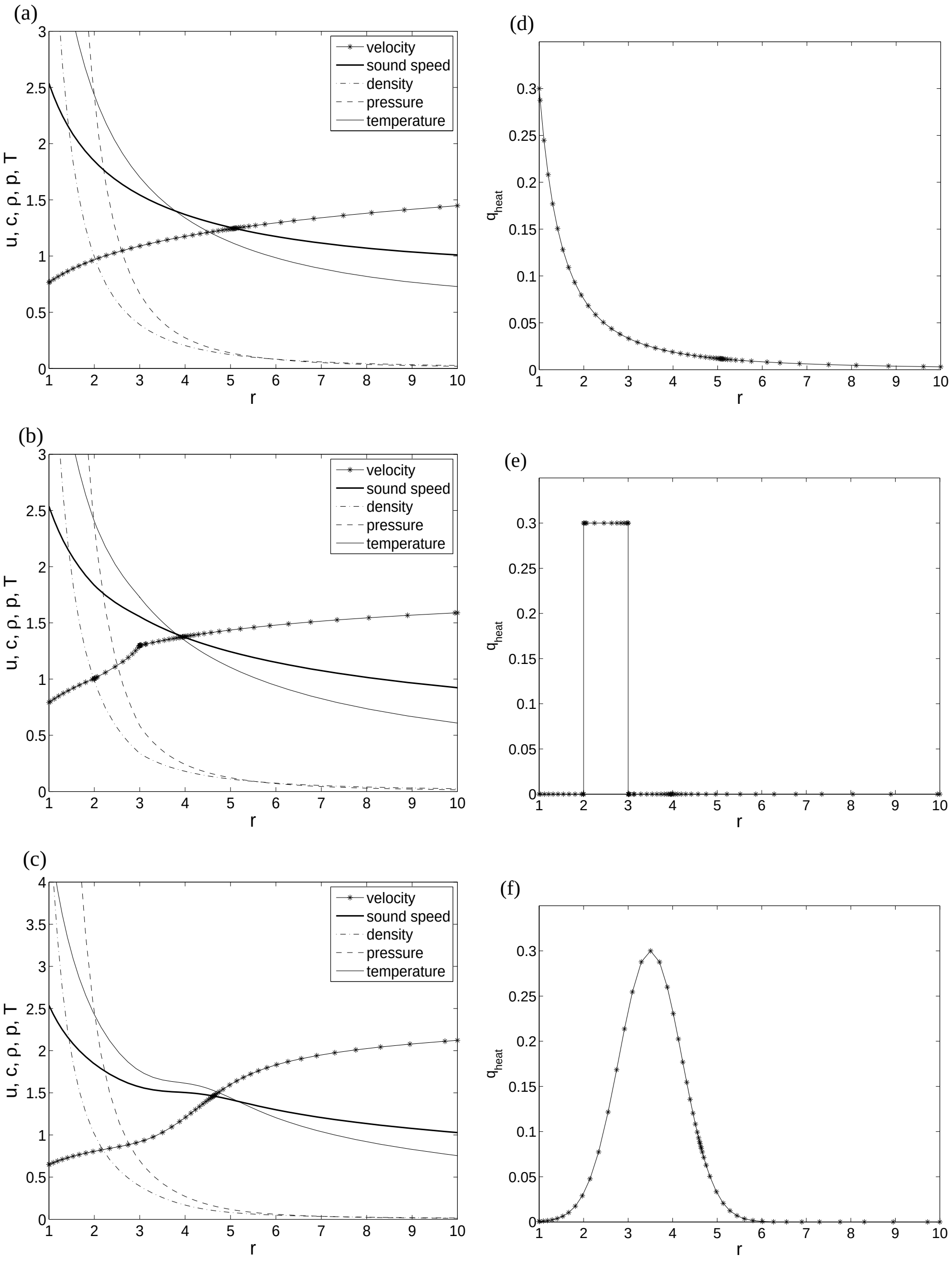

Fig. 6. Radial transonic flow solutions for various heating profiles obtained by the NCP method. The heating profiles are: (d) $q_{heat}(r) = 0.3/r^2$, (e) $q_{heat}(r) = 0.3$ for $r \in [2, 3]$ and $q_{heat}(r) = 0$ elsewhere, and (f) $q_{heat}(r) = 0.3 \exp(-(r - 3.5)^2)$.

Section.

4 Example application: accretion onto a black hole with a shock

In this section we describe how the NCP method can be applied to calculate a transonic solution branch that makes a transition from subsonic inflow away from the central object to supersonic inflow close to it, and how NCP can be extended to handle flows with shocks. We find numerical solutions for a simple model of an accretion disk around a black hole, as described in [14,15].

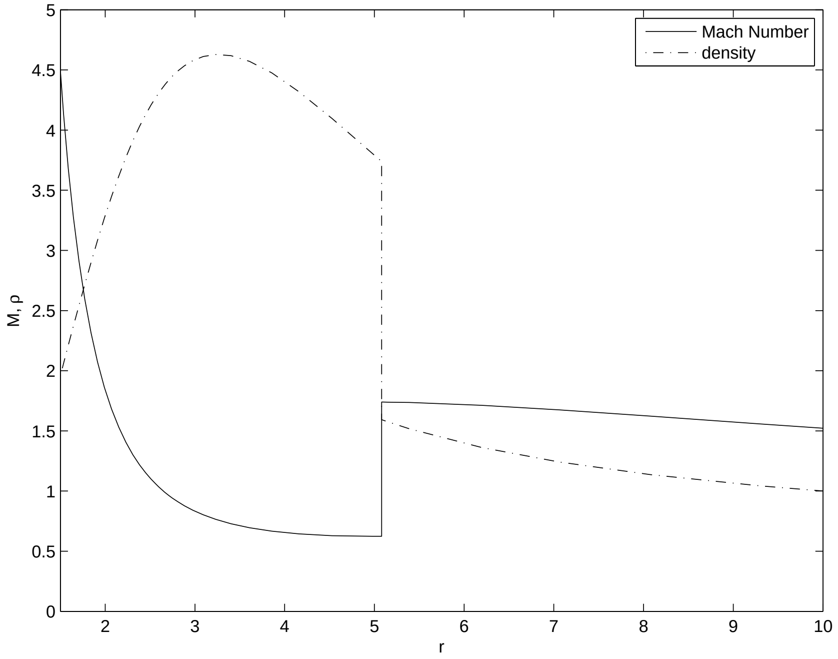

Fig. 7. Numerically obtained accretion flow solution onto a black hole with a shock. Matter flows into the black hole from the right. At $r = 10$, the incoming flow is supersonic. It becomes subsonic at a shock transition ($r \approx 5$), and then makes a transition from subsonic to supersonic inflow through a critical point that is located at $r \approx 2.5$. The boundary conditions at $r = 10$ are $\rho = 1$, $p = 0.01428$, $u_r = -0.21$, and $u_\theta = 0.1785$.

We solve stationary Euler system

$$\frac{\partial}{\partial r}\begin{bmatrix}\rho u_r r\\ \rho u_r^2 r + p\, r\\ \rho u_r u_\theta r^2\\ u_r(e+p)r\end{bmatrix} = \begin{bmatrix}0\\ p + \rho u_\theta^2 + \rho g(r) r\\ 0\\ \rho g(r) u_r r\end{bmatrix}, \tag{27}$$

with pseudo-Newtonian gravity field $g(r) = -(r-1)^{-2}/2$ [14,15]. Here, u_r and u_θ are the radial and azimuthal velocity components in the plane of the disk, and $e = p/(\gamma - 1) + \rho(u_r^2 + u_\theta^2)/2$. The radial distance, r, is measured in units of Schwarzschild radius. The adiabatic index $\gamma = 4/3$. The system has cylindrical symmetry around the axis of the accretion disk, and the equations describe the flow in the equatorial plane.

Fig. 7 shows an accretion solution, in which matter flows into the black hole from the right. At $r = 10$, the incoming flow is supersonic. It becomes subsonic at a shock transition ($r \approx 5$), and then makes a transition from subsonic to supersonic flow through a critical point that is located at $r \approx 2.5$. The boundary conditions at $r = 10$ are $\rho = 1$, $p = 0.01428$, $u_r = -0.21$, and $u_\theta = 0.1785$. We now proceed with a critical point analysis, which, together with the Rankine-Hugoniot relations at the shock, allows us to use the NCP method for calculating flow profiles like the one in Fig. 7. We can solve the first and third equations in (27) to obtain the two constants of motion

$$F = \rho u_r r, \qquad \mu = u_\theta r.$$

These are constant both for continuous and discontinuous flow. The remaining two equations can also be integrated in this simple case without further source terms [15], but for our purposes we want to write them as a system of ODEs for u_r and T. Using the constants and the equation of state, $p = \rho T$, one obtains

$$\begin{aligned} \frac{du_r}{dr} &= \frac{1}{(u_r^2 - \gamma T)} \left(-\frac{u_r}{2(r-1)^2} + \frac{\mu^2}{r^3} u_r + \frac{T\gamma u_r}{r} \right), \\ \frac{dT}{dr} &= \frac{-(\gamma - 1)\, T}{(u_r^2 - \gamma T)} \left(-\frac{1}{2(r-1)^2} + \frac{\mu^2}{r^3} + \frac{u_r^2}{r} \right). \end{aligned} \tag{28}$$

This system of ODEs has a critical point (of saddle-point type) when

$$\begin{aligned} T_{crit} &= \frac{r_{crit}}{2\gamma(r_{crit} - 1)^2} - \frac{\mu_{crit}^2}{\gamma r_{crit}^2}, \\ u_{r,crit} &= \sqrt{\gamma T_{crit}}. \end{aligned} \tag{29}$$

At the shock, the Rankine-Hugoniot relations express that $\rho u_r^2 + p$ and $u_r(e+p)$ remain constant, in addition to F and μ. With $\mathcal{M} = u_r/c$ denoting the normal Mach Number, and the subscripts 1 and 2 denoting the states on the two sides of the shock, it can be derived that

$$\begin{aligned} \frac{\rho_2}{\rho_1} &= \frac{u_{r,1}}{u_{r,2}} = \frac{(\gamma+1)\mathcal{M}_1^2}{(\gamma-1)\mathcal{M}_1^2 + 2}, \\ \frac{T_2}{T_1} &= 1 + \frac{2(\gamma-1)(\gamma\mathcal{M}_1^2 + 1)}{(\gamma+1)^2\mathcal{M}_1^2}(\mathcal{M}_1^2 - 1). \end{aligned} \tag{30}$$

The solution of Fig. 7 can then be calculated using the NCP method as follows. First, the four boundary conditions at the inflow boundary, $r = 10$, fully determine F and μ in the whole simulation domain. The NCP method then matches the location of the critical point, r_{crit}, and the location of the shock, r_{shock}, with the two remaining boundary conditions. Indeed, knowledge of r_{crit} and μ determine T_{crit} and $u_{r,crit}$ according to Eq. (29). Once the state at the critical point is known, the transonic trajectory from the critical point up to the shock can be calculated as in Section 3. At the shock location, the

Table 2
Newton method convergence for the NCP method applied to the problem of Fig. 7.

Newton step k	error $\|\mathbf{B}^{(\mathbf{k})} - \mathbf{B}^*\|_2$
1	0.44190821577274
2	0.27186715877756
3	0.00501067043758
4	0.00001344201187
5	0.00000020814331

values to the right of the shock can be calculated using Rankine-Hugoniot relations (30), and the trajectory to the right of the shock can be calculated using straightforward ODE integration. The mismatch at the right boundary is then used to improve the current guess for r_{crit} and r_{shock} using Newton's method. Table 2 shows that the Newton procedure converges fast.

5 Conclusion

We have presented the Newton Critical Point method for fast and accurate calculation of 1D transonic solutions of the steady compressible Euler equations with gravity and heating terms. The method has two main ingredients. The first ingredient is adaptive integration outward from the critical point, which is a technique that exploits the dynamical system formulation of the problem and is in fact well-known in dynamical systems research. The second ingredient follows from some more dynamical system analysis and a simple idea: a Newton method is employed that allows the critical point to vary within the manifold of possible critical points, thus driving the iterative approximation toward the particular transonic solution and critical point that match the inflow boundary conditions. It was shown that the method converges fast and that the solution is accurate, in the sense that the solution in low Mach-number regions and at steep gradients is as accurate as in other parts of the flow. In fact, the solution procedure uses a rigorous error estimator and the integration step is chosen adaptively such that the error per step is bounded by a pre-specified error tolerance. The method can be extended easily to handle flows with shocks. This was illustrated for an example flow problem that models accretion onto a black hole with a shock.

The method as described in this paper deals with transonic flows in one dimension. The ideas behind the algorithm, however, are quite general. Application of these ideas to problems in multiple spatial dimensions is a promising topic of further research.

References

[1] Feng Tian, Owen B. Toon, Alexander A. Pavlov, and H. De Sterck, 'Transonic Hydrodynamic Escape of Hydrogen from Extrasolar Planetary Atmospheres', Astrophysical Journal 621, 1049-1060, 2005.

[2] A.J. Watson, T.M. Donahue, and J.C.G. Walker, 'The dynamics of a rapidly escaping atmosphere - Applications to the evolution of Earth and Venus', Icarus 48, 150-166, 1981.

[3] James A. Kasting and James B. Pollack, 'Loss of Water from Venus. I. Hydrodynamic Escape of Hydrogen', Icarus 53, 479-508, 1982.

[4] Feng Tian, Owen B. Toon, Alexander A. Pavlov, and H. De Sterck, 'A Hydrogen-Rich Early Earth Atmosphere', Science 308, 1014-1017, 2005.

[5] E.N. Parker, 'Dynamics of the Interplanetary Gas and Magnetic Fields', Astrophysical Journal 128, 664-676, 1958.

[6] T. Sakurai, 'Magnetohydrodynamic solar/stellar wind models', Comput. Phys. Rep. 12, 247-273, 1990.

[7] R. Keppens and J. P. Goedbloed, 'Numerical simulations of stellar winds: polytropic models', Astron. Astrophys. 343, 251-260, 1998.

[8] R.J. LeVeque, 'Finite volume methods for hyperbolic problems', Cambridge University Press, Cambridge, 2002.

[9] B. van Leer, W.T. Lee and P.L. Roe, 'Characteristic time-stepping or local preconditioning of the Euler equations', AIAA Paper 91-1552,1991.

[10] D.L. Darmofal and K. Siu, 'A Robust Multigrid Algorithm for the Euler Equations with Local Preconditioning and Semi-coarsening', Journal of Computational Physics 151, 728-756, 1999.

[11] S.R. Cranmer, 'New views of the solar wind with the Lambert W function', American J. Phys., 72, 1397-1403, 2004.

[12] Bernd Krauskopf, Hinke M. Osinga, Eusebius J. Doedel, Michael E. Henderson, John M. Guckenheimer, Alexander Vladimirsky, Michael Dellnitz and Oliver Junge, 'A survey of methods for computing (un)stable manifolds of vector fields', Int. J. Bifurcation & Chaos 15, 763-791, 2005.

[13] Lars Elden, Linde Wittmeyer-Koch, Hans Bruun Nielsen, 'Introduction To Numerical Computation', Studentlitteratur AB, Lund, 2004.

[14] G. Toth, R. Keppens, and M.A. Botchev, 'Implicit and semi-implicit schemes in the Versatile Advection Code: numerical tests', Astronomy and Astrophysics 332, 1159-1170, 1998.

[15] D. Molteni, G. Tóth, and O.A. Kuznetsov, 'On the Azimuthal Stability of Shock Waves around Black Holes', Astrophysical Journal 516, 411-419, 1999.