2-D stationary gas dynamics in a barred galaxy

W.A. Mulder[a]

[a] *Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Geoscience & Engineering, P.O.Box 5048, 2600 GA Delft, Netherlands*

Abstract

A code for solving the 2-D isothermal Euler equations of gas dynamics in a rotating disk is presented. The gravitational potential represents a weak bar and controls the flow of gas. A damped Newton method solves the second-order upwind discretisation of the equations for a steady-state solution, using a consistent linearisation and a direct solver. Successive grid refinement, starting from a finite-volume grid with 8 by 8 cells, is applied to find solutions on subsequently finer meshes. On coarser meshes, a first-order spatial discretisation is used. The method obtains quadratic convergence once the solution approaches the steady state. The initial searching is quick with the first-order scheme and slower with the second-order discretisation, up to 256 by 256 cells. Beyond, on 512 by 512, the number of iterations becomes too large to be useful. Potential causes are discussed. The code can be applied as practical tool for generating flow models if used on to not too fine meshes.

Keywords: galactic flow, 2-D gas dynamics, shock waves, spiral barred galaxy, Newton's method

1. Introduction

Codes for simulating galactic hydrodynamics come in two flavours. Finite-difference, finite-volume or finite-element codes solve the problem on a mesh, using an accurate discretisation of a set of equations, varying from the Euler equation of gas dynamics in a given gravitational potential to magneto-hydrodynamic viscous flows with self gravity and heating and cooling mechanisms. Smoothed particle hydrodynamics methods [17, 10] attempt to mimic the relevant physics by colliding particles or kernels. They are fairly easy to code and all kinds of seemingly realistic collision rules can be incorporated, but the statistical mechanics required to find the partial differential equations they represent, if any, is difficult or sometime impossible. Both approaches require substantial compute power if large contrasts and different scales need to be handled. There are several public-domain codes for either approach.

In this paper, a 2-D finite-volume code is described, implemented in Matlab [18] and available from [20]. The second-order upwind spatial discretisation of [19] is taken. Following Occam's razor, the number of parameters and the

amount of physics is kept limited. The 2-D stationary isothermal Euler equations in a power-law potential with an ellipsoidal bar are solved. In the early 1980s, this could only be accomplished on relatively coarse grids and with an iterative method. Developments in hardware and numerical algorithm now allow for finer meshes, storage of the fully consistent linearisation of the second-order spatial discretisation for Newton's method, and the use of a direct solver. The output of the code, a density and velocity distribution, can be used for the construction of fairly detailed galactic rotation curves, as done in for instance [23] or, more recently, in [25] with a particle method that captures different physics but appears to lead to qualitatively similar spiral patterns. Matching the resulting rotation curves to the observed ones will provide constraints on the large-scale structure of the background gravitational potential.

Least-squares inversion for model parameters that describe the gravitational potential in terms of its radial decay and the strength of the bar as well as for the parameters that describe the effective properties of the gas, in particular the sound speed, might be feasible, although data quality, coverage and resolution as well as the limitiations of a simplified physical model are likely to pose challenges. A more common approach is some type of manual inversion, in which models for a range of parameters are generated and compared to velocity and density data for a given galaxy by inspection [26, 5, 23, 7, 2, 27, 16, 24, 3, 9, 11, 30, 8, 4, 28, 15, 25, 14, e.g.].

The next section reviews the method. The main difference with earlier work [19] is the use of a consistent Jacobian for the second-order upwind scheme and the use a direct solver for the damped Newton method instead of an iterative multigrid approach with the intention to improve the code's robustness. Section 3 shows one example of a solution obtained with the code. The results are discussed in section 4.

2. Method

A simple description of galactic hydrodynamics is offered by the isothermal Euler equation of gas dynamics in an external gravitational potential. Newton's method can be applied to compute a stationary solution. In the 1-D case, a damped Newton method can be constructed by having a fully implicit time-stepping scheme with a time step that increases when the residual, which measures the rate of change of the evolution equations, decreases [22]. This approach attains quadratic convergence when a consistent Jacobian is applied and the solution approaches the steady state. In the 2-D, an iterative solver may solve the large sparse system related to the fully implicit time-stepping scheme, which can be interpreted as a damped version of the Jacobian of the stationary set of discrete equations [19]. Progress in hardware and solvers allows for the use of direct solver instead. Whereas earlier, a second-order discretisation of the Euler equations was solved with an iterative method for a first-order discretisation, similar to defect correction [29] and approximate Newton methods, the present code employs a consistent Jacobian of the second-order discretisation. The code is implemented in Matlab [18], despite its inherently meagre performance. Since

about half the compute time is spent in a library call to the solver [6], this is not a major drawback. The governing equations and their discretisation are the same as in an earlier paper [19]. The gravitational potential corresponds to a power-law gravitational density distribution with an ellipsoidal bar [21].

Details will be given next. Starting from the 3-D case and solutions for the axisymmetric case, without the ellipsoidal bar, the equations for the 2-D stationary problem will given. This is followed by some implementation details, a description of the code and a list of user parameters.

Governing equations

The 3-D isothermal Euler equations — before simplifying to 2D — in cylindrical coordinates and in a rotating frame are

$$\frac{\partial \mathbf{w}}{\partial t} = \mathbf{r}(\mathbf{w}) = \mathbf{s} - \frac{\partial \mathbf{f}}{\partial R} - \frac{\partial \mathbf{g}}{\partial \phi} - \frac{\partial \mathbf{h}}{\partial z}, \tag{1}$$

where

$$\mathbf{w} = R\rho \begin{pmatrix} 1 \\ u \\ v \\ w \end{pmatrix}, \quad \mathbf{f} = u\mathbf{w} + R \begin{pmatrix} 0 \\ \rho c^2 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{g} = \frac{v}{R}\mathbf{w} + \begin{pmatrix} 0 \\ 0 \\ \rho c^2 \\ 0 \end{pmatrix},$$

$$\mathbf{h} = w\mathbf{w} + R \begin{pmatrix} 0 \\ 0 \\ 0 \\ \rho c^2 \end{pmatrix}, \quad \mathbf{s} = \rho \begin{pmatrix} 0 \\ -R\frac{\partial V}{\partial R} + c^2 + (v + \omega R)^2 \\ -\frac{\partial V}{\partial \phi} - u(v + 2\omega R) \\ -R\frac{\partial V}{\partial z} \end{pmatrix}.$$

The dependent variables are the density ρ, radial velocity u and angular velocity v in a co-rotating frame with angular velocity ω around the z-axis. They depend the radial distance R, angle ϕ and height z, and also on time t until the solution has reached stationarity. The pressure $p = \rho c^2$, where the sound speed c is constant in the isothermal case.

External gravitational field

The background density responsible for the gravitational field is $\rho_g = \rho_{g,0} m^p$, where $m^2 = (x/a)^2 + (y/b)^2 + (z/c)^2$, with $c < b < a$ and $a = 1$. This can be expressed in spherical harmonics as $\rho_g(r, \theta, \phi) = \rho_{g,0} r^p \sum_{n,m} a_{nm} P_n^m(\mu) \cos(m\phi)$, $\mu = \cos(\theta)$, n and m even, $n \geq m$. The gravitational potential V obeys $\Delta V = 4\pi G \rho_g$ and becomes

$$V = (4\pi G \rho_{g,0}) r^{p+2} \sum_{n=0}^{\infty} \sum_{m=0}^{n} c_{n,m}(r) P_n^m(\mu) \cos(m\phi),$$

with $r = \sqrt{R^2 + z^2}$, $\mu = z/r$, n and m even, and $P_n^m(\mu)$ associated Legendre polynomials. Details are given in [21]. The code uses an approximation with terms up to $n = 4$ and $m = 2$, which should suffice for weak bars.

Axisymmetric case.

One solution is $u = 0$, $w = 0$,

$$\rho(R, z) = \rho_a(R) e^{[V(R,0)-V(R,z)]/c^2},$$

and

$$v_{\rm rot} = v + \omega R = c\sqrt{R\left(\frac{1}{c^2}\frac{dV(R,0)}{dR} + \frac{d\log\rho_a(R)}{dR}\right)}.$$

In the axisymmetric case, only the $c_{n,0}$ are involved and these do not depend on $r = \sqrt{R^2+z^2}$. Define $c_0 = \sum_n c_{n,0}(r)P_n^0(0) = c_{0,0} - \frac{1}{2}c_{2,0} + \frac{3}{8}c_{4,0} - \frac{5}{16}c_{6,0} + \ldots$. Then, $V(R, z = 0) = c_0 R^{p+2}$ and

$$v_{\rm rot} = \sqrt{f_0^2 R^{p+2} + c^2 R\frac{d\log\rho_a(R)}{dR}},$$

with $f_0 = \sqrt{(p+2)c_0}$. One choice is $\rho_a(R) = \rho_0$, a constant, leading to $v_{\rm rot} = f_0 R^{1+p/2}$ and $v_0(R) = v_{\rm rot} - \omega R = f_0 R^{1+p/2} - \omega R$. At the other extreme, $v_{\rm rot} = 0$, $\rho_a(R) = \rho_0 e^{-V(R,0)/c^2}$, so $\rho(R, z) = \rho_0 e^{-V(R,z)/c^2}$, which describes hydrostatic equilibrium.

2-D stationary solution

In the 2-D case, we only consider the plane $z = 0$ where $w = 0$, so we can drop the z-derivative of the flux $\mathbf{h}$ as well as the fourth component from the 3-D equations 1. Then, $\mathbf{w} = \mathbf{w}(t, R, \phi)$ and we look for a stationary solution which does not depend on time t by solving

$$\frac{\partial \mathbf{w}}{\partial t} = \mathbf{r}(\mathbf{w}) = \mathbf{s} - \frac{\partial \mathbf{f}}{\partial R} - \frac{\partial \mathbf{g}}{\partial \phi}, \tag{2}$$

where now

$$\mathbf{w} = R\rho\begin{pmatrix}1\\u\\v\end{pmatrix}, \quad \mathbf{f} = u\mathbf{w} + R\begin{pmatrix}0\\\rho c^2\\0\end{pmatrix}, \quad \mathbf{g} = \frac{v}{R}\mathbf{w} + \begin{pmatrix}0\\0\\\rho c^2\end{pmatrix},$$

and the source term

$$\mathbf{s} = \rho\begin{pmatrix}0\\-R\frac{\partial V}{\partial R} + c^2 + (v+\omega R)^2\\-\frac{\partial V}{\partial \phi} - u(v + 2\omega R)\end{pmatrix}$$

Newton's method updates the solution according to

$$\mathcal{A}(\mathbf{w}^n)(\mathbf{w}^{n+1} - \mathbf{w}^n) = \mathbf{r}(\mathbf{w}^n),$$

where $\mathcal{A} = -\partial\mathbf{r}/\partial\mathbf{w}$ is minus the Jacobian matrix of $\mathbf{r}$ with respect to $\mathbf{w}$. An implicit backward Euler scheme will update the solution by

$$\left[\frac{1}{\Delta t} + \mathcal{A}(\mathbf{w}^n)\right](\mathbf{w}^{n+1} - \mathbf{w}^n) = \mathbf{r}(\mathbf{w}^n).$$

This can be interpreted as a damped Newton scheme, if we assume that A is an M-matrix, that is, does not have negative eigenvalues. For a timestep that is inverse proportional to the inverse of a scaled residual norm [22], such a method was given the name Switched Evolution/Relaxation scheme [13]. This residual norm per cell (i, j), where i indexes the ϕ-coordinate and j the radial direcions, is given by

$$\sigma_{ij} = \max\left(\frac{|r_{1ij}|}{w_{1ij}}, \frac{|r_{2ij}|}{|w_{2ij}| + w_{1ij}c}, \frac{|r_{3ij}|}{|w_{3ij}| + w_{1ij}c}\right). \tag{3}$$

If we assume that a local volume factor $\Delta\phi\Delta_j R$ is contained in the numerator, then the overall residual norm is defined by $\max_{ij}[\sigma_{ij}/(\Delta\phi\Delta_j R)]$.

Units

For a unit of length r_0, the unit of velocity is $r_0\sqrt{4\pi G\rho_{g,0}}$. The code sets $4\pi G\rho_{g,0} = 1$ and uses the sound speed c as input parameter. If the gravitational potential V is scaled by $1/c^2$, the velocities u and v are effectively scaled by the sound speed c, which then is set to 1 in the code. The unknowns in the code are $\mathbf{q} = (\log\rho/\rho_{\rm ref}, u/c, (v - v_0)/c)^{\mathsf{T}}$ and the conserved quantities $\mathbf{w} = (R\rho, R\rho u/c, R\rho v/c)^{\mathsf{T}}$. An example of how to convert these internal units to physical values can be found elsewhere [23].

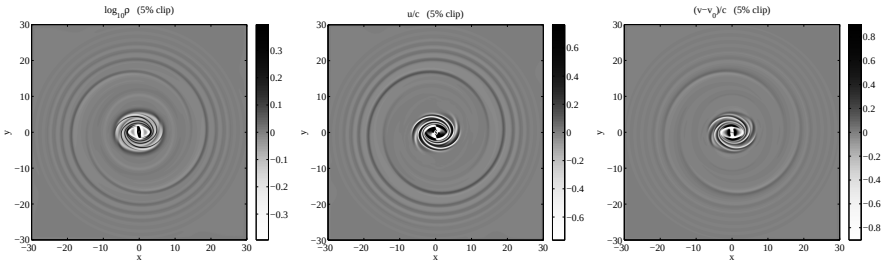
Stretched grid

The $N_\phi \times N_R$ cell centres on the finite-volume grid centres are indexed by (i, j) for ϕ_i and R_j, respectively. The radial grid is gradually stretched such that $\xi = R^{1+\kappa\, p/2}$ is equidistant with constant $\Delta\xi = (\xi_{\max} - \xi_{\min})/N_R$, with $\xi_{\min}$ and $\xi_{\max}$ corresponding to $R_{\min}$ and $R_{\max}$, respectively. Cell boundaries are placed at $\xi_{j-1/2} = \xi_{\min} + (j-1)\Delta\xi$, $j = 1, \ldots, N_R + 1$, from which the $R_{j-1/2}$ follow. Cell centres are placed at $R_j = \frac{1}{2}(R_{j-1/2} + R_{j+1/2})$, $j = 1, \ldots, N_R$. The grid spacing in the ϕ-direction is a constant $\Delta\phi = \pi/N_\phi$. The code assumes that $N_\phi = N_R = n$ and that both are a power of 2. Periodicity and symmetry imply that only the range $\phi \in [0, \pi)$ needs to be considered.

Successive grid refinement is carried out from an initial grid with n_i^2 cells to a final one with n_f^2, where $n_f \gg n_i$.

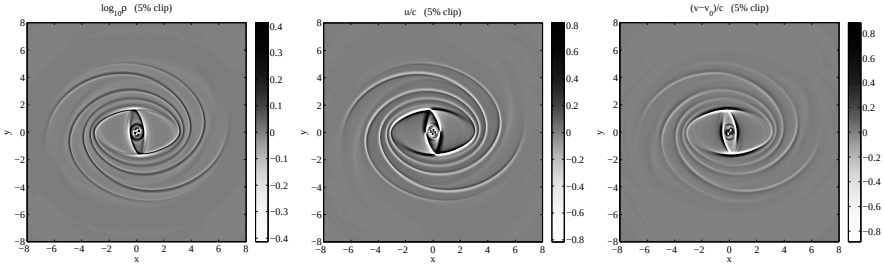
Code

Table 3 lists the various parameters, their meaning and their default values. The default parameters are defined by `gpar = g2setparms();` and can be modified by editing the contents of `g2myparms.m`. An example is shown in Table 1. The code can then be started by entering `g2run` at the Matlab command line. Table 2 shows how to run on a machine with 8 processors if the source code is contained in a directory `../g2/` on a Unix-type operating system.



(a) (b) (c)

Figure 1: Stationary solution: (a) base 10 logarithm of the density, (b) the radial velocity, and (c) the deviation of tangential velocity from the axisymmetric case. To emphasise the spiral pattern, the difference with a smoothed version is plotted and the values are clipped at 10% of their maximum absolute values.



(a) (b) (c)

Figure 2: The central part of Figure 1.

Table 1: Example of the file `g2myparms.m`, used to generate the figures in this paper.

```
% g2myparms.m
% the user may modify the default parameters over here
% ------------------------------------------------------
gpar.nf = 512; gpar.label = 'G02';
%EOF
```

Table 2: Matlab script showing how to run the code on a machine with 8 cores, when the Matlab functions are stored in the directory `../g2`. The default parameters can be changed by editing `g2myparms.m` in the run directory, as shown in Table 1.

```
% please edit g2myparms.m for parameters
%
% path to source files:
addpath('../g2','-end');
% max nr of processors:
maxNumCompThreads(8);
% start run (calls g2myparms.m)
g2run;
%EOF
```

3. Results

Figures 1a–c display a stationary solution obtained with the default parameters of the code, as listed in Table 3. Shown is the difference with respect to the axisymmetric solution. The contrast has been enhanced by means of a spatial high-pass filter and clipping of larger values. The filtered image was obtained by projection of the solution from the polar coordinate system to cartesian coordinates with a grid spacing of 0.04, using simple bilinear interpolation despite the presence of shocks. The result was convolved 128 times by the dyadic or Kronecker product $(1,2,1)^{\mathsf{T}}(1,2,1)$ to obtain a smooth image, which was then subtracted from the original. Figure 2 shows an enlargement of the central part, now using a grid spacing of 0.01. Again, 128 of convolutions steps were used to determine a smooth image, which was subsequently subtracted. Co-rotation occurs at a radius of 8.36 model units, as defined in section 2.

The convergence history is displayed in Figure 3. The residual norm is a weighted version of the solution's time derivative and is defined in the text just below equation (3). Continuation in grid size, known as successive grid refinement, as well as the order of the spatial discretisation was used to obtain a stationary solution with a second-order discretisation on a finite-volume grid with 512×512 cells, equidistant in $\phi \in [0, \pi)$ and stretched in the radial coordinate $R \in [R_{\min}, R_{\max}]$. The other half, $\phi \in [\pi, 2\pi]$, follows by symmetry. The initial grid had 8×8 cells and the solution was computed with a first-order accurate scheme. The stationary result after 17 iterations was interpolated to a

Table 3: Code parameters.

symbol	name	content	default
	`label`	name used for naming output files	`G01`
		physical parameters	
c	`c`	sound speed	0.035
ρ_0	`rhoinit`	initial density in the interior	1
$\rho_{1/2}$	`rhoinner`	average density at inner boundary	1.e2
$\rho_{N_R+1/2}$	`rhoouter`	average density at outer boundary	1
$R_{\min}$	`Rmin`	radius of inner boundary	0.25
$R_{\max}$	`Rmax`	radius of outer boundary	30
		gravitational potential	
p	`pp`	power for gravitational density as a function of radius	-1.8
c/a	`axs`	ratio of short z- to long x-axis	0.5
b/a	`axi`	ratio of intermediate y- to long x-axis	0.8
ω	`om`	frame rotation rate	0.1
	`cutoff`	0: no cut-off; 1: bar cut-off at co-rotation; 2: cut-off at the Outer Lindblad Resonance	1
	`ii`	power for cut-off function	10
		numerical parameters	
n_i	`ni`	initial mesh has n_i^2 cells for domain $[R_{\min}, R_{\max}] \times [0, \pi)$	8
n_f	`nf`	final mesh has n_f^2 cells; n_i and n_f should be powers of 2	256
κ	`kappa`	R grid is equidistant in $\xi = R^{1+\kappa\, p/2}$	1
	`order`	spatial discretisation of order 1 or 2	2
		iterative scheme	
	`idtfactor`	scaling factor for $1/\Delta t$	1
	`relchange`	maximum relative change in solution per step	0.9
	`nstep`	maximum number of Newton iterations per successive grid refinement level	4000
	`nsave`	number of iterations between saving intermediate results	50
	`norderswitch`	switch from 1st to 2nd order when the number of grid cells in one coordinate equals `norderswitch`	64
	`resfactor1`	desired residual reduction for 1st order when 2nd order is needed	1.e$-$8
	`resfactor2`	desired residual reduction for 2nd order scheme	1.e$-$12
	`print`	extra printout for values ≥ 0	-1

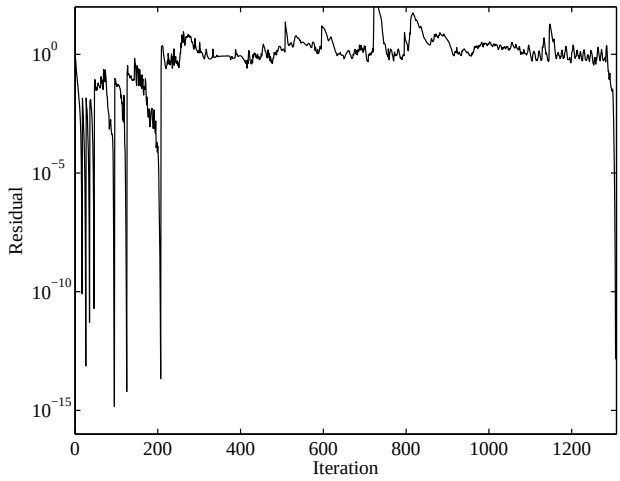


Figure 3: Convergence history. Note the quadratic convergence once the solution has been reached.

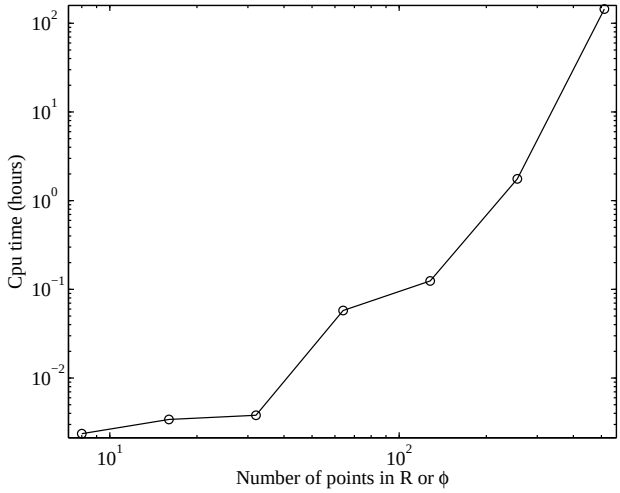


Figure 4: Performance. Shown is the compute time in hours required to obtain the solutions on subsequently finer grids, from 8×8 to 512×512, doubling the number of cells in each coordinate at each refinement. On a grid with 64×64 cells, the solution for a first-order discretisation was computed first and used as an initial guess for a second-order discretisation. Only the latter was used on the finer meshes.

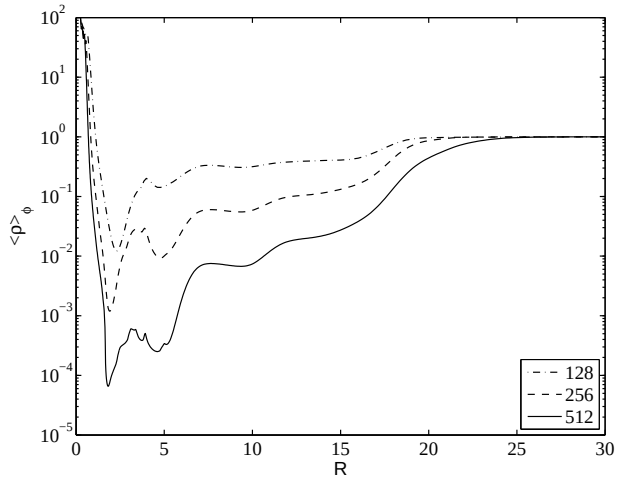


Figure 5: Average of density over ϕ as a function of radius on a sequence of grids with 128^2, 256^2 and 512^2 cells, respectively. The density contrast increases under grid refinement.

a grid with 16×16 cells and the stationary solution now required 8 iterations. This procedure was continued up to a grid with 64×64, where the first-order stationary solution, obtained after 10 iterations, was used as an initial guess for a second-order discretisation. Convergence of the latter required another 48 iterations. The successive grid refinement with the second-order scheme was continued to the grid with 512×512 cells. At that level, however, the number of required iterations jumped to the order of thousand. Note that in all case, quadratic convergence was obtained when the solution approached the stationary one.

Figure 4 shows the cpu time in hours needed to compute the solution on the sequence of grids. The solver used 8 threads on a 3rd generation 2.6 GHz quadcore Intel i7-3720QM (Ivy Bridge). Beyond 256×256, the code takes too long to be of practical use.

4. Discussion

The code is fairly robust up to meshes of size 256×256, both with a first-order and a second-order spatial discretisation. With 512×512 cells, convergence became extremely slow. This may have several causes, related either to the properties of the solution or to the solver, or both. Typical causes for convergence problems with Newton's method related to the formulation of the problem are: lack of smoothness of the multi-dimensional function, oscillatory behaviour, a starting solution that is too far from the root of the equations, or non-existence or non-uniqueness of the stationary solution. The direct solver may also adversely affect convergence if it introduces too much numerical noise.

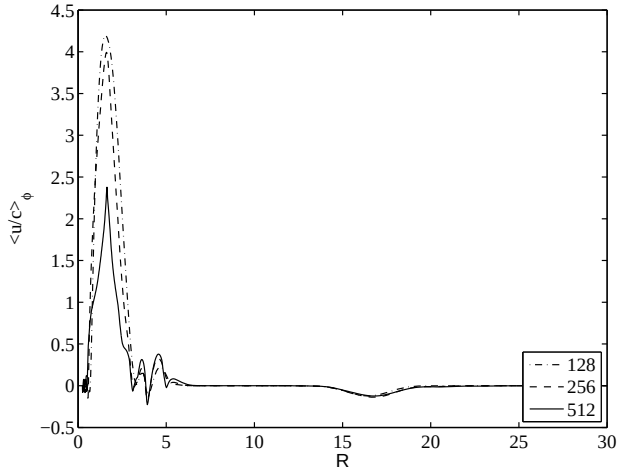


Figure 6: Average radial Mach number u/c as a function of radius.

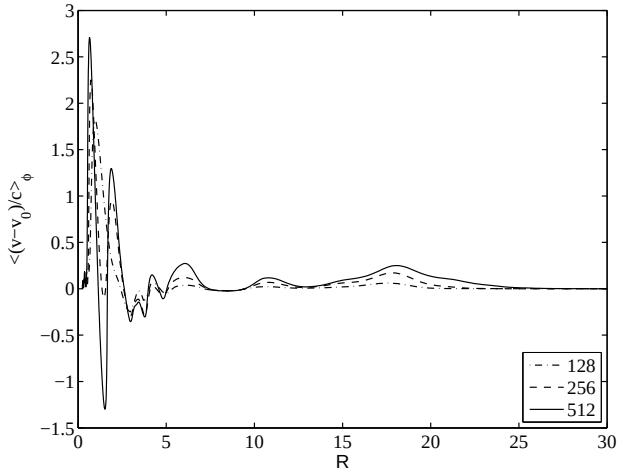


Figure 7: Average tangential velocity $(v - v_0)/c$ relative to the axisymmetric solution v_0.

As far as lack of smoothness is concerned: the code employs flux-vector splitting [12], which is inherently smooth, and a smooth limiter [1] for the linear interpolation within cells, needed to obtain second-order accuracy away from discontinuities. The successive refinement should provide a starting solution close to the stationary solution on the next finer grid. This will not work, however, if the solution does not converge to a unique one under grid refinement. Figures 5–7 display the average of the solution over the angle ϕ on subsequently finer grids. The asymptotic range, where the difference between subsequent meshes decreases with cell size, has clearly not been reached. The density contrasts become increasingly larger and this may pose a challenge for the direct solver. Also, the amount of detail in Figure 2 appears to increase under grid refinement, raising the question whether or not convergence under grid refinement will set in on finer meshes. If the amount of detail would keep increasing, the problem is ill-posed in a mathematical sense. In that case, some kind of regularisation may make it well-posed, for instance, by adding viscous terms, by dropping the isothermal assumption, by adding sources and sinks for the density or by going to three space dimensions. The limitations of the current code do not allow a further investigation into the question whether or not convergence under grid refinement can be attained. For practical purposes, a mesh size of 256×256 should very well suffice to generate a flow model for a given set of parameters in one or two hours.

5. Conclusions

A Matlab code for 2-D isothermal galactic gas dynamics in a weakly barred gravitational potential is capable of providing stationary solutions in a reasonable number of iterations for grid with 256 by 256 cells. On finer meshes, it becomes too slow for practical purposes because the number of required iterations becomes very large.

In practice, a grid with 256 by 256 cells should suffice for constructing galactic velocity models. The results can then be used for any of the applications described in the list of references. In addition, the code can serve educational purposes or be a starting point for further development.

References

[1] van Albada, G.D., van Leer, B., Roberts, W.W., 1982. A comparative study of computational methods in cosmic gas dynamics. Astronomy and Astrophysics 108, 76–84.

[2] Athanassoula, E., 1992. The existence and shapes of dust lanes in galactic bars. Monthly Notices of the Royal Astronomical Society 259, 345–364. doi:10.1093/mnras/259.2.345.

[3] Athanassoula, E., Bureau, M., 1999. Bar diagnostics in edge-on spiral galaxies. II. Hydrodynamical simulations. The Astrophysical Journal 522, 699–717. doi:10.1086/307677.

[4] Bissantz, N., Englmaier, P., Gerhard, O., 2003. Gas dynamics in the Milky Way: second pattern speed and large-scale morphology. Monthly Notices of the Royal Astronomical Society 340, 949–968. doi:10.1046/j.1365-8711.2003.06358.x.

[5] Combes, F., Gerin, M., 1985. Spiral structure of molecular clouds in response to bar forcing – A particle simulation. Astronomy and Astrophysics 150, 327–338.

[6] Davis, T.A., 2004. Algorithm 832: UMFPACK V4.3 – an Unsymmetric-pattern Multifrontal Method. ACM Transactions on Mathematical Software 30, 196–199. doi:10.1145/992200.992206.

[7] England, M.N., 1989. Dynamical models - The barred spiral galaxy NGC 1300. The Astrophysical Journal 344, 669–684. doi:10.1086/167833.

[8] Englmaier, P., Shlosman, I., 2000. Density waves inside the inner Lindblad resonance: Nuclear spirals in disk galaxies. The Astrophysical Journal 528, 677–686. doi:10.1086/308201.

[9] Fux, R., 1999. 3D self-consistent N-body barred models of the Milky Way. II. gas dynamics. Astronomy and Astrophysics 345, 787–812.

[10] Gingold, R.A., Monaghan, J.J., 1977. Smoothed particle hydrodynamics – Theory and application to non-spherical stars. Monthly Notices of the Royal Astronomical Society 181, 375–389. doi:10.1093/mnras/181.3.375.

[11] Lee, C.W., Lee, H.M., Ann, H.B., Kwon, K.H., 1999. Smoothed particle hydrodynamic simulations of galactic gaseous disk with bar: Distribution and kinematic structure of molecular clouds toward the galactic center. The Astrophysical Journal 513, 242–251. doi:10.1086/306846.

[12] van Leer, B., 1982. Flux-vector splitting for the Euler equations, in: Krause, E. (Ed.), Eighth International Conference on Numerical Methods in Fluid Dynamics. Springer Berlin Heidelberg. volume 170 of *Lecture Notes in Physics*, pp. 507–512. doi:10.1007/3-540-11948-5_66.

[13] van Leer, B., Mulder, W.A., 1985. Relaxation methods for hyperbolic conservation laws, in: Angrand, F., Dervieux, A., Desideri, J., Glowinski, R. (Eds.), Numerical Methods for the Euler Equations of Fluid Dynamics, SIAM, Philadelphia. pp. 312–333.

[14] Lin, L.H., Wang, H.H., Hsieh, P.Y., Taam, R.E., Yang, C.C., Yen, D.C.C., 2013. Hydrodynamical simulations of the barred spiral galaxy NGC 1097. The Astrophysical Journal 771, 8. doi:10.1088/0004-637X/771/1/8.

[15] Lina, L.H., Yuan, C., Buta, R., 2008. Hydrodynamical simulations of the barred spiral galaxy NGC 6782. The Astrophysical Journal 684, 1048–1061. doi:10.1086/590247.

[16] Lindblad, P.A.B., Kristen, H., 1996. Hydrodynamical simulations of the barred spiral galaxy ngc 1300. dynamical interpretation of observations. Astronomy and Astrophysics 313, 733–749.

[17] Lucy, L.B., 1977. A numerical approach to the testing of the fission hypothesis. Astronomical Journal 82, 1013–1024. doi:10.1086/112164.

[18] MATLAB, 2014. Version 8.3.0 (R2014a). The MathWorks Inc., Natick, Massachusetts.

[19] Mulder, W.A., 1986. Computation of the quasi-steady gas flow in a spiral galaxy by means of a multigrid method. Astronomy and Astrophysics 156, 354–380.

[20] Mulder, W.A., 2015. URL: `https://github.com/Someone789/galax2d.git`.

[21] Mulder, W.A., Hooimeyer, J.R.A., 1984. Periodic orbits in a rotating triaxial potential. Astronomy and Astrophysics 134, 158–170.

[22] Mulder, W.A., van Leer, B., 1985. Experiments with implicit upwind methods for the Euler equations. Journal of Computational Physics 59, 232–246. doi:10.1016/0021-9991(85)90144-5.

[23] Mulder, W.A., Liem, B.T., 1986. Construction of a global gas-dynamical model for our Galaxy. Astronomy and Astrophysics 157, 148–158.

[24] Patsis, P.A., Grosbol, P., Hiotelis, N., 1997. Interarm features in gaseous models of spiral galaxies. Astronomy and Astrophysics 323, 762–774.

[25] Rodriguez-Fernandez, N.J., Combes, F., 2008. Gas flow models in the Milky Way embedded bars. Astronomy and Astrophysics 489, 115–133. doi:10.1051/0004-6361:200809644.

[26] Sanders, R.H., Prendergast, K.H., 1974. The possible relation of the 3-kiloparsec arm to explosions in the galactic nucleus. The Astrophysical Journal 188, 489–500. doi:10.1086/152739.

[27] Sempere, M.J., Garcia-Burillo, S., Combes, F., Knapen, J.H., 1995. Determination of the pattern speed in the grand design spiral galaxy NGC 4321. Astronomy and Astrophysics 296, 45–63.

[28] Slyz, A.D., Kranz, T., Rix, H.W., 2003. Exploring spiral galaxy potentials with hydrodynamical simulations. Monthly Notices of the Royal Astronomical Society 346, 1162–1178. doi:10.1111/j.1365-2966.2003.07166.x.

[29] Stetter, H.J., 1978. The defect correction principle and discretization methods. Numerische Mathematik 29, 425–443. doi:10.1007/BF01432879.

[30] Weiner, B.J., Sellwood, J.A., 1999. The properties of the galactic bar implied by gas kinematics in the inner Milky Way. The Astrophysical Journal 524, 112–118. doi:10.1086/307786.