Marshall Plan Scholarship
Final Report

Exponential integrators for Vlasov-type equations

Lukas Einkemmer

Introduction

The following presentation is divided into three chapters.

In the first chapter an exposition of the research that has been conducted while at the University of California is given. It is the intention to submit this as a paper to an applied mathematics journal (such as Computer Physics Communication, for example).

The second chapter outlines possibilities of further research that are well suited to be conducted in the framework of the collaboration with M. Tokman and others from the University of California. Furthermore, a review of literature that is relevant for that endeavor is provided.

In the third chapter (the appendix) we summarize and explain in more detail some plasma physics concepts that are necessary to understand this report. It is our hope that this makes the work presented here more easily accessible and reasonably self-contained.

The references are collected at the end of each chapter.

CHAPTER 1

Preprint "An evaluation of the performance of exponential integrators for magnetohydrodynamics"

An evaluation of the performance of exponential integrators for magnetohydrodynamics

Lukas Einkemmer[a,*], Mayya Tokman[b]

[a] *Department of Mathematics, University of Innsbruck, Austria*
[b] *University of California, Merced, United States of America*

Abstract

In this paper we consider the application of exponential integrators to three different problems of plasma physics. These simulation are carried out using the resistive magnetohydrodynamics (MHD) equations. Compared to the previous examples that have been investigated in the literature, the resistive MHD equation are formulated as partial differential equations in 8 variables that include a physically important, but relatively weak, diffusion.

The performance of the fifth-order EpiRK method developed in Tokman 2010 is compared to that of the CVODE library. It is found that the exponential integrator provides equal or superior performance in most circumstances. Furthermore, we study the dependence of the performance characteristics as a function of the the resistivity (the inverse of the Lundquist number).

*Corresponding author
Email addresses: lukas.einkemmer@uibk.ac.at (Lukas Einkemmer), mtokman@ucmerced.edu (Mayya Tokman)

1. Introduction

The method of choice for the time integration of stiff ordinary differential equations and the spatial semi-discretization of partial differential equations are implicit methods, such as the Backward difference formula (BDF). This is due to the fact that for such problems explicit integrators are usually forced to take excessively small time steps.

However, in recent years exponential integrators have emerged as a promising alternative for the time integration of partial differential equations. A significant body of research has been accumulated in which these methods are investigated both from a theoretical (see e.g. [7] for a review article) as well as from a numerical point of view (see e.g. [18] and [9]). This research includes the construction of exponential integrators that have been tailored to a given differential equation (see e.g. [5]) as well as such integrators that can be applied to a class of problems (see e.g. [6]). For example, in [9] the Exponential Integrator Collection (EPIC), a framework written in C++, is used to demonstrate that in many test problems superior or equal performance can be achieved by a class of exponential integrators that are called EpiRK methods, as compared to the BDF implementation found in the CVODE library [1].

It is generally believed that in real world applications, except in the case where hand tailored integrators can be constructed for a given problem, exponential integrators are most promising, as a viable candidate to outperform explicit methods, if no good preconditioners are available.

However, most numerical work conducted in the literature do consider test problems of small to medium complexity (see, for example, [9], [8], or [2]) where in many instances efficient preconditioners can be constructed. It is therefore the goal of this paper to apply the exponential integrators found in the EPIC library to the problem of solving the resistive magnetohydrodynamics (MHD) equations. Preconditioners for such systems have, for example, been investigated in [13] and . It is found that the performance gain from

preconditioning in the two and a half-dimensional examples considered there is rather modest under most circumstances.

Our goal in this paper is to compare the performance of the exponential integrators in the EPIC library to that of the BDF scheme that is implemented in the CVODE library. Due to the discussion above we will perform this comparison using the CVODE library but without any preconditioner.

In section 2 we provide an introduction to exponential integrators in general as well as to the specific EpiRK method that is implemented in EPIC. In section 3 we describe the MHD equations which are solved numerically in section 4 for a reconnection problem, in section 5 for the Kelvin-Helmholtz instability, and for an arcade model in the section (6). Implementation details are discussed in section 7. Finally, we conclude in section 8.

2. Exponential integrators

Let us consider the following initial value problem

$$\begin{aligned} y'(t) &= F(y(t)) \\ y(0) &= y_0 \end{aligned} \tag{1}$$

which is assumed to be large and stiff. Before preceding to state the class of numerical methods we are interested in, let us extract the the Jacobian $J(y(t)) = D_y F(y(t))$ from the right hand side of (1); this gives

$$y'(t+h) = F(y(h)) + J(y(t))\left[y(t+h) - y(t)\right] + R(y(t+h)),$$

where the remainder is given by

$$R(y(t+h)) = F(y(t+h)) - F(y(t)) - J(y(t))\left[y(t+h) - y(t)\right].$$

Now using the Gröbner-Alekseev formula, we obtain

$$y(t+h) = y(t) + \left(e^{hJ} - I\right)(hJ)^{-1} hF(y(t)) + \int_0^1 e^{hJ(1-s)} R(y(t+sh))\,\mathrm{d}\theta.$$

This representation is still exact. However, to obtain a practical numerical method the integral has to be approximated in a suitable manner. This is the essential idea of exponential integrators. By using a polynomial approximation we obtain a linear combination of entire functions defined by

$$\begin{aligned}\varphi_0(hJ) &= e^{hJ} \\ \varphi_k(hJ) &= \int_0^1 e^{(1-\theta)hJ} \frac{\theta^{k-1}}{(k-1)!}\,\mathrm{d}\theta \quad \text{for } k \geq 1.\end{aligned}$$

Thus, to implement exponential integrators efficiently we have to compute the application of the $\varphi_k(hJ)$ functions to a (large) vector. In order to accomplish this goal, a number of algorithms have been proposed in the literature (see e.g. [7]).

The most common approach is to approximate the application of the φ_k functions to a vector b by a projection on a Krylov subspace. Thus, let us define the Krylov subspace of dimension m by

$$K_m(hJ, b) = \operatorname{span}\left\{b, (hJ)b, \ldots, (hJ)^{m-1}b\right\}.$$

As a next step we have to compute an orthonormal basis of $K_m(hJ, b)$. This can be done, for example, by using the Arnoldi iteration algorithm. The result of that algorithm is a matrix V_m that consists of columns which form a basis of $K_m(hJ, b)$. The product of a matrix function applied to a vector, i.e. $\varphi_k(hJ)b$, can then be approximated as follows

$$f(hJ)b \approx V_m \left(V_m^T \varphi_k(hJ) V_m\right) V_m^T b.$$

Furthermore, we approximate the term in parenthesis by $f(H_m)$, where $H_m = hV_m^T JV_m$. This yields the final approximation

$$f(hJ)b \approx V_m \varphi_k(H_m) V_m^T b.$$

Note that H_m is a matrix of dimension $m \times m$ and therefore the application of $\varphi_k(H_m)$ to a vector can be easily computed by using any of the standard methods (such as Pade approximation or methods based on polynomial interpolation). For a more detailed discussion see [7], for example.

An alternative to that procedure is to employ a polynomial approximation of φ_k. To that end we have to chose m interpolation nodes on some compact set $K \subset \mathbb{C}$. Chebyshev nodes are an obvious choice, however, they suffer from the disadvantage that to compute the interpolation for $m+1$ points we have to reevaluate all the matrix-vector products already computed. To remedy this, Leja points have been proposed which share many of the favorable properties of the Chebyshev nodes but can be generated in sequence.

The advantage of interpolation at Leja points is that less memory is used as the matrix V_m has to be stored in memory. This makes it an attractive alternative for computer systems where memory is limited (such as graphic processing units, see [2]). The main disadvantage of the method, however, is that (at least) some approximation of the spectrum of J has to be available.

In this paper we will exclusively use the Krylov based approach. However, for more detailed information on the usage of polynomial interpolation for the computation of matrix functions in the context of high performance computing see [10], for example.

Similar to explicit Runge-Kutta methods, a general EpiRK method can be

written down as

$$Y_i = y_n + a_{i1}\psi_{i1}(g_{i1}hJ_n)hF_n + \sum_{j=2}^{i-1} a_{ij}\psi_{ij}(gIjhJ_n)h\Delta^{(j-1)}R(y_n),$$

$$y_{n+1} = y_n + b_1\psi_{s1}(g_{s1}hJ_n)hF_n + \sum_{j=2}^{s} b_j\psi_{sj}(g_{ij}hJ_n)h\Delta^{(j-1)}R(y_n)$$

where the stage index i ranges from 1 to $s-1$ and the $\psi_{ij}(z)$ functions are linear combinations of the φ_k functions given in the following form

$$\psi_{ij}(z) = \sum_{k=1}^{s} p_{ijk}\varphi_k(z).$$

The number of stages is given by s and the divided differences $\Delta^{(h-1)}R(y_n)$ can be computed by using the nodes $y_n, Y_1, \ldots, Y_{s-1}$.

As in the Runge-Kutta case, to construct an efficient integrator the coefficients a_{ij}, g_{ij}, b_j and p_{ijk} have to be chosen subject to the appropriate order conditions. In this paper, we use the EpiRK5P1 method that has been derived in [18] (which is employed in the performance comparison given in [9]) and uses the Adaptive Krylov algorithm from [12] and a variable time stepping approach. The corresponding coefficients are listed in Table 7 [9].

3. The resistive magnetohydrodynamics equations

The most fundamental theoretical description of a classical plasma comes from the kinetic equation. This so called Vlasov equation (the collisionless case) or Boltzmann equation (when collisions are of physical significance) describes the time evolution of a particle-density in the 3 + 3 dimensional phase space (the first three dimensions correspond to the space dependence while the remaining do correspond to the velocity dependence of the particle-density). While a number of simulations of different plasma phenomena have

been conducted within this approach, due to the high dimensional phase space a lower-dimensional approximation is usually used to render such simulations feasible. A similar discussion holds true for the gyrokinetic approximation that can be employed in plasmas where a strong external magnetic field along a given axis is present (in this case the phase space of the Vlasov equation is reduced to $3 + 2$ dimensions by averaging over the gyro motion; this procedure yields a good approximation under the assumption of low frequency as compared to the cyclotron frequency).

However, in many applications (such as magnetic confined fusion, spheromak experiments, and astrophysical plasmas) the timescales of interest are sufficiently long and/or the full three dimensional model is necessary to model physical phenomena. In that case the kinetic approach is usually not feasible (even on modern day supercomputers) and thus further simplifications have to be introduced. For the magnetohydrodynamics (MHD) equations, which we will describe in the remainder of this section, the assumption is made that, to a good approximation, the distribution in the velocity direction is Maxwellian; that is, that each sufficiently small volume in the plasma is in thermodynamic equilibrium.

For a more detailed discussion of gyrokinetic models, see e.g. [4] or [3]. For a general overview of kinetic and MHD models (including a derivation of the MHD equations from the Vlasov equation) see e.g. [11]. Numerical computations in the context of the MHD model discussed in this paper are performed, for example, in [14], [15], and [13].

If the assumption is made that the plasma considered is in thermodynamic equilibrium, the equations of motion in a three dimensional phase space (a so called fluid model) can be derived. These are given by (in dimensionless

units)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{2}$$

$$\rho \frac{\partial \boldsymbol{v}}{\partial t} + \rho \boldsymbol{v} \cdot \nabla \boldsymbol{v} = \boldsymbol{J} \times \boldsymbol{B} - \nabla p \tag{3}$$

which we refer to as the continuity and momentum equation respectively. They are described in terms of the density ρ, the fluid velocity $\boldsymbol{v}$, the magnetic field $\boldsymbol{B}$, the electric current density $\boldsymbol{J}$, and the pressure p. The dynamics is determined by the Lorentz force (the $\boldsymbol{J} \times \boldsymbol{B}$ term) and the pressure gradient force (the ∇p term). These fluid equations have to be coupled to an appropriate model of the electric field. Note, however, that if the (ideal) Ohm's law is assumed to hold, i.e.

$$\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B} = 0,$$

then the electric field $\boldsymbol{E}$ can be eliminated from Maxwell's equation. This leaves us with the following three equations

$$\begin{aligned} \frac{\partial \boldsymbol{B}}{\partial t} &= -\nabla \times (\boldsymbol{v} \times \boldsymbol{B}) \\ \boldsymbol{J} &= \nabla \times \boldsymbol{B} \\ \nabla \cdot \boldsymbol{B} &= 0. \end{aligned} \tag{4}$$

The first equation yields the time evolution of $\boldsymbol{B}$, the second can be used to eliminate $\boldsymbol{J}$ from equation (3), and the third is the familiar selenoidal constraint imposed on the magnetic field.

A commonly employed approach to close these equations (see e.g. [15]) is to supplement them with the following equation of state

$$e = \frac{p}{\gamma - 1} + \frac{\rho}{2} v^2 + B^2,$$

where the time evolution of the energy density e is given by

$$\frac{\partial e}{\partial t} = \nabla \cdot \left((e + p + \tfrac{1}{2}B^2)\boldsymbol{v} - \boldsymbol{B}(\boldsymbol{B} \cdot \boldsymbol{v}) \right). \tag{5}$$

Collectively, the equations (2)-(5) in the variables $(\rho, \boldsymbol{v}, \boldsymbol{B}, e)$ yield a first-order system of 8 differential equations in 8 variables and are called the ideal magnetohydrodynamics equations (or the ideal MHD equations).

For the purpose of performing the spatial discretization, these equations are often cast into the so-called divergence form. Then, the equations of motion read as (see e.g. [15])

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0$$

with state vector

$$U = \begin{bmatrix} \rho \\ \boldsymbol{v} \\ \boldsymbol{B} \\ e \end{bmatrix}$$

and

$$F(U) = \begin{bmatrix} \rho\boldsymbol{v} \\ \rho\boldsymbol{v} \otimes \boldsymbol{v} + (p + \frac{1}{2}B^2)I - \boldsymbol{B} \otimes \boldsymbol{B} \\ \boldsymbol{v} \otimes \boldsymbol{B} - \boldsymbol{B} \otimes \boldsymbol{v} \\ (e + p + \frac{1}{2}B^2)\boldsymbol{v} - \boldsymbol{B}(\boldsymbol{B} \cdot \boldsymbol{v}) \end{bmatrix}, \tag{6}$$

where we have denoted the tensor product by using the $\otimes$ symbol.

In this paper, as in [14] and [15], we will consider a slightly more general class of equations which, in addition to the dynamics discussed so far, include dissipative effects (due to particle collisions in the plasma). To that end the hyperbolic flux vector $F(U)$ is extended in [14] by a diffusive part given by (where we have assumed a spatially homogeneous viscosity μ, resistivity η,

and thermal conductivity κ)

$$F_d(U) = \begin{bmatrix} 0 \\ Re^{-1}\boldsymbol{\tau} \\ S^{-1}\left(\eta\nabla B - \eta(\nabla B)^{\mathrm{T}}\right) \\ Re^{-1}\boldsymbol{\tau}\cdot\boldsymbol{v} + \frac{\gamma}{\gamma-1}Re^{-1}Pr^{-1}\nabla T + S^{-1}\left(\frac{1}{2}\nabla(\boldsymbol{B}\cdot\boldsymbol{B}) - B(\nabla\boldsymbol{B})^{\mathrm{T}}\right) \end{bmatrix} \tag{7}$$

with

$$\boldsymbol{\tau} = \nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T - \tfrac{2}{3}\nabla\cdot\boldsymbol{\tau}I,$$

where the Reynolds number is given by $Re = \rho_0 v_A l_0/\mu$, the Lundquist number by $S = \mu_0 v_A l_0/\eta$ and the Prandtl number by $Pr = c_p\mu/\kappa$, for a characteristic density ρ_0, a characteristic length scale l_0, and the Alfven velocity v_A (as usual the permeability of free space is denoted by μ_0, $\kappa = 5/3$, and c_p is the specific heat of the fluid).

Note that in all the simulation conducted dimensionless units are used. Thus, we can rewrite (7) more conveniently in terms of the (dimensionless) viscosity $\mu = Re^{-1}$, the (dimensionless) resistivity $\eta = S^{-1}$, and the (dimensionless) thermal conductivity $\kappa = Pr^{-1}$. Then, the form of the resistive MHD equations used for the spatial discretization is

$$\frac{\partial U}{\partial t} + \nabla\cdot F(U) = \nabla\cdot F_d(U)$$

with $F(U)$ given by equation (6) and where $F_d(U)$ is given by

$$F_d(U) = \begin{bmatrix} 0 \\ \mu\boldsymbol{\tau} \\ \eta\left(\nabla B - (\nabla B)^{\mathrm{T}}\right) \\ \mu\boldsymbol{\tau}\cdot\boldsymbol{v} + \frac{\gamma\mu\kappa}{\gamma-1}\nabla T + \eta\left(\frac{1}{2}\nabla(\boldsymbol{B}\cdot\boldsymbol{B}) - B(\nabla\boldsymbol{B})^{\mathrm{T}}\right) \end{bmatrix}.$$

For the comparison that is conducted here, we employ the MHD code developed in [14]. This code has been used to conduct plasma physics simulations

(see e.g. [14], [15], and [13]) as well as to construct more efficient preconditioners in the context of implicit time integrators (see [15]). The code employs a finite difference approximation of order two and is constructed in such a way that if the selenoidal property, i.e. $\nabla \cdot \boldsymbol{B} = 0$, is satisfied for the initial value, then this is true for all later times. In [15] this is shown to be true for an implict method with a matrix-free inexact Newton–Krylov algorithm. However, the simulation conducted here do suggest that this behavior is also true if an exponential integrator is used for the time integration.

4. The reconnection problem

The examples in this and the next section are drawn from [14] and [15], respectively. We start with a reconnection problem for which the initial value of the magnetic field is given by

$$B_0(x, y, z) = \begin{bmatrix} \tanh(2y) - \psi_0 k_y \cos(k_x x) \sin(k_y y) \\ \psi_0 k_x \sin(k_x x) \cos(k_y y) \\ 0 \end{bmatrix},$$

where as in [14] we have chosen $k_x = \pi/x_r$, $k_y = \pi/(2y_r)$, $\psi_0 = 0.1$, and the computational domain is given by $[-x_r, x_r] \times [-y_r, y_r]$ for $x_r = 12.8$ and $y_r = 6.3$. Essentially this implies that the magnetic field reverses direction from pointing along $\boldsymbol{e}_x$ to pointing along $-\boldsymbol{e}_x$ abruptly at $y = 0$. Furthermore, we impose a density that is given by

$$\rho = 1.2 - \tanh^2(2y)$$

and a pressure that is proportional to the density; to be more precise $p = 0.5\rho$ (from which the energy is determined that is employed as an independent variable in the computation). A vanishing velocity in both space directions is prescribed.

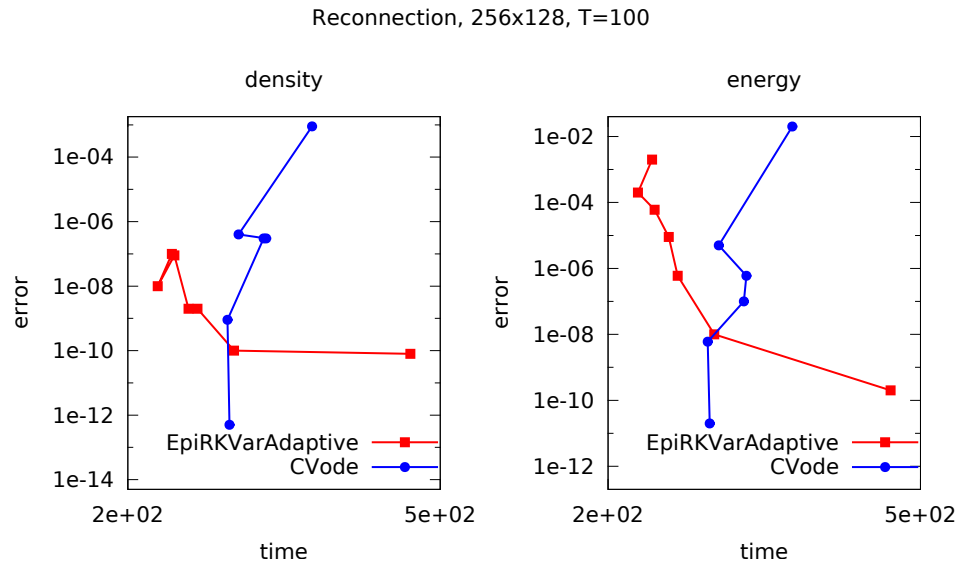

Figure 1: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time of $T = 100$. We have used 256 grid points in the x-direction and 128 grid points in the y-direction (this corresponds to the configuration investigated in [14]). The viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resistivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

First, we consider the configuration that is investigated in [14]; that is, we employ 256 grid points in the x-direction and 128 grid points in the y-direction. From the numerical simulations conducted, we draw the conclusion that for the problem under consideration the error in the energy (or equivalently the error in the pressure) dominates the error present in the remaining variables. Furthermore, we note that the performance of the exponential integrator is superior to the BDF method employed in the CVODE library if the desired accuracy is larger than 10^{-8} with respect to the energy or larger than 10^{-10} with respect to the density (see 1).

Next, let us conduct a simulation with 256 grid points in both the x- as well as the y-direction. In that case, as can be seen from Figure 2, the performance of the exponential integrator is comparable to that of the CVODE

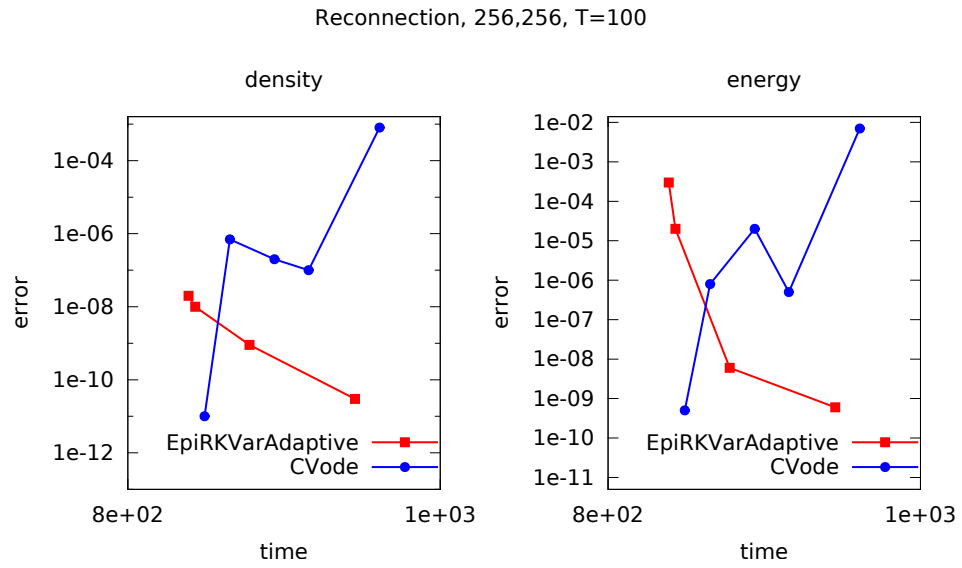

Figure 2: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time of $T = 100$. We have used 256 grid points in both spatial directions. The viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resistivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

implementation. Note, that in the exponential integrator case the run time and precision does actually increase if the prescribed tolerance is increased whereas almost the exact opposite can be observed for the CVODE implementation. This is, most likely, a consequence of the adaptive time stepping algorithm used.

Now, let us increase the stiffness of the problem by choosing a space discretization with 512 grid points in the x-direction. In that case we have to employ a relatively small tolerance in order to obtain a method that is numerically stable. However, if this stability is achieved the computed solution is correct within an error of 10^{-6} for the energy and 10^{-8} for the density (see Figure 3).

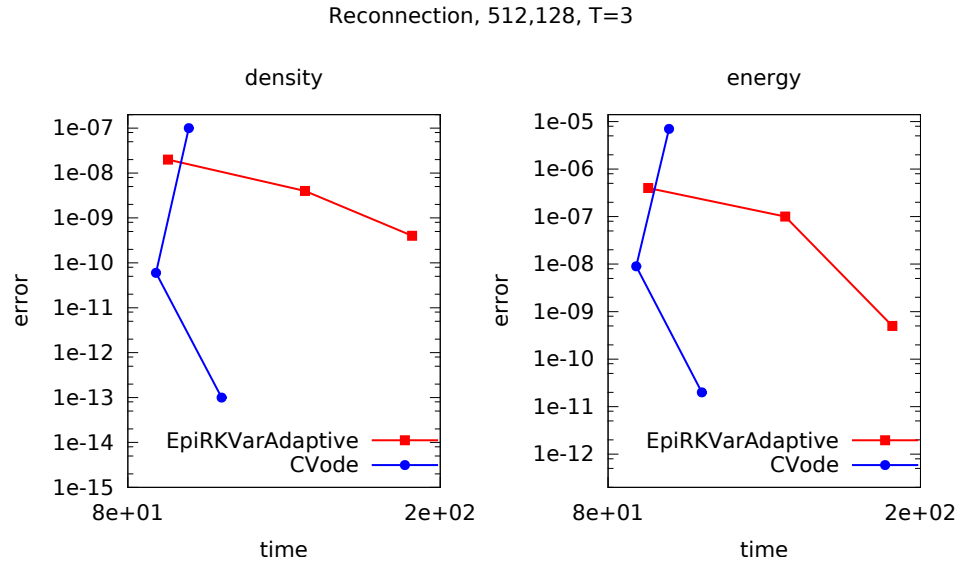

Figure 3: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time $T = 3$. We have used 512 grid points in the x-direction and 128 grid points in the y-direction. The viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resistivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

From Figure 1 and 2 we can deduce that the exponential integrator shows superior performance as compared to the CVODE implementation, if the states of relatively low accuracy are accessible. Since this is not the case in the simulation under consideration (where 512 grid points have been used in the x-direction), we might expect that there is no region where the exponential integrators yields superior performance. This is confirmed by the results displayed in Figure 3. However, let us note that for the accuracy requirement of most practical simulation the performance of both integrators is almost identical.

To conclude this section let us investigate the dependence of the (relative) performance of the exponential integrator under consideration on the

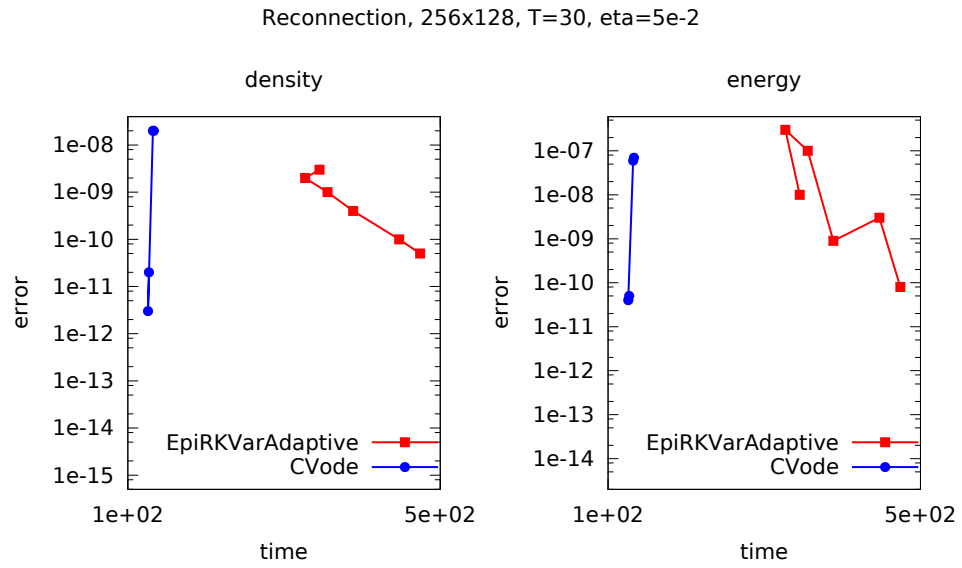

Figure 4: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time $T = 30$. We have used 256 grid points in the x-direction and 128 grid points in the y-direction. The viscosity is given by $\mu = 5 \cdot 10^{-1}$, the resisitivity by $\eta = 5 \cdot 10^{-2}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

resistivity η. In all the simulations conducted, we have chosen a viscosity that is an order of magnitude smaller than the resistivity (i.e. $\mu = 10^{-1}\eta$). Furthermore, we have chosen to keep the thermal conductivity fixed.

From the results displayed in Figure 4 (using a resistivity $\eta = 5 \cdot 10^{-2}$ and a viscosity $\mu = 5 \cdot 10^{-1}$) we see that for small Reynolds and Lundquist numbers the CVODE implementation does retain a significant advantage in run time as compared to the EpiRK method.

As the Reynolds and Lundquist numbers are increased, the relative performance of the exponential integrator does increase significantly and a perfor-

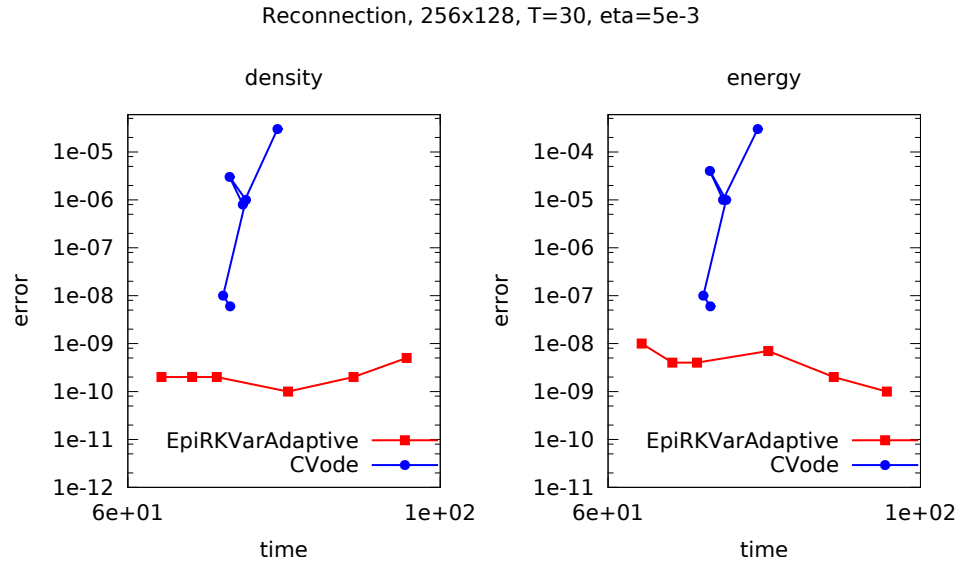

Figure 5: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time $T = 30$. We have used 256 grid points in the x-direction and 128 grid points in the y-direction. The viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resisitivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

mance that exceeds or matches that of the BDF implementation found in the CVODE library is observed (see Figures 5 and 6).

5. The Kelvin-Helmholtz instability

As a second example we consider the Kelvin-Helmholtz instability. The KH instability considered here is triggered by superimposing the following perturbation

$$\epsilon_x \cos\left(\tfrac{2\pi\omega_x}{L}x\right) + \epsilon_y \sin\left(\frac{2\pi\omega_y}{L}x\right)$$

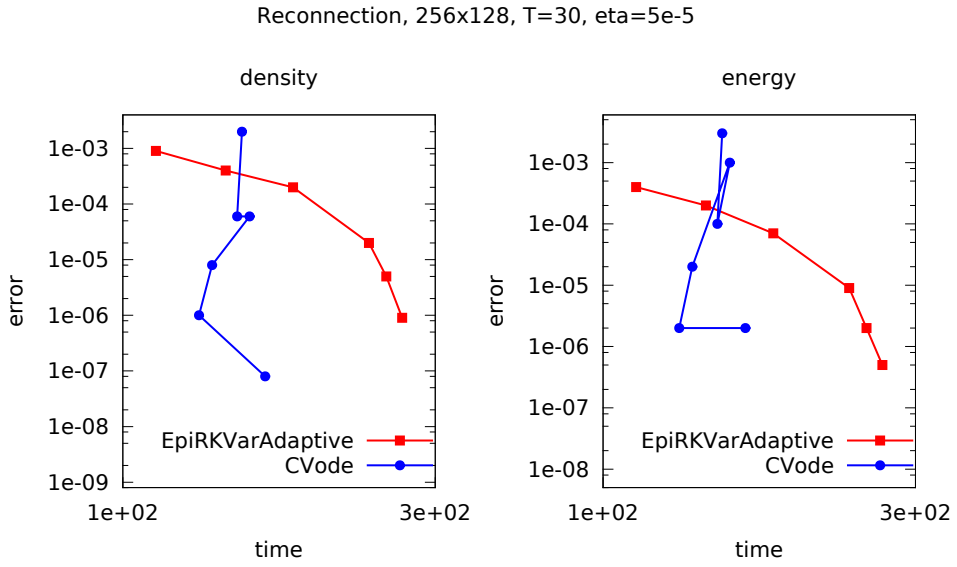

Figure 6: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time $T = 30$. We have used 256 grid points in the x-direction and 128 grid points in the y-direction. The (dimensionless) viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resisitivity by $\eta = 5 \cdot 10^{-5}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

parameter	value
ϵ_x, ϵ_y	0.1
v_0	0.5
ω_x, ω_y	2
pressure p	0.25
B_z	10
B_x	0.1

Table 1: Parameters for the initial value of the Kelvin-Helmholtz instability.

in the x-direction on the velocity field

$$\boldsymbol{v} = \begin{bmatrix} v_0 \tanh(\frac{y}{\lambda}) \\ 0 \\ 0 \end{bmatrix}.$$

The density is initialized to unity and a uniform pressure is chosen. The magnetic field is initialized to be uniform in both the x- and z-direction with strength B_x and B_z, respectively, and is assumed to vanish in the y-direction. All the parameters used to determine the numerical value of the initial value are listed in Table 1.

The result of the numerical simulation for 256 grid points is shown in Figure 7. We observe that the run time of the the CVODE implementation is about 25% faster as compared to the exponential integrator. In addition, we note that this difference in performance is consistent no matter what accuracy is desired.

If the number of grid points is increased, a similar picture is obtained (see Figure 8 for the case with 512 grid points).

A further investigation, however, reveals that in the case of the Kelvin-Helmholtz instability turning off the adaptive algorithm described in [12]

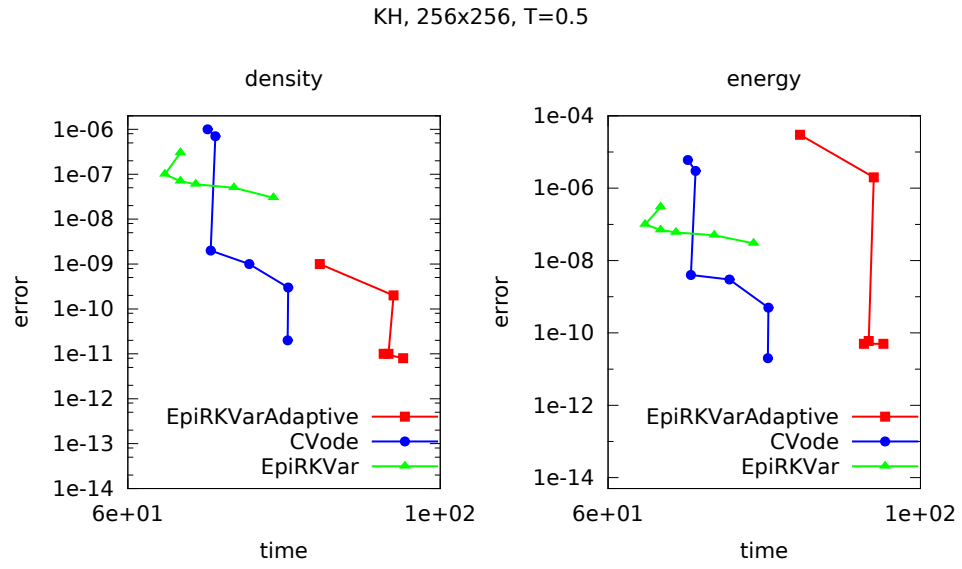

Figure 7: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time $T = 0.5$ and 256 grid points are used in both the x- and y-direction. The (dimensionless) viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resistivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

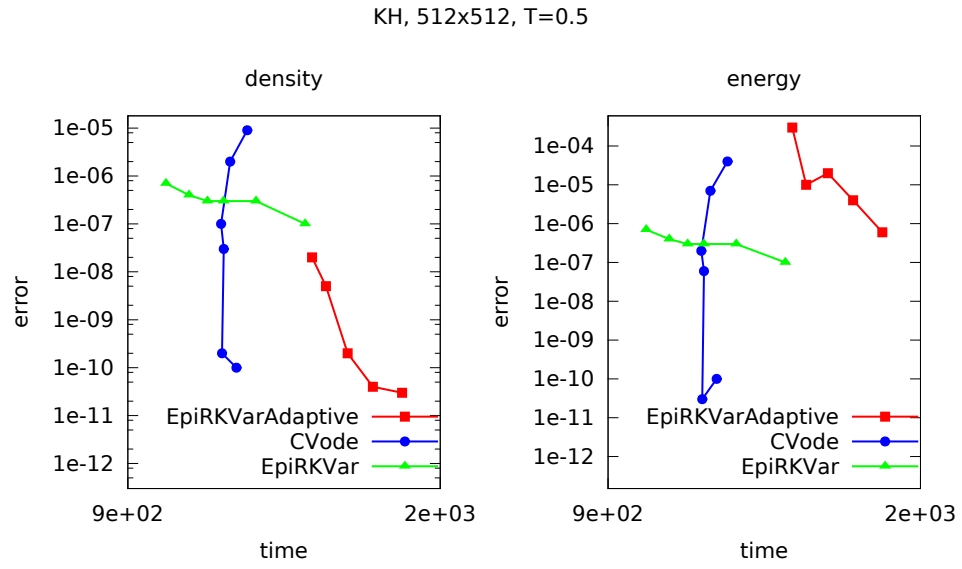

Figure 8: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-14}) is shown as a function of the run time. The simulation is conducted up to a final time $T = 0.5$ and 256 grid points are used in both the x- and y-direction. The (dimensionless) viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resistivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

does result in significantly improved performance for tolerances of up to 10^{-6} (both in density as well as in energy). The increase in run time for higher accuracy is, as is to be expected, significantly steeper and thus the non-adaptive version is obviously ill suited for higher precision requirements. However, this clearly indicates that the degraded performance of the adaptive EpiRK method, as compared to the reconnection problem described in the previous section, is due to an an insufficiency in the adaptive algorithm as opposed to a deficiency of the exponential integrator under consideration.

To conclude this section, let us duly note that this is in contrast to the numerical simulations conducted for the reconnection problem (as stated in the previous section). In case of the reconnection problem we observe that the adaptive algorithm does in fact result in a significant speedup that enables the EpiRK method to achieve superior performance as compared to the CVODE implementation.

6. A model involving time dependent boundary conditions

The third example is a two-dimensional model of a solar arcade (see [16] and [17] for the extension to a three dimensional model). The initial magnetic field is given by

$$\boldsymbol{B}_0(x,y) = \begin{bmatrix} B_0 e^{-k_1 y}\cos(k_1 x) \\ -B_0 e^{-k_1 y}\sin(k_1 x) \\ B_z \end{bmatrix},$$

where $B_0 = 1$, $B_z = 0.1$, and $k_1 = 2\pi/L_y$. Furthermore, we assume a unit density and a vanishing velocity field. The computational domain is $[-L_x, L_x] \times [-L_y, L_y]$ for $L_x = 1$ and $L_y = 6$.

Note that compared to [16], the model we solve here is more involved in that it takes into account the pressure as well as diffusive effects. For the initial value we chose $p = 1$ which corresponds to a significantly higher beta than is usual in astrophysical plasmas. This is necessary, as due to the structure of

the initial value, which does not take pressure effects into account, a realistic value value for astrophysical plasmas (i.e. $\beta \approx 10^{-2}$) would result in a pressure that is too small to be handled correctly by our code. The configuration presented here, nevertheless, is an interesting and sufficiently complex test case for the time integrators considered in this paper.

However, what most distinguishes this model from the previous two we have so far considered is that we impose a time dependent boundary condition on the lower boundary. More specifically, we assume that

$$v_1 = -V_0 f(t) \sin(k_1 x),$$

where $V_0 = 0.01$, $f(t) = t/t_R$ for $t < t_r$, and $f(t) = 1$ otherwise. In addition, we have set $t_r = 200$.

Snapshots of the time evolution of the magnetic field in this configuration are presented in Figure 9. We see that the arcade structure present in the initial configuration does undergo a relatively slow evolution until approximately $t = 12$, where the formed magnetic island (the structure where the magnetic field lines do form a loop) does detach from the bottom boundary and travels upwards.

Note that due to the higher plasma beta, a significant faster time evolution than reported in [16] is observed. Nevertheless, both configurations exhibit similar dynamics in that at some point in time the arcade does detach from the bottom boundary.

Even though the magnetic field does represent the physically interesting quantity in this observation, we remark that the error made in the magnetic field is always lower than both the error of the density and the energy. Therefore, we once again use the density and the energy as our variables to investigate the performance of the CVODE algorithm as compared to our exponential integrator.

The result of the simulation for 128 grid points in both the x- and the y-

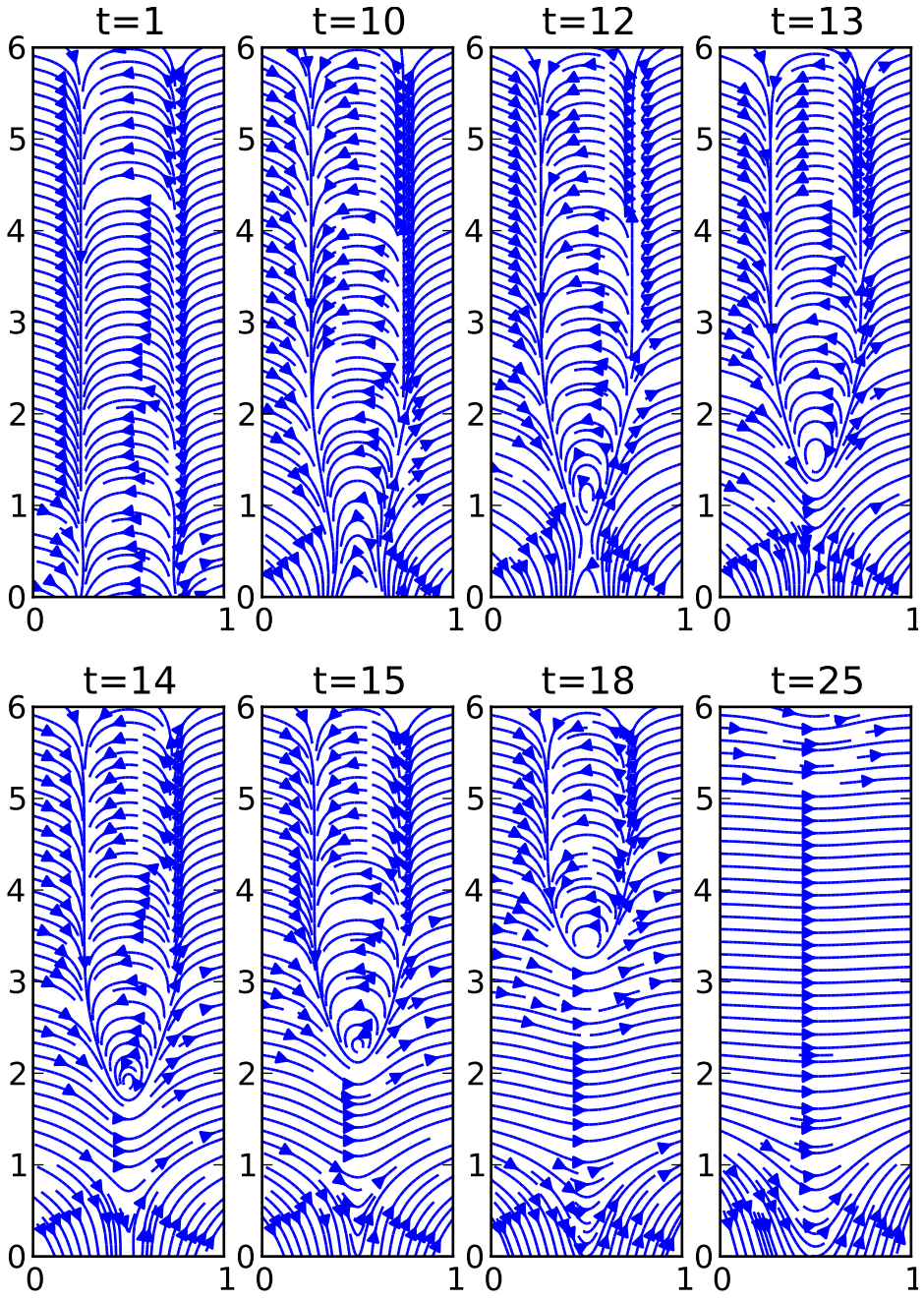

Figure 9: The field lines of the magnetic field in the $x-y$ plane are shown for different times of the simulation.

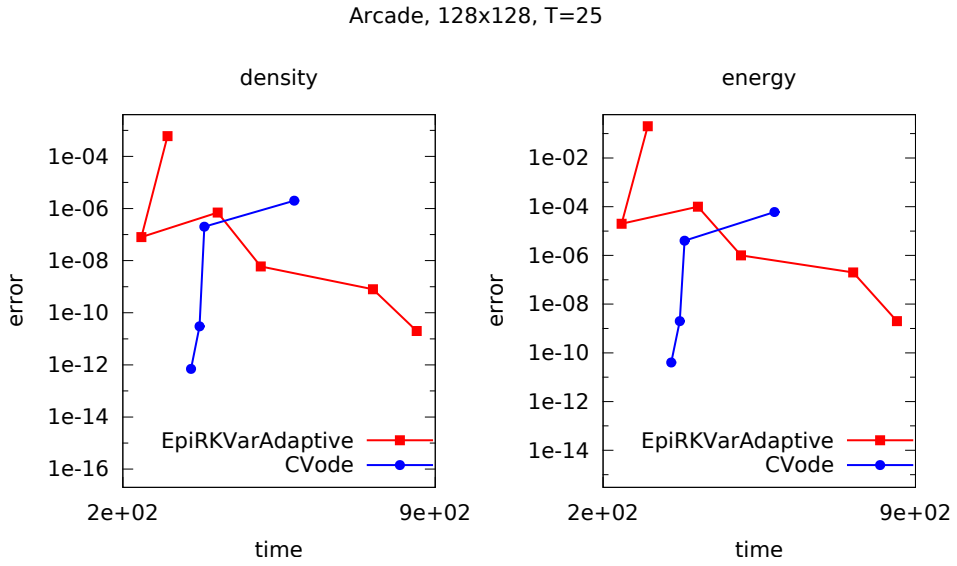

Figure 10: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-12}) is shown as a function of the run time. The simulation is conducted up to a final time of $T = 25$ and 128 grid points are used in both the x- and y-direction. The viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resistivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

direction is shown in Figure 10. In this instance, the run time of the CVODE implementation does not change significantly with a lowering of the achieved tolerance as long as at least an accuracy of 10^{-5} is required. This is in contrast to the EpiRK implementation where the run time increases, if increased accuracy is required. Note that as in the previous two examples we can observe that the exponential integrator does prove advantageous for low precision requirements.

Now, let us consider the same problem with 256 grid points in both the x- and the y-direction. In that case similar results are obtained (see Figure 11) where as before exponential integrators are superior in terms of run time consideration if an accuracy of 10^{-5} or less is required. If that is not the case, we observe that the curve of the CVODE implementation is almost vertical and thus a significant run time advantage results as compared to the

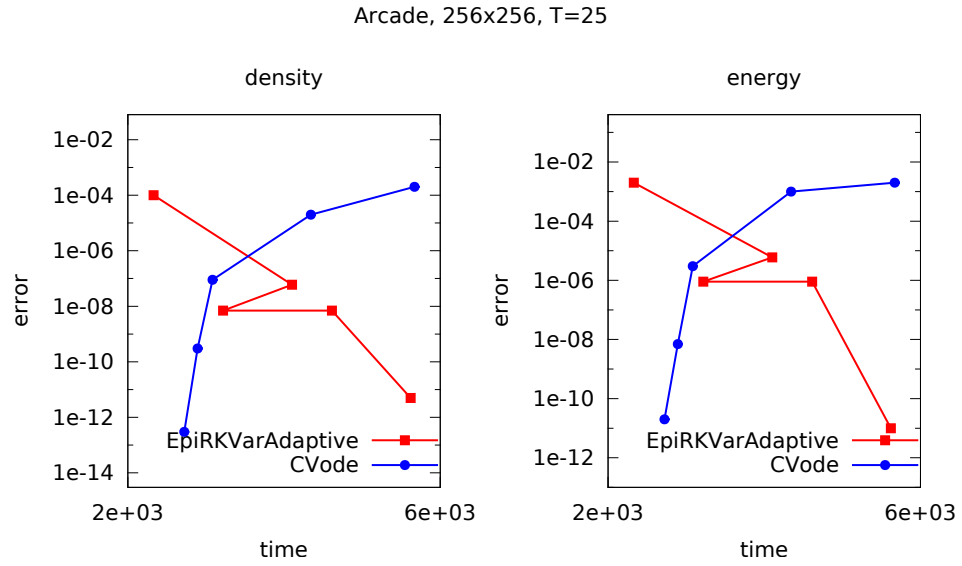

Figure 11: The error in the density as well as the energy (compared to a reference solution that is computed using CVODE with an absolute tolerance of 10^{-12}) is shown as a function of the run time. The simulation is conducted up to a final time of $T = 25$ and 256 grid points are used in both the x- and y-direction. The viscosity is given by $\mu = 5 \cdot 10^{-2}$, the resistivity by $\eta = 5 \cdot 10^{-3}$, and the thermal conductivity by $\kappa = 4 \cdot 10^{-2}$.

CVODE implementation.

7. Implementation

We have chosen to base our implementation on the 2.5 dimensional[1] MHD code developed by Daniel R. Reynolds et al.. A large number of numerical simulations have been conducted (see e.g. [14] and [15]) and the performance of preconditioners for Newton-Krylov based implicit methods has been investigated (see e.g. [15]). Furthermore, the extension of the code to three dimensional problems as well as to non-square geometries (such as a Tokamak geometry) have been investigated in [13].

[1] All quantities depend on only two spatial variables; however, the direction of the velocity as well as the magnetic fields are three dimensional vectors.

The code is implemented in the Fortran programming language and can be build using both the gfortran[2] as well as the Intel Fortran compiler. It employs the FCVODE module that provides a Fortran binding to the CVODE library.

The second component of our implementation is the EPIC library (see [9]) which implements a fifth-order EpiRK method using automatic step size control and an adaptive algorithm to evaluate the φ_k functions as described in [12]. This library is written in the C++ programming language and does use the Intel Math Kernel Library (Intel MKL).

We have chosen to make small modifications to the Fortran program to expose a clean C interface that computes the right hand side of the MHD equations under consideration. Then a C/C++ program was written that acts as a driver and controls the execution of the program employing either the EpiRK method or the CVODE library using the standard C interface.

One detail warrants further discussion: In the original Fortran program the CVODE interface is used to approximate the Jacobian by a simple forward difference stencil. In our implementation, however, we provide a custom function to compute an approximation to the Jacobian. This is necessary for the EPIC library, which has only been used in the context of problems where an exact Jacobian is available (as in [9]).

However, to compute an exact Jacobian for the significantly more complex MHD problem considered in this paper is infeasible. Therefore, we have to approximate $J(a)v$, i.e. the application of the Jacobian J at position a to a vector v, where both a and v do depend on the specific numerical algorithm as well as the initial values under consideration. We note here, that there is a qualitative difference in the norm of the vector v that depends on the numerical time integration scheme used. In the EpiRK the norm of v is close to unity whereas in the BDF method the value is often significantly below

[2]The Fortran compiler gfortran is part of the GNU Compiler Collection.

$\sqrt{\epsilon}$ (where we use $\epsilon \approx 10^{-16}$ to denote machine precision). Therefore, care is taken to scale vectors to the norm $\sqrt{\epsilon}$ only if the initial norm is above $\sqrt{\epsilon}$ in magnitude. Then the same implementation can be used for both methods and a difference in performance due to an internally optimized function to compute the Jacobian (as is provided by the CVODE library) is precluded.

In all the computations conducted in this paper, we have used the infinity norm to scale v to its appropriate size. In general, we have found that the performance and accuracy of the computation is not significantly altered if the scaling is performed to some value that is reasonably close to $\sqrt{\epsilon}$.

8. Conclusion

We have shown that for the magnetohydrodynamics (MHD) problems considered in this paper, exponential integrators do constitute a viable alternative to the more commonly employed BDF method (as implemented in the CVODE library). In almost all instances equal or superior performance has been observed for the adaptive and variable time stepping fifth order EpiRK method for low to medium accuracy requirements, despite the fact that the EPIC package is a relatively straight forward implementation and therefore, does not include excessive computer science optimizations. This is in contrast to the CVODE library which has been in active development for an extended period of time.

However, the results presented here, especially in the context of the Kelvin-Helmholtz instability, do suggest that an improvement in the adaptive algorithm could further increase the performance of the EpiRK method investigated in this paper.

Furthermore, we note that the results presented here is somewhat different from the simpler models considered in [9]. In that case order of magnitude speedups are observed for the EpiRK method as compared to the CVODE implementation. Whether it is generally true that the relative performance of

exponential integrators degrades if more complex problems are investigated or if that is a due to the specific system of partial differential equations considered here, does require numerical simulation of a sufficiently complex problem that is drawn from a different domain. Also it is not clear if the approximate computation of the Jacobian does contribute to the observed differences (in all the problems investigated in [9] an analytical form of the Jacobian has been used).

Acknowledgements

We would like to take the opportunity to thank J. Loffeld for providing the code of the EPIC library and D. R. Reynolds for providing the code of the Fortran MHD solver. Furthermore, we would like to thank both J. Loffeld and D. R. Reynolds for the helpful discussion.

The first author was supported by the Marshall scholarship of the Austrian Marshall plan foundation (http://www.marshallplan.at/) and by the Fonds zur Förderung der Wissenschaften (FWF) – project id: P25346.

This work was supported by the Austrian Ministry of Science BMWF as part of the UniInfrastrukturprogramm of the Focal Point Scientific Computing at the University of Innsbruck.

References

[1] S.D. Cohen and A.C. Hindmarsh. CVODE , a stiff/nonstiff ODE solver in C. *Computers in physics*, 10(2), 1996.

[2] L. Einkemmer and A. Ostermann. Exponential integrators on graphic processing units. *High Performance Computing and Simulation HPCS , 2013 International Conference on*, 2013.

[3] M.R. Fahey and J. Candy. GYRO : A 5-D Gyrokinetic-Maxwell Solver. *Proceedings of the ACM/IEEE SC2004 Conference*, 2008.

[4] E. Frieman and L. Chen. Nonlinear Gyrokinetic Equations For Low-frequency Electromagnetic Waves In General Plasma Equilibria. *Phys. Fluids*, 25, 1982.

[5] M. Hochbruck and C. Lubich. Exponential integrators for quantum-classical molecular dynamics. *BIT Numerical Mathematics*, 39(4), 1999.

[6] M. Hochbruck and A. Ostermann. Explicit exponential Runge–Kutta methods for semilinear parabolic problems. *SIAM J. Numer. Anal.*, 43(3), 2005.

[7] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19(1), 2010.

[8] C. Klein and K. Roidot. Fourth order time-stepping for Kadomtsev-Petviashvili and Davey-Stewartson equations. *SIAM J. Sci. Comput.*, 33(6), 2011.

[9] J. Loffeld and M. Tokman. Comparative performance of exponential , implicit , and explicit integrators for stiff systems of ODEs. *J. Comput. Appl. Math.*, 2012.

[10] A. Martinez, L. Bergamaschi, M. Caliari, and M. Vianello. A massively parallel exponential integrator for advection-diffusion models. *Journal of computational and applied mathematics*, 231(1), 2009.

[11] D.R. Nicholson. *Introduction to plasma theory.* Cambridge University Press, 1983.

[12] J. Niesen and W.M. Wright. Algorithm 919: a Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators. *ACM Transactions on Mathematical Software TOMS*, 38(3), 2012.

[13] D.R. Reynolds, R. Samtaney, and H.C. Tiedeman. A fully implicit Newton–Krylov–Schwarz method for tokamak magnetohydrodynamics:

Jacobian construction and preconditioner formulation. *Computational Science & Discovery*, 5(1), 2012.

[14] D.R. Reynolds, R. Samtaney, and C.S. Woodward. A fully implicit numerical method for single-fluid resistive magnetohydrodynamics. *J. Comput. Phys.*, 219(1), 2006.

[15] D.R. Reynolds, R. Samtaney, and C.S. Woodward. Operator-based preconditioning of stiff hyperbolic systems. *SIAM J. Sci. Comput.*, 32(1), 2010.

[16] M. Tokman. *Magnetohydrodynamic modeling of solar magnetic arcades using exponential propagation methods*. PhD thesis, 2001.

[17] M. Tokman and P. Bellan. Three-dimensional Model of the Structure and Evolution of Coronal Mass Ejections. 1, 2000.

[18] M. Tokman, J. Loffeld, and P. Tranquilli. New adaptive exponential propagation iterative methods of Runge–Kutta type EPIRK. *submitted , available online at http://faculty. ucmerced. edu/mtokman/publications*, 2010.

CHAPTER 2

Future directions of research

1. Scaling of the EPIC library

Even though the results published in [8] do suggest that the EPIC library does scale at least as well as the CVODE framework for a range of test problems would suggest that similar results can be obtained for the MHD equations.

However, this is not the case as can be seen from Figure 1 and 2, however, we observe that the weak scaling behavior deviates quite significantly from that of CVODE.

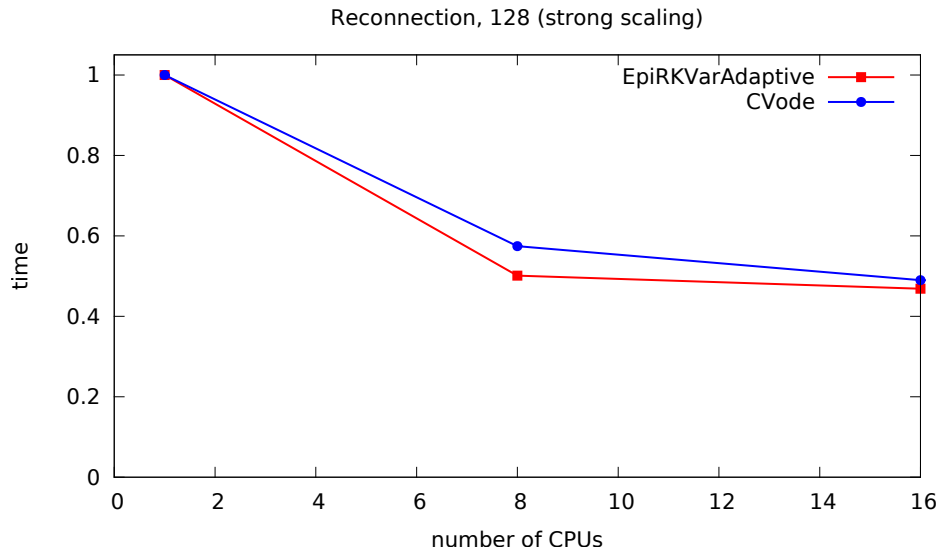

Figure 1: The weak scaling of the reconnection problem for 128 grid points in both the x- and y- directions for up to 16 cores is shown.

Furthermore, for the Kelvin-Helmholtz instability this issues significantly worsens (see Figure 3).

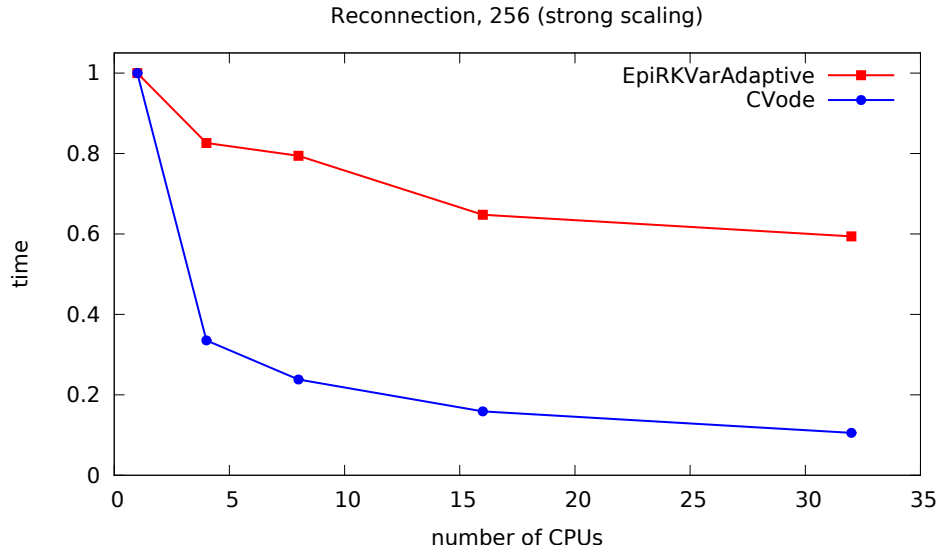

Figure 2: The weak scaling of the reconnection problem for 128 grid points in both the x- and y- directions for up to 32 cores is shown.

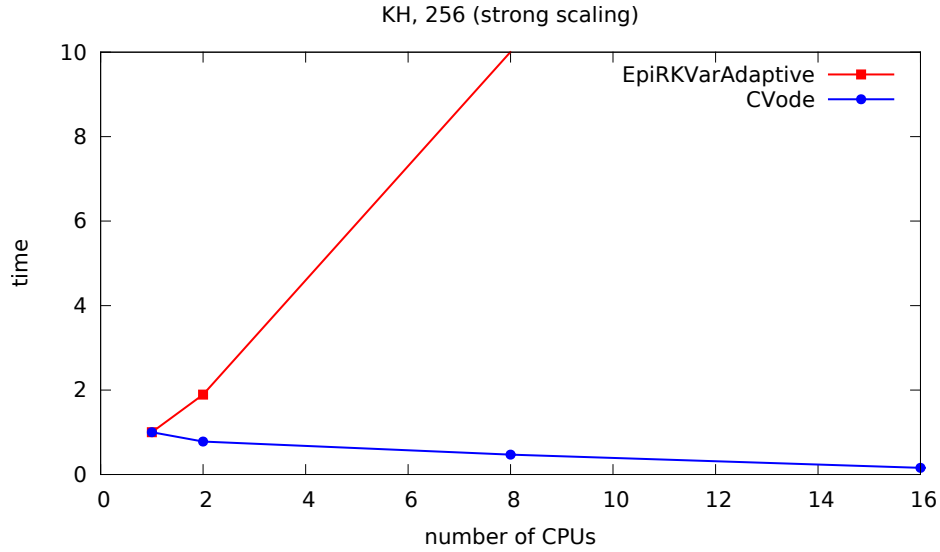

Figure 3: The strong scaling of the Kelvin-Helmholtz problem for 256 grid points in both the x- and y- directions for up to 16 cores is shown.

In this case the time steps significantly decrease as a function of the number of cores which results in a scheme that is not competitive. Note, however, that this is not purely a problem within the automatic step size control, i.e. the step is actually necessary to ensure numerical stability.

Since parallelization is of vital importance if more realistic plasma physics problems are to be tackled, the exact cause of this phenomenon is currently under investigation.

2. A literature overview of the physics of coronal mass ejections

Numerical simulations conducted in the literature (in the context of coronal mass ejections) can be divided into two different categories:

- to construct a three dimensional representation of the dependent variables (density, velocity field, magnetic field, and pressure) from measurement data. Such data are usually available in the form of line-of-sight magnetograms that give the magnetic field at the surface of the sun (there physical effects such as the Zeeman splitting of iron in combination with the resulting electromagnetic emissions can be used to determine the field strength).

- to investigate possible plasma configurations that exhibits both the relatively long stability of solar arcades, while still providing a mechanism that can result in fast eruptions.

The first problem is intimately connected with the question of what (quasi-)equilibrium states are accessible to a plasma. In the seminal paper of Taylor [10], the class of linear force free state has been identified as the result of relaxation under constraints. Taylor was able to explain the reversed field topology that appears naturally in some toroidal plasmas.

The idea has been extended to a number of long lived plasma states. However, the validity of this approach has been called into questions. First, on the ground that the application of low plasma beta, while usually valid in the corona, might not be applicable above 1.2 solar radii (see [4]). In a number of papers the approach was therefore extended to either nonlinear force free states or states that are more general than linear force free states (such as linear combination of these). See, for example, [9],[2], [5], [12], and [3]. It has been found that for model problems the magnetic fields could be determined accurately even if only the lower boundary condition was specified. It is, however, unclear if these model, which usually include a number of parameters that are fitted in order to obtain the correct data on the domain boundary, provide any explanatory power from a physical point or view.

On the other hand a number of MHD simulations and theoretical investigations have been conducted to model a number of arcade type structures in the context of solar plasmas. In [11] a time dependent MHD simulation gives similar results as compared to astronomical observations (assuming an initial flux tube that leads to a kink instability).

In [13] the free energy in bipolar and quadruple magnetic fields is investigated. It is found that states with energy twice as high as for open field lines (which have previously considered to be the upper limit for certain classes of fields). In [2] the possibility that arcade type magnetic fields are in fact MDR (minimum dissipation rate) states (in a two fluid model) is investigated. The paper demonstrates the plausibility using an order of magnitude estimation but does not provide any numerical simulation and no follow up papers are found in the literature.

It is generally believed that reconnection does play a major role in the dynamics even at low plasma beta. However, there is a significant discrepancy in the speed of the dynamics that is observed in astronomical observations and the one obtained from numerical simulations (see, for example, [14] for a review article). In that context the reconnection of different kinds of flux

tubes have been investigated (see [7] and [6]).

Even though in recent years more advanced MHD simulations have been attempted (see e.g. [1]) a thorough understanding of solar arcades has proved elusive.

References

[1] T. Amari, J. Aly, J. Luciani, Z. Mikic, and J. Linker. Coronal Mass Ejection Initiation by Converging Photospheric Flows: Toward a Realistic Model. *The Astrophysical Journal Letters*, 742(2), 2011.

[2] R. Bhattacharyya, M. Janaki, B. Dasgupta, and G. Zank. Solar arcades as possible minimum dissipative relaxed states. *Sol. Phys.*, 240(1), 2007.

[3] P. Chen. Coronal mass ejections: models and their observational basis. *Living Rev. Solar Phys.*, 8, 2011.

[4] G.A. Gary. Plasma beta above a solar active region: Rethinking the paradigm. *Sol. Phys.*, 203(1), 2001.

[5] G.A. Gary. Evaluation of a Selected Case of the Minimum Dissipative Rate Method for Non-Force-Free Solar Magnetic Field Extrapolations. *Sol. Phys.*, 257(2), 2009.

[6] M. Linton. Reconnection of nonidentical flux tubes. *Journal of Geophysical Research: Space Physics 1978–2012*, 111(A12), 2006.

[7] M. Linton and S. Antiochos. Magnetic flux tube reconnection: tunneling versus slingshot. *The Astrophysical Journal*, 625(1), 2005.

[8] J. Loffeld and M. Tokman. Comparative performance of exponential , implicit , and explicit integrators for stiff systems of ODEs. *J. Comput. Appl. Math.*, 2012.

[9] C.J. Schrijver, M.L. Derosa, T.R. Metcalf, Y. Liu, J. Mctiernan, S. Régnier, G. Valori, M.S. Wheatland, and T. Wiegelmann. Nonlinear force-free modeling of coronal magnetic fields part I: A quantitative comparison of methods. *Sol. Phys.*, 235(1-2), 2006.

[10] J.B. Taylor. Relaxation of toroidal plasma and generation of reverse magnetic fields. *Phys. Rev. Lett.*, 33(19), 1974.

[11] T. Török and B. Kliem. Confined and ejective eruptions of kink-unstable flux ropes. *The Astrophysical Journal Letters*, 630(1), 2005.

[12] T. Wiegelmann and T. Sakurai. Solar Force-free Magnetic Fields. *arXiv preprint arXiv:1208.4693*, 2012.

[13] R. Wolfson, C. Drake, and M. Kennedy. Maximizing magnetic energy storage in the solar corona. *The Astrophysical Journal*, 750(1), 2012.

[14] M. Yamada, R. Kulsrud, and H. Ji. Magnetic reconnection. *Rev. Modern Phys.*, 82(1), 2010.

Appendix

1. Tensor manipulation

First, let us note that (where ϵ_{ijk} is the Levi-Civita symbol and we employ the Einstein summation convention for repeated indices)

$$\begin{aligned}\nabla \times (\boldsymbol{v} \times \boldsymbol{B}) &= \nabla \times (\epsilon_{.jk} v_j B_k)_i \\ &= \epsilon_{imn}\epsilon_{njk}\partial_m v_j B_k \\ &= \epsilon_{imn}\epsilon_{njk} v_j \partial_m B_k + \epsilon_{imn}\epsilon_{njk} B_k \partial_m v_j\end{aligned}$$

now separating the two cases we get

$$\epsilon_{imn}\epsilon_{njk} v_j \partial_m B_k = \epsilon_{imn}\epsilon_{jkn} v_j \partial_m B_k = v_i \partial_j B_j - v_j \partial_j B_i$$

and

$$-\epsilon_{imn}\epsilon_{kjn} B_k \partial_m v_j = -B_i \partial_j v_j + B_j \partial_j v_i$$

Since

$$\nabla \cdot (\boldsymbol{v} \otimes \boldsymbol{B} - \boldsymbol{B} \otimes \boldsymbol{v})_i = \partial_j (v_i B_j - B_i v_j) = v_i \partial_j B_j + B_j \partial_j v_i - B_i \partial_j v_j - v_j \partial_j B_i$$

we follow that

$$\nabla \cdot (\boldsymbol{v} \otimes \boldsymbol{B} - \boldsymbol{B} \otimes \boldsymbol{v}) = \nabla \times (\boldsymbol{v} \times \boldsymbol{B}).$$

Similar derivations can be conducted for the other equations of motion.

2. Parameters in a coronal plasma

From measurement data it is determined that the solar corona has a temperature of approximately 10^7 Kelvin and a density of 10^9 electrons/cm^3

(see, for example, [2]). Assuming that the plasma consists mainly of hydrogen atoms and using the quasi-neutrality assumption gives a density of $1.6 \cdot 10^{-20} kg/cm^3$.

To cast this quantities in the dimensionless units used in the actual simulation we have to determine a characteristic length and time scale. From [gary2009] we can use $l_0 = 10^9$ cm as the characteristic length scale (which is between the size of the entire domain of size 10^{10} cm and the resolution of a single pixel which is on the order of 10^8 cm). For the characteristic magnetic field strength $B_0 = 0.01$ T (also taken from [3]) we get the Alfven speed (see e.g. [4])

$$v_A = \frac{B_0}{\sqrt{\mu_0 \rho_0}} = 7 \cdot 10^7 m/s = 0.2c,$$

which we take as the characteristic speed. Then it is easy to determine the characteristic time to be (sometimes called the Alfven time)

$$t_A = \frac{l_0}{v_A} = 0.15\text{s}.$$

On the other hand the characteristic pressure has the same units ($\frac{kg}{ms^2}$ or $\frac{kg}{m^3}\frac{m^2}{s^2}$) as does $B_0^2/(2\mu_0)$. This gives a characteristic pressure

$$p_0 = 39.8 \text{ Pa}.$$

The pressure can be determined from the ideal gas law which yields (for the data used here)

$$p = 0.1 \text{ Pa}.$$

Note that since in the dimensionless units used in the simulations B_0^2 and p_0 have the same units as they represent the magnetic and kinetic pressure, respectively. As a consistency check we can thus compute the plasma beta

(the fraction of pressure to magnetic pressure) which is given by

$$\beta = \frac{p}{B_0^2} = 0.0025$$

which is on the order of 10^{-3} and thus consistent with the data reported for coronal plasmas (see e.g. [2]).

3. Stability of force free plasmas

If the pressure force is negligible (i.e. the plasma beta is much smaller than unity) then the only driving force in the MHD model is the Lorentz force. It is then argued in the physical literature (see e.g. [1], [5], and [6]) that the plasma will tend to an equilibrium state, i.e. a state where the Lorenz force vanishes. Such a state is called a force-free state and is characterized by

$$\boldsymbol{j} \times \boldsymbol{B} = 0. \tag{1}$$

Now, for the ideal MHD equations Ohm's law implies that

$$\boldsymbol{j} = \nabla \times \boldsymbol{B}$$

and thus we can rewrite equation (1) as

$$(\nabla \times \boldsymbol{B}) \times \boldsymbol{B} = 0$$

which implies that there exists an α such that

$$\begin{aligned} \nabla \times \boldsymbol{B} &= \alpha \boldsymbol{B} \\ (\nabla \alpha) \cdot \boldsymbol{B} &= 0. \end{aligned}$$

If the parameter α is homogeneous in space then we will it a *linear force free state* otherwise it is refereed to as a *non-linear force free state*.

In many instances we are further interested in determining if a force-free state $\boldsymbol{B} = \boldsymbol{B}_0 + \boldsymbol{B}_1$ is stable if a perturbation $\boldsymbol{B}_1$ is applied. To that end we investigate the change in energy a perturbation causes, i.e. if

$$W = \int \boldsymbol{B}^2 - \boldsymbol{B}_0^2 \, dV \tag{2}$$

is positive then the perturbation is stable. For the present discussion, we only consider linear stability. Linearizing Faraday's law of induction gives

$$\frac{\partial \boldsymbol{B}_1}{\partial t} = -\nabla \times (v_1 \times \boldsymbol{B}_0). \tag{3}$$

Since $\boldsymbol{B}_0$ is constant (in time) integrating equation (3) in time shows that the perturbation is of the form

$$\boldsymbol{B}_1 = -\nabla \times (\boldsymbol{\xi} \times \boldsymbol{B}_0),$$

for $\boldsymbol{\xi}$ appropriately defined. Substituting the formula for $\boldsymbol{B}_1$ into the change in energy (given by equation (2)) yields

$$W = \int (\nabla \times (\boldsymbol{\xi} \times \boldsymbol{B}_0))^2 - 2(\nabla \times (\boldsymbol{\xi} \times \boldsymbol{B}_0)) \cdot \boldsymbol{B}_0 \, dV.$$

Now let us assume that B_0 is a force free state; then after some algebraic manipulation (see e.g. [6]) we get

$$W = (\nabla \times (\boldsymbol{\xi} \times \boldsymbol{B}_0))^2 - (\nabla \times (\boldsymbol{\xi} \times \boldsymbol{B}_0)) \cdot (\boldsymbol{\xi} \times (\nabla \times \boldsymbol{B}_0)) \; dV$$

and for $\boldsymbol{A}_1 = \boldsymbol{\xi} \times \boldsymbol{B}_0$ we have

$$W = \int (\nabla \times \boldsymbol{A}_1)^2 - \alpha \boldsymbol{A}_1 \cdot (\nabla \times \boldsymbol{A}_1) \, dV.$$

From this expression a number of interesting consequences are immediate.

First, a state with α , which corresponds to a potential field, is stable. No such conclusion can be drawn in general for different force-free states. However, if we analyze the orders of magnitude we find that the first term scales as A_1^2/l^2, where l is a characteristic length scale, and the second term scales as $\alpha A_1^2/l$. Therefore, if

$$\alpha < \frac{1}{l}$$

then the corresponding force free state is stable. This observation is called Shafranov's limit.

4. The dependence of the dynamics on the initial pressure

In the paper we have presented numerical results for the arcade type initial that were computed using an initial pressure equal to unity. In fact, we would expect that if we decrease the initial pressure than the speed of the dynamics would decrease as well. To verify this we have chosen $p = 0.25$. The results are shown in Figure 1.

Furthermore, in Figure 2 we show a side by side comparison, up to the final time $t = 0$, for $p = 0.25$ and $p = 0.0625$. We can clearly see that the magnetic island detaches from the bottom at an earlier time in the latter case; thus confirming our previous observation that the speed of the dynamics is increased.

References

[1] R. Bhattacharyya, M. Janaki, B. Dasgupta, and G. Zank. Solar arcades as possible minimum dissipative relaxed states. *Sol. Phys.*, 240(1), 2007.

[2] G.A. Gary. Plasma beta above a solar active region: Rethinking the paradigm. *Sol. Phys.*, 203(1), 2001.

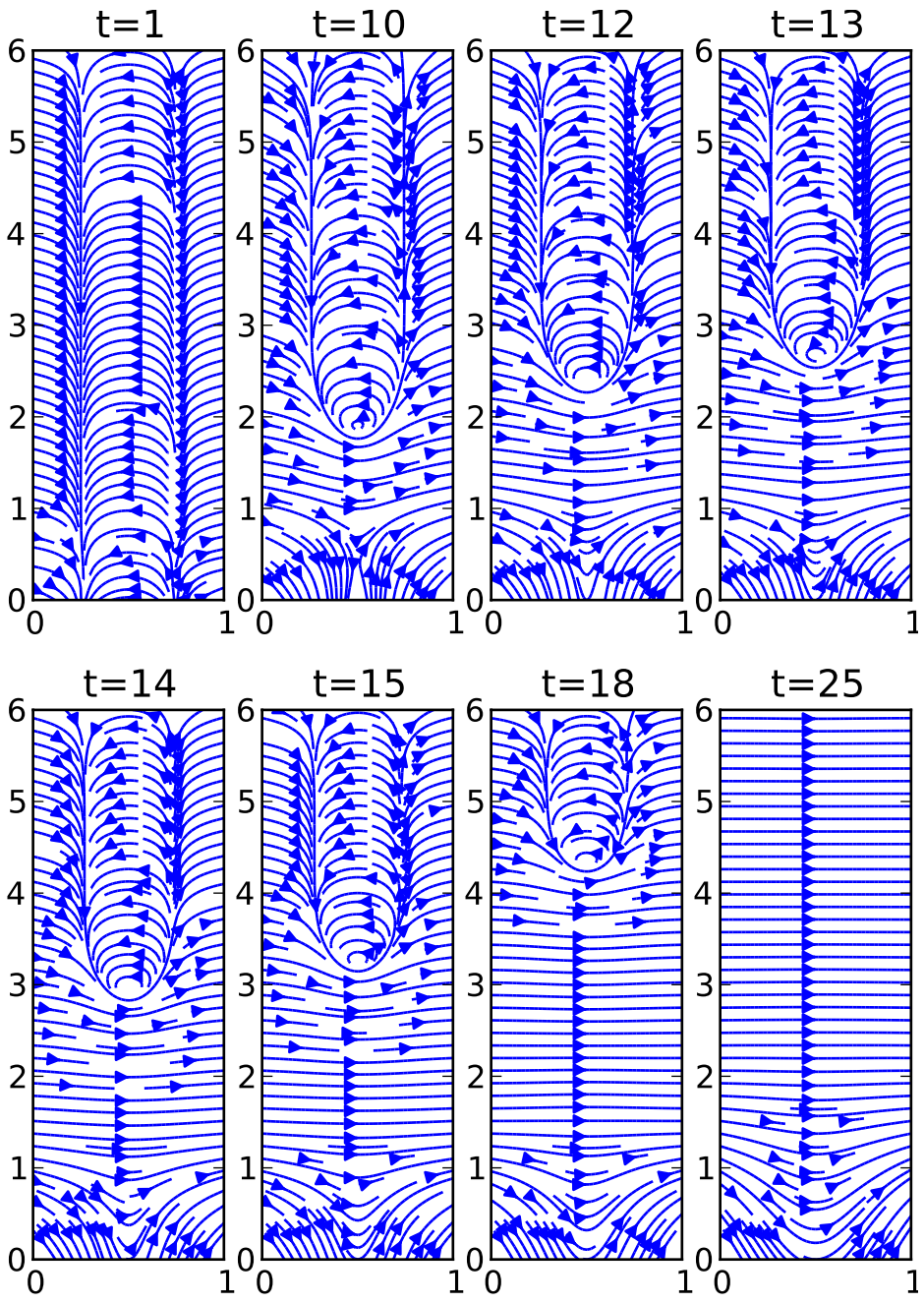

Figure 1: The snapshots in time are the same as those that are shown in the preprint. The initial value and all parameters are the same except that initial pressure of $p = 0.25$ was chosen.

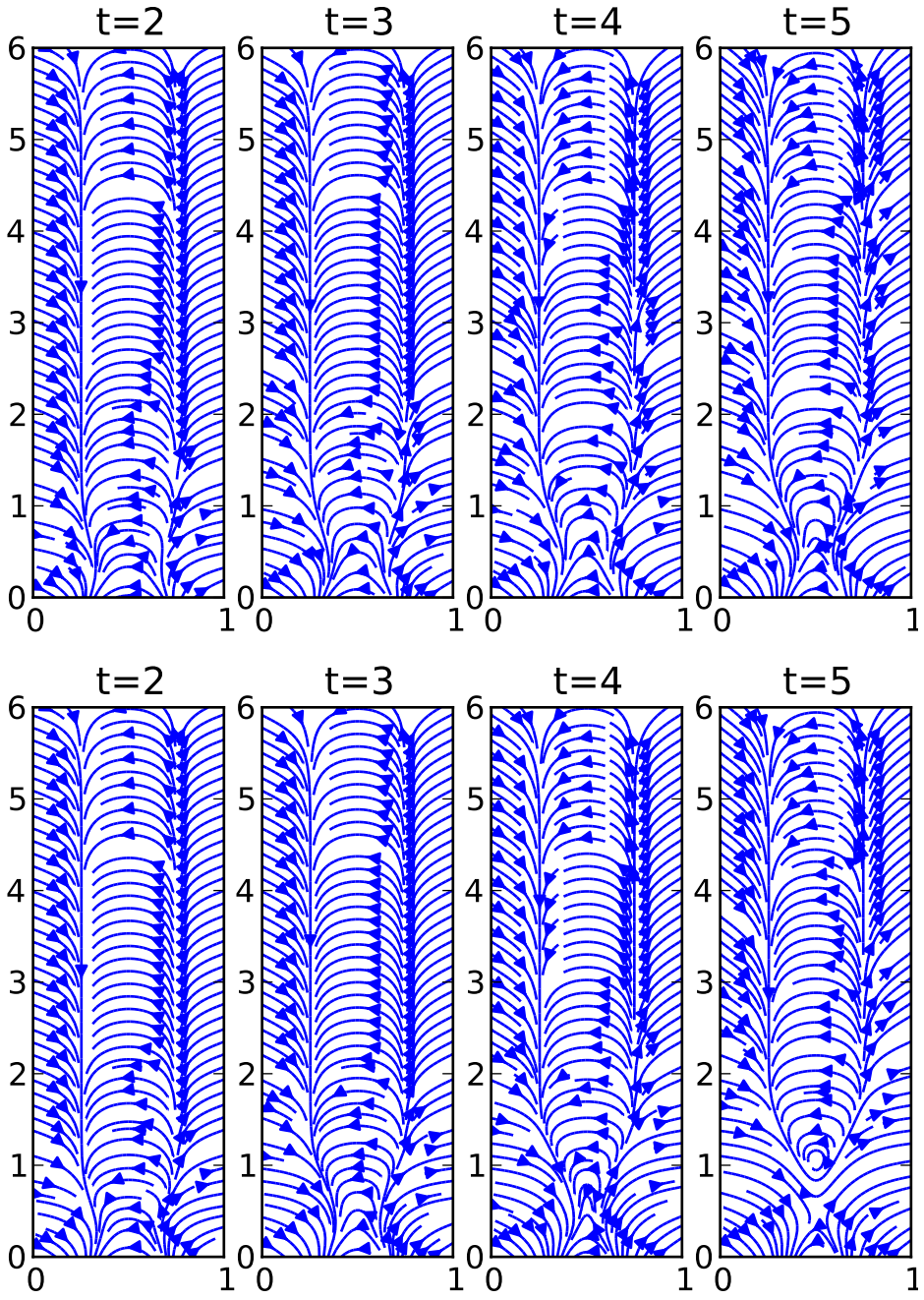

Figure 2: The snapshots in time for $t = 2$, 3, 4, 5 are shown for $p = 0.25$ (top) and $p = 0.0625$ (bottom) are shown. The initial value and all parameters are the same as those chosen in the preprint

[3] G.A. Gary. Evaluation of a Selected Case of the Minimum Dissipative Rate Method for Non-Force-Free Solar Magnetic Field Extrapolations. *Sol. Phys.*, 257(2), 2009.

[4] D.R. Nicholson. *Introduction to plasma theory.* Cambridge University Press, 1983.

[5] C.J. Schrijver, M.L. Derosa, T.R. Metcalf, Y. Liu, J. Mctiernan, S. Régnier, G. Valori, M.S. Wheatland, and T. Wiegelmann. Nonlinear force-free modeling of coronal magnetic fields part I: A quantitative comparison of methods. *Sol. Phys.*, 235(1-2), 2006.

[6] T. Wiegelmann and T. Sakurai. Solar Force-free Magnetic Fields. *arXiv preprint arXiv:1208.4693*, 2012.