INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **28**: 1–22 (1998)

A METHOD FOR COMPUTING COMPRESSIBLE, HIGHLY STRATIFIED FLOWS IN ASTROPHYSICS BASED ON OPERATOR SPLITTING

A. HUJEIRAT[b] AND R. RANNACHER[a,*]

[a] *Institut für Angewandte Mathematik, Universität Heidelberg, D-69120 Heidelberg, Germany*
[b] *Astronomisches Institut der Universität Würzburg, D-97074 Würzburg-Am Hubland, Germany*

SUMMARY

A new numerical approach based on consistent operator splitting is presented for computing compressible, highly stratified flows in astrophysics. The algorithm is particularly designed to search for steady or almost steady solutions for the time-dependent Navier–Stokes equations, describing viscous flow under the influence of a strong gravitational field. The algorithm proposed is multidimensional and works in Cartesian, cylindrical or spherical co-ordinates. It uses a second-order finite volume scheme with third-order upwinding and a second-order time discretization. An adaptive time step control and monotonic multilevel grid distribution has been incorporated to speed up convergence. This method has been incorporated into a hydrodynamical code by which, for the first time, for two-dimensional models the dynamics of the boundary layer in the accretion disk around a compact star could be computed over the whole viscous time scale. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: accretion disk; compressible Navier–Stokes equations; stratified flow; operator splitting; hydrodynamic code; boundary layer

1. INTRODUCTION

Most astrophysical fluid flows are highly turbulent, compressible and time-dependent. They are also viscous, therefore, part of their kinetic energy is converted into heat and part of this heat is transported away via radiation. To simulate the thermo- and the hydrostructure of the occurring phenomena, multidimensional radiation-hydrodynamical codes are needed. A prototype problem is the formation of steady state or almost steady state structures in the boundary layer between a rapidly rotating accretion disk which surrounds a highly compressed and slowly rotating white dwarf (cf. Figure 1). If the secondary 'normal' star is found in an expansion phase of evolution, the outflowing material from the outer layers will soon reach the first turning point of the equi-potential envelope of the binary, and will be captured afterwards by the gravitational pull of the companion. However, as this captured material possesses angular momentum, the Coriolis forces and viscosity will prevent this matter from falling freely onto the white dwarf, and alternatively, an accretion disk will evolve. Astrophysically, the structure of the boundary layer is of great importance for understanding the short and long time evolution of the accretion disk and possibly the mechanism leading to novae explosions (see e.g. References [1–3] for a detailed description of the physical backgrounds).

* Correspondence to: Institut für Angewandte Mathematik, Universität Heidelberg, D-69120 Heidelberg, Germany.

CCC 0271–2091/98/110001–22$17.50
Received 28 December 1995
Revised 29 March 1997

In modeling such complex phenomena, there are various physical and numerical difficulties. Although viscosity plays an essential role, it is rather small, so that its effects are felt only on very long time scales. In fact, this viscosity arises from modeling turbulent momentum transport, while the molecular viscosity of the gas is extremely small, so that normally this type of flow would not be considered as viscous at all. Physically consistent initial and boundary conditions are mostly unknown. Enlarging the computational domain and carrying the calculations out on the viscous time scale may reduce the numerical artefacts resulting from imposing non-consistent boundary conditions and may also ensure dissolution of artificial viscous perturbations that might be generated initially.

Until recently, only 2D explicit time-dependent calculations were performed to simulate viscous accretion flow. Therefore, unless viscosity was taken to be very large, steady structures could not be reached. Additional limitations were due to a poor spatial resolution, a constant viscosity prescription, the relatively short physical time scale, and many other simplifications. This situation made it necessary to develop a new numerical approach. If steady or quasi-steady viscous solutions are of interest, the use of explicit methods is prohibitive because of the long viscous time scale and the extremely small time steps enforced by stability restrictions. Furthermore, in view of the large number of time steps required even within a fully implicit scheme, most commonly used direct and iterative solution methods also appear to be inefficient.

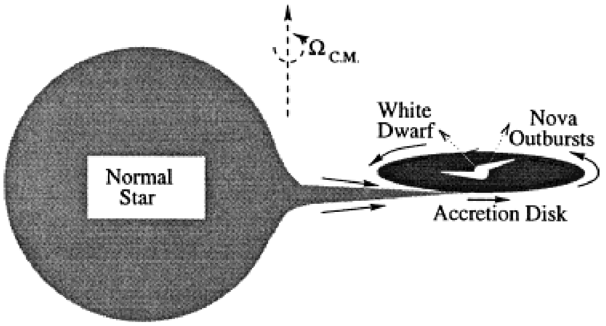


Figure 1. The accretion phenomenon around a compact star.

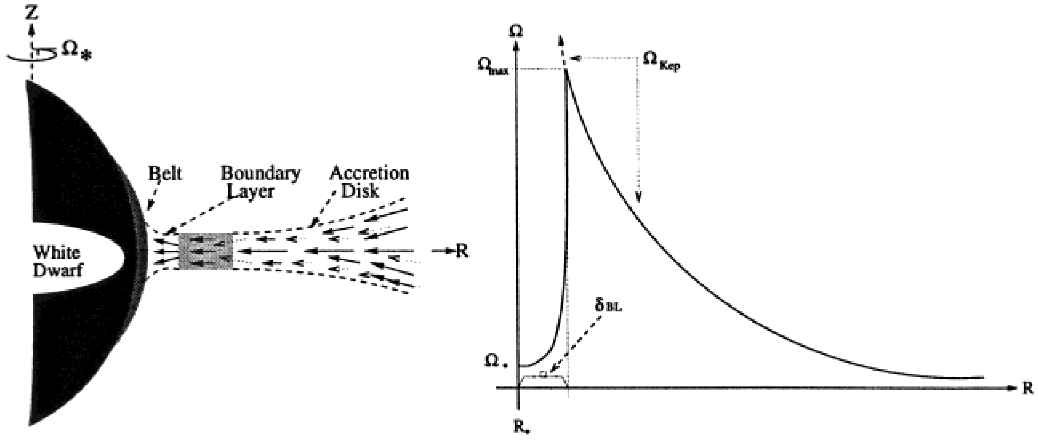


Figure 2. The geometrical location of the boundary layer.

In this paper, a new numerical algorithm is presented to address these problems. This algorithm has been incorporated into a 2D hydrodynamical code which has been successfully applied to the boundary layer problem (see References [4,5]). Our approach is based on the well-known concept of consistent physical and dimensional splitting which, however, has to be applied with care in solving the underlying problem. The discretization is at least second-order-accurate in time and space while preserving stability and efficiency of the numerical code. This approach has proven to be successful in dealing with the various inherent difficulties of the physical problem: the strong source terms, the extreme variations in density and velocity and the different relevant time scales. The method has been tested on several one- and two-dimensional problems. It proved to be accurate, stable and capable of capturing strongly time-dependent flows with areas of strong recirculation and those of only weak compressibility. What seems even more important, is that the resulting code fully vectorizes and is efficient enough to carry out long time simulations even on standard workstations. Below, the essential features of the underlying algorithm will be discussed in more detail.

2. THE MATHEMATICAL MODEL

A treatment of the full astrophysical problem described above is out of reach at present-day computational methods. To make the problem tractable, significant simplifications are necessary:

- The 3D flow is assumed to possess axi-symmetry.
- Magnetic effects and chemical processes are neglected.
- The heating and cooling of matter is modeled in the limit of the diffusion approximation.

In this case, using, e.g. cylindrical co-ordinates centered at the source of gravitation, the non-dimensional compressible 3D axi-symmetric time-dependent Navier–Stokes equations take the following form:

1. Continuity equation:

$$\partial_t \rho + \epsilon r^{-1} \partial_r (r\rho u) + \epsilon \partial_z (\rho v) = 0.$$

2. Radial momentum equation:

$$\begin{aligned} &\partial_t(\rho u) + \epsilon r^{-1}\partial_r(r\rho u^2) + \epsilon\partial_z(\rho u v) \\ &= -\epsilon\partial_r P + \epsilon^{-1}\rho r(\Omega^2 - \bar{r}^{-3}) + \epsilon r^{-1}\partial_r r\sigma_{rr} + \epsilon\partial_z\sigma_{rz} - \epsilon r^{-1}\sigma_{\phi\phi}. \end{aligned}$$

3. Vertical momentum equation:

$$\partial_t(\rho v) + \epsilon r^{-1}\partial_r(r\rho u v) + \epsilon\partial_z(\rho v^2) = -\epsilon\partial_z P - \epsilon^{-1}\rho z\bar{r}^{-3} + \epsilon r^{-1}\partial_r r\sigma_{rz} + \epsilon\partial_z\sigma_{zz}.$$

4. Angular momentum equation:

$$\partial_t(r\rho v_\phi) + \epsilon r^{-1}\partial_r(r^2\rho u v_\phi) + \epsilon\partial_z(r\rho v v_\phi) = \epsilon r^{-1}\partial_r(r^2\sigma_{r\phi}) + \epsilon\partial_z(r\sigma_{\phi z}).$$

5. Energy equation:

$$\begin{aligned} &\partial_t(\rho\mathscr{E}) + \epsilon r^{-1}\partial_r(r\rho u\mathscr{E}) + \epsilon\partial_z(\rho v\mathscr{E}) \\ &= -\epsilon P(r^{-1}\partial_r(ru) + \partial_z v) + \epsilon f_{\mathrm{diff}} r^{-1}\partial_r(r\breve{\kappa}\partial_r T) + \epsilon f_{\mathrm{diff}}\partial_z(\breve{\kappa}\partial_z T) + \epsilon f_\Phi \Phi. \end{aligned}$$

Here,

$$\mathscr{E} = \frac{C_V T}{\gamma}, \qquad P = \frac{\rho T}{\gamma}, \qquad \breve{\kappa} = \frac{\lambda_f \xi T^3}{(\kappa + \sigma)\rho}, \qquad \lambda^f = \frac{6}{6 + 3R_f + R_f^2},$$

$$R_f = \frac{2.12 \times 10^{-4}}{(\kappa + \sigma)\rho} \frac{|\nabla T^4|}{T^4},$$

where f_{diff}, f_Φ are switch on/off coefficients. The parameter

$$\epsilon = \frac{\text{Typical Sound Speed}}{\text{Keplerian Speed}}$$

is used to define the relevant time scale in the model. For the meaning of λ^f and R_f see, e.g. References [4,6]. The value $\bar{r} = (r^2 + z^2)^{1/2}$ is set, while all other symbols have their usual meaning. The components of the stress tensor are:

$$\sigma_{rr} = 2\eta \partial_r u - \tfrac{2}{3}\eta \nabla \cdot \vec{V}, \quad \sigma_{zz} = 2\eta \partial_z v - \tfrac{2}{3}\eta \nabla \cdot \vec{V}, \quad \sigma_{\phi\phi} = 2\eta r^{-1} u - \tfrac{2}{3}\eta \nabla \cdot \vec{V},$$

$$\sigma_{r\phi} = \eta r \partial_r \Omega, \qquad \sigma_{rz} = \eta(\partial_r v + \partial_z u), \qquad \sigma_{z\phi} = \eta r \partial_z \Omega.$$

Further, Φ has the form $\Phi = 2\rho \nu_{\text{tur}} \langle D_{rr}^2 + D_{zz}^2 + D_{\phi\phi}^2 + 2D_{rz}^2 + 2D_{r\phi}^2 + 2D_{z\phi}^2 \rangle - (2/3)\eta(\nabla \cdot \vec{V})^2$. The turbulent viscosity ν_{tur} is defined by

$$\nu_{\text{tur}} = \alpha V_s H_h,$$

where H_h is the harmonic mean of the disk thickness and the radial pressure scale height, and $\alpha \in [0.1, 1]$ is a parameter. The elements of the deformation tensor are

$$D_{rr} = \partial_r u, \qquad D_{zz} = \partial_z v, \qquad D_{\phi\phi} = r^{-1} u,$$

$$D_{rz} = \tfrac{1}{2}(\partial_r v + \partial_z u), \quad D_{r\phi} = \tfrac{1}{2} r \partial_r \Omega, \quad D_{z\phi} = \tfrac{1}{2} r \partial_r \Omega,$$

where u, v, and v_φ $(= r\Omega)$ are the r, z, and φ components of the velocity vector $\vec{V}$.

Instead of the cylindrical co-ordinates, Cartesian or spherical co-ordinates could have been used. The computational approach described below applies in analogous ways to all three cases.

Here it is assumed $\epsilon \approx 10^{-2}$, which is a relevant value when accretion onto compact objects is considered. In this case, the flow field described by the above equations may exhibit the following characteristics (see Figure 2):

- The Mach number can vary in the range $0 \leq \mathscr{M} \leq 100$.
- The source terms are about $\epsilon^{-2} \approx 10^4$ times larger than the advection and viscous terms. Therefore, the flow is highly turbulent and perturbations may easily be amplified generating a dynamically unstable flow. Also, the density waves may easily steepen into shock waves.
- The internal energy in the outer flow region is about four orders of magnitude less than the kinetic energy, i.e. the internal energy could be of the same order or less than the truncation error of the numerical method and negative temperatures may thus occur.
- The order of the non-dimensional viscous time scale is $\mathscr{T}_{\text{visc}} \approx 10^4$, while the dynamical time scale is only about $\mathscr{T}_{\text{dyn}} \approx 10^{-2}$. Therefore, in order to resolve the flow structure on the viscous time scale, several thousand time steps are required even for strongly implicit methods.
- The flow may undergo rapid spatial changes in its macro-structure, i.e. from incompressible to pressure-free to highly compressed (by shocks) and to hydrostatic equilibrium. Therefore, the numerical method is expected to deal with four types of fluids: highly viscous flow,

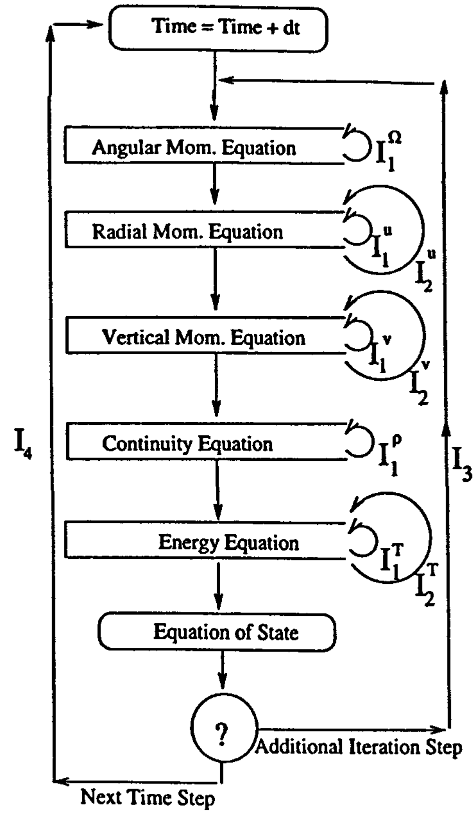


Figure 3. The operator splitting scheme.

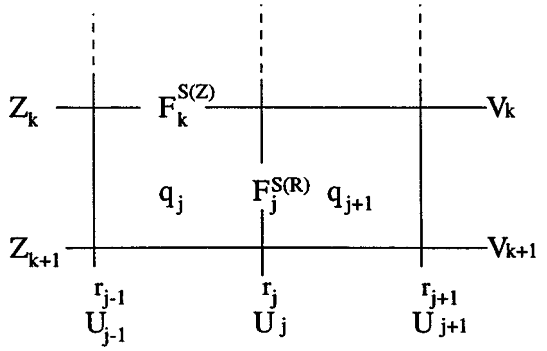


Figure 4. The staggered grid finite volume discretization.

highly supersonic flow with shock formation, corona type flow with circulation, and quasi-hydrostatic flow.

- Consistent initial conditions are not known, and also some of the boundary conditions have to be artificially prescribed.

In view of these facts, even a qualitative simulation of such flows is a challenge for numerical methods. In order to judge the reliability of any computational results, a careful study of the dependence of the solution obtained with respect to the various discretization parameters is necessary. This requires frequent recalculations on different levels of resolution, which leads to prohibitive CPU times unless the code is highly efficient.

3. THE NUMERICAL ALGORITHM

The requirements that a numerical algorithm should meet are now summarized, in order to simulate correctly those complex viscous flows as described above:

- Strongly refined grids with grid spacing varying by four to six orders of magnitude.
- Conservative treatment of all equations.
- At least second-order discretization in space and time.
- Robust and highly efficient numerical solver in each (implicit) time step.

In the following, the steps of the approach towards satisfying these conditions are described in detail. The main problem is to ensure that the computed trajectory starting from an inconsistent initial value finally enters an attracting neighborhood of the true steady state or almost steady state exact solution.

A consistent advection of a 2D flow requires a full coupling of the velocity field to the density. However, all other quantities enter as source terms and instabilities that may arise from viscous decoupling act on a longer time scale than the dynamical one. Therefore, operator splitting is suggested for an efficient approach to solve this problem. As this concept of 'physical splitting' is the key part of our method, we begin with the discussion of this approach for the compressible Navier–Stokes equations in the situation considered. The other ingredients such as spatial discretization, time discretization, treatment of convective terms, and viscosity prescription are rather standard and will be described afterwards. It is emphasized that due to the physical conditions, the time stepping must be implicit, particularly when a steady state limit is to be calculated. However, in the applications considered, it is not clear at the beginning whether the solution will become stationary or almost stationary. Hence, the algorithm must be capable of working efficiently over very long intervals of time.

3.1. Physical splitting

The key idea in this approach is to solve the set of equations successively, in a Gauss–Seidel-like manner, according to a natural physical ordering. Each time step is split into several substeps. Within each substep one equation is solved after the other and the resulting solution is used to update the corresponding variables in the next equations. Thus, each time a sparse matrix corresponding to a scalar problem with only a few entries per line has to be inverted, the complexity and huge memory requirements associated with a direct or indirect inversion of the large block-matrices appearing in the fully coupled approach are avoided. This process is embedded into a defect correction iteration, which allows the control of the effect of decoupling by monitoring the global residual in each time step. The whole process may be viewed as a type of non-linear block-Gauss–Seidel method for solving the fully coupled problem. This method is schematically shown in Figure 3, where I_4 indicates the time stepping, I_3 the defect iteration for the fully coupled system, I_2 the non-linear iteration in each substep, and I_1 the inner linear solver. In the simplest case of a Cartesian tensor product mesh, the linear problems may be solved by dimensional splitting, which reduces the total work to solving a sequence of only point-tridiagonal systems.

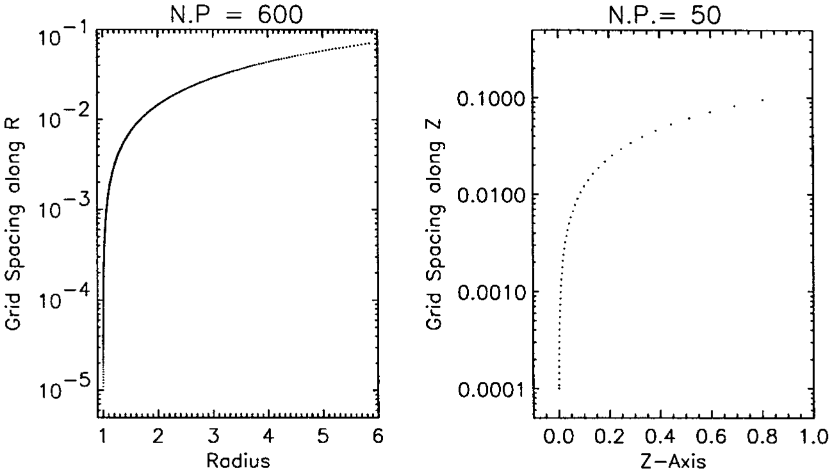


Figure 5. Distribution of grid points in the 1D and 2D computations.

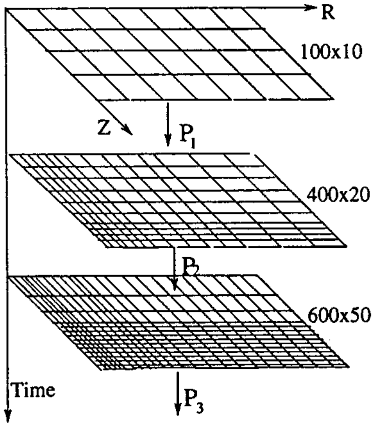


Figure 6. Sequence of meshes within the multilevel time stepping.

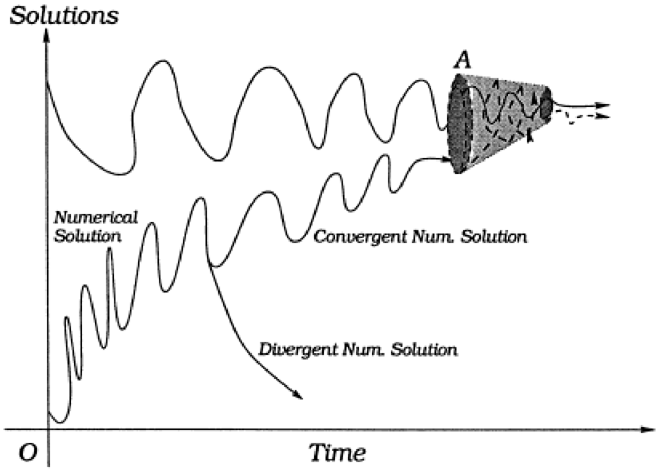


Figure 7. The transient solution process.

In practice, the successful application of this splitting method depends on a physically consistent ordering of the equations and on a careful conservative treatment. This allows the use of very moderate time step sizes which may be several orders of magnitude larger than the corresponding ones required in a fully explicit time stepping process. The only limitation is due to accuracy requirements and to ensuring convergence of the inner non-linear iterations. The possible destabilizing effect of the physical decoupling may be controlled by increasing the number of global defect correction steps within each time step. This issue will be discussed in more detail below.

An essential feature of the physical splitting approach is the reformulation of the set of equations in conservative form in terms of the conservation variables ρ, $m(=\rho u)$, $n(=\rho v)$, $\ell(=\rho r^2\Omega)$, $\mathscr{E}(=\rho e)$. It has been observed in these calculations (e.g. forward facing step) that using the conservative variables in the context of finite volume philosophy enlarges the range of convergence, enhances mathematical consistency, and yields more accurate solutions than otherwise possible. Moreover, using conservative variables reduces the solution procedure of the continuity equation down to only an algebraic equation, and thus considerably reduces the overall computational work.

The ideas underlying the order of the physical splitting for the type of problem considered is now discussed in more detail. To be physically consistent, two important aspects have to be taken into account: causality and hierarchy. Causality, means whether the variation of an individual variable has a direct effect on the main variable of the next equation. For example, solving the continuity equation after the energy equation is meaningless, because ρ and T are thermo- and hydrodynamically-independent, or equivalently, any change of T cannot cause a change in ρ without going through the velocity field. The same reasoning applies for ρ and Ω due to the axi-symmetric assumption made above. In any case, computing the temperature requires an updating of all variables involved in the system. Therefore, it is reasonable to solve for T as the final step of the solution cycle in each time step. A straightforward conclusion is to solve the continuity equation prior to the energy equation. On the other hand, the order in which the momentum equations ought to be solved is determined by the so-called hierarchy strategy. Its construction is purely dependent on the physical processes considered, because it requires the different time scales involved to be investigated, the rate of variation of each variable as well as the way in which the non-linear terms of the equations can be recovered.

For example, applying this strategy to the 1D shock tube problem, the consistent order appears to be straightforward: (I) the momentum equation ⇒ (II) continuity equation ⇒ (III) energy equation.

Similarly, the equations describing a 1D polytropic gas accreted onto a compact object may be solved in the following order: (I) angular momentum equation ⇒ (II) radial momentum equation ⇒ (III) continuity equation. This ordering is not only physically consistent, but also yields convergence of the numerical solution process, provided that the non-linearity is recovered by higher-order extrapolation in time or by a Newton-type iteration.

Extending this approach to the 2D configuration, convergence was observed whenever the equations were solved in the following order: (I) angular momentum equation ⇒ (II) radial momentum equation ⇒ (III)vertical momentum equation ⇒ (IV) continuity equation ⇒ (V) energy equation. This ordering is consistent with the evolution of the physical processes involved in a thin accretion disk:

$$\begin{pmatrix}\text{change}\\ \text{in }\Omega\\ \text{due to } v\end{pmatrix} \Rightarrow \begin{pmatrix}\delta\Omega\text{ causes}\\ \text{drifts in}\\ u\text{ and }v\end{pmatrix} \Rightarrow \begin{pmatrix}\delta u, \delta v\text{ cause}\\ \text{changes}\\ \text{in }\rho\end{pmatrix} \Rightarrow \begin{pmatrix}\delta\Omega, \delta u, \delta v, \delta\rho\\ \text{cause change}\\ \text{in energy }\mathscr{E}\end{pmatrix}.$$

Test calculations have shown that, for stability reasons, the angular momentum equation must come first. Otherwise, time accuracy requires Ω to be extrapolated, generating (relatively) small errors in v_φ which, since $v_\varphi \gg u$, cause large errors in the whole system, unless the time step is taken to be unacceptably small. Thus, our ordering in the physical splitting follows the general rule, well-known from error propagation analysis, that ill-conditioned operations should be ordered before well-conditioned ones.

If the physical solution is strongly time-dependent, the operator splitting may introduce inherent errors and the numerical solution may diverge whenever the time step size exceeds a certain limit. This effect can be controlled either by choosing the time step sufficiently small or by increasing the number of outer coupling iteration steps I_3. In the solution procedure described below the inversion of each submatrix is relatively cheap, therefore, two to four global iterations in each time step are easily affordable, while the step size is still many orders of magnitude larger than that of a corresponding explicit scheme. In practice, the number of global iterations I_3 may adaptively be adjusted according to how small the time-independent part of the global residual is. It reduces to one when the solution approaches steady state, but it may be up to five for strongly non-stationary flows.

3.2. Time discretization and time step control

The operator splitting described above may be combined with any of the usual time stepping schemes. However, if time-accurate simulations are to be carried over the viscous time scale, only schemes of at least second-order accuracy should be used. A dampened version of the Crank–Nicolson scheme, which is known to have relatively low numerical dissipation, was chosen for this study. To determine appropriate time step sizes, an adaptive process based on estimating the local truncation error within a predictor-corrector scheme was used. This approach is rather standard, therefore it is described briefly.

Let the vector variable q be a solution of the differential equation

$$\frac{\mathrm{d}q}{\mathrm{d}t} = F(t, q).$$

Following Rannacher [7] and Heywood and Rannacher [8], a modified Crank–Nicolson scheme is applied, and defined as:

$$\frac{q^{n+1} - q^n}{\delta t} - \theta F^{n+1} = (1-\theta)F^n,$$

where $\theta = (1 + \alpha\delta t)/2$ and $F^n = F(t_n, q^n)$. The term $\alpha\delta t/2$ has been added to enhance the dampening property of the Crank–Nicolson scheme, while still preserving its second-order-accuracy.

Having calculated F^n, a second-order predictor-corrector scheme can be constructed without much additional work. An explicit Adams–Bashford step is taken as predictor

$$\frac{q_{\text{exp}}^{n+1} - q^n}{\delta t} = \frac{3}{2}F^n - \frac{1}{2}F^{n-1},$$

which costs only some more storage. The step size may preliminarily be chosen as $\delta t = \delta_{n-1}$. With the exact solution $q^{n+1} = q(t_{n+1})$ at time level $t_{n+1} = t_n + \delta t$, we then have

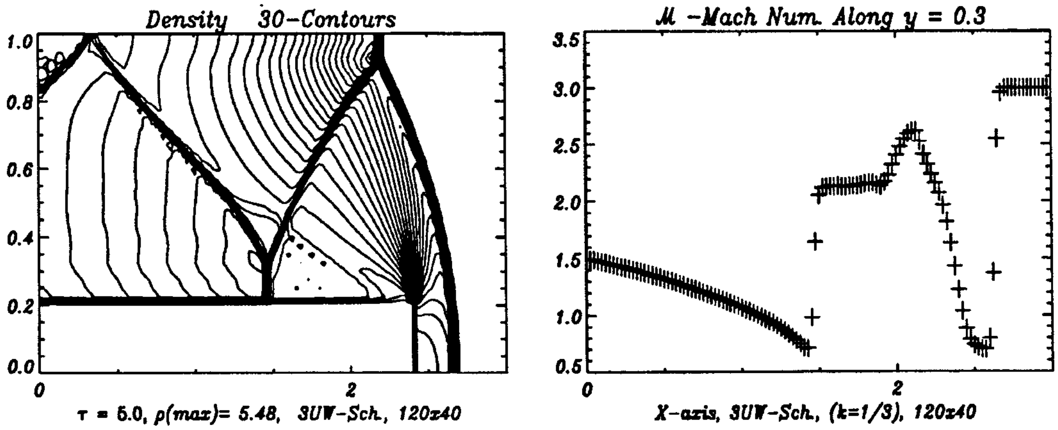


Figure 8. The solution of the forward facing step problem.

$$q_{\exp}^{n+1} - q^{n+1} = C_1 q_{tt} \delta t^2 + O(\delta t^3),$$

where $C_1 = 5/12$, for the Adams–Bashford scheme. Furthermore,

$$q_{\mathrm{imp}}^{n+1} - q^{n+1} = C_2 q_{tt} \delta t^2 + O(\delta t^3),$$

where $C_2 = -1/12$, for the Crank–Nicolson scheme. Here, as usual in estimating the local truncation error, the previous error at time t_n is assumed to be negligible. Thus, eliminating q^{n+1}, an estimate for the derivative term q_{tt} is obtained:

$$q_{tt} = \frac{q_{\mathrm{imp}}^{n+1} - q_{\exp}^{n+1}}{C_2 - C_1} \frac{1}{\delta t^2} =: \frac{\mathrm{Rel}}{\delta t^2}.$$

Prescribing now the relative error in each time step to be bounded by some small tolerance ϵ, the required time step size $\tilde{\delta t}$ is predicted from the relation

$$\frac{\| q_{\mathrm{imp}}^{n+1} - q^{n+1} \|}{\| q_{\mathrm{imp}}^{n+1} \|} \approx C_2 \frac{\mathrm{Rel}}{\tilde{\delta t}^2} \tilde{\delta t}^2 \frac{1}{\| q_{\mathrm{imp}}^{n+1} \|} \leq \epsilon$$

as

$$\tilde{\delta t} \leq \delta t \sqrt{\frac{\epsilon \, \| q_{\mathrm{imp}}^{n+1} \|}{C_2 \mathrm{Rel}}}.$$

Then, the new time step size is usually chosen according to

$$\delta t_n = \begin{cases} \delta t \times \sqrt{2} & \text{if} \quad \tilde{\delta t} > \delta t \\ \delta t / \sqrt{2} & \text{if} \quad \tilde{\delta t} \leq \delta t \end{cases}.$$

3.3. Spatial discretization

To be consistent with the conservative formulation of the Navier–Stokes equations, a (staggered grid) finite volume formulation is adopted. The scalar quantities are located at the cell center while the fluxes live at the cell surface. For the present problem, density, pressure, temperature, viscosity and any combination of them are defined at the cell center, while u, v and v_φ are defined at the cell surface (cf. Figure 4). Furthermore, to avoid numerical instabilities that may result from non-consistent interpolation of the pressure and angular velocity in steep gradient regions, the cell shifting method is used in the evaluation of the momentum equations. For the problems under consideration, which are mostly posed on rectangular domains, the choice of (not necessarily uniform) Cartesian tensor-product meshes is most natural.

In calculating the advective terms, the MUSCL scheme is adopted within the context of the finite volume philosophy, to determine the fluxes on the cell surfaces. The radial flux, for example reads:

$$F_j^{S(R)} = U_{|r=r_j}\vec{q} = U_j \begin{cases} q_{j+1} + \dfrac{\alpha_j}{4}[(1-k\alpha_j)\Delta^+ + (1+k\alpha_j)\Delta^-]q_{j+1} & \text{if } U_j \le 0 \\ q_j + \dfrac{\alpha_{j+1}}{4}[(1+k\alpha_{j+1})\Delta^+ + (1-k\alpha_{j+1})\Delta^-]q_j & \text{otherwise} \end{cases},$$

where

$$\Delta^+ q_j = q_{j+1} - q_j, \qquad \Delta^- q_j = q_j - q_{j-1}, \qquad \alpha_j = \frac{2\Delta^+ q_j \Delta^- q_j}{(\Delta^+ q_j)^2 + (\Delta^- q_j)^2 + \epsilon}.$$

The spatial accuracy obviously depends on the parameter k. For instance, $k = -1, 0, 1/3, 1$ correspond to fully upwinded, symmetric, third-order bias, and to central advection schemes, respectively. In these calculations, $\alpha = 0$ for the transient solution phase, and $k = 1/3$ in the final phase on the finest grid level. It should be noted that the computational work required for second-order-accuracy is only 40% higher than that for the low-order scheme. Hence, the use of higher order discretization using the cell-shifting method within the context of finite volumes is strongly suggested.

3.4. *Artificial viscosity*

A modified second-order version of the Von Neumann artificial viscosity is adopted here. The underlying idea is that flow regions which are in homogeneous contraction or expansion phase are free of any kind of artificial viscosity. Only in regions with real physical compression such as shock fronts, is artificial viscosity acting (shock capturing). A similar approach has already been developed by Tscharnuter and Winkler [9]. Furthermore, the artificial viscosity is only applied to the radial and vertical momentum equations, thus avoiding artificial transport of angular momentum. The coefficient of the artificial viscosity can then be formulated as follows:

$$\nu_{\text{art}} = \begin{cases} \alpha\alpha_a l_a^2(\nabla\cdot\vec{V}) & \text{if } \nabla\cdot\vec{V} \le 0 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where l_a was set to δr_j in the one-dimensional calculations and to 0.01 in the 2D case, $\alpha_a > 0$ is a constant parameter of order one, and α is a viscosity parameter. Thus, the total viscosity coefficient acting in the time-implicit solution process is the sum of a real physical and an artificial viscosity coefficient, $\nu_{\text{tot}} = \nu_{\text{tur}} + \nu_{\text{art}}$.

3.5. *Grid generation*

In generating the grid distribution, two aspects must be taken into account: first, the physical forces acting on the flow confine most of the dynamics to regions near the equator. Second, the geometrical location of the boundary layer is well defined. Thus, plane-parallel motions are expected to govern the flow in the region of interest. Consequently, a tensor-product grid distribution is used where the finest cells are located at the equator and near the star surface, while they gradually grow in size outwards in both directions. Specifically, given the required sizes of the finest grids, $\delta r_{\min}$ and $\delta z_{\min}$, the prefixed numbers JR and KZ of the grid points in both directions and the absolute global length $\Delta R_{\text{in}}^{\text{out}}$ $(=R^{\text{out}}-R^{\text{in}})$, $\Delta Z_{\text{in}}^{\text{out}}$ $(=Z^{\text{out}}-Z^{\text{in}})$, it is assumed that $\delta r_j/\delta r_{j+1}$ and $\delta z_k/\delta z_{k+1}$ have constant outgrowing ratios ξ_r and ξ_z in the r- and z-directions, respectively. Their values are determined by solving the following two algebraic equations:

$$\xi_r^{JR-1}+(\xi_r-1)\frac{\Delta R_{\text{in}}^{\text{out}}}{\delta r_{\min}}-1=0,$$

$$\xi_z^{KZ-1}+(\xi_z-1)\frac{\Delta Z_{\text{in}}^{\text{out}}}{\delta z_{\min}}-1=0.$$

These equations are solved by a simple fixed point iteration to an accuracy of order 10^{-14}. Once ξ_r and ξ_z are found, all other grid spacings are uniquely determined. In Figure 5, the distributions of the grid points in the one- and two-dimensional calculations are plotted.

3.6. *A modified dimensional splitting*

The approximate factorization method of Beam and Warming [10] is used to split the two-dimensional set of equations into two one-dimensional sets of equations. This approximates the inversion of a generally pentadiagonal matrix by two successive inversions of tridiagonal matrices. To enhance accuracy, this process may be iterated by defect correction. However, the flow field is not necessarily isotropic, and therefore it has preferable directions that are determined by the net forces on the right-hand-side of each particular equation. The effective gravity and the pressure gradient at the right-hand-side of the radial momentum equation determine how fast the flow in the radial directions must be. Consequently, this equation is solved in the radial direction first and then vertically. Similar arguments also apply to the vertical momentum equations. Determining the preferable directions in the other equations is problem-dependent. For example, for accretion disk problems, the main flow direction is the radial one, while most of the energy is transported vertically. Thus, it is reasonable to solve the continuity and the angular momentum equations radially while the energy equation is solved in the z-direction, followed by the radial one.

The following describes how the physical and the space splitting are applied to the 2D boundary layer problem:

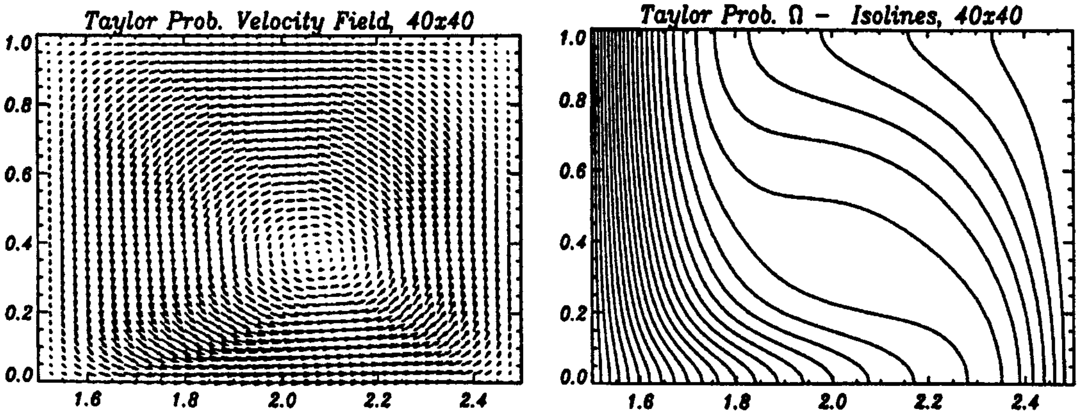


Figure 9. The solution of the Taylor problem.

- In the first step, the angular momentum equation is solved:
  1. Radial: $\{I+\delta t[L_r(u^{\text{ex}})+L_{rr}(v^{\text{ex}})]\}\ell^*=\ell^n$,
  2. Vertical: $\{I+\delta t[L_z(u^{\text{ex}})+L_{zz}(v^{\text{ex}})]\}\ell^{n+1}=\ell^*$,
- In the second step, the radial momentum equation is solved:
  1. Radial: $\{I+\delta t[L_r(u^{\text{ex}})+L_{rr}(v^{\text{ex}})]\}m^*=m^n+\delta t[L_{rz}(v^{\text{ex}}n^{\text{ex}})+f^u_{\text{ex}}(\ell^{n+1},\vec{r})]$,
  2. Vertical: $\{I+\delta t[L_z(u^*)+L_{zz}(v^{\text{ex}})]\}m^{n+1}=m^*$.
- In the third step, the vertical momentum equation is solved:
  1. Vertical: $\{I+\delta t[L_z(v^{\text{ex}})+L_{zz}(v^{\text{ex}})]\}n^*=n^n+\delta t[L_{rz}(v^{\text{ex}}m^{n+1})+f^v_{\text{ex}}(\vec{z})]$,
  2. Radial: $\{I+\delta t[L_r(u^{n+1})+L_{rr}(v^{\text{ex}})]\}n^{n+1}=n^*$.
- In the fourth step, the continuity equation is solved:
  1. Radial: $\{I+\delta t L_r(u^{n+1})\}\rho^*=\rho^n$,
  2. Vertical: $\{I+\delta t L_z(v^{n+1})\}\rho^{n+1}=\rho^*$.
- In the fifth step, the energy equation is solved:
  1. Vertical: $\{I+\delta t[L_z(v^{n+1})+L_{zz}(f(\rho,\kappa,T)^{n+1,\text{ex}})]\}\mathscr{E}^*=\mathscr{E}^n+\delta t[\Phi(\rho,u,v,\Omega)^{n+1}+f_T^{n+1}(\vec{z})]$,
  2. Radial: $\{I+\delta t[L_r(u^{n+1})+L_{rr}(f(\rho,\kappa,T)^{n+1,*})]\}\mathscr{E}^{n+1}=\mathscr{E}^*$.

Here, ‘ex’ denotes time-extrapolated values.

3.7. The combined solution procedure

The structure of the solution process described above is now summarized. Within each single time step, a set of fully coupled non-linear algebraic equations $A_{rz}(q)q=b$ for a vector-variable q must be solved. This is done the following nested iteration:

- Apply physical splitting, resulting in a sequence of equations

$$\begin{cases} A^1_{rz}q_1=b_1\\ A^2_{rz}q_2=b_2\\ \quad\cdot\\ \quad\cdot\\ \quad\cdot\\ A^L_{rz}q_L=b_L, \end{cases}$$

  and proceed in each substep, for $l=1,\ldots,L$, as follows.
- Linearize the scalar equations to $\tilde{A}^l_{rz}q_l=b_l$.
- Construct the appropriate preconditioning for each matrix $\tilde{A}^l_{rz}$, i.e.:

$$\tilde{A}^l_{rz}\rightarrow\tilde{A}^l_r\tilde{A}^l_z \quad\text{or}\quad \tilde{A}^l_z\tilde{A}^l_r.$$

- Use all the available values of the variables to compute the (non-linear) defect

$$d_l=:b_l-A^l_{rz}q^i_l.$$

- Solve for the correction μ^i_l:

$$\tilde{A}^l_r\tilde{A}^l_z\mu^i_l=d_l\Rightarrow\mu^i_l.$$

- Update q^{i+1}_l:

$$q^{i+1}_l=q^i_l+\mu^i_l.$$

- Perform additional global iteration steps to enhance the coupling, or proceed with the next time step.

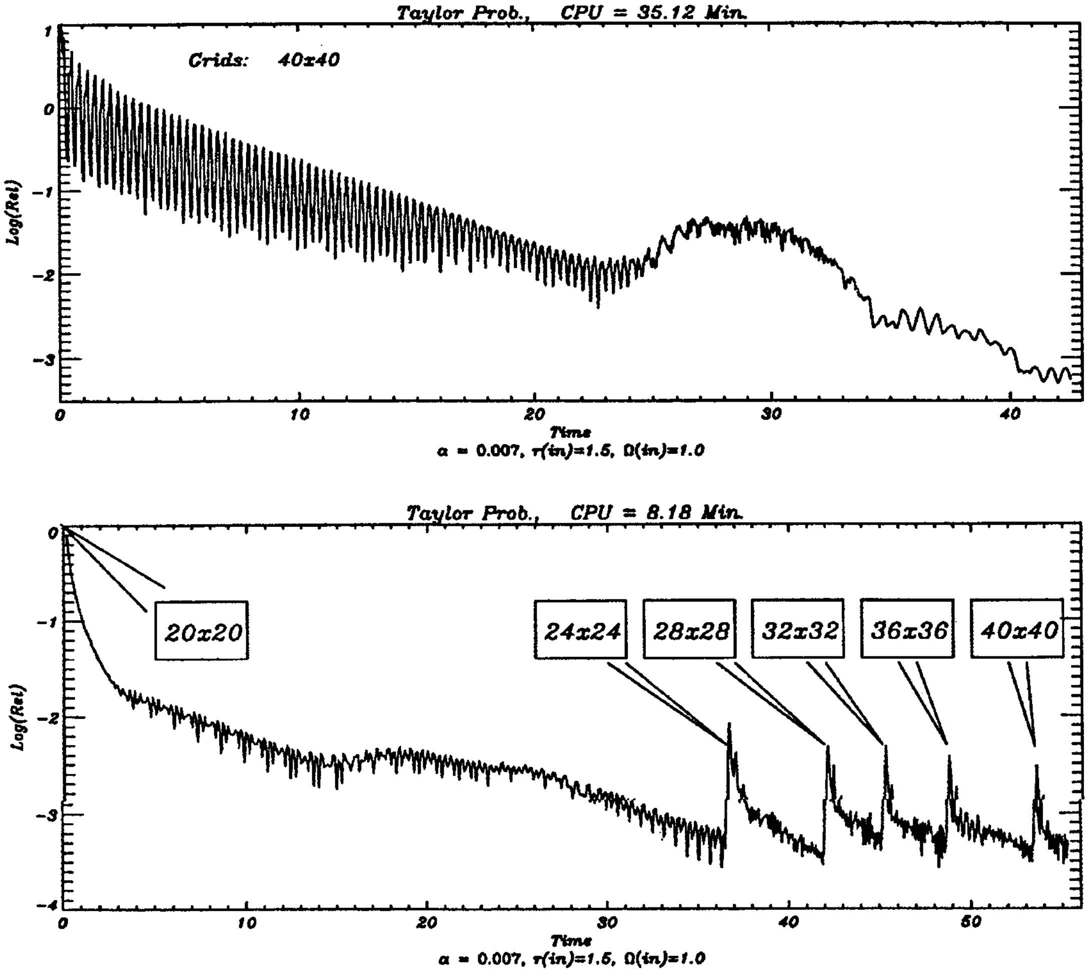


Figure 10. The convergence history of the time stepping for the Taylor problem.

To accelerate convergence to the limit state, stationary or possibly time-periodic, a monotonic nested grid strategy (multilevel time stepping) has been adopted. The calculations are started on the coarsest grid level. Then, having the corresponding solution satisfy some given criteria that examine whether the numerical solution has entered a steady or quasi-steady state phase, the solution is prolongated on the next finer grid. This procedure is repeated until the solution on the finest grid is obtained (see Figure 6).

Since the distribution of the grid points on each level is independently constructed, the ratio of grid size from one level to the next is not necessarily 1/2, as would be typical in standard multigrid algorithms. Therefore, in order to reduce the error generated by the prolongation process, it is essential to use at least a second-order interpolation scheme (e.g. bilinear interpolation). The prolongation is organized in a directional splitting-like way, which in these tests, appeared to increase the convergence property of the scheme. To clarify the idea, suppose there are three hierarchical grid levels (10 × 10), (30 × 30) and (80 × 80). The directional splitting prolongation procedure is then:

$$(10 \times 10) \rightarrow (30 \times 10) \rightarrow (80 \times 10) \rightarrow (80 \times 30) \rightarrow (80 \times 80).$$

Furthermore, it was found that in some test cases convergence was increased whenever a fast and successive prolongation was applied. The idea here is to carry out the calculations on the coarsest grid with only first-order discretization in space and time until the prolongation criterion is fulfilled. The solution is then prolongated and carried only over a few time steps on the next grid level. This fast prolongation is repeated until the finest grid level is reached, where the higher order of space and time discretization is switched on, and the coupling iteration I_3 is accordingly increased.

Having elucidated the main ingredients of the numerical algorithm, the practical evolution of the calculations applied to the various problems may now be described. For a given starting value O (see Figure 7) the aim is to obtain, with minimal numerical effort, a numerical solution within a certain neighborhood A of the real solution. In practice, this transient solution from O to A is obtained by using only a first-order discretization in space and time, with moderate time step size δt and with a minimum number of iterations within each time step. The starting mesh and δt are chosen to ensure convergence according to given physical criteria. Once the numerical solution has entered region A, the multilevel prolongation procedure can be applied, higher-order discretization in space and time are switched on, and the number of local and global iterations within each time step can be increased. By this adapted solution strategy the CPU time for obtaining a physically relevant solution can be drastically reduced (e.g. down to 5% for the accretion disk problem) compared with the time required to obtain the same solution when the calculation is carried from the beginning on the finest grid level with second-order-accuracy in space and time.

4. APPLICATIONS

In this section, the tests which have been performed in the development of the code at several model problems are reported and finally some computational results for the 2D accretion disk problem are presented. Most of the computations were carried out on a workstation SUN Sparc-10/20, which has a measured performance of about 6 MFlops (double precision).

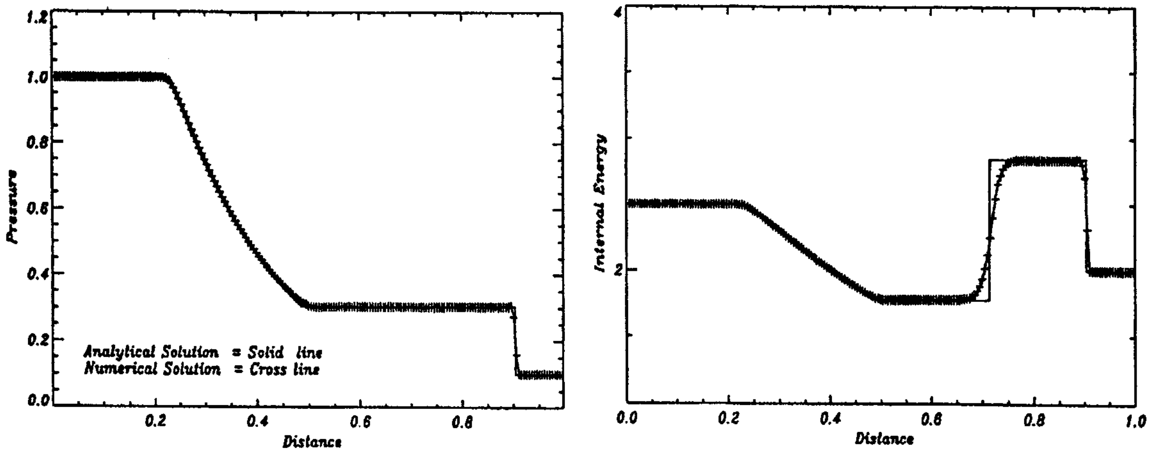


Figure 11. The solution of the shock tube problem.

In order to solve the accretion disk problem, the code must be capable of accurately resolving stationary as well as non-stationary areas of steep gradients. Furthermore, it has to cope with flows with moderate viscosity as well as with those with almost vanishing viscosity. Finally, it must be robust with respect to large variations of the Mach number, from highly supersonic to nearly incompressible ($\mathscr{M} \approx 0$). To examine these properties, the code has been tested on a series of standard model problems. In all calculations it has been assured by systematic variation of the discretization parameters that the time and space discretization is in the range of second-order accuracy. It is emphasized that these test cases (Tests 1–4) are not meant to be examples for typical applications of the code, as much more accurate special-purpose solvers are available for each particular problem. The main goal of the code is the solution of more complex problems (Problems 1–2) presented at the end.

4.1. Test 1: Euler flow over a forward facing step

This is a standard test problem for 2D high resolution Euler codes. The stationary solution develops a non-trivial shock structure, which can efficiently be computed by explicit schemes. Here, the aim is to test whether the fully implicit code can also accurately produce this solution in acceptable time, despite the fact that it uses a Cartesian mesh which is not well adapted to the structure of the shocks. An additionally aim is to examine how well the scheme advects a non-linear wave, if the quantities are actually conserved and if the artificial viscosity provides the correct jump conditions. Figure 8 shows the density distribution obtained on a mesh with 120×40 cells using a third-order upwind advection scheme combined with the MUSCL scheme, with $k = 1/3$ and for an inflow Mach number $\mathscr{M} = 3$. The quality of the shock resolution compares well with that reported (e.g. Reference [11]) for a uniform mesh with 240×80 cells.

4.2. Test 2: Taylor flow

The 2D flow between two concentric counter-clockwise rotating cylinders (Taylor problem) is a classical test problem for codes designed for solving the incompressible Navier–Stokes equations. For moderate Reynolds numbers, there is a stationary solution which exhibits the well-known Taylor cells. Here, the aim is to test whether the code is robust enough to work in the incompressible limit case as well, i.e. for almost vanishing $\mathscr{M}$. Additionally, correct transport of angular momentum has to be justified. Furthermore, the strong rotational structure of the solution may pose problems for algebraic solvers being based on dimensional splitting. The calculation has been performed for a test case with $Re = 500$ and $\mathscr{M} \approx 0.02$. The solution (one cell) shown in Figure 9 has been obtained on a uniform mesh with 200×40 points (five Taylor cells) and with a CPU time of 900 s. This compares well with standard theoretical and experimental results reported for this problem (e.g. Reference [12]). The convergence history towards steady state is displayed in Figure 10 for one cell, first on a fine mesh with 40×40 points and then on a sequence of successively refined meshes. The gain in CPU time is about a factor of 1/4.

4.3. Test 3: shock tube problem

The 1D gas flow in a shock tube is a standard test case for non-stationary Euler solvers. Here, the aim is to test the accuracy, mainly in time of our algorithm to capture time-dependent solutions, particularly moving shocks. In Figure 11 the results obtained on a uniform mesh with 200 cells compared with the analytical solution is plotted. The obtained accuracy

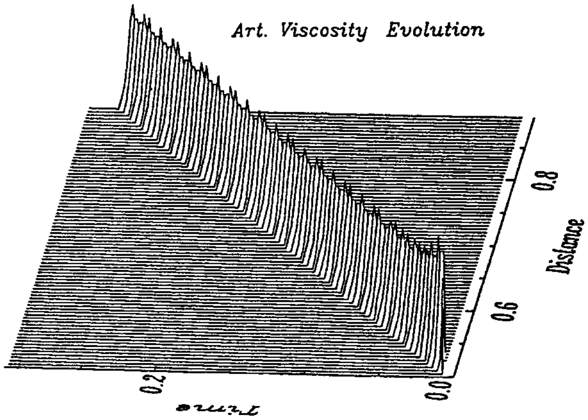


Figure 12. The development of the artificial viscosity in the shock tube problem.*

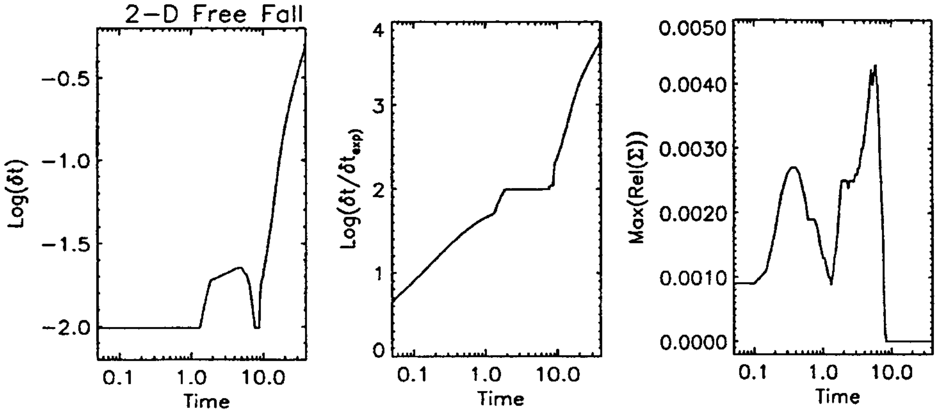


Figure 13. Development of the time step size in the '2D free fall problem'.

compares well with that reported (e.g. Reference [13]). Figure 12 shows the advancement of the artificial viscosity (shock capturing term) in time and space. As the time step size is relatively small, it appears that recovering the non-linearity by a time-extrapolation procedure provides better accuracy than extending the non-linear iteration I_3, thereby largely enhancing the efficiency of the scheme.

4.4. *Test 4: the 2D free fall problem*

The configuration of the 2D accretion disk problem is considered, but the Dirichlet condition is not imposed at the inner boundary. Then, under the influence of a high gravitational force, the flow undergoes a strong radial acceleration but no accumulation of matter occurs. Therefore, whatever enters the domain through the right boundary, leaves it on the left, and thus a steady state structure of the flow is expected. The purpose of this test is to examine the robustness of the code and the performance of the automatic step size control which should let the time step size grow when the flow converges to complete steady state. As can be seen from Figure 13, the time step size actually grows exponentially with time when the flow approaches steady state, becoming almost four orders of magnitude larger than that admissible in an explicit scheme.

4.5. Application 1: the 1D boundary layer problem

Averaging the cylindrical Navier–Stokes equations in the z-direction, a one-dimensional model is obtained for the accretion disk flow, which may be simplified further by assuming a

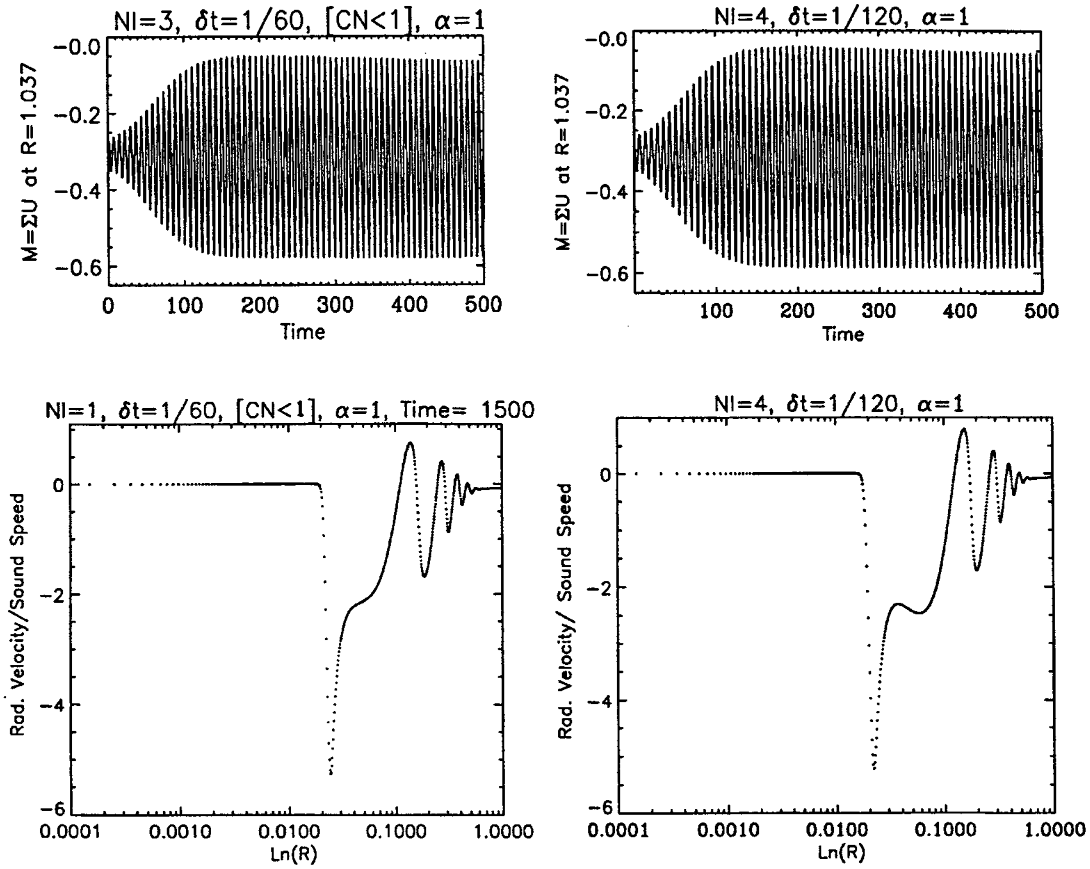


Figure 14. The time variation of $\mathscr{M}$ and the spatial distribution of the radial Mach number for the cases $\delta t = 1/60$ and 1/120.

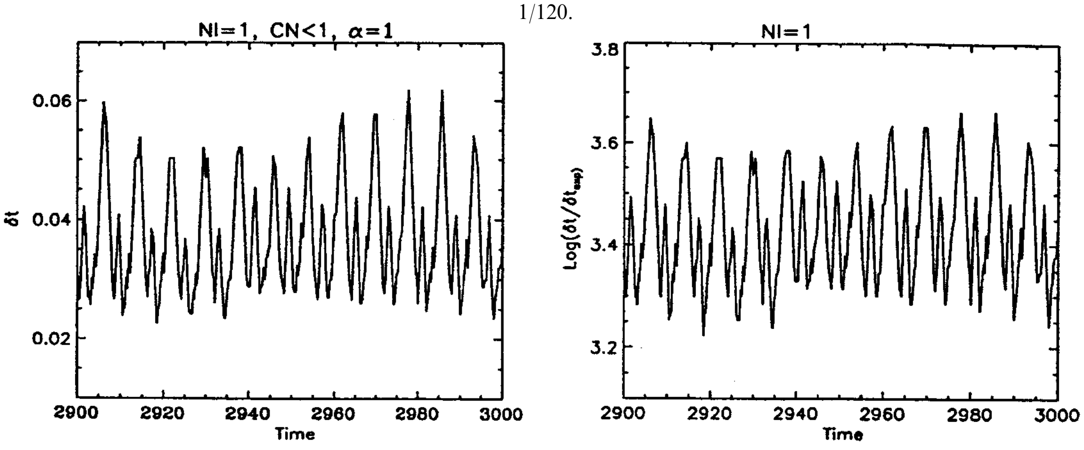
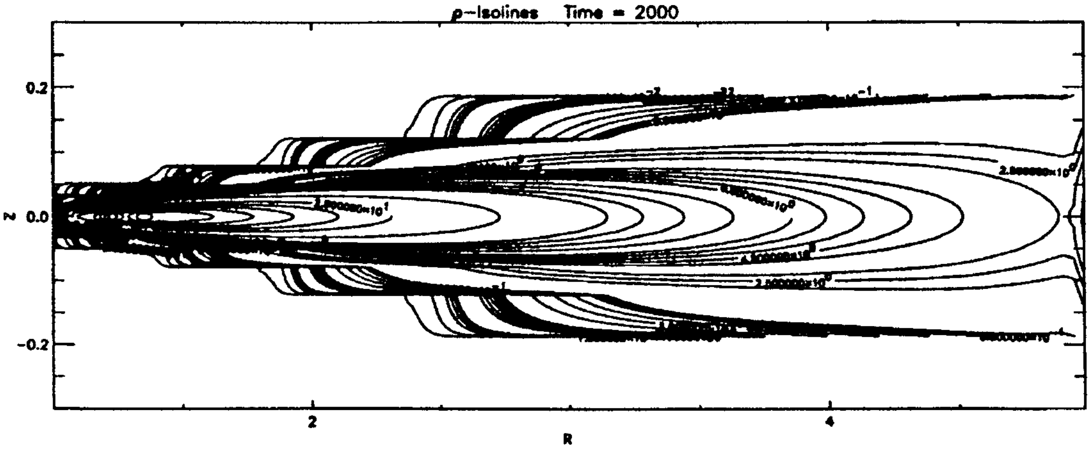


Figure 15. Development of the time step size δt and the ratio $\delta t/\delta t_{\mathrm{exp}}$.



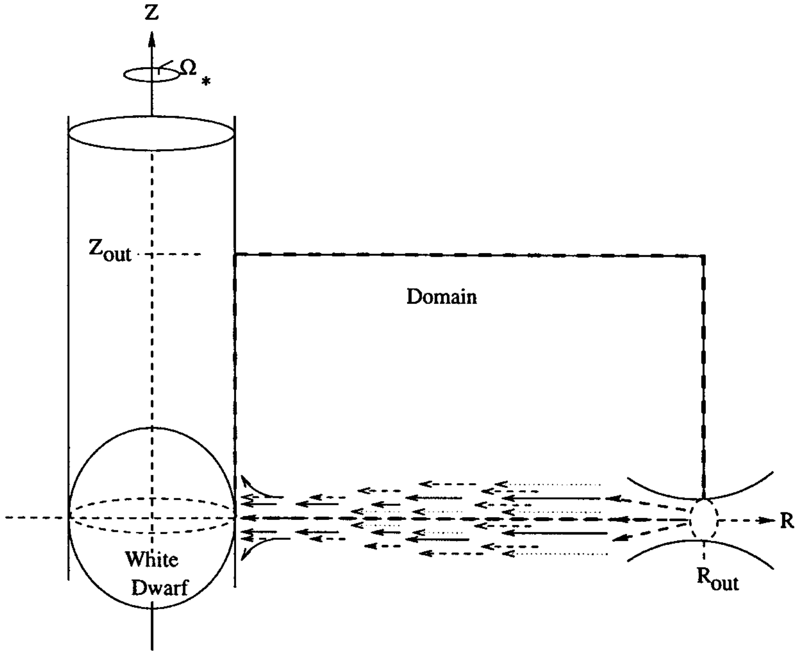


Figure 16. The geometry of the numerical domain of the 2D accretion disk calculations.

polytropic material behavior. This model is still rich enough to produce solutions with a strong oscillatory shock. The main purposes in this test are, (1) to demonstrate the capability of this method to capture a highly irregular flow on a highly non-uniform mesh, and (2) to examine the dependence of the numerical solution on the discretization parameters, mainly on the time step size. For a more detailed discussion of this model and the computational results see Reference [4]. In this problem, a numerical difficulty arises whenever the modified viscosity prescription is used (with $\alpha = 1$), as the viscosity drops in the boundary layer down to an extremely small value. Here, our method, as any other method, will become unstable.

However, this problem can be overcome if the grid size is reduced sufficiently to resolve the boundary layer. To this end, a strongly refined mesh was used, consisting of 800 points with grid spacing varying between 5×10^{-6} and 8×10^{-2}. To test the time dependence of the solution, two calculations were carried out, one with uniform time step size $\delta t = 1/60$ and the other with $\delta t = 1/120$. The corresponding results, the time variation of $\mathcal{M}$ and the spatial distribution of the radial Mach number, are displayed in Figure 14. These numerical solutions are independent of the numerical parameters, and the computed oscillations are physical. The performance of the implemented self-adaptive step size control is demonstrated in Figure 15, which shows the evolution of the time step size, as well as its size in units of the corresponding time step size admissible in an explicit scheme. Although the physical solution is strongly time-dependent, the achievable time step size is still three orders of magnitude larger than that of an explicit scheme.

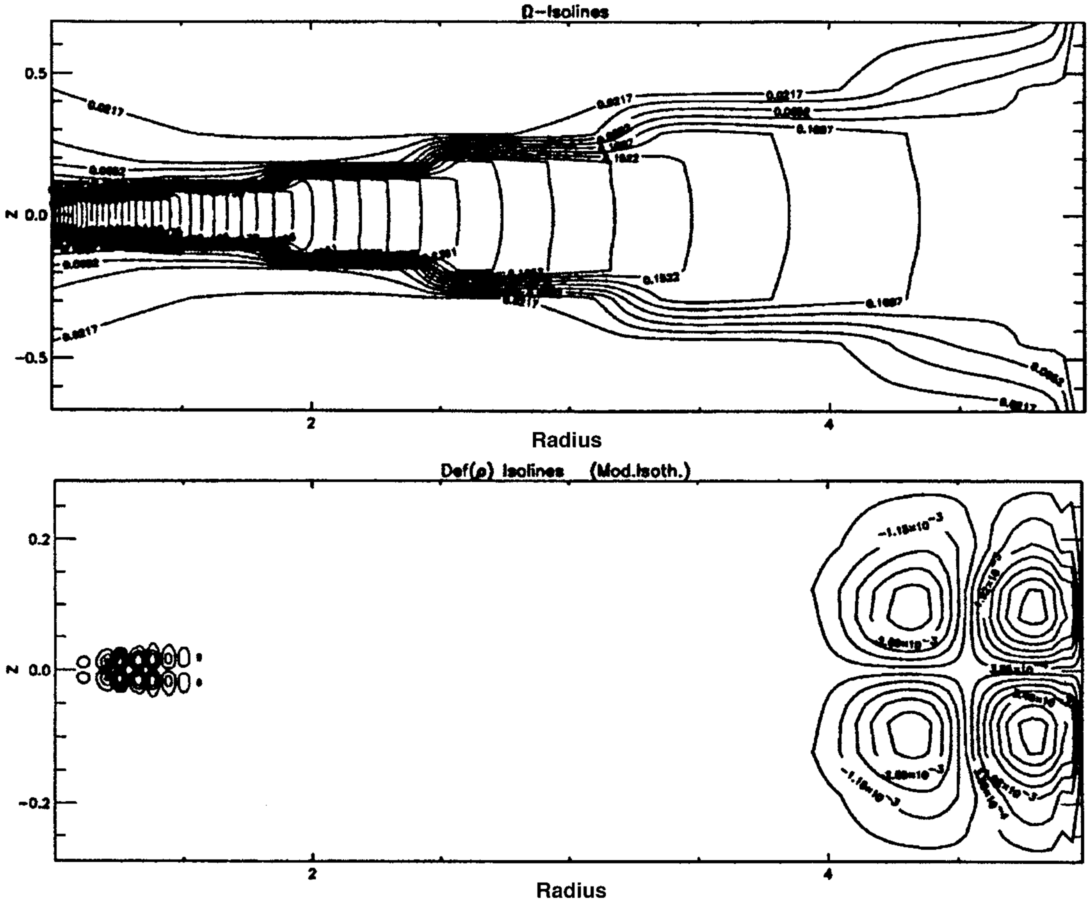


Figure 17. Isolines of density, angular velocity, and residual of the continuity equation for the 2D boundary layer computation.

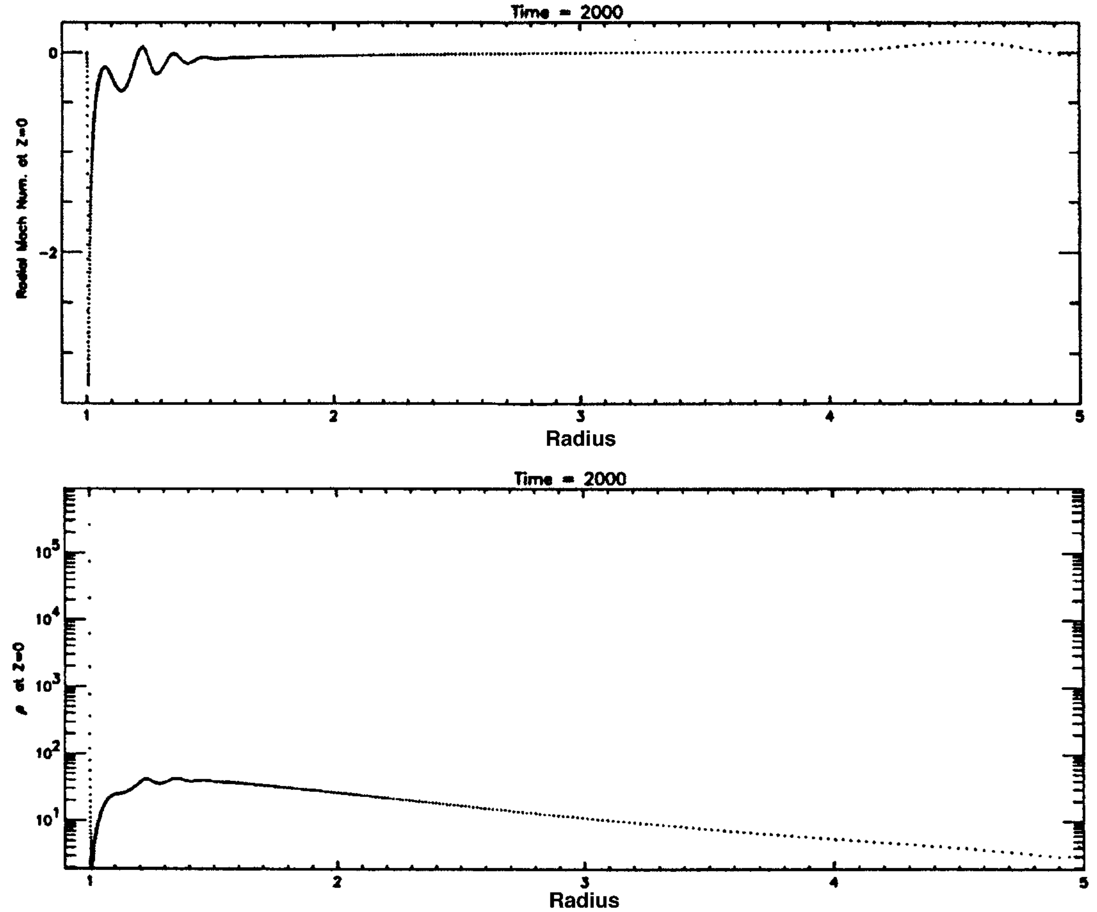


Figure 18. Radial Mach number and density along the equator in the 2D boundary layer problem.

4.6. Application 2: the 2D boundary layer problem

The main purpose in developing the described code was to solve the 2D boundary layer problem where all the numerical difficulties described above occur at the same time. Only some of the obtained results are displayed here (again with $\alpha = 1$). For their physical interpretation, see Reference [5].

The calculations were carried out on a finest mesh of 600 × 20 cells with grid spacing along the Z-axis ranging from 10^{-4} to 0.3. It took about 10 h CPU time on the SUN Sparc 10/20 to reach the quasi-steady state. Note that this is less than 5% of the time which would be required to achieve the same results without employing the self-adaptive time step control and the multilevel strategy. In Figures 16 and 17 the geometry of the computational domain and the resulting distributions of density, angular velocity, and defect of the continuity equation are displayed. Finally, Figure 18 shows the distribution of the radial Mach number and angular velocity along the equator. These 2D results are consistent with those of the 1D computation. Therefore, it seems physically meaningless to carry out 2D calculations without a more refined treatment of energy transfer by radiation. In fact, most of the dissipated energy in the boundary layer-accretion disk is expected to be transported away from the two sides of the disk. The implementation of a more realistic energy transfer model will be one of the next steps in the further development of this code.

5. SUMMARY

In this paper an efficient multidimensional numerical solver is presented for computing viscous, compressible, highly stratified flows. The underlying principle is that of a physically consistent physical and dimensional splitting. The resulting algorithm works well on various types of steady or strongly time varying flow problems, including inviscid flows associated with shock formation, highly viscous boundary layers, flows under high gravitational forces as well as almost incompressible fluids. Convergence is significantly enhanced by employing a multilevel time stepping approach combined with an automatic step size control. The method was implemented in a hydrodynamical code which fully vectorizes and allows the efficient computation of highly complex astrophysical flows over the full viscous time scale, even on workstations.

ACKNOWLEDGMENTS

This research has been supported by the Deutsche Forschungsgemeinschaft (DFG) through the SFB 359 'Reaktive Strömungen, Diffusion und Transport' at the University of Heidelberg. The first author also thanks the DFG for the financial support through the Schwerpunktprogramm 'Physik der Sternenentstehung' at Würzburg.

REFERENCES

1. G. Hensler, 'Hydrodynamical calculations of accretion disks in close binary systems, I, II', *Astron. Astrophys.*, **114**, 309–327 (1982).
2. J. Frank, A.R. King, D.J. Raine, *Accretion Power in Astrophysics*, Cambridge University Press, Cambridge, 1985.
3. M. Shara, 'Recent progress in understanding the eruptions of classical novae', *Publ. Astron. Soc. Pac.*, **101**, 5–31 (1989).
4. A. Hujeirat, 'Hydrodynamical calculations towards steady state structures in boundary layers in accretion disks, I. 1-D polytropic boundary layers', *J. Astron. Astrophys.*, **295**, 249–267 (1995).
5. A. Hujeirat, 'Hydrodynamical calculations towards steady state structures in boundary layers in accretion disks, II. 2-D models', *J. Astron. Astrophys.*, **295**, 268–288 (1995).
6. C.D. Levermore and G.C. Pomraning, 'A flux limited diffusion theory', *Astrophys. J.*, **248**, 321–334 (1981).
7. R. Rannacher, 'On the stabilization of the Crank–Nicolson scheme for long-time calculations', *Technical Report*, Universität Saarbrücken, 1986.
8. J.G. Heywood, and R. Rannacher, 'Finite-element approximation of the nonstationary Navier–Stokes problem, Part IV. Error analysis for second-order time descretization, *SIAM J. Numer. Anal.*, **27**, 353–384 (1990).
9. W. Tscharnuter and K.H.A. Winkler, 'A method for computing selfgravitating gas flow with radiation', *Comp. Phys. Commun.*, **18**, 171–199 (1978).
10. R.M. Beam and R.F. Warming, 'An implicit factored scheme for the compressible Navier–Stokes equations', *AIAA J.*, **16**, 393–402 (1978).
11. P.R. Woodward, and P. Collela, 'The numerical simulation of two-dimensional fluid flow with strong shocks', *J. Comput. Phys.*, **54**, 115–173 (1984).
12. R.C. DiPrima and H.L. Swinney, 'Instabilities and transition in flow between concentric rotating cylinders', in H.L. Swinney and J.P. Gollub (eds.), *Hydrodynamics and Tensions to Turbulence*, 2nd edn., Springer, New York, 1985, pp. 139–180.
13. G.A. Sod, 'A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws', *J. Comput. Phys.*, **27**, 1–17 (1978).