A&A manuscript no.
(will be inserted by hand later)
Your thesaurus codes are:
03(03.13.4; 02.13.2; 02.08.1; 02.19.1;)

ASTRONOMY AND ASTROPHYSICS

Implicit and semi-implicit schemes in the Versatile Advection Code: numerical tests*

G. Tóth[1], R. Keppens[2], and M. A. Botchev[3]**

[1] Astronomical Institute, Utrecht University, P.O. Box 80000, 3508 TA Utrecht, The Netherlands, toth@fys.ruu.nl
[2] FOM-Institute for Plasma-Physics Rijnhuizen, P.O. Box 1207, 3430 BE Nieuwegein, The Netherlands, keppens@rijnh.nl
[3] Mathematical Institute, Utrecht University, P.O. Box 80010, 3508 TA Utrecht, The Netherlands, botchev@math.ruu.nl

Received August 22, 1997/ Accepted

Abstract. We describe and evaluate various implicit and semi-implicit time integration schemes applied to the numerical simulation of hydrodynamical and magnetohydrodynamical problems. The schemes were implemented recently in the software package Versatile Advection Code, which uses modern shock capturing methods to solve systems of conservation laws with optional source terms. The main advantage of implicit solution strategies over explicit time integration is that the restrictive constraint on the allowed time step can be (partially) eliminated, thus the computational cost is reduced.

The test problems cover one and two dimensional, steady state and time accurate computations, and the solutions contain discontinuities. For each test, we confront explicit with implicit solution strategies.

Key words: methods: numerical – magnetohydrodynamics – hydrodynamics – shock waves

1. Introduction

The Versatile Advection Code (VAC) has been developed (Tóth 1996, 1997) as a general purpose software package for astrophysical numerical simulations[1]. VAC can solve the hydrodynamic (HD) and magnetohydrodynamic (MHD) equations with additional source terms in different geometries. Under the extreme conditions, astrophysical gases and plasmas can develop shock waves and other types of discontinuities, which require the use of high resolution shock capturing numerical schemes, thus total variation diminishing (TVD) and flux corrected transport (FCT) techniques are implemented in VAC. Recently, Keppens et al. (1997, hereafter Paper I) added a great variety of implicit and semi-implicit time integration schemes to the software package. These new time integration schemes are compatible with the various spatial discretizations and boundary conditions, and second order spatial and temporal accuracy can be maintained for the smooth part of the solutions.

In this paper we will show how the use of appropriate implicit techniques can dramatically reduce the computational cost by (partially) eliminating the restrictive stability constraint on the allowed time step. We demonstrate the obtainable efficiency for four astrophysically oriented numerical tests, where implicit and semi-implicit time integrations clearly outperform an explicit code. The newly introduced Minimum Residual Predictor-Corrector (MR-PC, Botchev et al. 1997) time integration scheme is tested on astrophysical problems for the first time.

The explicit approach can be very inefficient for the solution of steady state problems. The usual strategy is to solve the time dependent equations and "march" towards the steady state solution. The convergence, however, often stagnates and the physical time corresponding to the final solution may be huge, while the size of the explicit time step is limited by numerical stability conditions, therefore an excessive number of explicit time steps are needed. Even in time accurate calculations, it may occur that the variables evolve slowly relative to the numerically allowed time step. For example, even if the fast magnetosonic waves are not induced in an MHD simulation, it still restricts the time step. Another possibility is that a strongly elliptic source term, e.g. resistivity or viscosity, imposes a limiting stability condition due to the short diffusion time scale, while, in reality, the diffusion is balanced by other processes, and the flow evolves slowly.

The optimal solution strategy also depends on the dimensionality of the problem, thus the four selected tests described in Sect. 4 represent the four possible combinations of steady state versus time accurate and one dimensional versus two dimensional simulations. The tests are hydrodynamic and magnetohydrodynamic problems representative for astrophysical research. We model accretion onto a black hole, oscillations of a high density plasma sheet in a very low density surroundings, the formation of a steady bow shock, and reconnection of magnetic field lines. The test problems also demonstrate that the implicit schemes combined with high resolution spatial discretiza-

* This work was performed as part of the project 'Parallel Computational Magneto-Fluid Dynamics', funded by the Dutch Science Foundation (NWO) Priority Program on Massively Parallel Computing. See http://www.fys.ruu.nl/~mpr/.
** *Present address:* Department of Atomic Physics, Eötvös Loránd University, Puskin u. 5-7, Budapest 1088, Hungary
[1] See http://www.fys.ruu.nl/~toth/ for more information.

tion can handle discontinuities efficiently. All the problems addressed here are governed by the HD and MHD equations. Section 3 gives the conservation form for these equations and defines the additional source terms.

The spatial discretization schemes were described by Tóth and Odstrčil (1996, hereafter TO96), while the newly added implicit and semi-implicit time integration schemes are discussed in detail in Paper I. To make this paper self-contained, Sect. 2 summarizes the numerical schemes, the various options, their advantages and limitations, and we provide guidelines for selecting the optimal solution strategy.

2. Numerical schemes

VAC is a general purpose software package using modern high-resolution, shock-capturing schemes for solving equations of the following type

$$\partial_t \boldsymbol{U} = \boldsymbol{R}(\boldsymbol{U}) = -\sum_i \partial_i \boldsymbol{F}_i(\boldsymbol{U}) + \boldsymbol{S}(\boldsymbol{U}, \partial_i \boldsymbol{U}, \partial_i \partial_j \boldsymbol{U}, \boldsymbol{x}, t). \quad (1)$$

Such a system generally describes the evolution of conservative variables $\boldsymbol{U}(\boldsymbol{x}, t)$, where t is the time coordinate, i and j run over 1, 2 or 3 components of the spatial coordinate $\boldsymbol{x}$, and the evolution of $\boldsymbol{U}$ is governed by the right hand side $\boldsymbol{R}$ containing the source or sink terms $\boldsymbol{S}$ and the derivatives of the fluxes $\boldsymbol{F}_i$. In case of the MHD equations the conservative variables are mass, momentum, total energy density and the magnetic field components. Source terms may then represent external gravitational forces, or dissipative effects like viscosity, resistivity, and thermal conduction, among others.

2.1. Spatial discretization and boundary conditions

The conservative variables $\boldsymbol{U}(\boldsymbol{x}, t)$ are spatially discretized on a structured grid in a finite volume sense: for each cell we store the cell averaged values. Structured grids can be regarded as a continuous mapping from a regular mesh, they include uniform and non-uniform Cartesian and polar grids as special cases. VAC allows 1, 2, and 3 dimensional structured grids.

Boundary conditions are implemented in VAC by the use of two layers of *ghost cells* surrounding the computational domain. Before each step of the multi-step time integration algorithms, the ghost cells are updated based on $\boldsymbol{U}$ in the computational domain and on the type of the boundary condition represented by the ghost cell. The most frequently used boundary types are *fixed*, i.e. the ghost cells contain always the same value, *continuous* or zero-gradient, i.e. the ghost cells are copied from the nearest physical cell, *symmetric* or *anti-symmetric* across the boundaries, and *periodic*. Fixed and continuous boundaries are usually used for inflow and outflow, respectively. The symmetric boundary types can represent a solid wall as well as the physical symmetry of the problem, while periodic boundaries are useful for polar grids and for studying small scale or periodic phenomena in a theoretically infinite system.

2.2. Shock-capturing schemes

In this paper we will use three types of Total Variation Diminishing schemes, two of them, TVD (Harten 1983) and TVD-MUSCL (van Leer 1979), with a Roe-type Riemann solver (Roe 1981), and a Lax-Friedrichs type scheme TVDLF (Yee 1989, improved version in TO96). All these methods calculate smooth flows to second order accuracy, and are able to resolve discontinuities, but they differ in their diffusive and dispersive character near such discontinuities. See TO96 for a detailed description and comparison.

The one-step TVD method can sharply resolve discontinuities. The spatial and temporal second order accuracy is achieved by second order correction terms corresponding to a piece-wise linear approximation. The TVD property is ensured by applying limiters on the jumps allowed in each of the characteristic wave fields.

In the TVD-MUSCL and TVDLF methods one limits the slopes, obtained from the conservative variables $\boldsymbol{U}$, to achieve the TVD property close to discontinuities and sharp gradients, while second order spatial accuracy is maintained where the solution is smooth. TVD-MUSCL also makes use of the Riemann solver-based decomposition into characteristic wave fields, thus it has similar, maybe slightly worse, resolution than the TVD method. The TVDLF scheme does not solve the approximate Riemann problem, it only uses the fastest characteristic wave speed $c^{\max}$. Consequently TVDLF is computationally less expensive and somewhat more diffusive than the other two methods, but it is very robust.

For explicit time integration both TVD-MUSCL and TVDLF use a Hancock predictor step to ensure temporal second order accuracy. In the two-step schemes, it is easy to incorporate source terms with second order temporal accuracy, while the same is more difficult with the one-step TVD method.

There are several choices for the slope limiter functions as well. We will only use two, the more diffusive *minmod* limiter and the sharper *Woodward* limiter (Collela & Woodward 1984). As usual, the more diffusive limiter is more robust, it maintains positivity of thermal pressure and density better than the sharper limiters.

All three spatially second order schemes use a 5 point stencil in one dimension. In the implicit schemes we will also use the spatially first order accurate versions of TVD-MUSCL and TVDLF, these apply no limiters, and only the closest neighbouring cells are needed, which corresponds to a 3 point stencil in 1D. The first order TVDMU1 scheme is an upwind scheme for each characteristic variable, while the first order TVDLF1 method differs from the Lax-Friedrich scheme only in the coefficient of the diffusive term, which is $c^{\max}$ instead of the ratio of the grid spacing and the time step $\Delta x/\Delta t$.

For multi-dimensional explicit calculations it is quite common and efficient to use a Strang-type (Strang 1968) dimensional splitting, where the fluxes $\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots$ are applied one by one in consecutive *sweeps*. In case of implicit steady state calculations, however, it is better to add all fluxes at the same time, so that the appropriate terms can cancel. The TVD-MUSCL and

TVDLF schemes can be used in this dimensionally unsplit way as well. Similarly, it is usually better to use temporally first order schemes in steady state calculations, so that the change from time level n to $n+1$ is simply the contribution of fluxes and sources multiplied by the time step Δt. This ensures that $U^{n+1} = U^n$ if the discretized fluxes and sources cancel each other. In case of TVDLF and TVD-MUSCL, using no predictor step will make the schemes linear in Δt. For the TVD scheme, we follow Yee's (1989) suggestion, and drop all second order terms proportional to $a^2 = (c\Delta t/\Delta x)^2$ in Eqs.(42-47, 49) of TO96. We call this temporally first, but spatially second, order scheme TVD1.

2.3. Implicit schemes

There are three basic choices for the implicit time integration to advance the solution of the system of equations (1) from time level n to $n+1$: the temporally first order *backward Euler* scheme for steady state calculations, the second order *trapezoidal* method for implicitly treated elliptic source terms, and the second order *Backward Differentiation Formula* (BDF2, see in Hairer et al. 1987) for advection dominated time accurate integration. All of these methods can be regarded as a special case of the following general two parameter time discretization

$$\begin{aligned} \boldsymbol{U}^{n+1} &= \boldsymbol{U}^n + \Delta t_n \boldsymbol{R}(\boldsymbol{U}^n) \\ &+\alpha \Delta t_n \left[\frac{\boldsymbol{U}^n - \boldsymbol{U}^{n-1}}{\Delta t_{n-1}} - \boldsymbol{R}(\boldsymbol{U}^n) \right] \\ &+\beta \Delta t_n \left[\boldsymbol{R}_{\rm impl}(\boldsymbol{U}^{n+1}) - \boldsymbol{R}_{\rm impl}(\boldsymbol{U}^n) \right] , \qquad (2) \end{aligned}$$

where $\boldsymbol{R}$ is a conservative high-order discretization (TVD, TVD-MUSCL, or TVDLF), $\boldsymbol{R}_{\rm impl}$ contains all the implicitly treated terms, and the α and β parameters may vary between 0 and 1. The scheme is three-level whenever the parameter $\alpha \neq 0$, while for $\alpha = 0$, it only uses time levels n and $n+1$. The backward Euler scheme corresponds to $\alpha = 0, \beta = 1$, and the trapezoidal scheme to $\alpha = 0, \beta = 1/2$. The BDF2 method with a varying time step requires $\alpha = \Delta t_n/(\Delta t_n + 2\Delta t_{n-1}), \beta = 1 - \alpha$. In the first time step, when $\boldsymbol{U}^{n-1}$ is not available, one has to use the backward Euler or the trapezoidal method.

In Paper I, we showed that second order spatial accuracy can be maintained even if a first order discretization is used for $\boldsymbol{R}_{\rm impl}$ in Eq. (2). We also discussed in detail the different semi-implicit options, i.e. when only some of the variables, or some of the terms are treated implicitly.

To solve Eq. (2) for $\boldsymbol{U}^{n+1}$, we linearize it by

$$\begin{aligned} \boldsymbol{R}_{\rm impl}(\boldsymbol{U}^{n+1}) &= \boldsymbol{R}_{\rm impl}(\boldsymbol{U}^{n+1}_{\rm expl}, \boldsymbol{U}^n_{\rm impl}) \\ &+ \frac{\partial \boldsymbol{R}_{\rm impl}}{\partial \boldsymbol{U}_{\rm impl}} (\boldsymbol{U}^{n+1}_{\rm impl} - \boldsymbol{U}^n_{\rm impl}) + \mathcal{O}(\Delta t^2) \qquad (3) \end{aligned}$$

where $\boldsymbol{U}_{\rm impl}$ contains the implicitly treated variables. The explicitly treated variables $\boldsymbol{U}_{\rm expl}$, if any, are advanced by an explicit time stepping scheme before Eq. (3) is applied.

The linearized fully implicit backward Euler scheme, for example, results in the linear system

$$\left[\frac{\hat{I}}{\Delta t} - \frac{\partial \boldsymbol{R}_{\rm impl}}{\partial \boldsymbol{U}} \right] (\boldsymbol{U}^{n+1} - \boldsymbol{U}^n) = \boldsymbol{R}(\boldsymbol{U}^n), \qquad (4)$$

which can be solved by one of the linear system solvers discussed in Sect. 2.5.

2.4. Evaluation of the Jacobian matrix

The *Jacobian matrix*, $\partial \boldsymbol{R}_{\rm impl}/\partial \boldsymbol{U}_{\rm impl}$ in Eq. (3), is evaluated numerically by using the various spatial discretizations in VAC. There are three options available, we describe their basic properties and their applicability in this section, for more details see Paper I.

The *matrix-free* evaluation does not calculate the elements of the Jacobian matrix, only the action of the Jacobian on a vector $\Delta \boldsymbol{U}$, when this is needed by the iterative schemes. We simply take the directional derivative of $\boldsymbol{R}$ with respect of $\boldsymbol{U}$ by taking a difference $[\boldsymbol{R}(\boldsymbol{U} + \epsilon\Delta U) - \boldsymbol{R}(\boldsymbol{U})]/\epsilon$ where ϵ is an appropriately chosen small parameter. This matrix-free method is independent of the spatial discretization and requires very little storage. Unfortunately, the matrix vector multiplication can become computationally expensive, and direct linear solvers and preconditioners cannot be used. Even the iterative solvers may fail to converge due to the fact that the matrix is effectively perturbed at each iteration by the error in the numerical evaluation of the matrix vector product. The matrix-free approach can be applied for treating elliptic-type source terms (e.g. resistivity in MHD) implicitly and in combination with the MR-PC strategy (see Sect. 2.6).

The *grid masking algorithm* calculates the individual matrix elements numerically in a fairly general way. Since storage and computational demands for the Jacobian matrix can be excessive, we restricted the implementation of this method to spatial discretizations that use the nearest neighbour cells only, in our case these are the TVDMU1 and TVDLF1 schemes. The idea of the grid masking algorithm is to perturb each implicitly treated variable of $\boldsymbol{U}$ in certain spatial patterns, and then to read off the matrix elements from the numerical evaluation of $\boldsymbol{R}$. The perturbations only affect the neighbouring grid cells, so one can determine all matrix elements in $1 + (2D+1)N_{\rm impl}$ evaluations of $\boldsymbol{R}$, where D is the number of spatial dimensions and $N_{\rm impl}$ is the number of implicitly treated variables.

A more *efficient*, but less general algorithm calculates the Jacobian matrix elements directly from the numerically taken partial derivatives $\partial \boldsymbol{F}_i/\partial \boldsymbol{U}$ and $\partial \boldsymbol{S}/\partial \boldsymbol{U}$. The spatial discretization is restricted to TVDLF1, and all the sources have to be local with the exception of Powell's source terms in Eq. (10). With these restrictions, however, the efficient method can calculate the Jacobian matrix about $2D+1$ times faster than the general grid masking algorithm.

Once the Jacobian matrix elements are calculated, which is a computationally expensive step, direct solvers, preconditioners, and iterative schemes can be applied efficiently.

2.5. Linear system solvers

There is a large selection of linear system solvers implemented in VAC. Here we only discus the most commonly used ones.

The *direct block tridiagonal solver* can be very efficiently used in one dimensional calculations. It requires that the Jacobian matrix elements are calculated directly (with the grid masking or the efficient algorithm of Sect. 2.4) and that only a three-point stencil is used for the spatial discretization of $\boldsymbol{R}_{\rm impl}$ in Eq. (2).

In more than one dimensions, or in combination with the matrix-free evaluation of the Jacobian, the linear system has to be solved by iterative methods. For symmetric positive definite matrices, e.g. resistive source terms on a Cartesian grid, the Conjugate Gradient (CG) scheme is the most efficient, both in terms of memory and CPU cost. For all other types of matrices, we found the Stabilized Bi-Conjugate Gradient (Bi-CGSTAB) algorithm (van der Vorst 1992) to be robust and efficient with a relatively small amount of storage requirement.

In multi-dimensional advection dominated problems preconditioning is vital to accelerate the convergence of the iterative schemes. The Modified Block Incomplete LU (MBILU) preconditioner (van der Ploeg et al. 1997) implemented in VAC can efficiently precondition penta-diagonal and hepta-diagonal matrices resulting from the nearest neighbor discretization in 2 and 3 dimensions, respectively.

2.6. MR-PC strategy

When direct computation of the Jacobian elements is not desirable (e.g. due to high computational cost), we can use the matrix-free approach (Sect. 2.4) in combination with the newly introduced Minimum Residual Predictor-Corrector time integration strategy (Botchev et al. 1997)[2].

In essence, the MR-PC method uses a restricted number of iterations with a minimum residual method, such as the Generalized Minimal RESidual algorithm (GMRES, Saad & Schultz 1986), which allows the time step to be much greater than for an explicit scheme. For time accurate calculations, the predictor step is a second order explicit scheme and the corrector step is a second order implicit scheme, e.g. BDF2. For steady state calculations, no predictor step is necessary, one can start the limited number of iterations from $\boldsymbol{U}^n$, and the corrector step is the Backward Euler scheme. The order of the local error in MR-PC is the minimum of those of the predictor and corrector steps, and it does not depend on how accurately the linear system is solved. The corrector step is mainly used to stabilize the scheme.

Since the MR-PC scheme is not unconditionally stable, it is important to adjust the time step correctly. This is achieved by extracting the spectral information delivered by minimal residual iterations, and requires a negligible amount of extra work. For details see Botchev et al. (1997).

3. Equations

Several equation modules are available in VAC, all written in a dimension independent notation, with $1 \le D \le 3$ spatial dimensions and C vector components with $D \le C \le 3$. The D=1, C=2 or 3, and the D=2, C=3 cases are usually referred to as 1.5D and 2.5D, respectively. Although the equation modules are implemented independently, physically they can be regarded as special cases of the MHD equations, which we write in terms of density ρ, momentum density $\rho\boldsymbol{v}$, total energy density e, and magnetic field $\boldsymbol{B}$, with additional source terms,

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\boldsymbol{v}\rho) = S_\rho \qquad (5)$$

$$\frac{\partial\rho\boldsymbol{v}}{\partial t} + \nabla\cdot(\boldsymbol{v}\rho\boldsymbol{v} - \boldsymbol{B}\boldsymbol{B}) + \nabla p_{\rm tot} = \boldsymbol{S}_{\rho v} \qquad (6)$$

$$\frac{\partial e}{\partial t} + \nabla\cdot(\boldsymbol{v}e + \boldsymbol{v}p_{\rm tot} - \boldsymbol{B}\boldsymbol{B}\cdot\boldsymbol{v}) = S_e \qquad (7)$$

$$\frac{\partial\boldsymbol{B}}{\partial t} + \nabla\cdot(\boldsymbol{v}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{v}) = \boldsymbol{S}_B, \qquad (8)$$

where we used the total pressure $p_{\rm tot} = p + \boldsymbol{B}^2/2$, with thermal pressure p related to the total internal energy density e as

$$p = (\gamma - 1)\left(e - \rho\boldsymbol{v}^2/2 - \boldsymbol{B}^2/2\right), \qquad (9)$$

for an ideal gas with adiabatic index γ. Units of $\boldsymbol{B}$ are chosen such that the magnetic permeability is 1.

The source terms S_ρ, $\boldsymbol{S}_{\rho v}$, S_e and $\boldsymbol{S}_B$ can be arbitrary functions defined by the user, but some frequently used source terms are implemented for convenience. Currently available in VAC are sources for external gravity, viscosity, resistivity, and thermal conduction

$$\begin{array}{lcllll} \boldsymbol{S}_{\rho v} &=& -\delta(\nabla\cdot\boldsymbol{B})\,\boldsymbol{B} & -\nabla\cdot\left(\nu\hat{\Pi}\right) & +\rho\boldsymbol{g} & \\ S_e &=& -\delta(\nabla\cdot\boldsymbol{B})\,\boldsymbol{B}\cdot\boldsymbol{v} & -\nabla\cdot\left(\boldsymbol{v}\cdot\nu\hat{\Pi}\right) & +\rho\boldsymbol{g}\cdot\boldsymbol{v} & \\ & & & +\nabla\cdot(\boldsymbol{B}\times\eta\boldsymbol{J}) & +\nabla\cdot(\kappa\nabla T) & \\ \boldsymbol{S}_B &=& -\delta(\nabla\cdot\boldsymbol{B})\,\boldsymbol{v} & -\nabla\times(\eta\boldsymbol{J}), & & \end{array} \qquad (10)$$

where $\boldsymbol{g}$, ν, η, and κ are the external gravitational field, and the viscosity, resistivity and thermal conduction coefficients. All the coefficients may be functions of t, $\boldsymbol{x}$, or $\boldsymbol{U}$. Further, current density is given by $\boldsymbol{J} = \nabla\times\boldsymbol{B}$, and the temperature is $T = p/\rho\mathcal{R}$ with gas constant $\mathcal{R}$. The tensor $\hat{\Pi}$ is the traceless part of the total kinetic pressure dyadic

$$\hat{\Pi} = -\left(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^{\rm T}\right) + \frac{2}{3}\hat{I}\,(\nabla\cdot\boldsymbol{v}), \qquad (11)$$

where $\hat{I}$ is the identity tensor.

The source terms proportional to $\nabla\cdot\boldsymbol{B}$ are analytically zero, but numerical errors may introduce a non-zero magnetic field divergence in multidimensional MHD simulations. These correction terms were suggested by Powell (1994). In combination with his modified eight-wave Riemann solver, they introduce an additional divergence wave which is advected by the flow velocity $\boldsymbol{v}$, and they stabilize the TVD and TVD-MUSCL type algorithms. Tóth and Odstrčil (1996) found that the source terms can also improve the accuracy of the FCT and TVDLF numerical schemes, which do not employ a Riemann solver. The coefficient δ is 1 when these corrective source terms are included, and 0 otherwise.

Powell's source terms, in general however, are insufficient to keep the numerically generated $\nabla\cdot\boldsymbol{B}$ negligible. Therefore a projection scheme (Brackbill & Barnes 1980) is implemented in VAC to remove the divergence of the magnetic field before each

[2] See also http://www.math.ruu.nl/people/botchev/.

time step. We use iterative schemes to solve the Poisson equation $\nabla^2\phi = \nabla\cdot\boldsymbol{B}$ and subsequently correct the magnetic field to $\boldsymbol{B}' = \boldsymbol{B} - \nabla\phi$. The boundary conditions for the Poisson equation are chosen carefully, so that the gradient of the solution $\nabla\phi$ changes $\boldsymbol{B}$ consistently with the boundary conditions for $\boldsymbol{B}$. We also note that the numerical representation of the Laplace operator is obtained by the repeated application of the discretized ∇ operator, rather than by the usual centered difference formulae.

In explicit integration schemes the time step is restricted by the Courant-Friedrichs-Levy (CFL) condition

$$\Delta t \le C \min_i \left(\frac{\Delta x_i}{c_i^{\max}}\right), \tag{12}$$

where C is the Courant number and $c_i^{\max}$ is the maximum propagation speed of information in direction i. For the hyperbolic system of ideal MHD equations this speed is

$$\begin{aligned} c_i^{\max} &= |v_i| + c_i^{\mathrm{fast}} = |v_i| \\ &+ \frac{1}{\sqrt{2}}\left[\frac{\gamma p + \boldsymbol{B}^2}{\rho} + \sqrt{\left(\frac{\gamma p + \boldsymbol{B}^2}{\rho}\right)^2 - 4\frac{\gamma p B_i^2}{\rho^2}}\,\right]^{1/2}, \end{aligned} \tag{13}$$

where c_i^{fast} is the speed of the fast-mode MHD wave relative to the fluid in the i direction. For hydrodynamics c_i^{fast} simplifies to the sound speed $c_{\mathrm{s}} = \sqrt{\gamma p/\rho}$. In our units the Alfvén speed is simply $c_{\mathrm{A}} = |B|/\sqrt{\rho}$.

For resistive, viscous, and/or thermally conductive plasmas the time step is further limited by the diffusion time scale

$$\Delta t \le C_{\mathrm{diff}} \min_i \left[\frac{(\Delta x_i)^2}{\max(\eta,\nu,\kappa)}\right]. \tag{14}$$

As the grid resolution increases and Δx decreases, the diffusive time scale becomes more restrictive than the CFL condition, because it scales with Δx^2 rather than Δx. In such situations a semi-implicit approach that treats the source terms implicitly, can be highly effective.

For explicit time integration schemes the maximum allowable value for the Courant number C and the coefficient C_{diff} depend slightly on the spatial discretization, but in general $C < 1$ and $C_{\mathrm{diff}} < 0.5$ has to be used. The conditions (12) and (14) have to be satisfied everywhere in the computational domain.

4. Model problems

We present a selection of four test problems. For each problem we provide a physical description and the corresponding numerical representation, and then we compare the efficiency of implicit and explicit time integration strategies. We follow the general guidelines given in Sect. 2 to find the optimal solution strategy in each case. Test problems 1, 3, and 4 were timed on a DEC Alpha/400 work station, while the second test was done on an SGI Power Challenge.

4.1. Steady state accretion onto a black hole

4.1.1. Problem description and solution

We use a simple model of an accretion disk around a black hole, which was studied analytically by Chakrabarti (1989). The system has a cylindrical symmetry around the axis of the accretion disk and we assume that in the equatorial plane the gradients normal to this plane are negligible. The model is one dimensional with radial and azimuthal velocity components only, thus we solve the hydrodynamical ($\boldsymbol{B} \equiv 0$) equations (5-7) in 1.5D with the inclusion of a pseudo-relativistic external gravity $g_r = -(r-1)^{-2}/2$ (Paczýnski & Wiita 1980), where the radial distance r is measured in units of the Schwartzschild radius. Gravity is the only source term incorporated. The adiabatic index is $\gamma = 4/3$.

Chakrabarti (1989) derived some theoretical predictions regarding the attainable steady state solutions from the conservation of mass, angular momentum, and energy. For the one dimensional accretion problem, the position of a transonic point can be determined. The solutions within this ‘outer’ sonic point can either remain purely supersonic, or contain a shock and an extra ‘inner’ sonic point. Although the mathematics allows for two possible shock locations, subsequent numerical investigations by Chakrabarti and Molteni (1993) revealed that only the shock location furthest away from the black hole is stable.

Numerical experiments showed that conservation of the angular momentum is crucial to get an accurate solution, thus the azimuthal component of the momentum equation (6) is cast in terms of specific angular momentum $r\rho v_\varphi$ instead of the momentum ρv_φ. This required only minor modifications of the difference formulae.

The black hole is at the origin $r = 0$, and we model the region between 1.5 and 10 Schwartzschild radii, which is resolved by a grid of 204 cells including the 4 ghost cells. We set density $\rho = 1$, thermal pressure $p = 0.01428$, radial velocity $v_r = -2.1/r$, and specific angular momentum $r\rho v_\varphi = 1.785$ in the whole grid initially. At $r = 10$ a supersonic inflow is imposed by fixing the variables in the ghost cells. Since the corresponding sound speed c_{s} is only 0.138, the inflow is supersonic in both the radial and tangential directions, with the radial Mach number $M_r = |v_r|/c_{\mathrm{s}}$ being approximately 1.5.

We are interested in recovering the steady state solution containing the shock wave. Since the flow is supersonic at the inner boundary, an outflow boundary condition can be realized by extrapolating the variables in the last grid cell of the physical domain into the ghost cells. However, numerical experiments showed that using an open boundary from the beginning leads to the fully supersonic steady state solution. To obtain the shocked solution, we use a reflective boundary representing a solid wall temporarily, thus the infalling material bounces off this inner boundary, and a shock front forms. Before this shock moves too far outwards, we switch to the outflow boundary condition at the inner boundary, so that the shock can settle into its steady state position. The obtained solution is shown in Fig. 1, where the density and the radial Mach number are plotted.

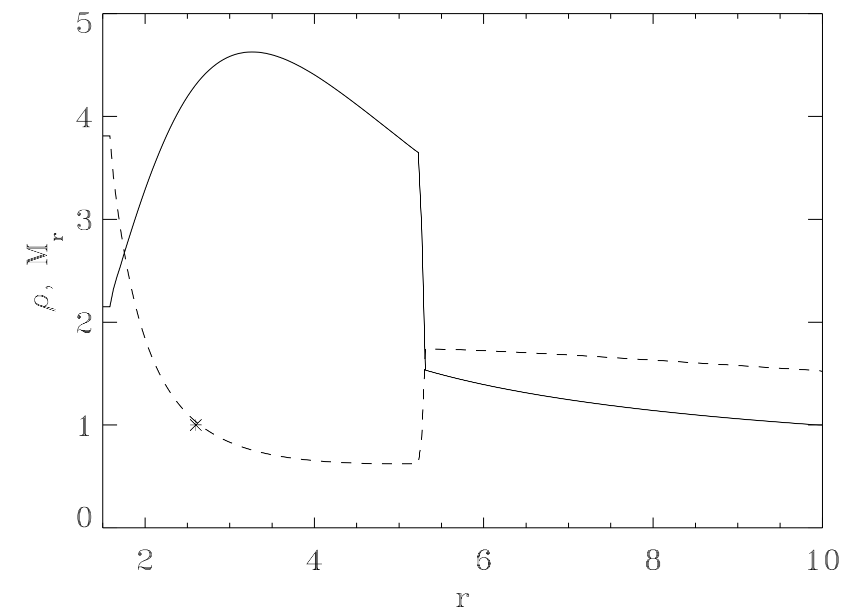

Fig. 1. Steady state accretion onto a black hole located at $r = 0$. Density is shown as a solid line, the radial Mach number M_r as a dashed line, and the inner sonic point is at the $*$

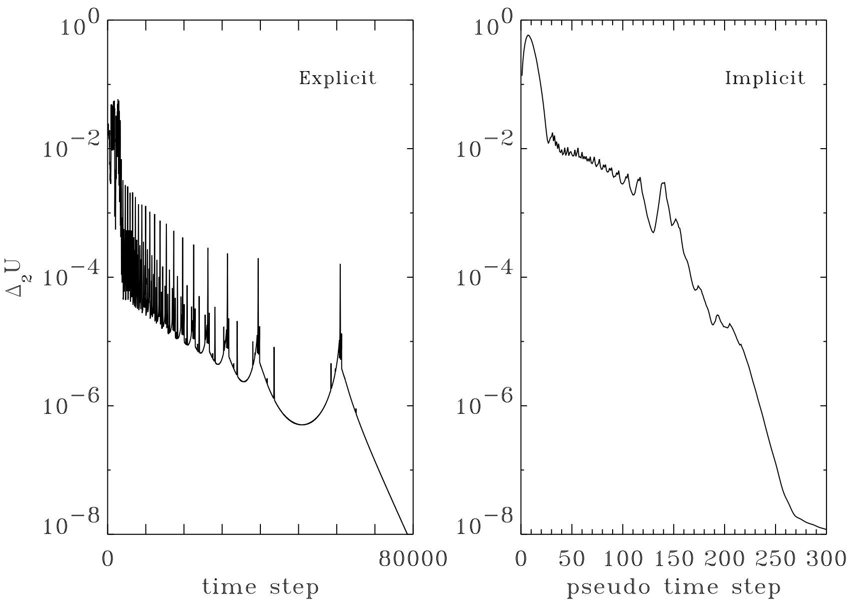

Fig. 2. Convergence history for the explicit (left) versus the implicit (right) solution. We plot the relative residual change $\Delta_2 U$ versus the number of (pseudo) time steps

4.1.2. Implicit versus explicit

Since we solve the time-dependent equations numerically, we demand that in steady state, the difference between subsequent time steps as measured by the relative change from one time level to the next

$$\Delta_2 U \equiv \sqrt{\frac{1}{N_{\rm var}} \sum_{u=1}^{N_{\rm var}} \frac{\sum_{\rm grid}(U_u^{n+1} - U_u^n)^2}{\sum_{\rm grid}(U_u^n)^2}} \qquad (15)$$

drops below 10^{-8}. Note that the normalization is done per component u of the conserved variables $\boldsymbol{U}$. This is necessary since the physical units of the different variables may differ by several orders of magnitudes.

When an explicit high order method is used to solve the outlined problem, the method stagnates as it evolves very slowly to the final steady state after the fast transient phase is completed. Using the Riemann solver-based TVD1 scheme with a Woodward limiter, we arrive at steady state with $\Delta_2 U < 10^{-8}$ after about 78500 CFL-limited explicit time steps. This takes about 750 seconds.

To overcome the stagnation of the explicit scheme, we first perform 200 explicit time steps (2 CPU seconds) using the reflective boundary conditions, and then switch to an implicit approach. We obtain the spatially second order accurate solution implicitly by performing a pseudo time stepping (Eq. 4), where we evaluate $\boldsymbol{R}$ using the high-resolution TVD1 method with a

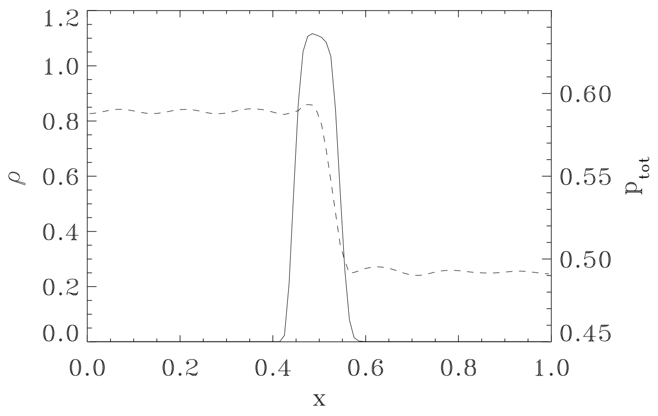

Fig. 3. Density ρ (solid line) and total pressure $p_{\rm tot}$ are plotted at time $t = 1$. Note the different scaling of the vertical axes: ρ at left and $p_{\rm tot}$ at right. The small amplitude magnetosonic waves, visible in $p_{\rm tot}$, are induced by the oscillation of the plasma sheet

Woodward limiter. The Jacobian matrix is calculated with the efficient algorithm using the TVDLF1 scheme. The problem is 1.5 dimensional, thus the linear system can be efficiently solved with the direct block tridiagonal solver. For stability reasons, we increase the Courant number in a continuous fashion from $C = 5$ to about $C = 1000$ in 314 implicit pseudo time steps to obtain a nicely converged solution with $\Delta_2 U < \mathcal{O}(10^{-8})$. The implicit calculation is completed in only 17 CPU seconds. The overall gain is thus a factor of 44 relative to the explicit scheme, or a factor of 40, if we include the CPU time spent on setting up the initial condition with the reflective inner boundary.

The convergence histories of the explicit and implicit calculations are shown in Fig. 2. It can be seen that the explicit method converges rapidly at first, but subsequently stagnates. Each time the shock passes another grid cell the error fluctuates slightly, until the shock reaches its final steady state position.

4.2. Oscillating plasma sheet

4.2.1. Problem description and solution

This test demonstrates the problems encountered in numerical simulations of astrophysical MHD phenomena surrounded by magnetized vacuum. The vacuum can be well represented by a low density, low pressure plasma, but that implies an extremely high magnetosonic speed, which poses significant problems to explicit time integration schemes. Another difficulty is to maintain the positivity of density and pressure in the "vacuum".

For sake of simplicity, here we model the oscillation of a plasma sheet surrounded by vacuum and placed between two ideally conducting walls. The problem is studied in 1D, with reflective boundary conditions at both ends. The sheet is modeled as a plasma of high density $\rho = 1$ and thermal pressure $p = 0.3201$, relative to the surrounding "vacuum", where $\rho = 0.001$ and $p = 0.0001$ are set. The magnetic field $\boldsymbol{B}$ is parallel to the plasma sheet with $B_y = 0.6$ inside the sheet, $B_y = 1.1$ to the left and $B_y = 1.0$ to the right from the plasma slab. Therefore the total pressure $p + B^2/2 = 0.5001$ inside the high density region is in balance with the pressure in the "vacuum" to the right, but it is 10% less than the pressure on the left. The initial velocity $\boldsymbol{v}$ is zero everywhere. The computational domain is $x \in [0, 1]$ within which the plasma sheet is initially positioned at $x \in [0.45, 0.55]$. We take $\gamma = 1.4$ and for this one dimensional ideal MHD simulation there are no source terms. We use a total of 100 grid cells.

The high magneto-sonic speed $c^{\rm fast} \approx v_{\rm A} \approx 33$ in the low density surroundings of the plasma sheet limits the time step for an explicit scheme. However, we expect that the plasma sheet moves away from its initial position relatively slowly due to the small jump in the total pressure, and that a slow oscillation will set in around an equilibrium position governed by the conservation of magnetic flux in the left and right vacuum regions. From an elementary calculation the oscillation frequency is expected to be $\omega = \sqrt{2B^2/(mL)} \approx 6.48$ with an amplitude $\Delta L \approx (\Delta B/B)L = 0.0238$, where $L \approx 0.5$ is the distance between the sheet and the perfectly conducting walls, and the total mass of the sheet is $m = 0.1$. Thus the maximum of the total momentum mv_x is $m\omega\Delta L \approx 0.015$ and the oscillation period is $2\pi/\omega \approx 0.97$. Therefore $c^{\rm max}$, which sets the time step in the CFL condition, is about 200 times larger than the actual flow speed characterizing the evolution of the system. In Fig. 3, we show a snapshot of the time evolution. The plasma sheet is clearly visible in the density profile, while the total pressure imbalance across the sheet governs the oscillation. We also plot the time evolution of the total momentum in Fig. 4, which is in excellent agreement with the analytical predictions.

4.2.2. Implicit versus explicit

For this simple problem, we can easily compare the effects of using different spatial discretizations in both the explicit and the implicit time integration schemes. For each case, we summarize our findings for the TVDLF and the TVD-MUSCL scheme, in combination with both the minmod and the Woodward limiter. We solve this 1.5D problem fully implicitly using the spatially and temporally second order accurate BDF2 scheme, and a spatially first order Jacobian matrix in the linearized scheme. The first implicit time step is evaluated with backward Euler. The linear problems are solved with the direct block tridiagonal solver.

For the explicit two-step TVDLF scheme with the minmod limiter, a Courant number $C = 0.8$ can be used. We obtain a rather diffusive solution up to time $t = 2$ in roughly 50 seconds. Using the Woodward limiter to reduce diffusion, we needed to lower the Courant number to maintain positivity of pressure: with $C = 0.1$, an explicit solution up to time $t = 2$ takes 512 seconds, needing about 93000 time steps. When we solve for the time evolution implicitly, we note that we can use the efficient Jacobian evaluation for the first order TVDLF1 scheme since there are no source terms present. We can combine this

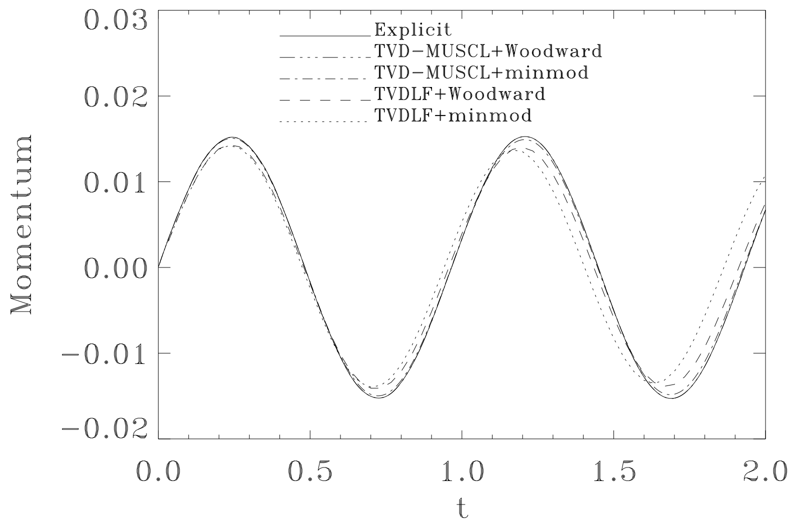

Fig. 4. Comparison of the sharpest explicit solution (TVD-MUSCL with Woodward limiter) against four implicitly obtained solutions using the time evolution of the total momentum

low order Jacobian with an evaluation of $\boldsymbol{R}$ appearing in the right hand side using TVDLF and either limiter: after 1.5 seconds we arrive at time $t = 2$ with 40 steps of constant $\Delta t = 0.05$. This corresponds to a Courant number of approximately $C \approx 170$. The gain is thus at least a factor of 30 in CPU time. The implicit solution with fixed time step $\Delta t = 0.05$ using the TVDLF scheme is slightly more diffusive than the explicit solution with the same limiter. For smaller time steps $\Delta t = 0.01$, the implicit calculation can be as good as the explicit one, while still gaining an order of magnitude in CPU time (6.6 seconds versus 50 seconds for the minmod limiter).

When using the TVD-MUSCL scheme, an explicit solution with either the minmod or the Woodward limiter needs a Courant number $C = 0.4$, and 159 seconds on average to take a total of about 17000 time steps. This scheme is less diffusive, so the plasma sheet maintains its original width. In fact, when we apply the Woodward limiter, we can even detect small amplitude fast magneto-acoustic waves in the surrounding low density plasma (these can be seen in Fig. 3). In the implicit solution with the second order three-level BDF2 scheme, we use the high order TVD-MUSCL scheme in the right hand side and its first order variant to evaluate the Jacobian matrix. As explained in Sect. 2.4, this first order upwind Jacobian can be calculated easily using the general grid masking algorithm. Our experiments showed that when we use the minmod limiter at right, we can allow a maximal time step of $\Delta t = 0.03$ (corresponding to $C > 100$) to obtain a time accurate solution up to $t = 2$ in 7.7 seconds. The gain is thus a factor of 20, but the implicit solution is again somewhat more diffusive. This can be remedied partly by going to lower Courant numbers. However, since the general Jacobian calculation is rather time-consuming, we can only gain in efficiency compared to the explicit scheme up to a Courant number of $C = 10$, when the implicit solution needs 80 seconds (factor of 2 gain) to complete the necessary 700 time steps to reach $t = 2$. The situation is even less favourable to the implicit scheme if we try to use the sharper Woodward limiter in the right hand side evaluation of $\boldsymbol{R}$. We found a time-accurate, stable, and fully implicit solution at a maximum Courant number $C = 4$, but this required 198 seconds to complete so that nothing could be gained compared to the explicit approach. The MR-PC(5) (i.e. 5 GMRES iterations per step) scheme turns out to be the fastest in this case, but the gain factor with respect to explicit scheme is only about 1.5. It does not make any difference for MR-PC whether a first order (TVDMU1) or second order (TVDMU) Jacobian is used, the CPU time remains approximately the same.

These findings are illustrated in Fig. 4, where we plot the time evolution of the total momentum mv_x for five different second order time accurate solution strategies. The solid line corresponds to the least diffusive explicit solution obtained with the TVD-MUSCL scheme and the Woodward limiter ($C = 0.4$). The amplitude of the resulting oscillation is observed to decrease in time if the plasma sheet broadens by diffusing into the surrounding 'vacuum'. Therefore, we can quantitatively compare the effects of different spatial discretizations in this figure. The four implicitly obtained solutions shown are (ordered by their diffusive character): (i) first order upwind TVDMU1 for $\boldsymbol{R}_{\mathrm{impl}}$ in Eq. (2) and high order TVD-MUSCL with Woodward limiter for $\boldsymbol{R}$ ($C = 4$), indistinguishable from the explicit solution; (ii) TVDMU1 for $\boldsymbol{R}_{\mathrm{impl}}$ with TVD-MUSCL and minmod limiter for $\boldsymbol{R}$ ($C = 10$); (iii) TVDLF1 combined with TVDLF with Woodward limiter ($C = 100$); and (iv) the rather diffusive solution with TVDLF1 and TVDLF with minmod limiter ($C = 100$).

For this problem there is a trade-off between accuracy and computational cost. The sharper implicit schemes have difficulties with maintaining the positivity of pressure and density at the edges of the plasma sheet, and this reduces the allowable time step considerably.

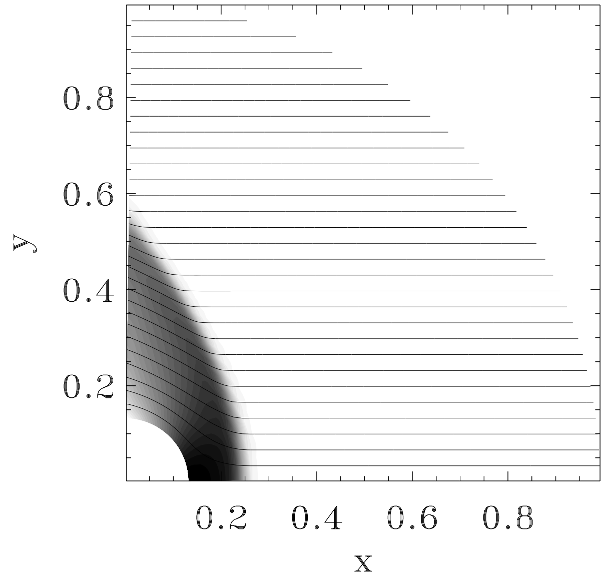

Fig. 5. Steady state bow-shock around a perfectly conducting cylinder, the physical system is symmetric with respect to the x-axis. The density (grey-scale) and the magnetic field lines are shown

4.3. Formation of a steady bow shock

4.3.1. Problem description and solution

This test demonstrates how a two-dimensional steady-state MHD problem with a shock wave can be solved. We model super-fast flow around a perfectly conducting cylinder. Our interest is in the final steady state solution which contains a shock front, akin to the bow shock around planets in the solar wind. Figure 5 shows the steady-state density and magnetic field. This problem is also briefly described by van der Ploeg et al. (1997), where the emphasis was on the efficiency of the MBILU preconditioner.

We solve the 2D ideal MHD equations on a 60×60 polar grid (r, ϕ), and take the surface of the cylinder at $r = 0.125$ and the outer boundary at $r = 1$. Initially, the velocity is constant with $v_x = -4$ and $v_y = 0$, and the magnetic field is parallel to the velocity field with $B_x = -1$ and $B_y = 0$. Density ρ and pressure p are 1, and the adiabatic index is $\gamma = 5/3$. The Mach number is $v/c_\mathrm{s} = 3.1$, while the Alfvénic Mach number is $v/c_\mathrm{A} = 4$. We make use of the symmetry at $\phi = 0$, and only model a quarter of the full polar grid. The boundary conditions keep the initial values at $r = 1$ fixed (supersonic inflow), and impose the symmetry at $\phi = 0$. At $r = 0.125$, the perfectly conducting cylinder is impenetrable for the flow and the magnetic field, which is represented by the appropriate symmetric and anti-symmetric boundary conditions, while at $\phi = \pi/2$ a zero gradient is maintained in all variables for the outflow.

4.3.2. Implicit versus explicit

Just as in Sect. 4.1, this steady state problem needs many (on the order of 2500) explicit time steps to arrive at a steady state solution where $\Delta_2 U \le 2 \times 10^{-5}$. This explicit steady state solution was found using the TVDLF scheme and the minmod limiter, and takes about 2800 seconds to complete. To enforce a zero $\nabla \cdot \boldsymbol{B}$, we included the Powell source terms in Eq. (10), combined with a projection scheme applied before each time step. It turns out that this divergence cleaning takes up roughly 50% of the total CPU time. Interestingly, further reduction of the residual could not be obtained, most likely due to numerical difficulties related to the magnetic field and its divergence. If we do the divergence cleaning every 10 time steps only, the CPU time spent on it reduces significantly, however, the residual does not drop below 2×10^{-4}.

Again, after 100 initial explicit steps that take care of the initial transients, we can accelerate the convergence towards the final steady state using an implicit pseudo time stepping scheme Eq. (4) with the time step determined by the maximum Courant number $C = 100$. The first order TVDLF1 scheme is used for $\boldsymbol{R}_\mathrm{impl}$. The resulting block penta-diagonal Jacobian is strongly nonsymmetric, for which the conjugate gradient-type iterative schemes hardly converge. Therefore, the MBILU-preconditioner is applied, and the Jacobian matrix is calculated by the efficient algorithm. For the evaluation of $\boldsymbol{R}$ in Eq. (4), we use the second order TVDLF scheme with a minmod limiter, as in the explicit calculation.

In each pseudo time step, the iterative scheme is stopped when the residual drops by a factor of 10^{-5}. Using GMRES we need only 47 pseudo time steps with typically 20 iterations per time step. This takes about 530 seconds. Experiments with other iterative schemes showed that only little can be gained from the use of a different scheme. The best result came from Bi-CGSTAB, which requires for 47 pseudo time steps also, but only 502 seconds, i.e. Bi-CGSTAB is computationally cheaper than GMRES for this problem. We also experimented with limiting the number of iterations per time step. This approach is similar to MR-PC, but here we use preconditioning, so the Jacobian matrix needs to be calculated. For instance, the Bi-CGSTAB algorithm with a maximum of 5 iterations in every pseudo step can achieve convergence in 401 CPU seconds in 50 pseudo time steps. The gain is due to the decrease in the total number of iterations by about a factor of 2, which is partially compensated by the increase in the number of pseudo time steps, which require calculating the Jacobian matrices. Overall, we can get a total speedup relative to the explicit scheme of a factor of 5 to 7, and the convergence behaviour is much better.

Noteworthy is our approach to keep $\nabla \cdot \boldsymbol{B} = 0$ when solving this 2D MHD problem implicitly: Powell's source terms given in Eq. (10) are included into the low order evaluation of the Jacobian matrix, which proves to be important for the convergence properties of the iterative scheme. In conjunction with these extra source terms, the projection scheme is applied before each pseudo time step. In view of the low number of pseudo time steps taken, the extra CPU cost of divergence cleaning is now negligible.

We also experimented with the MR-PC(5) approach in combination with the matrix-free Jacobian evaluation using the second order TVDLF scheme. For this steady state problem no predictor step is used, while the corrector step is backwards Euler. The residual drops below 10^{-4} after 317 pseudo-time steps, which requires about 1.7 times less CPU time than the explicit scheme that reaches the same accuracy in 1585 steps. Further reduction of the residual below 10^{-5} could not be achieved with MR-PC, which is similar to our experience with the explicit time integration.

4.4. *MHD reconnection*

4.4.1. Problem description and solution

To show how (semi-)implicit time-accurate solutions for the resistive MHD equations are obtained, we set forth to simulate the evolution towards a Petschek-type magnetic field annihilation (Petschek 1964). We start from an isothermal equilibrium configuration with two regions of oppositely directed field lines. A localized anomalous resistivity initiates the onset of the reconnection process between the two regions, and an X-type neutral point forms. This is followed by a gradual buildup towards the familiar Petschek steady-state solution containing a region around the X-type neutral point where diffusion dominates and stationary slow magnetoacoustic shock fronts. The anomalous resistivity is required to break the initial symmetry and to maintain the position of the X-point, but the annihilation is mostly due to the unavoidable numerical resistivity. Note, however, that the Petschek solution depends only logarithmically on the value of η.

We solve the MHD equations on a non-uniform Cartesian grid in 2D. Due to the symmetry of the problem, only one quarter of the reconnection region is represented, thus the X-point is at the lower left corner of Fig. 6. The non-uniform grid greatly improves the resolution at the neutral point, while at the boundaries, which are sufficiently far to have a minimal influence on the reconnection, the grid spacing is coarser. The cell sizes increase by a constant factor $\Delta x_{j+1}/\Delta x_j = \Delta y_{k+1}/\Delta y_k = 1.02$ in both directions. The computational domain $x \in [0,1]$ and $y \in [0,6]$ is resolved by a 100×100 grid.

In Eq. (10), $\boldsymbol{g} = 0$, $\nu = 0$, $\kappa = 0$, while $\delta = 1$, and we impose an anomalous resistivity given by

$$\eta(\boldsymbol{x}) = \eta_0 \exp\left[-(x/\ell_x)^2 - (y/\ell_y)^2\right], \quad (16)$$

where the constants η_0, ℓ_x, and ℓ_y determine the resistivity at the origin, and the width of the resistivity profile in the x and y directions, respectively.

Initially, $v_x = 0$, $v_y = 0$, and $B_x = 0$ everywhere, while $B_y = \tanh(x/L)$ and the length scale L determines the width of the region where the magnetic field smoothly reverses its sign. The isothermal equilibrium configuration implies that the thermal pressure and density profiles are $p(x) = p_{\rm tot} - B_y^2(x)/2$ and $\rho(x) = \rho(1)p(x)/p(1)$, respectively. The value of the $p_{\rm tot}$ parameter determines the plasma beta $2p/\boldsymbol{B}^2$ at $x = 1$, which is an important parameter for the Petschek reconnection. The ratio of specific heats is set to $\gamma = 5/3$.

The boundary conditions are symmetric for $\rho, \rho v_x, e$, and B_y and anti-symmetric for ρv_y and B_x along the x-axis, while along the y-axis the variables $\rho, \rho v_y, e$, and B_x are symmetric, and ρv_x and B_y are anti-symmetric. At the top boundary, all physical quantities are extrapolated into the ghost cells with a zero gradient. At right, we fix $\rho, \rho v_y, e$, and B_y to their initial values, while the inflow mass flux ρv_x is fixed at to supply the reconnection region with plasma and magnetic flux. The initially zero B_x is extrapolated continuously from the last cell row into the ghost cells, so that the field lines can bend.

For parameter values $\eta_0 = 0.0001$, $\ell_x = 0.05$, $\ell_y = 0.1$, $L = 0.1$, $p_{\rm tot} = 1.25$, $\rho(1) = 1$, and $\rho(1)v_x(1) = -0.04$ we show in Fig. 6 four snapshots of the time evolution towards the steady state configuration. The final picture corresponds to the steady state.

4.4.2. Implicit versus explicit

We solve the 2D resistive MHD equations using the TVDLF method with the Woodward limiter without dimensional splitting. The divergence of $\boldsymbol{B}$ is kept below 0.01 by the projection scheme. The explicit scheme is stable with Courant number 0.8, and it requires 18556 seconds to reach time $t = 22$ in 10214 time steps. The time evolution is extremely slow in the beginning, but after the jet forms, which accelerates matter upwards with high speed, the dynamic time scale reduces. We stop the calculation at this time, because the jet head is about to reach the upper boundary at $y = 6$, which results in some interaction with the boundary. However, in the part of the domain plotted in Fig. 6, the flow has settled down to the Petschek solution.

To speed up the calculation, the second order BDF2 implicit time integration scheme is used. The rather weak resistive source terms are treated explicitly, so that we can apply the efficient Jacobian calculation algorithm with the TVDLF1 method. If the source terms were also implicit, we would have to use the general grid masking method, since the resistive sources involve spatial derivatives. The time step is limited by accuracy considerations. There are no magnetosonic waves present, but the flow evolves by simple advection at the fluid speed $\boldsymbol{v}$ which is driven by the resistive reconnection at the origin. Therefore we limit the time step to $\Delta t < 0.5$ to capture the effect of resistivity, and we put an additional constraint $\Delta t < \min(\Delta x/v_x, \Delta y/v_y)$ to follow the evolution of the flow accurately.

The implicit approach needs 3890 seconds to reach $t = 22$ in 221 time steps, which is a factor of 4.8 improvement over the explicit scheme. A closer inspection of the solution reveals that the implicit solution is less accurate, there are small wiggles in the field lines, although the overall picture agrees well. We may also notice that the time step reduces sharply to $\Delta t \approx 0.04$ after $t \approx 17$ due to the increasing speed of the jet. This is about 20 times bigger than the explicit time step, while the CPU cost of an implicit time step is about 10 times more than that of the explicit step, so our gain in CPU time is only a factor of 2 at the last stage of the computation. Therefore we may switch back to the explicit method at time $t = 15.4$ after just 57 implicit steps, and finish the calculation with 3500 explicit time steps. The total CPU time required by this hybrid approach is 7360 CPU seconds, which is a factor of 2.5 gain over the fully explicit scheme, and the result is just as accurate. For this problem

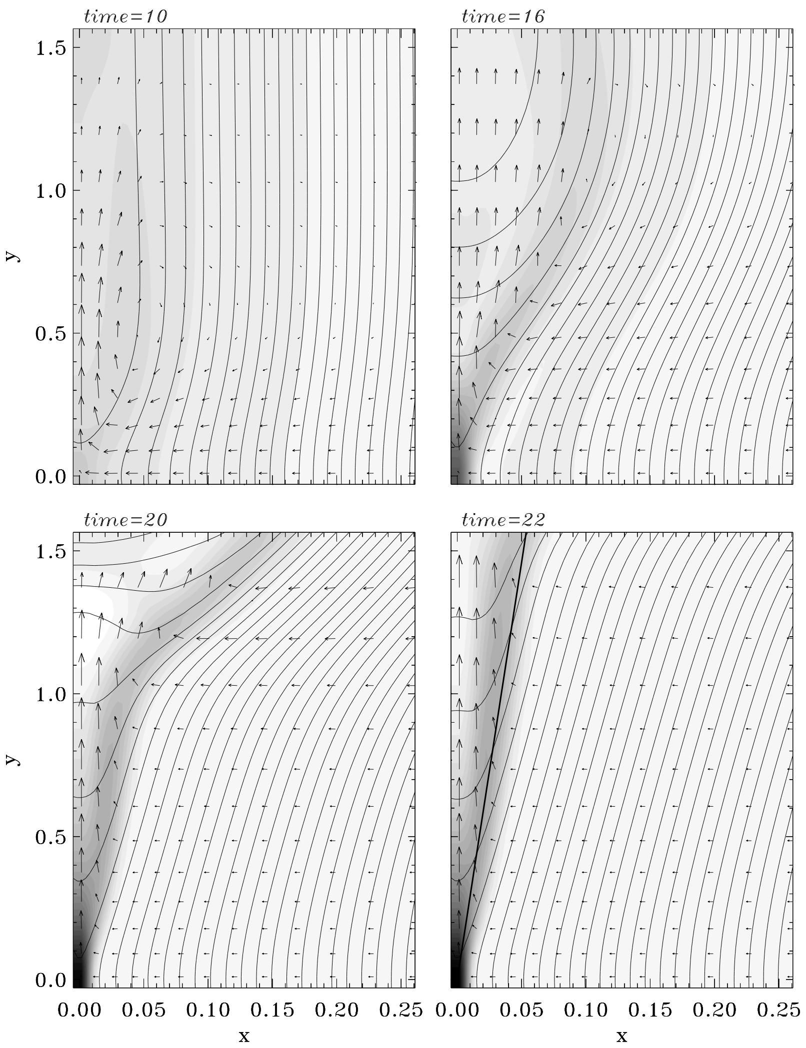

Fig. 6. The field lines, the velocity field (arrows), and the current density J_z (grey scale) for four snapshots of the magnetic reconnection problem. In the last snapshot the theoretical opening angle is indicated by a thick line. Only the lower left quarter of the computational domain is shown

the MR-PC(5) method gave only a factor of 1.2 speed up with respect to the explicit scheme.

5. Conclusions

The time scales present in hydrodynamic and magnetohydrodynamic systems may vary by several orders of magnitude. Using explicit time integration, the shortest of these time scales governs the allowed time steps to ensure numerical stability. For steady state calculations, or when the actual dynamics occur on one of the longer time scales in the system, (semi-)implicit time discretizations can be much more efficient. In this paper, we discussed different semi-implicit integration schemes recently implemented in VAC, and we illustrated the obtainable efficiency gain in a variety of test problems. We outlined general solution strategies depending on the type of problem at hand, and on its dimensionality.

We have shown how one can obtain steady states containing shocks using modern high resolution schemes (TVD-type schemes) in a fully implicit manner. This was done for a 1.5D HD problem modeling accretion onto a black hole, and in a 2D MHD simulation of a bow shock around a perfectly conducting cylinder. In the accretion problem, we found that the steady state solution with a stationary shock front could best be obtained fully implicitly. Explicit schemes were found to stagnate. The speed up amounted to a factor of 40. In case of the superfast flow impinging on a cylinder, preconditioning proved to be essential for an efficient implicit solution. The zero magnetic field divergence was ensured by combining Powell's source terms in the calculation of the Jacobian matrix with a projection scheme applied after each pseudo time step. This is more efficient than the divergence cleaning used in the explicit scheme, where the projection is applied after every explicit time step. The implicit solver is a factor of 7 faster, and it showed superior convergence properties over the explicit scheme.

The time-accurate simulation of a plasma sheet in a low density surroundings showed that we can gain an order of magnitude in speed at the price of losing somewhat in accuracy relative to the sharpest explicit results. In the two dimensional time-accurate calculation of the reconnection problem the speed up was most noticeable at the early stages of the calculation, when the flow evolves slowly. It seems to be useful to switch back to an explicit time integration scheme when the CPU cost becomes comparable.

Our first attempts to use the implicit methods are encouraging, although we admit that it took considerable time to find the best way to solve the problems. With the experience gained here, we hope to have good recipes to tackle new problems efficiently.

Future applications could address stationary HD and MHD wind solutions, together with time accurate calculations of jet-type flows. In stellar wind modeling, one may again expect stagnation problems when the simulated flow region contains both subsonic and supersonic flow regions. For jets, one must be able to resolve discontinuities. The model problems clearly indicated how stagnation is avoided and that implicit schemes are able to deal with shock waves.

Acknowledgements. This work was sponsored by the Dutch National Computing Facilities Foundation (NCF) for the use of supercomputer facilities. G.T. thanks Prof. D. Molteni and Dr. D. Innes for suggesting and discussing test problems 1 and 4, respectively. G.T. also acknowledges partial support from the Hungarian Science Foundation, grant F 017313. R.K. thanks FOM, M.A.B. gratefully acknowledges collaboration with G.L.G. Sleijpen and H.A. van der Vorst.

References

Boris J.P., Book D.L., 1973, Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works. J. Comput. Phys., 11, 38

Botchev M.A., Sleijpen G.L.G., van der Vorst H.A., 1997, Low-Dimensional Krylov subspace iterations for enhancing stability of time-step integration schemes. Department of Mathematics, Utrecht University, Preprint 1004, March

Brackbill J.U., Barnes D.C., 1980, The effect of nonzero $\nabla \cdot \boldsymbol{B}$ on the numerical solution of the magnetohydrodynamic equations. J. Comput. Phys., 35, 426

Chakrabarti S.K., 1989, ApJ, 347, 365

Chakrabarti S.K., Molteni D., 1993, ApJ, 417, 671

Collela P., Woodward P.R., 1984, The piecewise parabolic method (PPM) for gas-dynamical simulations. J. Comput. Phys., 54, 174

Harten A., 1983, High resolution schemes for hyperbolic conservation laws. J. Comput. Phys., 49, 357

Hairer E., Nørsett S.P., Wanner G., 1987, Solving Ordinary Differential Equations I. Nonstiff Problems. Springer series in Computational Mathematics, Springer–Verlag, vol. 8

Keppens R., Tóth G., Botchev M.A., van der Ploeg A., 1997, Adding Implicit and Semi-Implicit Schemes to the Versatile Advection Code. submitted to J. Comput. Phys. (Paper I)

Molteni D., Lanzafame G., Chakrabarti S.K., 1994, ApJ, 425, 161–170

Paczýnski B., Wiita P.J., 1980, A&A, 88, 23

Petschek H.E., 1964, Magnetic Field Annihilation. In: AAS/NASA Symposium on the Physics of Solar Flares. NASA Spec. Publ., SP-50, p. 425–439

Powell K.G., 1994, An approximate Riemann solver for magnetohydrodynamics (that works in more than one dimension). ICASE Report No 94-24, Langley, VA

Roe P.L., 1981, Approximate Riemann solvers, parameter vectors, and difference schemes. J. Comput. Phys., 43, 357

Saad Y., Schultz M.H., 1986, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 7(3), 856-869

Strang G., 1968, On the construction and comparison of difference schemes. SIAM J. Numer. Anal., 5, 506

Tóth G., 1996, A general code for modeling MHD flows on parallel computers: Versatile Advection Code. Astrophys. Lett. & Comm., 34, 245

Tóth G., 1997, Versatile Advection Code. In: Hertzberger B., Sloot P. (eds.) High Performance Computing and Networking Europe 1997 (Vienna, Austria). Lecture Notes in Computer Science, Springer-Verlag, vol. 1225, p. 253–262

Tóth G., Odstrčil D., 1996, Comparison of some Flux Corrected Transport and Total Variation Diminishing Numerical Schemes for Hydrodynamic and Magnetohydrodynamic Problems. J. Comput. Phys., 128, 82

Yee H.C., 1989, A class of high-resolution explicit and implicit shock-capturing methods. NASA TM-101088

van der Vorst H.A., 1992, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Statist. Comput., 13, 631-644

van der Ploeg A., 1996, Reordering strategies and LU-decomposition of block tridiagonal matrices for parallel processing. Technical Report NM-R9618, Centrum voor Wiskunde en Informatica, October

van der Ploeg A., Keppens R., Tóth G., 1997, Block Incomplete LU-preconditioners for Implicit Solution of Advection Dominated Problems. In: Hertzberger B., Sloot P. (eds.) High Performance Computing and Networking Europe 1997, (Vienna, Austria) Lecture Notes in Computer Science, Springer-Verlag, vol. 1225, p. 421-430

van Leer B., 1979 Towards the ultimate conservative difference scheme. V. A Second order sequel to Godunov's method. J. Comput. Phys., 32, 101