Magnetohydrodynamic Modeling of Solar Magnetic Arcades Using Exponential Propagation Methods

A Thesis by

Mayya Tokman

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2001

(Submitted October 13, 2000)

Acknowledgements

I would like to thank my advisor Dan Meiron for providing a direction for my research and for all the help and support throughout my graduate student career. I am very grateful to him for proofreading my thesis and making many helpful suggestions. I would like to extend my gratitude to the faculty and staff of the Applied Mathematics department for teaching, helping and allowing me to pursue my research interests. I would like to thank the members of the defense committee Professors D. Meiron, P. Bellan, O. Bruno and T. Hou for their time (special thanks to Prof. Oscar Bruno for making the adjustments in his schedule).

My deepest gratitude goes to my "semi"-advisor Paul Bellan. I am greatly indebted to him for all the long hours spent discussing my research, for many valuable insights, which helped me find the right way, for encouraging me to present my research results at conferences and for teaching me a lot about science. The collaboration with Paul truly inspired me and I would like to thank him for making me a part of his research "family".

Throughout most of my stay at Caltech I was supported by the Department of Energy Computational Science Graduate Fellowship. I greatly appreciate the honor of being a fellow of this program and would like to thank the DOE and the staff of Krell institute, which administers the fellowship, for enhancing my graduate experience and giving me an opportunity to fully concentrate on my research.

My life at Caltech would be much less pleasant without my friends, who added a large positive constant to a cosine-like curve of a graduate student experience. Many thanks to my close friend and officemate of five years Gang Hu, my friend officemate and countryman in one, Yalchin Efendiev. I am grateful to Oleg Pariser for his help and support in my decision to come to Caltech. Big thanks to a fabulous system administrator and a good friend Chad Schmutzer, who made my computer life so much easier and provided a lot of much needed material for laughter decompression. And of course, the numerous scientific and very unscientific discussions at lunches, coffee breaks and dinner parties wouldn't be half as enjoyable without (in order of appearance) Danny Petrasek, John Pelesko, Patrick Guidotti, Randy Paffenroth, Dave Amudsen and Eusebius Doedel.

The deep belief in me from my family always supported me, for which I am eternally grateful. I feel very lucky to have the love of my wonderful mom Adel, my greatest sister Lena, my spirited grandmother Ida and my dear grandfather Shimon, who is responsible for my scientific inclinations. I would like to dedicate this work to two people who are and always will be a big part of who I am - my great grandmother and my father.

Finally, I would like to thank Dave Bernstein for his love and patience, for dinners brought late to my office and scientific and nonscientific debates, for being there and caring about me, and for making my life not always easier and much more interesting.

Abstract

Advanced numerical methods based on exponential propagation have been applied to magnetohydrodynamic (MHD) simulations. This recently developed numerical technique evolves the system of nonlinear equations using exponential propagation of the Jacobian matrix. The exponential of the matrix is approximated by projecting it onto the Krylov subspace using the Arnoldi algorithm. The primary advantage of the exponential propagation method is that it allows time steps exceeding the Courant-Friedrichs-Lewy (CFL) limit. Another important aspect is faster convergence of the iteration computing the Krylov subspace projection compared to solving an implicit formulation of the system with similar iterative methods. Since the time scales in the resistive MHD equations are widely separated the exponential propagation methods are especially advantageous for computing the long term evolution of a low-beta plasma. We analyze several types of exponential propagation methods and highlight important issues in the development of such techniques. Our analysis also suggests new ways to construct schemes of this type. Implementation issues, including scalability properties of exponential propagation methods, and performance are also discussed.

In the second part of this work we present numerical MHD models which are constructed using exponential propagation methods and which describe the evolution of the magnetic arcades in the solar corona. Since these numerical methods have not been used before for large evolutionary systems like resistive MHD, we first validate our approach by demonstrating application of the exponential schemes to two existing magnetohydrodynamic models. We simulate the reconnection process resulting from shearing the footpoints of two-dimensional magnetic arcades and compute the three-dimensional linear force-free states of plasma configurations. Analysis of these calculations leads us to new insights about the topology of the solutions. The final chapter of this work is dedicated to a new three-dimensional numerical model of the dynamics of coronal plasma configurations. The model is motivated by observations and laboratory experiments simulating the evolution of solar arcades. We analyze the results of numerical simulations and demonstrate that our numerical approach provides an accurate and stable way to compute the solution to the zero-β resistive MHD

system. Based on comparisons of the simulation results and the observational data we offer an explanation for the observed structure of eruptive events in the corona called coronal mass ejections (CME). We argue that the diversity of the images of CMEs obtained by the observational instruments can be explained as two-dimensional projections of a unique three-dimensional plasma configuration and suggest an eruption mechanism.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

As the complexity of problems tackled by scientists grows, computer simulations of physical processes become an ever more important part of research efforts in many scientific fields. A research project which integrates advanced numerical modeling with experimental and theoretical studies can bring especially valuable insights into the nature of physical processes. The main results of this thesis are twofold. First, we demonstrate how the recently introduced numerical technique of exponential propagation can be used to efficiently integrate large stiff systems of equations. Second, we present a theory of the solar magnetic arcades, motivated by computer simulations performed using these new numerical techniques, that describes the structure and evolution of active magnetic regions in solar atmosphere. In the first part of this thesis (Chapters 2,3) we provide a detailed description of the main building blocks of numerical exponential propagation methods, describe the advantages of this technique and demonstrate its performance on a test problem. Chapters 4 and 5 concentrate on numerical models of physical processes occurring in the plasma which comprises the solar atmosphere. While material in chapters 2 and 3 would be of interest to a reader with some background in numerical analysis, chapters 4 and 5 could be read independently if the primary subject of interest is the results of computer simulations of solar plasma configurations.

The primary objective of using exponential propagation methods is to avoid the restrictive Courant-Friedrichs-Lewy (CFL) condition which constrains the maximum allowed time step for explicit schemes [77]. This becomes important if a large system of differential equations is stiff. In this case the ratio of the largest to the smallest eigenvalues of the system's Jacobian is very large. Since the CFL condition is determined by the fastest modes of the system (these correspond to the largest eigenvalues) the time step in an explicit scheme has to be very small in order to obtain a stable method. In a system of partial differential equations the Jacobian comes from discretization of spatial differential operators, and the stiffness of the system grows as the spatial resolution is increased. Thus, the time step restriction becomes severe and, if the system has to be integrated over long periods of time, one is forced to seek an implicit numerical scheme which does not have the CFL restriction.

An integral part of an implicit scheme is solution of a linear system. Since the problems we are interested in are large, a Krylov-projection-based iterative method has to be used to invert a matrix in such a linear system [71]. Due to the stiffness of the Jacobian matrix we can expect the convergence of an iterative method to be very slow, so that the larger time step advantage of an implicit methods is overweighed by the number of iterations required to invert the matrix. Exponential propagation methods provide an alternative to avoid the limitations of both explicit and implicit methods. First, they allow a stable time integration to be performed with the time step greatly exceeding the CFL bound. Second, as was shown in [39], the convergence of the Krylov projection method used in an exponential propagation scheme exceeds that of the same Krylov projection technique used to invert a matrix in an implicit method. The latter statement will be clarified below as well as discussed in detail later in this work.

The two main concepts that the exponential numerical methods presented in this work are built upon are the exponential propagation and the Krylov approximation of functions of matrices. The former notion refers to the formal solution of the systems of equations describing time evolution of a physical process. If the system of first-order in time differential equations is written in an integral form, the dynamics of the problem is described by an exponential of the spatial differential operator integrated over the time interval of interest. In most cases the exponential cannot be computed or integrated exactly and has to be estimated using asymptotic techniques. The idea of computing the integral approximately and using the resulting formula as a numerical scheme to integrate the system of equations in time is not new and has been used to study small systems of differential equations as far back as the 1950's (see [17] and references therein). The use of this technique to integrate large systems of differential equations has been considered impractical, however, since computing an exponential of a general large non-symmetric matrix was prohibitively expensive. By the 1970's there were a number of algorithms proposed for calculating an exponential of a matrix. None of these methods, however, could significantly reduce the complexity of this task for problems with general large matrices [17]. The solution to this problem was found in the 1980's with the development of the Krylov subspace projection methods for large linear systems.

The Krylov subspace approximations to functions of matrices are based on an idea of projecting the function of a large matrix onto a small Krylov subspace, calculating this

function using the small projection matrix as an argument and then transforming the result back to the original linear space [71]. The advent of these methods revolutionized the solution of linear systems with large matrices. Initially they were used to solve linear systems of equations and to find eigenvectors of large matrices. In particular, Gear and Saad [32] used Krylov methods to solve the systems arising in the Newton iteration of implicit multistep schemes. In 1983 Nauts and Wyatt [60] successfully utilized a Krylov method for symmetric matrices (i.e. Lanczos algorithm) to compute the exponentials of discrete Hamiltonian operators for an application in chemical physics. Later this technique was used by Park and Light [63] to exponentially propagate the Schrödinger equation. The idea of approximating general functions of matrices using the Krylov subspace projection has also been proposed by Van der Vorst [21]. Combining exponential propagation and the Krylov methods for estimating the exponential of a large matrix was presented for the first time as a numerical technique to integrate general systems of non-linear ordinary differential equations by Friesner *et al.* [29]. They later extended this method to systems of partial differential equations [23] with particular application to the Navier-Stokes system. Gallopoulos and Saad [30] presented their version of these methods for linear parabolic equations with a forcing term and proved some results about the accuracy and stability of this type of exponential propagation techniques. Finally, Hochbruck and Lubich utilized the framework of Runge-Kutta methods to develop an efficient and robust exponential propagation method [40] and presented a numerical theory that provided an insight into the advantages of exponential propagation techniques and estimated the error of Krylov approximation for functions of matrices [39]. The main difference between the nonlinear exponential propagation methods proposed by Friesner *et al.* [29] and Hochbruck and Lubich [40] is the technique used to approximate the integral of the exponential propagator. Friesner *et al.* propose a multistep-type approach and develop an iterative technique for this purpose, while Hochbruck and Lubich advocate the framework of the Runge-Kutta method. Chapters 1 and 2 present a summary of the numerical theory of exponential propagation methods developed to date and highlights their advantages and limitations.

In Chapter 1 we introduce the main concepts of exponential propagation technique. We explain how the methods are constructed and then concentrate on the analysis of the Krylov approximation to the exponential of a general large matrix. We provide error estimates and present numerical analysis results for this approximation. In particular, we discuss

the theory presented in [39], which explains the faster convergence rates of the Krylov projection methods applied to approximating matrix exponential compared to the same procedure used to solve a linear system that involves this matrix. This gives an insight into why exponential propagation methods can be superior to implicit schemes. In the last section of the chapter we construct a simple second order exponential propagation method for linear systems of differential equations and discuss its accuracy and stability.

Chapter 2 is dedicated to studying exponential propagation techniques for general systems of nonlinear differential equations. First, we introduce the iterative exponential propagation methods of Friesner *et al.* Our analysis highlights the shortcomings of the technique and suggests important issues in construction of an exponential propagation method. We conclude that this iterative exponential method in the form presented in [29, 23] has many limitations and does not guarantee the accuracy of the computed solution. However, the understanding these problems gives insight into how the multistep-type approach can be used to develop a better method. While we plan to pursue this line of research in future work, here we turn to the Runge-Kutta-type exponential propagation methods proposed in [39, 40], that do not have these limitations. Sections 3.3 and 3.4 describe the construction of an exponential propagation method using the Runge-Kutta framework and discuss one of the main advantages of this approach - the automatic control mechanism, which can be directly transferred over to the exponential propagation method from Runge-Kutta schemes. Finally, in Section 3.5 we demonstrate the advantages of the exponential Runge-Kutta method over the explicit methods on a test problem. We solve Burgers equation using a fourth-order exponential Runge-Kutta method and a fourth-order explicit Runge-Kutta scheme. We use the same automatic error control mechanism for both techniques. The results of this test clearly demonstrate the advantages of exponential Runge-Kutta methods over explicit schemes as the spatial grid size increases and the stiffness of the problem grows. We also discuss ssues of practical implementation of the exponential propagation method. In particular, we mention that the algorithm is easily parallelizable, but has large memory requirements.

The second part of this work (Chapters 4,5) is dedicated to numerical modeling of physical processes that determine the behavior of the solar atmospheric plasma. The events occurring in the upper part of solar atmosphere, called the "corona", have a direct impact on the conditions in the Earth's geophysical environment; this is callled a space "weather". The

corona consists of rarefied plasma which is dominated by magnetic forces and is arranged into a complex network of magnetic arcades. Such plasma configurations can exist stably for long periods of time (days to months); some disappear eventually, others, however, suddenly erupt ejecting billions of tons of plasma into interplanetary space. One of the largest types of such spectacular plasma explosions are called the coronal mass ejections (CME) [19]. CMEs greatly enhance so-called "solar wind", the constant stream of charged particles from the Sun into space, which affects the environment around and on Earth. In particular, large solar storms can disrupt radio communications, damage spacecrafts, electric power grids and pipelines and endanger the astronauts working in Earth's orbit. Therefore, it is important to understand the mechanisms underlying CMEs and to be able to predict these events. At present, however, both the understanding and the predictive capabilities for CMEs are poor. We provide a more detailed overview of the challenges in understanding the dynamics of the coronal plasma configurations and the research in this area in Section 4.1.

Our models of coronal plasma configurations are based on resistive magnetohydrodynamics (MHD), a theory which describes the large scale behavior of plasma configurations. Many applications in fields ranging from fusion research to astrophysics are based on this theory. While a detailed treatment of MHD theory can be found in numerous references [11, 74, 67, 61], we present a brief discussion of the resistive MHD equations in Section 4.2. Due to the complexity of the system of resistive MHD equations, numerical simulations are often the only option to obtain a solution. Solving the system numerically, however, poses numerous challenges to a computational scientist. One of the difficulties in solving the equations of magnetohydrodynamics numerically is the widely separated time scales in the system. The fast modes in the system, represented by electromagnetic waves called Alfven waves, propagate much faster than the slowest, resistive, modes. This causes stiffness in the system of resistive MHD equations and consequent difficulties in numerical integration of the equations. The exponential propagation methods described in Chapters 2 and 3 help resolve this problem and allow the integration of the resistive MHD system with a time step exceeding the CFL bound and have better efficiency than a fully implicit scheme. In Section 4.3 we give an overview of previous work on numerical MHD, describe the challenges of this research area and show why exponential propagation methods are advantageous compared to other numerical techniques used for MHD simulations.

There have only been a few previous studies that explored the potential of exponential propagation methods. Aside from numerical integration of the Schrödinger equation in the context of quantum dynamical simulations, the methods have only been used for small test problems [40, 30] and the performance of these methods for large-scale three-dimensional systems of equations have not been studied in any detail. Thus, in order to validate these numerical schemes in the context of the resistive MHD system we use the exponential Runge-Kutta method for two commonly used models of coronal plasma processes and compare our results with previously published studies (Section 4.4). First, we simulate a magnetic reconnection process in two-dimensional magnetic arcades (Section 4.4.2). We show that our numerical solution agrees with the dynamics calculated using other numerical methods. We also obtain an interesting result about the topology of the reconnection process (Fig. 4.10), which, to our knowledge, have not been previously noted. The second model we present in Section 4.4.3 allows computation of three-dimensional force-free magnetic configurations, which are often used to model the stable coronal magnetic arcades. This problem highlights another application of the exponential propagation methods, namely, the use of these techniques for relaxation problems. So far we discussed the exponential propagation methods in a context of computing the dynamics of a physical system, i.e. problems, where the accurate numerical solution has to be obtained for the time interval of interest. In Section 4.4.3 we propose the use of the methods for steady state problems, where an accurate computation of the system dynamics at every moment of time is not required as long as the final equilibrium solution can be obtained accurately. We demonstrate that a simple second-order in time exponential propagation method is very well suited for these types of problems because it can integrate the equations with a large time step and offers a fast and efficient way to obtain the steady state. The excellent stability properties of the method are also discussed.

In Chapter 5 we present a three-dimensional model of the evolution of the coronal arcades inspired by observational data and also laboratory simulations of coronal magnetic configurations. Section 5.2 provides a detailed description of the model. In particular, the boundary conditions, that we formulate and use, are discussed at length. The conclusions drawn from the results of our three-dimensional numerical model are presented in Section 5.3. We demonstrate that the topology of the magnetic field obtained from simulations is in excellent agreement with the observational data of structure and dynamics of the eruptive

magnetic configurations. One of the major difficulties in understanding the structure of CMEs is the two-dimensional nature of the images obtained by the observational instruments. The variety of structures seen in these images does not allow simple categorization of the different eruptions and makes it difficult to explain these events as a result of the same physical process. The results from our model resolve this problem. Based on the magnetic topology obtained from the simulations we present a theory which interprets the differences in the observational images as different angle projections of a single, specific three-dimensional coronal structure. We verify this result by comparing the observations to the numerical results and demonstrate how the boundary conditions cause formation of such three-dimensional configurations. We present several simulations and discuss how the solution depends on the parameter values used in the computation. These simulations give much insight into the mechanism of evolution and eruption of coronal magnetic arcades. We also present several numerical studies which verify the accuracy of our numerical solution and demonstrate that it is robust with respect to the boundary conditions and that it is not polluted by numerical boundary effects.

Finally, we summarize the results obtained in this work and discuss possible future research objectives in Chapter 6.

Chapter 2 Linear exponential propagation method

In this chapter we introduce the linear exponential propagation method. We begin by describing how exponential propagation can be used to solve systems of differential equations. Then we discuss methods used to approximate functions of a matrix argument. In the final section we present the properties of the resulting exponential propagation scheme.

2.1 Exponential propagation

Consider the following initial value problem for the system of ordinary differential equations:

$$\begin{aligned} \frac{d\mathbf{U}}{dt} &= A\mathbf{U}, \\ \mathbf{U}(t_0) &= \mathbf{U}_0, \end{aligned} \tag{2.1}$$

where $\mathbf{U}$ is a vector in $\mathbb{R}^N$ and A is an $N \times N$ constant matrix in $\mathbb{R}^{N\times N}$. This system can also arise from the application of the method of lines to a system of linear partial differential equations (PDE's). In this case the vector $\mathbf{U}$ will have values of the unknown function on a given grid as its elements, the matrix A will represent the discrete form of the spatial differential operator of the PDE and N will be the number of grid points times the number of unknowns in the system.

The formal solution of problem (2.1) at time $t = t_0 + \Delta t$ is

$$\mathbf{U}(t) = \exp(A(t - t_0))\mathbf{U}_0 = \exp(A\Delta t)\mathbf{U}_0. \tag{2.2}$$

where the action of the operator $\exp(A\Delta t)$ on the vector $\mathbf{U}_0$ is defined as

$$\exp(A\Delta t)\mathbf{U}_0 = \mathbf{U}_0 + \frac{(A\Delta t)}{1!}\mathbf{U}_0 + \frac{(A\Delta t)^2}{2!}\mathbf{U}_0 + \ldots + \frac{(A\Delta t)^n}{n!}\mathbf{U}_0 + \ldots . \tag{2.3}$$

Suppose the matrix A is diagonalizable and we can calculate its eigenvalues $\lambda_1, \ldots, \lambda_N$ and eigenvectors $v_1, \ldots, v_N$. Then we can express A in terms of a matrix V which has eigenvectors of A as its columns $\left(V = \left[v_1, v_2, \ldots, v_N\right]\right)$, its inverse V^{-1} and a diagonal

matrix

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}$$

as $A = V\Lambda V^{-1}$ and easily calculate the exponential operator in (2.8) using

$$\exp(A\Delta t) = V\exp(\Lambda\Delta t)V^{-1} = V \begin{pmatrix} \exp(\lambda_1\Delta t) & 0 & \cdots & 0 \\ 0 & \exp(\lambda_2\Delta t) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \exp(\lambda_N\Delta t) \end{pmatrix} V^{-1}.$$

For most practical applications, however, A is a very large and full matrix. For instance, if problem (2.1) is the spatial discretization of a heat equation in three dimensions on a Cartesian grid with n points in each coordinate direction, matrix A is a discretized Laplacian operator of size $N \times N$ with $N = n^3$. Diagonalizing such a matrix directly is an expensive and inefficient way to solve the system (2.1). Instead we would try to solve it by approximating the time derivative in (2.1) with a finite difference and computing the approximate solution $\hat{\mathbf{U}}$ at time t by solving the resulting system of difference equations. Depending on which finite difference approximation to the time derivative is used the resulting numerical method is either explicit or implicit. For example, if we use simple backward difference and define the approximate solution at time t to be $\hat{\mathbf{U}}$ and at time t_0 to be $\hat{\mathbf{U}}_0$ the resulting scheme will be the Euler method:

$$\frac{\hat{\mathbf{U}} - \hat{\mathbf{U}}_0}{\Delta t} = A\hat{\mathbf{U}}_0,$$

or

$$\hat{\mathbf{U}} = \hat{\mathbf{U}}_0 + A\Delta t\hat{\mathbf{U}}_0 \tag{2.4}$$

If the forward difference approximation to the time derivative is used the implicit scheme

$$\frac{\hat{\mathbf{U}} - \hat{\mathbf{U}}_0}{\Delta t} = A\hat{\mathbf{U}},$$

or

$$\hat{\mathbf{U}} = (I - \Delta t A)^{-1}\hat{\mathbf{U}}_0 \tag{2.5}$$

is obtained. Suppose now that we need to calculate the solution of the system over a long period of time and the absolute value of the ratio of magnitudes between the largest λ_l and the smallest λ_s eigenvalues of A is very large, i.e. system (2.1) is *stiff*. It is well known [50, 77] that the time step in explicit calculations is restricted by the Courant-Friedrichs-Lewy (CFL) stability condition. For the Euler method (2.4) the CFL condition reduces to $|\lambda_l|\Delta t < 1$ if we are solving a system of ODEs and if for example, A is the second order discretized Laplacian in one dimension, the CFL condition is $|\lambda_l|\Delta t < (\Delta x)^2$. Clearly, if our system is stiff the restriction on the time step is quite severe, and if we want to integrate system (2.1) to a large t many intermediate time steps will be required.

Implicit methods, on the other hand, will allow us to perform calculations with larger time steps. In this case, however to calculate the solution at the next time step we have to invert the matrix $(I - \Delta t A)$. Since we are interested in large stiff systems of type (2.1) direct inversion of this matrix is infeasible. We have to apply iterative methods to solve the linear system (2.5). As we know from the Kaniel-Page theory [33, 71] the convergence rate of the iterative methods will be affected by the stiffness of the matrix and unless a good preconditioner is available the iteration will converge very slowly.

Given the limitations of the numerical schemes discussed in the previous paragraphs we would like to develop a new iterative numerical procedure which will allow us to accurately compute the solution using time steps larger than the CFL limit with faster convergence rate than that of an iterative method used to invert matrix $(I - \Delta t A)$ at each time step.

Consider the following numerical scheme arising from approximation (2.2) of the exact solution of the system (2.1):

$$\hat{\mathbf{U}} = \exp(A\Delta t)\hat{\mathbf{U}}_0, \tag{2.6}$$

or equivalently,

$$\hat{\mathbf{U}} = \hat{\mathbf{U}}_0 + \frac{\exp(A\Delta t) - I}{A} A\hat{\mathbf{U}}_0. \tag{2.7}$$

where the action of the operators $\exp(A\Delta t)$ or $\phi(A\Delta t) = (\exp(A\Delta t) - I)/A$ on vectors $\hat{\mathbf{U}}_0$ or $A\hat{\mathbf{U}}_0$ respectively is approximated using an iterative method. In the subsequent sections we will discuss this numerical scheme in detail and show how this method compares with the explicit and implicit techniques.

2.2 Approximating the matrix exponential

The most effective way of approximating the matrix exponential is Krylov subspace projections. This technique was first introduced in the chemical physics literature [60, 63]. Later Van der Vost proposed using this iteration to approximate general functions of a matrix [21]. The convergence analysis and error estimates for such an approximation to the matrix exponential appeared later in the articles by Saad and Gallopoulos [70, 30]. The derived estimates, however, did not fully explain the rapid error reduction in the projection iteration and its relation to the convergence of the iterative techniques for implicit methods. Later Hochbruck and Lubich [39, 40] generalized the convergence analysis to arbitrary functions of a matrix and obtained error bounds which give a better understanding of the properties of Krylov subspace approximations. In this section we will summarize these results and discuss their application.

2.2.1 Krylov subspace projection

Consider the problem of approximating the action of a matrix exponential operator on a vector. Recall the series expansion form of the matrix exponential $e^{\tau A}$ multiplied by a vector $\mathbf{U}_0$

$$\exp(\tau A)\mathbf{U}_0 = \mathbf{U}_0 + \frac{(\tau A)}{1!}\mathbf{U}_0 + \frac{(\tau A)^2}{2!}\mathbf{U}_0 + \ldots + \frac{(\tau A)^m}{m!}\mathbf{U}_0 + \ldots, \tag{2.8}$$

where $\tau \in \mathbf{R}$ is a number which corresponds to the time step in problems of type (2.1) and $A \in \mathbf{R}^{N\times N}$. Clearly, the vector $\exp(\tau A)\mathbf{U}_0$ belongs to an infinite dimensional space of vectors $\{\mathbf{U}_0, A\mathbf{U}_0, A^2\mathbf{U}_0, \ldots, A^m\mathbf{U}_0, \ldots\}$. To approximate (2.8) we will project the

operator $\exp(\tau A)$ and the vector $\mathbf{U}_0$ onto a finite dimensional Krylov subspace

$$S_{kry} = \text{span}\{\mathbf{U}_0, A\mathbf{U}_0, A^2\mathbf{U}_0, \ldots, A^{m-1}\mathbf{U}_0\}.$$

In order to do this we compute an orthonormal basis $\{v_1, \ldots, v_m\}$ of the subspace S_{kry} using the Arnoldi algorithm [10] as follows:

1. $v_1 = \mathbf{U}_0/\|\mathbf{U}_0\|_2$,
2. For $j = 1, \ldots, m$ do
   a) for $i = 1, \ldots j$ compute $h_{i,j} = (v_i, Av_j)$
   b) calculate $w = Av_j - \sum_{i=1}^{j} h_{i,j} v_i$
   c) compute $h_{j+1,j} = \|w\|_2$.
   d) if $h_{j+1,j} = 0$ then stop, else compute the next basis vector $v_{j+1} = w/h_{j+1,i}$.

Arnoldi iteration can be interpreted as the Gram-Schmidt orthogonalization procedure performed on the vectors $\{\mathbf{U}_0, A\mathbf{U}_0, A^2\mathbf{U}_0, \ldots, A^{m-1}\mathbf{U}_0\}$. If we define the matrix H with elements $h_{i,j} = (v_i, Av_j)$, $i, j = 1, \ldots, m$, we can rewrite the Arnoldi algorithm in a matrix form as

$$AV_m = V_m H + h_{m+1,m} v_{m+1} e_m^T, \tag{2.9}$$

where $e_m = [0 \ \ldots \ 0 \ 1]$ is a unit vector in $\mathbb{R}^m$ and $V_m = [v_1, \ldots, v_m]$ is a $m \times m$ matrix with vectors v_i as its columns. If the iteration stopped with $h_{m+1,m} = 0$ (2.9) will become

$$AV_m = V_m H, \tag{2.10}$$

which implies that the computed S_{kry} is invariant under A. Since $\{v_1, \ldots, v_m\}$ is an orthonormal basis of the Krylov subspace S_{kry}, it is clear that if V_m^T is the transpose of V_m, $V_m^T V_m$ is a $m \times m$ identity matrix and $V_m V_m^T$ is a projector from $\mathbb{R}^N$ onto S_{kry}. From the construction of the basis we can see that each vector v_i is orthogonal to all vectors Av_j with $j < i - 1$, which implies $h_{i,j} = (v_i, Av_j) = 0$ so that H is upper Hessenberg. Using $V_m^T v_{m+1} = 0$ from (2.9) we have

$$H = V_m^T A V_m. \tag{2.11}$$

The projection of a matrix A onto a Krylov subspace is then $V_m V_m^T A$ and similarly any vector v projected onto S_{kry} is $V_m V_m^T v$. So if we want to use projection onto S_{kry} to approximate Av we can write $Av \approx V_m V_m^T A V_m V_m^T v$, or substituting (2.11) gives $Av \approx V_m H V_m^T v$. Similarly we can approximate the action of any operator $f(A)$ on a vector $\mathbf{U}_0$ using the Krylov subspace projection as follows:

$$f(A)\mathbf{U}_0 \approx V_m f(H) V_m^T \mathbf{U}_0 = V_m f(H) V_m^T v_1 \|v\|_2 = \|v\|_2 V_m f(H) e_1, \tag{2.12}$$

where we used $v_1 = \mathbf{U}_0 / \|\mathbf{U}_0\|_2$ and $V_m^T v_1 = e_1$. In particular, to approximate the exponential in (2.8) we write

$$\exp(\tau A)\mathbf{U}_0 \approx \|\mathbf{U}_0\|_2 V_m \exp(\tau H) e_1. \tag{2.13}$$

Now suppose the error of the approximation (2.12) is within some predefined tolerance for a relatively small m ($m \ll N$). Then since H is $m \times m$ it is relatively cheap to diagonalize it $\left(H = E\Lambda E^{-1}\right)$ and compute $\exp(\tau H) = E \exp(\tau\Lambda) E^{-1}$. The approximation to a function of A then becomes:

$$f(A)\mathbf{U}_0 \approx \|\mathbf{U}_0\|_2 V_m E f(\Lambda) E^{-1} e_1. \tag{2.14}$$

In the following subsections we will discuss the convergence and error of such an approximation and will show that even for very large matrices arising from the discretization on a PDE on a three-dimensional grid it is possible in cases of interest to obtain desired accuracy by computing only 15-20 Krylov vectors.

2.2.2 Convergence and error of the Krylov approximation

First let us consider the case when the Arnoldi algorithm has stopped with $h_{m+1,m} = 0$. Then from (2.10) we have that for any $j \geq 1$

$$A^j V_m = V_m H^j. \tag{2.15}$$

If function f is analytic on a open set containing the spectrum of the matrix A we can

expand $f(A)$ in a Taylor series as in (2.8) and using (2.15) we have

$$f(A)v = \sum_{j=1}^{\infty} f^{(j)}(0)\frac{A^j}{j!}v = \sum_{j=1}^{\infty} f^{(j)}(0)V_m\frac{H_m^j}{j!}v = V_m f(H_m)v, \tag{2.16}$$

so that approximation (2.12) in this case is exact.

Now suppose that we performed m iterations of the Arnoldi algorithm and $\forall j \leq m$ $h_{j+1,j} \neq 0$. In this case we would like to understand the behavior of the error

$$\varepsilon_m = f(A)v - V_m f(H_m)v \tag{2.17}$$

We begin by following Saad's argument [70] to show that approximating a function of a matrix using Krylov iteration is equivalent to interpolating it by a polynomial. First, the following lemma can be proven by induction

Lemma 2.2.1 *If V_m and H_m are the matrices obtained by performing a Krylov subspace projection method (2.9) on the matrix A with the starting vector v_1, then for any polynomial q_j of degree $j \leq m-1$ the following holds:*

$$q_j(A)v_1 = V_m q_j(H_m)e_1. \tag{2.18}$$

From the theory of matrices [51] we know that a function f is said to be defined on the spectrum of A if for each eigenvalue λ_j of A with multiplicity n_j values

$$f(\lambda_j), f^{(1)}(\lambda_j), \ldots, f^{(n_j)}(\lambda_j)$$

exist and the following theorem holds:

Theorem 2.2.1 *If $g(\lambda)$ is the general Hermite interpolatory polynomial of the minimum degree determined by the values of f on the spectrum of A, then $f(A) = g(A)$.*

It is also known that if $h_{j+1,j} \neq 0$ for any $j = 1, \ldots, m-1$ in the Hessenberg matrix H_m then all the eigenvalues of H_m have multiplicity one. In this case Hermite interpolatory polynomial on the spectrum of H_m is just a Lagrange interpolant p_{m-1} of degree m and

theorem (2.2.1) gives us

$$f(H_m) = p_{m-1}(H_m). \tag{2.19}$$

Applying lemma (2.2.1) and theorem (2.2.1) now to the approximation (2.12) we get

$$V_m f(H_m)v = V_m p_{m-1}(H_m)v = V_m p_{m-1}(A)v, \tag{2.20}$$

where p_{m-1} is the Lagrange interpolation polynomial of degree $m-1$ on the eigenvalues of H_m.

As we can see from this result the accuracy of the approximation (2.12) will depend on how well the interpolatory polynomial p_{m-1} constructed on the eigenvalues of H_m will approximate the Hermite interpolatory polynomial g on the spectrum of A. It is also clear that the properties of the function f are also important. An interpolant for the exponential function $\exp(z)$, for instance, will have a smaller remainder for a given number of iterations than the interpolant for the rational function $1/(1-z)$. This will correspond to the error for the Krylov approximation of the exponential of a matrix e^A being reduced faster during projection iteration than the error of the same procedure applied to compute the inverse matrix $(I-A)^{-1}$.

Gallopoulos and Saad [30, 70] derived error estimates for the Arnoldi approximation based on the formula for the remainder of the interpolating polynomial. As was mentioned in these papers the derived bounds are too large to fully explain the error reduction speed observed in numerical experiments. Hochbruck and Lubich [39, 40] presented a theory which gives a deeper insight into the dependence of the error on the spectrum of the matrix and the function being approximated and provides sharper error bounds. Therefore here we will follow their approach.

The error estimates are derived from applying the Cauchy integral formula to the expression $f(A)v$. Suppose f is analytic in a neighborhood of the numerical range

$$F(A) = \{x^*Ax : x \in \mathbf{C}^N, \|x\| = 1\}$$

of a matrix A and Γ is a contour that surrounds $F(A)$. Then

$$f(A)v = \frac{1}{2\pi i}\int_{\Gamma} f(\lambda)(\lambda I - A)^{-1}v d\lambda. \tag{2.21}$$

Since $H_m = V_m^T A V_m$ then $F(H_m) \subset F(A)$. So if λ is outside $F(A)$ then neither $(\lambda I - A)$ nor $(\lambda I - H_m)$ are singular matrices and we can also write

$$V_m f(H_m)e_1 = \frac{1}{2\pi i}\int_{\Gamma} f(\lambda)V_m(\lambda I - H_m)^{-1}e_1 d\lambda. \tag{2.22}$$

Therefore the error in Arnoldi approximation can be written as

$$\varepsilon_m = f(A)v - V_m f(H_m)e_1 = \frac{1}{2\pi i}\int_{\Gamma} f(\lambda)\{(\lambda I - A)^{-1}v - V_m(\lambda I - H_m)^{-1}e_1\}d\lambda. \tag{2.23}$$

The error bounds are obtained by estimating the error of $(\lambda I - A)^{-1}v - V_m(\lambda I - H_m)^{-1}e_1$, multiplying this estimate by $|f(\lambda)|$ and integrating the result along the appropriate contour Γ. Proceeding in this manner will lead to the following estimate on the error of the Arnoldi approximation (2.9):

Theorem 2.2.2 (Hochbruck,Lubich) *Suppose E is a convex, closed bounded set in the complex plane which contains the numerical range of the matrix A and $G_\infty = \mathbf{C}\backslash E$ is its complement. Let $\phi(z)$ be a unique conformal mapping determined by the Riemann mapping theorem which carries G_∞ onto the exterior of the unit circle $\{|w| > 1\}$ and has the Laurent expansion $\phi(z) = c_1\lambda + c_0 + c_{-1}\lambda^{-1} + \ldots$ with $c_1 > 0$. Then for every polynomial q_{m-1} of degree at most $m-1$ and matrices V_m and H_m determined by (2.9) the following inequality holds:*

$$\|f(A)v - V_m f(H_m)e_1\| \leq \frac{M}{2\pi}\int_{\Gamma} |f(\lambda) - q_{m-1}(\lambda)| \cdot |\phi(\lambda)|^{-m} \cdot |d\lambda| \tag{2.24}$$

with $M = l(\partial E)/(d(\partial E) \cdot d(\Gamma))$, where $l(\partial E)$ is the length of the boundary curve ∂E of E and $d(S)$ is the minimal distance between numerical range $F(A)$ and a subset S of the complex plane. If E is a straight line segment or a disk, then (2.24) holds with $M = 6/d(\Gamma)$.

The detailed proof of this theorem can be found in [39]. Here we will only outline the

proof as follows. Note that (2.24) is derived by first observing that since

$$V_m^*(\lambda I - A)V_m = \lambda I - H_m,$$

we have

$$[(\lambda I - A)^{-1} - V_m(\lambda I - H_m)^{-1}V_m^T]v = \\ [(\lambda I - A)^{-1} - V_m(\lambda I - H_m)^{-1}V_m^T](v - (\lambda I - A)V_m y_m) \tag{2.25}$$

for any vector $y_m \in \mathbf{C}^m$. But $v-(\lambda I-A)V_m y_m$ can always be written as $p_m(A)v$, where p_m is a polynomial of degree $\leq m$ and $p_m(\lambda) = 1$. Therefore using $\|(\lambda I-A)^{-1}\| \leq dist(\lambda, F(A))^{-1}$ and $\|(\lambda I - H_m)^{-1}\| \leq dist(\lambda, F(A))^{-1}$ we get an estimate

$$\|(\lambda I - A)^{-1}v - V_m(\lambda I - H_m)^{-1}e_1\| \leq 2d(\Gamma)^{-1} \cdot \|p_m(A)\| \tag{2.26}$$

for every polynomial p_m of degree at most m with $p_m(\lambda) = 1$. Then using the Cauchy integral (2.21) representation of $p_m(A)$ and the optimality properties of Faber polynomials generated by the conformal mapping $\phi(z)$ described in the theorem (2.2.2) the estimate

$$\|(\lambda I - A)^{-1}v - V_m(\lambda I - H_m)^{-1}e_1\| \leq M \cdot \|\phi(\lambda)\|^{-m} \tag{2.27}$$

the bound on $\|p_m(A)\|$ is derived. Finally, the estimate (2.24) is obtained using lemma 2.2.1 and formulas (2.23) and (2.27).

Bounds in Theorem 2.2.2 can be made more precise for classes of matrices with a specific spectrum by choosing the appropriate mapping $\phi(z)$ and analyzing different functions f. Below are additional results from [39] for approximating the exponential of a matrix $\exp(\tau A)$ for special classes of matrices, where we include a parameter τ corresponding to the time step in the numerical schemes for the time dependent problems.

Lemma 2.2.2 *If A is a Hermitian negative semidefinite matrix with eigenvalues in the*

interval $[-4\rho, 0]$ *then the error* ε_m *in the Arnoldi approximation of* $\exp(\tau A)v$ *is bounded as*

$$\varepsilon_m \leq 10e^{-m^2/(5\rho\tau)}, \qquad \sqrt{4\rho\tau} \leq m \leq 2\rho\tau, \tag{2.28}$$

$$\varepsilon_m \leq 10(\rho\tau)^{-1}e^{-\rho\tau}\left(\frac{e\rho\tau}{m}\right)^m, \qquad m \geq 2\rho\tau. \tag{2.29}$$

And for general matrices:

Lemma 2.2.3 *If numerical range of a matrix* A *is contained in the disk* $|z+\rho| \leq \rho$*, then the error in the Arnoldi approximation of* $e^{\tau A}v$ *is bounded by*

$$\varepsilon_m \leq 12e^{-\rho\tau}\left(\frac{e\rho\tau}{m}\right)^m, \qquad m \geq 2\rho\tau. \tag{2.30}$$

These error bounds can now be used to assess how the numerical methods for solution of problems of type (2.1) compare with the explicit and implicit schemes. It is clear that Theorem 2.2.2 can also be used to derive the error of the Arnoldi approximation to the inverse matrix $(I-\tau A)^{-1}$. The results obtained in this way help explain why approximating the exponential of the matrix $\exp(\tau A)$ to a specified accuracy may be a faster iteration procedure than computing the approximation to $(I-\tau A)^{-1}$. For example, if the matrix A is symmetric negative definite then applying the Arnoldi approximation algorithm (2.9) to invert $(I-\tau A)$ is equivalent to using the conjugate-gradient method to solve $(I-\tau A)x = v$ and the error derived using Theorem 2.2.2 is

$$\|x - x_m\| \leq 2\sqrt{1+4\rho\tau}\left(1 - \frac{2}{\sqrt{1+4\rho\tau}+1}\right)^m. \tag{2.31}$$

Comparison of the error bounds (2.31) and (2.28-2.29) shows that the error in approximating the exponential of the function will be reduced faster in the course of Arnoldi iteration. This is demonstrated on specific examples in [39].

2.3 Practical implementation of the Arnoldi iteration

2.3.1 Residual and stopping criterion for the Arnoldi algorithm

In practice, in order to use the Arnoldi algorithm to approximate a function of a matrix we have to consider the following issues. First, we need to have a stopping criterion for the

Krylov projection iteration. Clearly, since we do not usually know the spectral radius of the matrix in question, the error bounds in the previous section cannot be used for that purpose. Let us recall the integral form of the error in the Arnoldi approximation to the function $f(\tau A)v$:

$$\varepsilon_m = f(\tau A)v - V_m f(\tau H_m)e_1 = \frac{1}{2\pi i}\int_\Gamma f(\lambda)\{(\lambda I - \tau A)^{-1}v - V_m(\lambda I - \tau H_m)^{-1}e_1\}d\lambda \tag{2.32}$$

Define

$$e_m(\lambda) = (\lambda I - \tau A)^{-1}v - V_m(\lambda I - \tau H_m)^{-1}e_1. \tag{2.33}$$

From this expression we can see that $e_m(\lambda) = x - x_m$ is exactly the error of the mth Arnoldi approximate $x_m = V_m(\lambda I - \tau H_m)^{-1}V_m^* v = V_m(\lambda I - \tau H_m)^{-1}e_1$ to the solution of the equation $(\lambda I - \tau A)x = v$. Let us define the residual of this approximation as

$$r_m(\lambda) = v - (\lambda I - \tau A)x_m. \tag{2.34}$$

Using these expressions we can rewrite the error $e_m(\lambda)$ as follows:

$$\begin{aligned} e_m(\lambda) &= (\lambda I - \tau A)^{-1}v - x_m \\ &= (\lambda I - \tau A)^{-1}\Big[v - (\lambda I - \tau A)x_m\Big] \\ &= (\lambda I - \tau A)^{-1}r_m(\lambda). \end{aligned} \tag{2.35}$$

So minimizing the norm of the residual $\|r_m(\lambda)\|$ will result in minimizing the error in the same norm $\|e_m(\lambda)\|$. As is usually done in iterative methods we can use estimates of the residuals to monitor the accuracy of the approximation. From the construction of the Arnoldi iteration we have

$$(\lambda I - \tau A)V_m = V_m(\lambda I - \tau H_m) + \|v\|_2 h_{m+1,m}v_{m+1}e_m^T \tag{2.36}$$

where $h_{m+1,m}$ and v_{m+1} are routinely computed in the course of iteration at no additional cost. Therefore we can calculate the residual as

$$\begin{aligned} r_m(\lambda) &= v - \left[V_m(\lambda I - \tau H_m) + \|v\|_2 h_{m+1,m} v_{m+1} e_m^T\right] \cdot (\lambda I - \tau H_m)^{-1} V_m^* v \\ &= -\|v\|_2 h_{m+1,m} v_{m+1} e_m^T (\lambda I - \tau H_m)^{-1} e_1^T \\ &= -\|v\|_2 h_{m+1,m} v_{m+1} [(\lambda I - \tau H_m)^{-1}]_{m,1}. \end{aligned} \tag{2.37}$$

The error of the Arnoldi approximation to $f(\tau A)v$ is

$$\varepsilon_m = f(\tau A)v - \|v\|_2 V_m f(\tau H_m) e_1 = \frac{1}{2\pi i} \int_\Gamma f(\lambda) e_m(\lambda) d\lambda. \tag{2.38}$$

It was proposed in [70, 39] to use the norm of the generalized residual defined by

$$\begin{aligned} \rho_m &= \frac{1}{2\pi i} \int_\Gamma f(\lambda) r_m(\lambda) d\lambda \\ &= -\frac{1}{2\pi i} \|v\|_2 h_{m+1,1} v_{m+1} \int_\Gamma f(\lambda) [(\lambda I - \tau H_m)^{-1}]_{m,1} d\lambda \\ &= -\|v\|_2 h_{m+1,1} [f(\tau H_m)]_{m,1} v_{m+1}. \end{aligned} \tag{2.39}$$

as an accuracy check. This quantity can be computed at no additional cost during Arnoldi iteration using (2.37).

As we can see from the expression for the residual and the error estimates in the previous section, in order to reduce the norm of the error or the residual we have the option of either increasing the number of iterations, i.e. the number of Krylov vectors computed, or decreasing the time step. The optimal combination should be determined based on the available computing resources. If we are performing computations on a computer with large memory we might want to store more Krylov vectors and take a large time step. If, on the other hand, only a limited amount of memory is available we can decrease the time step and complete fewer iterations.

It is appropriate at this point to mention the effect of finite precision on the Arnoldi iteration. The difficulty arises when in the process of computing the orthonormal basis $\{v_1, \ldots, v_m\}$ the use of finite precision computation results in the loss of orthogonality of the basis vectors and consequently loss of accuracy as the iteration proceeds. If this

becomes a problem the Arnoldi iteration can be augmented by either complete or partial reorthogonalization of each newly computed Krylov vector against its predecessors. The discussion of algorithms accomplishing this task can be found, for example, in [33, 13] and references therein. In our numerical experiments we found, however, that to achieve the desired accuracy with the appropriate time step we only needed to compute an average of 15 and a maximum of 30 Krylov vectors. There was no significant loss of accuracy in computing such a small Krylov basis for the operators of interest and the straightforward implementation of the Arnoldi algorithm provided us with a tool to compute an accurate approximation to the Krylov vectors and consequently to the exponential of a matrix.

2.3.2 Calculating operators e^{H_m} and $\phi(H_m) = \frac{e^{H_m}-I}{H_m}$

For linear problems of type (2.1) we have to decide whether to use the exponential propagation method formulated as

$$\mathbf{U}^{n+1} = \exp(A\Delta t)\mathbf{U}^n \tag{2.40}$$

or

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \frac{\exp(A\Delta t) - I}{A} A\mathbf{U}^n. \tag{2.41}$$

In these formulas $\mathbf{U}^n$ is the approximate solution computed at time $t_n \in [t_0, t_{final}]$. The product of the operator $e^{A\Delta t}$ with the vector $\mathbf{U}^n$ or the product of the operator $\phi(A\Delta t) = \frac{e^{A\Delta t}-I}{A}$ with the vector $A\mathbf{U}^n$ are computed using the Arnoldi algorithm with the starting vectors $\mathbf{U}^n$ or $A\mathbf{U}^n$ respectively. Formulation (2.41) as we will see later is also relevant in the generalization of exponential propagation methods to nonlinear differential systems of equations. Let us derive a formula which will link the error in the Arnoldi approximation of $\exp(A\Delta t)$ and $\phi(A\Delta t)$. To simplify the notation we define $v = \mathbf{U}^n$ and the errors in the Arnoldi approximations of $\exp(A\Delta t)$ and $\phi(A\Delta t)$ as

$$\begin{aligned} \epsilon_{exp} &= \exp(A\Delta t)v - V_m \exp(H_m \Delta t) V_m^T v \\ &= \exp(A\Delta t)v - \|v\|_2 \cdot V_m \exp(H_m \Delta t) e_1 \end{aligned} \tag{2.42}$$

and

$$\begin{aligned}\epsilon_{phi} &= \phi(A\Delta t)v - V_m\phi(H_m\Delta t)V_m^T v \\ &= \phi(A\Delta t)v - \|v\|_2 \cdot V_m\phi(H_m\Delta t)e_1. \end{aligned} \tag{2.43}$$

From the discussion of the Arnoldi iteration in the previous sections we recall the formula

$$AV_m = V_m H_m + h_{m+1,m}v_{m+1}e_m^T, \tag{2.44}$$

and the fact that the matrix V_m has the orthonormal basis $\{v_1, \ldots, v_m\}$ as its columns with the first vector being $v_1 = v/\|v\|_2$, which gives

$$V_m^T v = v_1^T v = \|v\|_2 \cdot e_1. \tag{2.45}$$

Using these expressions we can write

$$\begin{aligned} e^{\Delta tA}v &= [I + A(\frac{e^{\Delta tA} - I}{A})]v \\ &= v + A\phi(\Delta tA)v \end{aligned} \tag{2.46}$$

and

$$\begin{aligned} V_m e^{\Delta tH_m} V_m^T v &= V_m[I + H_m\phi(\Delta tH_m)]V_m^T v \\ &= \|v\|_2 \cdot V_m e_1 + \|v\|_2 \cdot V_m H_m \phi(\Delta tH_m)e_1 \\ &= v + \|v\|_2 \cdot (AV_m - h_{m+1,m}v_{m+1}e_m^T)\phi(\Delta tH_m)e_1 \\ &= v + AV_m\phi(\Delta tH_m)V_m^T v - \|v\|_2 \cdot h_{m+1,m}[\phi(\Delta tH_m)]_{m,1}v_{m+1}. \end{aligned} \tag{2.47}$$

Subtracting (2.47) from (2.46) we obtain

$$\epsilon_{exp} = A\epsilon_{phi} - \|v\|_2 \cdot h_{m+1,m}[\phi(\Delta tH_m)]_{m,1}v_{m+1}. \tag{2.48}$$

Noting now that the second term on the right hand side of the above formula is exactly the expression for the generalized residual ρ_m we derived in the previous subsection, we arrive at the following formula linking the errors for the Arnoldi approximations of $e^{\Delta tA}$

and $\phi(\Delta t A)$:

$$A\epsilon_{phi} = \epsilon_{exp} + \rho_m \tag{2.49}$$

This easily leads to an inequality

$$\|\epsilon_{phi}\| \leq \|A^{-1}\|(\|\epsilon_{exp}\| + \|\rho_m\|). \tag{2.50}$$

If we take $\|\cdot\|$ to be a two-norm $\|\cdot\|_2$ and remember that we expect $\rho_m \approx const \cdot \epsilon_{exp}$ we can see that, for example, in the case when even the magnitude λ_{min} of the smallest eigenvalue of a matrix A satisfies $\lambda_{min} \gg 1$ so that the spectral radius of A^{-1} is $\|A^{-1}\| = 1/\lambda_{min} \ll 1$ then ϵ_{phi} can be smaller than ϵ_{exp}, so that formulation (2.41) can be preferable over (2.40).

Finally, we should consider the issue of computing $f(\tau H_m)$. It was mentioned before that we expect H_m to be rather small. Hence, we have an option of either computing eigenvalues and eigenvectors of H_m and calculating $f(\tau H_m)$ using (2.14) or approximating it directly. If the matrix A is Hermitian and, consequently due to (2.11), matrix H_m is also Hermitian the former approach will be more efficient and we can diagonalize H_m using the symmetric version of the QR algorithm. In particular, in the case of a symmetric operator we used a diagonalization routine from the LAPACK package after completing the Arnoldi iterations. The time required to compute diagonalization of H_m in this way was insignificant compared to the total time required to integrate the equations over one time step since H_m is small ($m \approx 15 - 20$) . For a general nonsymmetric operator A and a larger number of computed Krylov vectors other strategies can be used. The description of such methods is given in [17], [33] . One of the most effective ways to accomplish this in case of computing the exponential of a matrix $H = \tau H_m$ is to use Padé approximation defined by

$$R_{pq}(H) = [D_{pq}(H)]^{-1} N_{pq}(H), \tag{2.51}$$

where

$$N_{pq}(H) = \sum_{j=0}^{p} \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} H^j \tag{2.52}$$

and

$$D_{pq}(H) = \sum_{j=0}^{q} \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!}(-H)^j. \tag{2.53}$$

The properties of this approximation are discussed in detail in [17]. Here we will just note that, due to roundoff errors, the computation via the Padé formula has to be applied to a scaled matrix $H/2^k$ rather than H, where k is chosen so that $\|H/2^k\| < 1/2$. Then, since $e^H = (e^{H/2^k})^{2^k}$, we can approximate as $e^H \approx [R_{pq}(H/2^k)]^{2^k}$. This is done because the roundoff error and the cost of computing Padé approximates both increase as $\|H\|$ grows. The optimal choice of Padé approximation for the matrix $H/2^k$ is the diagonal version of (2.51-2.53), i.e. $p = q$. Such an estimate provides us with the solution accurate to order $2q$ with qm^3 flops required for computation of the diagonal Padé approximate.

If we chose formulation (2.41) we have to calculate

$$\phi(H) = \frac{e^H - 1}{H}$$

rather than the exponential of H. We can use Padé approximation to the exponential of a matrix

$$\begin{bmatrix} H & e_1 \\ 0 & 0 \end{bmatrix}$$

and using the formula derived by Saad [70]

$$\exp \begin{bmatrix} H & e_1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \exp(H) & \phi(H)e_1 \\ 0 & 1 \end{bmatrix} \tag{2.54}$$

calculate $\phi(H)$. Alternatively, as proposed by Hochbruck and Lubich [40] we can use Padé approximation to the function $\phi(H/2^k)$ and the recurrence

$$\phi(2z) = \frac{1}{2}(e^z + 1)\phi(z) \tag{2.55}$$

$$e^{2z} = e^z e^z \tag{2.56}$$

to obtain the required approximation.

2.4 Accuracy and stability of the linear exponential propagation method

To solve a linear problem

$$\frac{d\mathbf{U}}{dt} = A\mathbf{U} \qquad (2.57)$$
$$\mathbf{U}(t_0) = \mathbf{U}_0$$

we will use the following numerical method

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \frac{\exp(A\Delta t) - I}{A} A\mathbf{U}^n \qquad (2.58)$$

where $\mathbf{U}^n$ is approximation to a solution $\mathbf{U}$ at time t_n, $t_0 < t_1 < ... < t_n < t_{n+1} < ... < t_{final}$ and the matrix-vector product

$$\phi(A\Delta t)\mathbf{U}^n = \frac{\exp(A\Delta t) - I}{A} A\mathbf{U}^n$$

is computed using the Arnoldi algorithm with the matrix $\phi(A\Delta t)$ and a starting vector $A\mathbf{U}^n$ as described in the previous sections. Clearly, if $\phi(A\Delta t)A\mathbf{U}^n$ is computed directly, the method would be exact for linear problems. Since we use Krylov projection to compute this operator the accuracy of the approximate solution will depend on the size of the final residual (2.39) in the Arnoldi iteration and the error in the Padé approximation of the operator $\phi(\Delta t H_m)$. The time stepping algorithm will proceed as follows: we determine the maximum number of Krylov vectors that we can store (this will be defined by the amount of memory available), then we proceed with the Arnoldi iteration and if after computing all the Krylov vectors the residual is not satisfactory the time step is reduced and the procedure is repeated with the new time step.

Before analyzing stability of the linear exponential propagation method we emphasize that the system (2.57) is assumed to be well-posed. This implies that all eigenvalues of the matrix A have negative real parts and therefore the inequality $\|e^{A(t-t_0)}\| \leq 1$ holds for any time $t \in [t_0, t_{final}]$. Since the exact evolution operator $\exp(A(t_n - t_{n-1}))$ is approximated by the Krylov subspace projection $V_m \exp(H_m(t_n - t_{n-1}))V_m^T$ and for unitary matrices $\|V_m\| = 1$ the numerical exponential propagation method will be stable if the inequality

$\|\exp(\Delta t H_m)\| \leq 1$ holds. Following Gallopoulos and Saad's argument [30] we address this problem in two parts. First, we consider the case when $\exp(\Delta t H_m)$ is calculated exactly and then talk about the stability of Padé approximation to this operator.

Assuming $e^{\Delta t H_m}$ is calculated exactly, let us bound the exponential of the matrix $\Delta t H_m$ in terms of the exponential operator of $\Delta t A$. To accomplish this we use properties of a logarithmic norm defined as

$$\mu(A) = \lim_{h \to 0+} \frac{\|I + hA\| - 1}{h}. \tag{2.59}$$

If the norm used in this definition is the two-norm then

$$\mu(A) = \lambda_{max}(\frac{A + A^*}{2}), \tag{2.60}$$

where λ_{max} is the maximum absolute value eigenvalue of the symmetric part of A. While the set of full properties of the logarithmic norm $\mu(A)$ can be found for example in [78] here we will only use the following inequality

$$\|e^A\| \leq e^{\mu(A)}. \tag{2.61}$$

Lemma A.3 in [30] proves that

$$\mu(H_m) \leq \mu(A) \tag{2.62}$$

and therefore we have

$$\|V_m e^{\Delta t H_m} V_m^*\| \leq \|e^{\Delta t H_m}\| = e^{\mu(\Delta t H_m)} \leq e^{\mu(\Delta t A)} = \Delta t \exp\left(\lambda_{max}(\frac{A + A^*}{2})\right). \tag{2.63}$$

Obviously, if matrix A is Hermitian and negative definite so that $\lambda_{max}(A) \leq 0$ then from the inequality (2.63) it follows that

$$\|V_m e^{\Delta t H_m} V_m^*\| \leq 1 \tag{2.64}$$

and therefore method (2.58) is stable.

Stability properties of the diagonal Padé approximation to the exponential $e^{\Delta t H_m} \approx$

$R_{pp}(\Delta t H_m)$ have been studied by many authors [30, 17, 34, 73]. The detailed discussion of this topic is beyond the scope of this work. Therefore we will refer the reader to the sources above for a thorough study of this issue and limit ourselves to stating the result which can be found, for example, in [30] that if $\mu(H_m) \leq 0$ then the diagonal Padé approximation is stable, i.e.

$$\|V_m R_{pp}(\Delta t H_m) V_m^T\| \leq 1. \tag{2.65}$$

To summarize, we have unconditional stability of the method when the matrix A is symmetric positive definite or if the maximum eigenvalue of the symmetric part of A is negative. Obviously, we cannot have stability for an arbitrary matrix A, but in all of our computations described in the later chapters we found that for the matrices of interest the method was stable and in fact exhibited unconditional stability in many cases. Therefore we would like to note that this method is particularly effective for relaxation problems, where we need to obtain some equilibrium state and are not interested in computing the dynamics accurately. We will reiterate this point later when we discuss computing the force-free plasma states using exponential propagation method.

Chapter 3 Exponential propagation methods for systems of nonlinear differential equations

In the previous chapter it was shown how exponential propagation together with the Krylov subspace projection algorithm can be used to solve systems of linear differential equations. Ultimately, however, the problems of interest are systems of nonlinear partial differential equations. In the following sections we will discuss how the concepts of Chapter 2 can be extended to solve such systems and compare the resulting methods with conventional numerical techniques.

3.1 Introduction and formulation

Let us consider the general autonomous nonlinear initial value problem

$$\frac{d\mathbf{U}(t)}{dt} = F(\mathbf{U}(t)) \tag{3.1}$$

$$\mathbf{U}(t_0) = \mathbf{U}_0. \tag{3.2}$$

If we have a system of ordinary differential equations then $\mathbf{U} \in \mathbb{R}^N$ is a vector of unknowns and $F(\mathbf{U})$ is a exact nonlinear operator acting on this vector. If a system of partial differential equations is to be solved then $\mathbf{U} \in \mathbb{R}^N$ is the vector consisting of the values of all the unknown variables on a given spatial grid and $F(\mathbf{U})$ is a discretized version of the spatial differential operator on this grid. Note that considering only autonomous problems does not impose any loss of generality since any non-autonomous system of type

$$\frac{d\mathbf{U}(t)}{dt} = F(t, \mathbf{U}(t)) \tag{3.3}$$

can be converted to an autonomous system by creating a new solution vector

$$\mathbf{W}(t) = \begin{bmatrix} \mathbf{U}(t) \\ t \end{bmatrix}$$

which includes the time variable as a component, and adding an equation for this variable to the system (3.1) to get

$$\begin{pmatrix} \frac{d\mathbf{U}(t)}{dt} \\ \frac{dt}{dt} \end{pmatrix} = \begin{pmatrix} F(t, \mathbf{U}(t)) \\ 1 \end{pmatrix}. \tag{3.4}$$

or if we define

$$G(\mathbf{W}(t)) = \begin{pmatrix} F(t, \mathbf{U}(t)) \\ 1 \end{pmatrix} \tag{3.5}$$

the autonomous problem is formulated as

$$\frac{d\mathbf{W}(t)}{dt} = G(\mathbf{W}(t)) \tag{3.6}$$

Now let us return to the original problem (3.1) - (3.2). Suppose that the Jacobian $DF(\mathbf{U}_0)$ exists and is continuous on the interval $[t_0, t_0 + t]$, and we want to compute a solution to (3.1) at time $t_0 + t$. Then we can linearize $F(\mathbf{U}(t_0 + t))$ around the initial state $\mathbf{U}_0$ and rewrite system (3.1) as

$$\begin{aligned} \frac{d\mathbf{U}(t_0 + t)}{dt} &= F(\mathbf{U}_0) + DF(\mathbf{U}_0)\Big(\mathbf{U}(t_0 + t) - \mathbf{U}_0\Big) \\ &\quad + \Big[F(\mathbf{U}(t_0 + t)) - F(\mathbf{U}_0) - DF(\mathbf{U}_0)\Big(\mathbf{U}(t_0 + t) - \mathbf{U}_0\Big)\Big]. \end{aligned} \tag{3.7}$$

To simplify the notation, let us define the Jacobian matrix as $A = DF(\mathbf{U}_0)$ and the nonlinear remainder

$$R(\mathbf{U}) = R(\mathbf{U}(t_0 + t)) = F(\mathbf{U}(t_0 + t)) - F(\mathbf{U}_0) - DF(\mathbf{U}_0)(\mathbf{U}(t_0 + t) - \mathbf{U}_0), \tag{3.8}$$

so that (3.7) becomes

$$\frac{d\mathbf{U}(t_0 + t)}{dt} = F(\mathbf{U}_0) + A(\mathbf{U}(t_0 + t) - \mathbf{U}_0) + R(\mathbf{U}(t_0 + t)). \tag{3.9}$$

We can write this equation in an integral form as follows:

$$\mathbf{U}(t_0 + t) = \mathbf{U}_0 + \frac{e^{At} - I}{A} F(\mathbf{U}_0) + \int_0^t e^{A(t-s)} R(\mathbf{U}(t_0 + s) - \mathbf{U}_0) ds. \tag{3.10}$$

In the discussion of the exponential propagation method for linear systems, i.e. the case of $R(\mathbf{U}) = 0$, we computed the solution using expression (3.10) and approximated the action of the operator

$$\frac{e^{At} - I}{A}$$

on a vector $F(\mathbf{U}_0)$ by projecting onto a Krylov subspace. If the contribution of the integral

$$\int_0^t e^{A(t-s)} R(\mathbf{U}(t_0 + s) - \mathbf{U}_0) ds$$

in (3.10) to the right hand side of (3.7) over the time interval $[t_0, t_0 + t]$ is significantly smaller than that of the rest of the terms, i.e. the dynamics of the system is dominated by the linear part of the right hand side, we can generalize the linear method (2.58) and use the scheme

$$\hat{\mathbf{U}} = \hat{\mathbf{U}}_0 + \frac{e^{A\Delta t} - I}{A} F(\hat{\mathbf{U}}_0) \tag{3.11}$$

to integrate the system to the next time step. Clearly, such a method will be second order in time. We turned to exponential propagation methods because we wanted to increase the time step compared with explicit methods. Since we want to take large time steps, second order accuracy in time might not be enough to accurately compute the time dynamics of the nonlinear system of interest. Therefore, the core of the discussion below will be dedicated to the development of methods which estimate the nonlinear integral in (3.10) to a higher order. There is another set of problems, however, for which we think the method (3.11) is very useful.

Suppose that we only want to compute a steady-state of the system and we have some independent of the time integration scheme criterion to verify whether the desired state is achieved. For example, suppose we want to obtain a final temperature distribution of a

body given some boundary conditions. Once we obtain such a distribution we can verify that it is an equilibrium state by computing its Laplacian with the appropriate boundary conditions and checking whether it is close to zero. In such problems we are not interested in computing the time dynamics accurately and we want to take a very large time step to march to a steady-state as quickly as possible. This is exactly the goal that can be achieved with the method (3.11). The discussion of stability from the previous chapter can be carried directly over to this method and we can expect it to be stable if all eigenvalues of the Jacobian matrix have negative real parts. If it takes a long time for the system to settle to a steady state, then using exponential propagation with a large time step could be very effective. As discussed in the next chapter, this method can be used to compute the nonlinear relaxed force-free states of solar magnetic arcades.

Now we return to the question of solving dynamical nonlinear problems with a large time step. In this case, the nonlinear integral in (3.10) has to be computed with high accuracy. Noting that the integrand in (3.10) has a familiar from the last Chapter form of an exponential operator e^{At} acting on a vector $R(\mathbf{U})$, we would like to use the favorable properties of the Krylov subspace projection iteration in approximating this integral. To our knowledge two different extensions of the exponential propagation methods to the nonlinear systems have been proposed in the literature. In the following sections we will describe these two approaches and discuss their limitations and advantages.

3.2 Iterative approximation to the nonlinear integral

3.2.1 Formulation

Friesner *et.al.* proposed an iterative process to approximate the nonlinear integral in (3.10). The authors applied this method to nonlinear systems of ordinary differential equations in [29] and later extended it to solve the Navier-Stokes system [23]. Below we describe their method and discuss its performance in the next subsection.

First let us simplify the notation by defining $b = F(\mathbf{U}_0)$ and formulating the integral form of the system (3.1) in terms of a shifted variable

$$u(t) = \mathbf{U}(t_0 + t) - \mathbf{U}_0. \tag{3.12}$$

The integral formula (3.10) can be then written as

$$u(t) = \frac{e^{At} - I}{A} b + \int_0^t e^{A(t-\tau)} R(u(\tau)) d\tau. \tag{3.13}$$

Friesner *et.al* proposed to use the following iteration to calculate the approximation to a solution at each time step

$$u^{(l+1)}(t) = \frac{e^{At} - I}{A} b + \int_0^t e^{A(t-\tau)} R(u^{(l)}(\tau)) d\tau \tag{3.14}$$

with the initial guess being the solution of the linear equation with $R(u) = 0$

$$u^{(0)}(t) = \frac{e^{At} - I}{A} b. \tag{3.15}$$

Once iteration (3.14) has completed, the solution $\mathbf{U}(t_0 + t)$ can be easily computed using (3.12), so for now we will just concentrate on the algorithm the authors use to compute (3.13). To approximate the nonlinear integral on the right-hand-side of (3.14) an approach of multistep type is used. The integral is computed by fitting a polynomial to the function $R(u^{(l)}(s))$. Since the nonlinear integral is the second order term in the solution, the lowest power in the polynomial is chosen to be two. $R(u(s))$ is evaluated on the grid of Chebyshev points $0 < \tau_1 < \tau_2 < ... < \tau_{J_l} = t$ with

$$\tau_i = \frac{t}{2}[1 - \cos(\frac{i\pi}{J_l})], \tag{3.16}$$

and the following J_l collocation equations are solved for the coefficients $r_j^{(l)}$:

$$R(u^{(l)}(\tau_i)) = \sum_{j=2}^{J_l+1} \tau_i^j r_j^{(l)}, \qquad i = 1, \ldots, J_l, \tag{3.17}$$

where r_j are the coefficient vectors of length N. The iteration now takes form

$$u^{(l+1)}(t) = \frac{e^{At} - I}{A} b + \int_0^t \sum_{j=2}^{J_l+1} e^{A(t-\tau)} r_j^{(l)} \tau^j d\tau. \tag{3.18}$$

Now the matrix-vector products $\frac{e^{At}-I}{A}b$ and $e^{A(t-\tau)}r_j^{(l)}$ are computed using the Arnoldi algorithm. Suppose that $\frac{e^{At}-I}{A}b$ is accurately approximated with m Krylov vectors and m_j such vectors were calculated to estimate $e^{A(t-\tau)}r_j^{(l)}$ for each j. We can write these Krylov projection iterations as

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \tag{3.19}$$

$$AV_{m_j}^{(j)} = V_{m_j}^{(j)} H_{m_j}^{(j)} + h_{m_j+1,m}^{(j)} v_{m_j+1}^{(j)} e_{m_j+1}^T. \tag{3.20}$$

To simplify the notation we drop the sub- and superscripts and define $H_j = H_{m_j}^{(j)}$, $V_j = V_{m_j}^{(j)}$, $H = H_m$ and $V = V_m$. Note that H is an $m \times m$ matrix and H_j are matrices of sizes $m_j \times m_j$. Now iteration (3.18) becomes

$$u^{(l+1)}(t) = V\frac{e^{Ht}-I}{H}V^T b + \int_0^t \sum_{j=2}^{J_l+1} V_j e^{H_j(t-\tau)} V_j^T r_j^{(l)} \tau^j d\tau. \tag{3.21}$$

After diagonalizing the Krylov projection matrices $H = E\Lambda E^{-1}$ and $H_j = E_j \Lambda_j E_j^{-1}$ and substituting these expressions in (3.18) we obtain

$$u^{(l+1)}(t) = VE\frac{e^{\Lambda t}-I}{\Lambda}E^{-1}V^T b + \int_0^t \sum_{j=2}^{J_l+1} V_j E_j e^{\Lambda_j(t-\tau)} E_j^{-1} V_j^T r_j^{(l)} \tau^j d\tau \tag{3.22}$$

or equivalently,

$$u^{(l+1)}(t) = VE\frac{e^{\Lambda t}-I}{\Lambda}E^{-1}V^T b + \sum_{j=2}^{J_l+1} V_j E_j \left[\int_0^t e^{\Lambda_j(t-\tau)}\tau^j d\tau\right] E_j^{-1} V_j^T r_j^{(l)}. \tag{3.23}$$

Here the integrals

$$I_j = \int_0^t e^{\lambda(t-\tau)}\tau^j d\tau \tag{3.24}$$

which represent the elements of the diagonal matrix $\int_0^t e^{\Lambda_j(t-\tau)}\tau^j d\tau$ are evaluated using the

recurrence

$$I_0 = \frac{e^{\lambda t} - 1}{\lambda} \tag{3.25}$$

$$I_j = \frac{jI_{j-1} - t^j}{\lambda}. \tag{3.26}$$

The algorithm proceeds by computing the iterant $u^{(l+1)}(t)$ at the Chebyshev points $t = \tau_i$ according to the formula (3.23).

An important point we want to emphasize here is how the parameters m, m_j and J are determined in the algorithm proposed in [29, 23]. These quantities are set in advance and do not vary from time step to time step. In other words, Friesner *et.al.* are using a fixed number of Krylov vectors for each Krylov subspace projection without computing the residuals and estimating this number based on the norm of the residuals. In the next section we will discuss how this affects the algorithm's performance.

Finally, we describe the automatic error control mechanism that is used to estimate the accuracy of the solution and determine whether the approximation to $u(t)$ is acceptable or if the time step should be reduced and the procedure repeated for smaller t. From (3.12) and (3.1) it follows that

$$\frac{du(t)}{dt} = \frac{d\mathbf{U}(t_0 + t)}{dt} = F(\mathbf{U}(t_0 + t)) = F(\mathbf{U}_0 + u(t)). \tag{3.27}$$

If $u^{(l+1)}(t)$ is the last approximation in the iteration (3.23), $du^{(l)}/dt$ can be computed by differentiating this expression and using $V_j V_j^T r_j^{(l)} = r_j^{(l)}$ to get

$$\frac{du^{(l+1)}}{dt}(t) = VEe^{Ht}E^{-1}V^T b + \sum_{j=2}^{J_l+1} V_j E_j \Lambda_j \left[\int_0^t e^{\Lambda_j(t-\tau)}\tau^j d\tau\right] E_j^{-1} V_j^T r_j^{(l)} + \sum_{j=2}^{J_l+1} t^j r_j^{(l)}. \tag{3.28}$$

Based on this formula the residual vector is defined via

$$\Delta u \equiv t\left[\frac{du^{(l+1)}(t)}{dt} - F(\mathbf{U}_0 + u^{(l+1)}(t))\right]. \tag{3.29}$$

If the predefined relative and absolute tolerances are ε_{rel} and ε_{abs} respectively, and

$$\delta = \frac{\varepsilon_{abs}}{\varepsilon_{rel}}$$

then the accuracy check used is

$$\|\Delta u\| \leq \varepsilon_{rel} \tag{3.30}$$

where the norm $\|\cdot\|$ is defined either as

$$\|\Delta u\| = \left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\Delta u_i}{|\mathbf{U}_{0_i}| + \delta}\right)^2\right]^{1/2} \tag{3.31}$$

or

$$\|\Delta u\| = \frac{\max\limits_{1\leq i\leq N}|\Delta u_i|}{\left(\frac{1}{N}\sum\limits_{i=1}^{N}\mathbf{U}_{0_j}^2\right)^{1/2}}. \tag{3.32}$$

Now the residual is computed at each of the Chebyshev points $\tau_1, \ldots, \tau_J$. If inequality (3.30) is satisfied at neither of the points τ_i, $1 \leq i \leq J$, then the time step is reduced by some factor C and the algorithm is repeated on the interval $[0, t/C]$. Otherwise, the approximation at the largest value of τ_p for which the inequality (3.30) is satisfied is accepted, the solution is advanced as $\mathbf{U}(t_0 + \tau_p) = u^{(l+1)}(\tau_p) - \mathbf{U}_0$ and the next time step is set to be $t = \tau_p$.

3.2.2 Performance

Several numerical experiments using the iterative exponential propagation method in its original form proposed in [29, 23] showed that the error in the approximation to the solution can be large. The method exhibited stability and therefore if the test system to be solved was supposed to evolve to an equilibrium state the numerical scheme (3.23) was also likely to eventually converge to this steady state. However, the time dynamics of the solution can be computed incorrectly and the error evolves to a large value before eventually settling to zero at the equilibrium. To understand this performance let us consider the error checking mechanism in the method and analyze why it does not necessarily reflect the behavior of the actual error.

The number of Krylov vectors is predetermined and is not changed from time step to time step based on the residuals in the iterative exponential propagation method. Therefore if the method is to perform well, we have to expect that the behaviour of the norm of the

residual vector Δu defined in (3.29) reflects the evolution of the actual error and forces time step reduction when the error becomes larger than a given tolerance. Let us investigate what exactly Δu represents and why a small norm of this vector does not guarantee small residuals in the Arnoldi algorithms used to approximate exponential operators in (3.18) or a small overall error.

To make the analysis clearer, let us first assume that $R(u(t)) = 0$, i.e. analyze the linear problem. Without loss of generality we can just consider advancing the solution over one time step from 0 to time t and assume zero initial value. If the value of the initial vector or the starting time are not zero we can just shift the variable as in (3.12) and compute the solution in terms of the new variable. So the system we are interested in solving is

$$\frac{du}{dt} = Au + b \tag{3.33}$$

$$u(0) = 0. \tag{3.34}$$

For this problem, the iterative exponential propagation method (3.23) reduces to the linear scheme described in Chapter 1:

$$\hat{u}(t) = V\frac{e^{Ht} - I}{H}V^T b = V\phi(Ht)V^T b, \tag{3.35}$$

where t represents the time step, $\hat{u}$ is the approximate solution at time t, V and H are the matrices obtained as the result of the Krylov subspace projection with the starting vector $v_1 = b/\|b\|_2$. Let us define the error of Arnoldi approximation to the function $\phi(At)$ as

$$\epsilon_\phi = \frac{e^{At} - I}{A} - V\frac{e^{Ht} - I}{H}V^T b = \|b\|_2 V\phi(Ht)e_1. \tag{3.36}$$

From expression (3.35) we can compute $d\hat{u}/dt$ to get

$$\frac{d\hat{u}}{dt} = Ve^{Ht}V^T b. \tag{3.37}$$

The residual vector used to check the accuracy of the approximation to the solution is defined by (3.29) as

$$\Delta\hat{u} = \left[\frac{d\hat{u}}{dt} - F(\hat{u})\right]t. \tag{3.38}$$

Now combining formulas (3.35-3.38) and using the matrix formulation of the Arnoldi iteration

$$AV = VH + h_{m+1,m} v_{m+1} e_m^T, \tag{3.39}$$

we express the residual as

$$\begin{aligned}
\Delta\hat{u} &= \left[Ve^{Ht}V^T b - A\hat{u} - b \right] t \\
&= \left[Ve^{Ht}V^T b - AV\phi(Ht)V^T b - b \right] t \\
&= \left[Ve^{Ht}V^T b - (VH + h_{m+1,m} v_{m+1} e_m^T)\phi(Ht)V^T b - b \right] t \\
&= \left[Ve^{Ht}V^T b - V(e^{Ht} - I)V^T b + \|b\|_2 h_{m+1,m} v_{m+1} e_m^T \phi(Ht) e_1 - b \right] t \\
&= \|b\|_2 h_{m+1,m} [\phi(Ht)]_{m,1} v_{m+1} t.
\end{aligned} \tag{3.40}$$

Recall the definition (2.39) of the generalized residual ρ_m derived in Chapter 2. Formula (3.40) then gives us

$$\Delta\hat{u} = -t\rho_m. \tag{3.41}$$

It is clear now that in the linear case the residual (3.40) is simply the residual of the Arnoldi approximation scaled by the time step. Since the residual ρ_m can be computed as a side product of the Krylov projection operator from

$$\rho_m = \|b\|_2 h_{m+1,m} [\phi(Ht)]_{m,1} v_{m+1} \tag{3.42}$$

calculation of $\Delta\hat{u}$ via formula (3.38) is unnecessary. As we can also see from (3.40) by fixing the number of Krylov vectors in advance as it is done in [29, 23], we limit ourselves to having only the option to reduce the time step to get an accurate solution, while the error can also be reduced by increasing the number of Arnoldi iterations. Despite these limitations and extra work we have to do to check the accuracy, however, it is still true that in the linear case the method of the previous section will provide us with the accurate solution and the residual $\Delta\hat{u}$ will capture the behavior of the actual error.

Now let us consider the general nonlinear problem

$$\frac{du}{dt} = Au + b + R(u) \tag{3.43}$$

$$u(0) = 0, \tag{3.44}$$

solved as described in the previous section using the method

$$\hat{u}^{(l+1)}(t) = V\frac{e^{Ht} - I}{H}V^T b + \sum_{j=2}^{J_l+1} V_j \left[\int_0^t e^{H_j(t-\tau)}\tau^j d\tau\right] V_j^T r_j^{(l)}. \tag{3.45}$$

Let us set $\hat{u}^{(l+1)}$ to be the final iteration and just define $\hat{u} = \hat{u}^{(l+1)}$, $J = J_l$ and $r_j = r_j^{(l)}$. To compute $d\hat{u}/dt$ we differentiate (3.45) and use $V_j V_j^T r_j = r_j$ to obtain

$$\frac{d\hat{u}}{dt} = Ve^{Ht}V^T b + \sum_{j=2}^{J_l+1} V_j H_j \left[\int_0^t e^{H_j(t-\tau)}\tau^j d\tau\right] V_j^T r_j + \sum_{j=2}^{J_l+1} t^j r_j. \tag{3.46}$$

We can also compute

$$\begin{aligned} F(\hat{u}) &= b + A\hat{u} + R(\hat{u}) \\ &= b + AV\phi(Ht)V^T b + A\sum_{j=2}^{J_l+1} V_j H_j \left[\int_0^t e^{H_j(t-\tau)}\tau^j d\tau\right] V_j^T r_j + R(\hat{u}). \end{aligned} \tag{3.47}$$

Now by subtracting (3.47) from (3.46) we can express the residual $\Delta\hat{u}$ as follows

$$\begin{aligned} \Delta\hat{u} =& Ve^{Ht}V^T b - bAV\phi(Ht)V^T b \\ &+ \sum_{j=2}^{J_l+1} V_j H_j \left[\int_0^t e^{H_j(t-\tau)}\tau^j d\tau\right] V_j^T r_j - A\sum_{j=2}^{J_l+1} V_j \left[\int_0^t e^{H_j(t-\tau)}\tau^j d\tau\right] V_j^T r_j \\ &+ \sum_{j=2}^{J_l+1} t^j r_j - R(\hat{u}). \end{aligned} \tag{3.48}$$

As we showed earlier in (3.40)

$$Ve^{Ht}V^T b - bAV\phi(Ht)V^T b = \rho_m(\phi(Ht)). \tag{3.49}$$

where $\rho_m(\phi(Ht)$ is the residual of the Arnoldi approximation to $\phi(Ht)$. Similarly by substituting

$$AV_j = V_j H_j + h^{(j)}_{m_j+1,m_j} v_{m_j+1} e^T_{m_j} \tag{3.50}$$

in (3.48) and defining the residual of the Arnoldi approximation to $\left[\int_0^t e^{A(t-\tau)} r_j \tau^j d\tau\right]$ as

$$\rho_{m_j} = \|r_j\|_2 h^{(j)}_{m_j+1,m_j} \left[\int_0^t e^{H_j(t-\tau)} \tau^j d\tau\right]_{m_j,1} v_{m_j+1} \tag{3.51}$$

we obtain

$$\begin{aligned} \Delta\hat{u} = &- \left[\rho_m(\phi(Ht)) + \sum_{j=2}^{J_+1} \rho_{m_j}\right] t \\ &+ \sum_{j=2}^{J_+1} t^j r_j - R(\hat{u}). \end{aligned} \tag{3.52}$$

Finally, if we define the error of the polynomial approximation to the nonlinear remainder function $R(\hat{u})$ as

$$\varepsilon_p = R(\hat{u}) - \sum_{j=2}^{J_+1} t^j r_j \tag{3.53}$$

we get the expression for the residual vector

$$\Delta\hat{u} = -\left[\rho_m(\phi(Ht))t + t\sum_{j=2}^{J_+1} \rho_{m_j} + \varepsilon_p\right]. \tag{3.54}$$

The last expression means that the residual is actually the sum of the errors of different approximation stages in the method. The residuals and the error in (3.54) can have either positive or negative signs. So if, for example, the norm of the error of the polynomial approximation to the nonlinear integral ε_p is large or the preset number of Arnoldi iterations was not enough to obtain adequate accuracy in computing the exponential terms and the residuals' norms $\|\rho_m(\phi(Ht))\|$ or $\|\rho_{m_j}\|$ are large we can still have $\Delta\hat{u}$ with a small norm due to all the errors canceling each other in the expression (3.54). Once we diagonalize the

matrices H and H_j, the expression for the residual will also contain this approximation, so that the residual $\Delta\hat{u}$ is even more unusable as a global accuracy test. The correct and cheaper way to check the actual error is to minimize $\|\rho_m(\phi(Ht))\|$, $\|\rho_{m_1}\|$ and $\|\varepsilon_p\|$ and advance the solution to the minimum time t which makes all these errors less than a predefined tolerance.

Our conclusions about this method are supported by the numerical tests that Friesner *et.al* presented in [29, 23]. They tested the method on three problems: the nonlinear system of ODEs called the Krogh model [31] for which the exact solution is available, a chemical reaction-diffusion system with no exact solution at hand and the axisymmetric Taylor vortex problem for which there is also no closed form solution. First, note that in all of these calculations the authors found that the behaviour of the error as well as the efficiency of the method is highly sensitive on the values of parameters m, m_j, J and the number L of global iterations in (3.23). In the Krogh problem, in order to find the set of these parameters that provides the best efficiency, a "manual" optimization had to be performed. Such an optimization would be impossible to perform for a general large problem, since completing so many runs to determine the parameters would be too expensive and impractical. This result also indicates that the values of parameters m, m_j, J, L could not be determined from the residual $\Delta\hat{u}$. This agrees with our conclusion that this residual contains information about all of the parameters, but the information is in the form of a summation which does not allow determining what are the local errors at each stage of the computation and consequently does not minimize the global error. Moreover, the exact error in the Krogh model computation actually increased when the number of the Krylov iterations for one of the nonlinear terms in the scheme was increased. If the polynomial fitting of the function in the nonlinear integral was providing a good approximation then we would see a decrease in the error since more Krylov vectors means a better approximation to an exponential matrix-vector product. In the Krogh problem the optimization of the parameters and calculation of the exact error are easy to accomplish since the exact solution is available. Let us know look at the reaction-diffusion problem results obtained in [29]. Not only was the approximated solution computed by the method sensitive to the parameters m, m_j, J, L, but apparently even the relative tolerance value affected the phase of the solution. So the authors were not able to perform any convergence study and mainly discussed the efficiency of the method without addressing the accuracy of the solution. A very similar discussion was presented

for the Taylor vortex problem in [23], so it is hard to make a judgement on whether the approximation to the solution computed by the iterative exponential propagation method was actually accurate. Since the importance of the parameters changes from problem to problem and the numerical solution is found to be so sensitive to their values it seems natural to optimize different parts of the algorithm separately, i.e. as we proposed above to compute errors at different stages and optimize the values of the parameters based on the errors at the particular stage involving these parameters. From the results of the numerical tests on the three problems from [29, 23] we can conclude that the automatic error control mechanism based on the residual $\Delta\hat{u}$ does not allow us to calculate the set of parameters m, m_j, J, L which would provide a consistent and accurate solution in the most efficient way.

Although the discussion above concludes that the iterative exponential propagation method, in the form it is introduced in [29, 23], has many limitations we believe that further research on the development of the multistep-type exponential propagation methods can lead to very interesting results and powerful numerical schemes. Having developed an intuition and understanding of the issues involved in developing such methods we plan to investigate this question further in our future research. In this work, however, we take a different approach and use the methods proposed by Hochbruck and Lubich in [39, 40], which combine a Runge-Kutta framework with the exponential propagation ideas to design numerical schemes. In the subsequent sections we will describe these methods and their advantages over the iterative method of Friesner *et.al.*

3.3 Runge-Kutta exponential propagation methods

3.3.1 Introduction and formulation

First we would like to discuss briefly the background and the motivation behind developing the Runge-Kutta exponential propagation methods. More detailed discussion of the individual topics discussed in this summary can be found, for example, in [50, 35, 34, 76]. It is well known that if we want to solve a stiff nonlinear system (3.1) the numerical method we use has to be tailored to handle the stiffness. The stiffness of the problem implies that the step length in the numerical scheme is constrained by the stability rather than accuracy requirements. This consideration led Dahlquist to introduce the notion of A-stability of a numerical method. By definition, a numerical scheme is called A-stable if its stability do-

main contains the entire left half of the complex plane $\mathbb{C}^-$. Dahlquist also proved that there are no A-stable multistep methods of order greater than 2 and that the implicit trapezoidal rule is the second order multistep method with the smallest error constant. It was, however, possible to derive multistep methods with good stability properties, using backward differentiation formulae (BDF). If we define a solution of the autonomous problem (3.1) at times t_n and t_{n+1} to be $\mathbf{U}^n$ and $\mathbf{U}^{n+1}$ respectively , the standard form of the s-step s-order BDF method is

$$\sum_{j=0}^{s} a_j \mathbf{U}_{n+j} = h b_s F(\mathbf{U}^{n+s}), \tag{3.55}$$

where $h = t_{n+1} - t_n$ is a time step and coefficients a_j and b_s are determined based on what order method we want to derive. Even though BDF methods do not possess A-stability their efficiency can compensate for this. As it is true for any multistep method, however, to change the time step in the course of integration is difficult and the time step control mechanism for this type of methods is expensive.

The alternative to multistep schemes is to use the Runge-Kutta framework to derive implicit methods. Kaps and Rentrop [45] showed that it is possible to derive implicit Runge-Kutta methods of order higher than 2 which are A-stable, efficient and have simple structure [76]. Another advantage of these types of methods is that it is possible to construct a robust time step control mechanism via embedding of methods of a different order. The latter feature will be described in detail in section 3.4. The general s-stage implicit Runge-Kutta method to evolve the system from time t_n to time t_{n+1} can be formulated as

$$\begin{aligned} k_i &= F\left(\mathbf{U}^n + h\sum_{j=1}^{s} a_{ij} k_j\right), \qquad i = 1, \ldots, s \\ \mathbf{U}^{n+1} &= \mathbf{U}^n + h\sum_{j=1}^{s} b_i k_i, \end{aligned} \tag{3.56}$$

where once again $h = t_{n+1} - t_n$ is the time step and coefficients a_{ij} and b_i are determined based on the order of the method we want to derive.

As we can see from formulas (3.55) and (3.56) both BDF and implicit Runge-Kutta methods require solving a nonlinear system using a Newton iteration, which adds greatly to the complexity of the methods. If we use the implicit Runge-Kutta method (3.56) with the

full $s \times s$ matrix (a_{ij}) then $(N \times s)$ simultaneous implicit nonlinear systems have to be solved, given N is the length of the vector $\mathbf{U}^n$. To reduce the complexity of such methods it was proposed in the early seventies [2, 20, 49, 62, 1] to make the matrix (a_{ij}) lower triangular, i.e. $a_{ij} = 0$ for $i \leq j$, so that the equations (3.56) can be solved in s stages with only one N-dimensional system to be solved at each stage. These constitute the so-called diagonally implicit (DIRK) Runge-Kutta methods. Finally, to decrease the operational count even further we can set

$$a_{ii} = \gamma \qquad i = 1, \ldots, s, \tag{3.57}$$

so that the method becomes

$$\begin{aligned} k_i &= F\left(\mathbf{U}^n + h\sum_{j=1}^{i-1} a_{ij}k_j + \gamma k_i\right) \qquad i = 1, \ldots, s \\ \mathbf{U}^{n+1} &= \mathbf{U}^n + h\sum_{j=1}^{s} b_i k_i, \end{aligned} \tag{3.58}$$

and is called the singly diagonally implicit method (SDIRK). Here we need to solve s nonlinear systems by Newton-type iteration. Since in this case at each of the s stages the Jacobian matrix for the Newton iteration is $(I - h\gamma\partial F/\partial \mathbf{U})$, we could perform LU-factorization of this matrix only once and then use it at each stage to solve linear systems of type

$$\left(I - h\gamma\partial F/\partial \mathbf{U}\right)x = y. \tag{3.59}$$

The development and success of the singly diagonally implicit Runge-Kutta methods led to the following observation. The numerical technique which performs well in solving stiff systems (3.1) uses not only function evaluations $F(\mathbf{U})$ but also information provided by the Jacobian matrix $\partial F/\partial \mathbf{U}$. Once this fact was recognized there were a number of methods proposed which tried to incorporate the Jacobian matrix directly into the formulation of the scheme rather then use it through the Newton iteration. Some of the successful methods of this type, which are also distinguished by a fairly simple structure, were proposed by Rosenbrock [69] and are called semi-implicit, or just Rosenbrock methods. Rosenbrock noticed that one Newton iteration to solve (3.58) with the starting value $k_i^{(0)}$ can be written

as the linear system

$$k_i = F\left(\mathbf{U}^n + h\sum_{j=1}^{i-1} a_{ij}k_j\right) + h\frac{\partial F}{\partial \mathbf{U}}\left(\mathbf{U}^n + \sum_{j=1}^{i-1} a_{ij}k_j\right)\gamma k_i. \tag{3.60}$$

So instead of continuing the Newton iteration he proposed to evaluate the Jacobian matrix at $\mathbf{U}^n$ and formulate the s-stage numerical scheme as

$$\begin{aligned} k_i &= F\left(\mathbf{U}^n + h\sum_{j=1}^{i-1} \alpha_{ij}k_j\right) + hA\sum_{j=1}^{i} \gamma_{ij}k_j, \qquad i = 1, \dots s \\ \mathbf{U}^{n+1} &= \mathbf{U}^n + h\sum_{j=1}^{s} b_j k_j, \end{aligned} \tag{3.61}$$

where $A = \partial F/\partial \mathbf{U}(\mathbf{U}^n)$ and the coefficients α_{ij}, γ_{ij} and b_j are determined based on the order we want to method to have. To simplify the method further and reduce the number of matrix factorizations to be performed at each of Runge-Kutta stages we can look for the methods with $\gamma_{ii} = \gamma$ for $i = 1, \dots, s$. In this case we can also solve for k_i in (3.61) and rewrite the method as

$$\begin{aligned} k_i &= (I - \gamma hA)^{-1}\left(F(u_i) + hA\sum_{j=1}^{i-1} \gamma_{ij}k_j\right), \qquad i = 1, \dots s \\ u_i &= \mathbf{U}^n + h\sum_{j=1}^{i-1} \alpha_{ij}k_j \\ \mathbf{U}^{n+1} &= \mathbf{U}^n + h\sum_{j=1}^{s} b_j k_j, \end{aligned} \tag{3.62}$$

In a number of numerical studies [76, 50, 35] it was shown that when the function evaluations $F(\mathbf{U})$ are relatively cheap, Runge-Kutta methods are typically more efficient than multistep methods for initial value problems. In general, Runge-Kutta methods are more robust, provide easy mechanisms to estimate local error at each time step, and allow the time step to be changed without the overhead of the time control process of the multistep methods. If the system to be solved is stiff, implicit methods with a variable time step have to be used. Then the major complexity of the integration of the system over one time step for either multistep or Runge-Kutta implicit method comes from solving the linear systems of type (3.59) inside Newton iterations. The function evaluations account for a lesser number

of total flops. So if exponential propagation ideas can reduce the time needed to solve the linear systems (3.59), then the considerations discussed in this subsection lead to an idea of using the Runge-Kutta formulation and augment it with exponential propagation to obtain an effective numerical method for stiff problems. This is exactly what Hochbruck and Lubich first proposed in [39] and later generalized in [40]. They combined the idea behind the formulation of the Rosenbrock methods (3.61) and the exponential Euler method (3.11) to obtain the following general exponential Runge-Kutta method for the stiff nonlinear systems of type (3.1)-(3.2)

$$
\begin{aligned}
k_i &= \frac{e^{\gamma hA} - I}{\gamma hA}\left(F(u_i) + hA\sum_{j=1}^{i-1}\gamma_{ij}k_j\right), \qquad i = 1, \dots s \\
u_i &= \mathbf{U}^n + h\sum_{j=1}^{i-1}\alpha_{ij}k_j \\
\mathbf{U}^{n+1} &= \mathbf{U}^n + h\sum_{j=1}^{s}b_j k_j.
\end{aligned}
\tag{3.63}
$$

As is evident from comparing the method above with the Rosenbrock methods, (3.63) is obtained by replacing the rational function of argument

$$\phi(z) = (1 - z)^{-1} \tag{3.64}$$

of argument $z = \gamma hA$ in (3.61) with the function

$$\phi(z) = \frac{e^z - 1}{z}. \tag{3.65}$$

Since we expect the matrix A to be large, iterative methods have to be used to approximate either of the functions. From the analysis in the previous chapter, however, we concluded that if we use Krylov subspace projection for this purpose, the error of the Arnoldi algorithm applied to approximate (3.65) can be reduced faster than if this algorithm is used to calculate (3.64). This gives us a hope that a method of type (3.63) can offer an efficient and robust technique to solve stiff nonlinear systems of differential equations.

3.3.2 Order of the methods

As we noted before, methods of a certain order of type (3.63) can be obtained by specifying the coefficients α_{ij}, γ,γ_{ij} and b_j. Since the structure of these methods is very similar to Runge-Kutta schemes we can use the same approach to determine these coefficients as for Rosenbrock methods. Suppose our goal is to derive a method of order p, i.e. we want the local error to satisfy

$$\mathbf{U}(t_n + h) - \mathbf{U}(t_n) = O(h^{p+1}). \tag{3.66}$$

One of the ways to achieve this is to substitute expressions (3.63) into (3.66), expand the resulting formula for $\mathbf{U}^{n+1}$ in Taylor series and chose the parameters in the method is such a way that the coefficients of the terms with h^q, $q \leq p$, vanish. To deal with the complexity of the expression resulting from this procedure Butcher [16] developed a general theory which used elementary differentials and their graphical representation as a rooted trees to derive the order conditions for the Runge-Kutta methods. The detailed description of this theory can be found in a number of references, see for example [50, 35]. Hochbruck and Lubich applied the results of this theory to Rosenbrock methods with the modification of the function $\phi(z)$ in (3.64) and derived order conditions for the methods (3.63). Below we present these conditions.

Let us restate the formulation of exponential propagation Runge-Kutta methods:

$$k_i = \frac{e^{\gamma hA} - I}{\gamma hA}\left(F(u_i) + hA\sum_{j=1}^{i-1}\gamma_{ij}k_j\right), \qquad i = 1, \dots s \tag{3.67}$$

$$u_i = \mathbf{U}^n + h\sum_{j=1}^{i-1}\alpha_{ij}k_j \tag{3.68}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + h\sum_{j=1}^{s}b_jk_j. \tag{3.69}$$

After completing the Taylor expansion of these formulas we will obtain the coefficients of h^q, $q \leq p$, in terms of elementary differentials of the function $F(\mathbf{U})$. Now we will state some definitions from [35] which will help us specify the order conditions in the simple form.

Definition 3.3.1 *Let A be an ordered chain of indices $A = \{j \leq k \leq l \leq m \leq \dots\}$ and denote by A_q the subset consisting of the first q indices.* **A rooted labeled tree of order**

q ($q \geq 1$) is the son-father mapping

$$t : A_q \backslash \{j\} \to A_q \tag{3.70}$$

such that $t(z) \leq z$ for all $z \in A_q \backslash \{j\}$. The set of all labeled trees of order q is denoted by LT_q. z is called the son of $t(z)$ and $t(z)$ the father of z. The order of the tree is equal to the number of its vertices and denoted by $q = \varrho(t)$.

Definition 3.3.2 *Two labeled trees t and w are* **equivalent** *if they have the same order q, and if there exists a permutation $\sigma : A_q \to A_q$, such that $\sigma(j) = j$ and $t\sigma = \sigma w$ on $A_q \backslash j$. An equivalence class of qth order labeled trees is called a* **rooted tree of order** *q.*

Now we can specify the correspondence between the elementary differentials and the labeled trees. The elementary differential can be written in tensor notation and defined as

Definition 3.3.3 *For a labeled tree $t \in LT_q$ we call*

$$f^J(\mathbf{U}) = \sum_{K,L,\ldots} F^J_{K,\ldots}(\mathbf{U}) F^K_{\ldots}(\mathbf{U}) F^L(\mathbf{U}), \tag{3.71}$$

the corresponding **elementary differential**. *The summation is over $q-1$ indices $K, L, \ldots$ and the summand is a product of q F's, where the upper index specifies the component and runs through all vertices of t and the lower indices are the corresponding sons. The lower subscript represents differentiation with respect to a component.*

Finally, let us set

$$\beta_{ij} = \alpha_{ij} + \gamma_{ij}. \tag{3.72}$$

and define

Definition 3.3.4 *Let t be a labeled tree of order q with root j, denote*

$$\Phi_j(t) = \sum_{k,l,\ldots} \varphi_{j,k,l,\ldots} \tag{3.73}$$

to be the sum over the remaining $q-1$ indices $k, l, \ldots$. The summand $\varphi_{j,k,l,\ldots}$ is a product of $q-1$ factors which are

$\beta_{kl} = \alpha_{kl} + \gamma_{kl}$ *if l is the only son of k;*

α_{kl} *if l is a son of k and k has at least two sons.*

Now by substituting formulas (3.67) - (3.69) into (3.66), expanding the resulting expression for $\mathbf{U}^{n+1}$ in Taylor series and setting the coefficients of the low powers of h to zero the following order conditions for (3.63) will be obtained:

Definition 3.3.5 *An exponential method (3.67)-(3.69) is of order p iff*

$$\sum_{j=1}^{s} b_j \Phi_j(t) = P_t(\gamma), \tag{3.74}$$

for elementary differentials t up to order p. $\Phi_j(t)$ is specified in Definition (3.3.4) and $P_t(\gamma)$ is the polynomial in γ which corresponds in the collection of terms involving γ in the Taylor expansion of (3.66).

Table 3.1 lists $\Phi_j(t)$ and the polynomials $P_t(\gamma)$ corresponding to the elementary differentials up to order $p \leq 5$ as given in [40].

It is clear that the second order method (3.11) is exact for the linear systems when the exponential of the Jacobian is computed directly without the Krylov projection. Consequently, if the linear problem is well-posed this method is A-stable. In general if the method of type (3.67) - (3.69) is exact for linear systems it is A-stable. The following theorem proved in [40] gives a condition under which methods (3.67) - (3.69) are exact for linear systems.

Theorem 3.3.1 *The method (3.67) - (3.69) is exact for linear differential equations iff for all $n = 1, 2, 3, ...$*

$$\sum b_{j_1} \beta_{j_1,j_2} \beta_{j_2,j_3} \dots \beta_{j_{n-1},j_n} = \frac{1}{n}(\frac{1}{n-1} - \gamma)(\frac{1}{n-2} - \gamma) \dots (1/2 - \gamma)(1 - \gamma). \tag{3.75}$$

As we can see from this theorem if γ is the reciprocal of some integer only a few of conditions (3.75) need to be fulfilled and for n large enough the rest of the conditions will be satisfied automatically. In the next subsection we will show that this choice of γ in fact produces the most effective methods of type (3.67) - (3.69).

Elementary differential t	$\Phi_j(t)$	$P_t(\gamma)$
F^J	1	1
$\sum_K F^J_K F^K$	$\sum_k \beta_{jk}$	$1/2(1-\gamma)$
$\sum_{K,L} F^J_{KL} F^K F^L$	$\sum_{k,l} \alpha_{jk}\alpha_{jl}$	$1/3$
$\sum_{K,L} F^J_K F^K_L F_L$	$\sum_{k,l} \beta_{jk}\beta_{kl}$	$1/3(1/2-\gamma)(1-\gamma)$
$\sum_{K,L,M} F^J_{KLM} F^K F^L F^M$	$\sum_{k,l,m} \alpha_{jk}\alpha_{jl}\alpha_{jm}$	$1/4$
$\sum_{K,L,M} F^J_{KM} F^K_L F^L F^M$	$\sum_{k,l,m} \alpha_{jk}\beta_{kl}\alpha_{jm}$	$1/8-\gamma/6$
$\sum_{K,L,M} F^J_K F^K_{LM} F^L F^M$	$\sum_{k,l,m} \beta_{jk}\alpha_{kl}\alpha_{km}$	$1/12-\gamma/6$
$\sum_{K,L,M} F^J_K F^K_L F^L_M F^M$	$\sum_{k,l,m} \beta_{jk}\beta_{kl}\beta_{lm}$	$1/4(1/3-\gamma)(1/2-\gamma)(1-\gamma)$
$\sum_{K,L,M,P} F^J_{KLMP} F^K F^L F^M F^P$	$\sum_{k,l,m,p} \alpha_{jk}\alpha_{jl}\alpha_{jm}\alpha_{jp}$	$1/5$
$\sum_{K,L,M,P} F^J_{KMP} F^K_L F^L F^M F^P$	$\sum_{k,l,m,p} \alpha_{jk}\beta_{kl}\alpha_{jm}\alpha_{jp}$	$1/10-\gamma/8$
$\sum_{K,L,M,P} F^J_{KP} F^K_{ML} F^L F^M F^P$	$\sum_{k,l,m,p} \alpha_{jk}\alpha_{kl}\alpha_{km}\alpha_{jp}$	$1/15$
$\sum_{K,L,M,P} F^J_{KP} F^K_L F^L_M F^M F^P$	$\sum_{k,l,m,p} \alpha_{jk}\beta_{kl}\beta_{lm}\alpha_{jp}$	$1/30-\gamma/8+\gamma^2/9$
$\sum_{K,L,M,P} F^J_{KM} F^K_L F^L F^M_P F^P$	$\sum_{k,l,m,p} \alpha_{jk}\beta_{kl}\alpha_{jm}\beta_{mp}$	$1/20-\gamma/8+\gamma^2/12$
$\sum_{K,L,M,P} F^J_K F^K_{LMP} F^L F^M F^P$	$\sum_{k,l,m,p} \beta_{jk}\alpha_{kl}\alpha_{km}\alpha_{kp}$	$1/20-\gamma/8$
$\sum_{K,L,M,P} F^J_K F^K_{LP} F^L_M F^M F^P$	$\sum_{k,l,m,p} \beta_{jk}\alpha_{kl}\beta_{lm}\alpha_{kp}$	$1/40-5\gamma/48+\gamma^2/12$
$\sum_{K,L,M,P} F^J_K F^K_L F^L_{MP} F^M F^P$	$\sum_{k,l,m,p} \beta_{jk}\beta_{kl}\alpha_{lm}\alpha_{lp}$	$1/60-\gamma/12+\gamma^2/9$
$\sum_{K,L,M,P} F^J_K F^K_L F^L_M F^M_P F^P$	$\sum_{k,l,m,p} \beta_{jk}\beta_{kl}\beta_{lm}\beta_{mp}$	$1/5(1/4-\gamma)(1/3-\gamma)(1/2-\gamma)(1-\gamma)$

Table 3.1: Order conditions for the exponential Runge-Kutta methods up to order 5

3.3.3 Reducing complexity

The s-stage exponential propagation method (3.67) - (3.69) requires s Krylov subspace projections and s function F evaluations to be performed to compute all the k_i's. As in the case of Friesner *et. al.*'s iterative exponential methods of the previous section, this may be prohibitively expensive to compute if the system if large and a higher order method is used. A special reformulation of the method proposed in [40] helps reduce the complexity and derive more efficient methods of type (3.67) - (3.69).

In order to reduce the number of Arnoldi approximations we need to perform to calculate matrix-vector products

$$\phi(\gamma hA)v = \frac{e^{\gamma hA} - I}{\gamma hA}v$$

we can use the recurrence relation which allows computing $\phi(jz)$, $j = 2, 3, \ldots$ once $\phi(z)$ is calculated. For example, $\phi(2z)$ and $\phi(3z)$ can be computed from

$$\phi(2z) = \frac{1}{2}\Big(z\phi(z) + 1\Big)\phi(z) + \frac{1}{2}\phi(z) \tag{3.76}$$

$$\phi(3z) = \frac{2}{3}\Big(z\phi(z) + 1\Big)\phi(2z) + \frac{1}{3}\phi(z). \tag{3.77}$$

In general, it is easy to verify by simple algebra that for arbitrary $n = 2, 3, \ldots$ the following formula allows computing $\phi(jz)$ recursively

$$\phi(nz) = \frac{n-1}{n}\Big(z\phi(z) + 1\Big)\phi((n-1)z) + \frac{1}{n}\phi(z). \tag{3.78}$$

Therefore, if we approximated $\phi(\gamma hA)v$ by the Krylov subspace projection to get

$$\phi(\gamma hA)v \approx V_m\phi(\gamma hH_m)V_m^T v, \tag{3.79}$$

then rather than performing the Krylov projection again to compute $\phi(j\gamma hA)v$ we can use

recurrence (3.78) and the fact that matrix H_m is small to cheaply calculate

$$\begin{aligned}
\phi(2\gamma hA) &\approx V_m \phi(2\gamma h H_m) V_m^T v \\
&= V_m \left[\frac{1}{2} \Big(\gamma h H_m \phi(\gamma h H_m) + I_m \Big) \phi(\gamma h H_m) + \frac{1}{2} \phi(\gamma h H_m) \right] V_m^T v \qquad (3.80) \\
&\vdots \\
\phi(j\gamma hA) &\approx V_m \phi(j\gamma h H_m) V_m^T v \\
&= V_m \left[\frac{j-1}{j} \Big(\gamma h H_m \phi(\gamma h H_m) + I_m \Big) \phi((j-1)\gamma h H_m) + \frac{1}{j} \phi(\gamma h H_m) \right] V_m^T v. \qquad (3.81)
\end{aligned}$$

We also employ another useful formula which will help us find the coefficients β_{ij} once we reformulate the method.

Lemma 3.3.1 *If $\phi(z) = (e^z - 1)/z$ then for $n = 1, 2, 3, \ldots$ the following relation holds:*

$$\phi(nz) = \phi(z) \left(1 + \frac{z}{n} \sum_{j=1}^{n-1} \phi(jz) j \right) \qquad (3.82)$$

Proof. By simple algebraic manipulation of the right hand side of (3.82) we obtain:

$$\begin{aligned}
\phi(z) \left(1 + \frac{z}{n} \sum_{j=1}^{n-1} \phi(jz) j \right) &= \frac{e^z - 1}{z} \left(1 + \frac{z}{n} \sum_{j=1}^{n-1} \frac{e^{jz} - 1}{jz} j \right) \\
&= \frac{e^z - 1}{z} \left(1 + \frac{1}{n} \sum_{j=1}^{n-1} (e^{jz} - 1) \right) \\
&= \frac{e^z - 1}{nz} \left(1 + \sum_{j=1}^{n-1} e^{jz} \right) \qquad (3.83) \\
&= \frac{1}{nz} \left(e^z - 1 + \sum_{j=1}^{n-1} e^{(j+1)z} - \sum_{j=1}^{n-1} e^{jz} \right) \\
&= \frac{e^{nz} - 1}{nz} \\
&= \phi(nz). \qquad \square
\end{aligned}$$

We define auxiliary vectors

$$d_i = F(u_i) - F(\mathbf{U}^n) - hA \sum_{j=1}^{i-1} \alpha_{ij} k_j. \qquad (3.84)$$

Since the expression for d_i corresponds to the first order Taylor expansion of F as more stages are completed we can expect vectors d_i become smaller in norm. The general method (3.67)-(3.69) is then reformulated in terms of the newly introduced vectors d_i in such a way that recurrences (3.78) and (3.82) are used, i.e.

$$k_i = k_1 + \phi(\gamma hA)d_i + \phi(\gamma hA)hA\sum_{j=1}^{i-1}\beta_{ij}k_j. \tag{3.85}$$

Now we set

$$\gamma = 1/p, \tag{3.86}$$

where p is some small integer, and

$$\alpha_{ij} = 0, \qquad j \leq i, \quad i = 1, \ldots, p, \tag{3.87}$$

so that

$$u_i = \mathbf{U}^n, \qquad i = 1, \ldots, p \tag{3.88}$$

and consequently

$$d_i = 0, \qquad i = 1, \ldots, p. \tag{3.89}$$

Since

$$k_1 = \phi(\gamma hA)F(\mathbf{U}^n) \tag{3.90}$$

the above choice of parameters reduces formula (3.85) to

$$k_i = \phi(\frac{h}{p}A)\left(F(\mathbf{U}^n) + hA\sum_{j=1}^{i-1}\beta_{ij}k_j\right), \qquad i = 1, \ldots, p. \tag{3.91}$$

In order to use the recurrence (3.82) we choose coefficients

$$\beta_{ij} = \frac{j\gamma}{i} = \frac{j}{ip}, \qquad j \leq i, \qquad i = 1, \ldots, p. \tag{3.92}$$

This choice of β_{ij} and the formula (3.82) transforms (3.91) to

$$k_i = \phi(\frac{i}{p}hA)F(\mathbf{U}^n), \qquad i = 1, \ldots, p. \tag{3.93}$$

We also want to use recurrence (3.82) for each of the other sets of p stages. To accomplish this let us consider each of these sets. That is, fix $l = 1, \ldots, s$ and derive parameters for each of the stages $pl + i, \quad i = 1, \ldots, p$. First set

$$\alpha_{pl+i,j} = \alpha_{pl+1,j}, \qquad i = 1, \ldots, p, \quad l, j \geq 1 \tag{3.94}$$

which implies given the definition of (3.84) that

$$u_{pl+i} = u_{pl+1}, \qquad i = 1, \ldots, p, \quad l \geq 1 \tag{3.95}$$

$$d_{pl+i} = d_{pl+1}, \qquad i = 1, \ldots, p, \quad l \geq 1. \tag{3.96}$$

Now analogous to (3.93) we would like to get

$$k_{pl+i} = k_1 + \phi(\frac{i}{p}hA)d_{pl+1}, \qquad i = 1, \ldots, p, \quad l \geq 1. \tag{3.97}$$

Here we will work in reverse and derive coefficients β_{ij} given (3.97). Let us split the sum in (3.85) to rewrite is as

$$\begin{aligned} k_{pl+i} &= k_1 + \phi(\frac{h}{p}A)d_{pl+1} \\ &\quad + \phi(\frac{h}{p}A)hA\left[\sum_{j=1}^{p}\beta_{pl+i,j}k_j + \sum_{j=p+1}^{pl}\beta_{pl+i,j}k_j + \sum_{j=pl+1}^{pl+i}\beta_{pl+i,j}k_j\right]. \end{aligned} \tag{3.98}$$

First, we set

$$\beta_{pl+i,j} = 0, \qquad j = 2, \ldots, p, \tag{3.99}$$

$$\beta_{pl+i,j} = 0, \qquad j = p+1, \ldots, pl. \tag{3.100}$$

and choose

$$\beta_{pl+i,j} = \frac{j-pl}{pi}, \qquad j = pl+1, \ldots, pl+i. \tag{3.101}$$

Then using these parameters and substituting (3.97) we reduce (3.98) to

$$\begin{aligned} k_{pl+i} &= k_1 + \phi(\frac{h}{p}A)d_{pl+1} \\ &\quad + \phi(\frac{h}{p}A)hA\left[\beta_{pl+i,1}k_1 + \sum_{j=pl+1}^{pl+i}\frac{j-pl}{pi}\left(k_1 + \phi(\frac{ih}{p}A)d_{pl+1}\right)\right]. \end{aligned} \tag{3.102}$$

Finally, to obtain (3.97) from (3.102) we set

$$\beta_{pl+i,1} = -\sum_{j=pl+1}^{pl+i}\frac{j-pl}{ip} = -\frac{i-1}{2p}. \tag{3.103}$$

To summarize, in this subsection we derived the general formulas (3.87), (3.94), (3.92), (3.99), (3.100), (3.101) and (3.103) for the coefficients of the reduced methods proposed in [40]:

$$k_i = \phi(\frac{i}{p}hA)F(\mathbf{U}^n), \qquad i = 1, \ldots, p \tag{3.104}$$

$$k_i = k_1 + \phi(\frac{i}{p}hA)d_{pl+i}, \qquad i = 1, \ldots, p, \quad l \geq 1. \tag{3.105}$$

Compared to the general form of the method (3.67) - (3.69) the new formulation reduces the number of the function evaluations and the applications of the Arnoldi algorithm by a factor p. Only once at every p stages do we need to perform the Krylov subspace projection to compute $\phi(\gamma hA)v$, while $\phi(i\gamma hA)v$, $i = 2, \ldots, p$, can be computed using recurrence (3.78). Furthermore, since, as was mentioned before, vectors d_i decrease in norm, we can expect the number of Krylov vectors needed to approximate $\phi(\gamma hA)d_i$ will decrease for each subsequent stage.

One more simplification to the reduced methods is made by expressing the scheme in

terms of shifted vectors

$$\tilde{k}_{pl+i} = k_{pl+i} - k_1, \qquad l \geq 1, \quad i = 1, \ldots, p, \tag{3.106}$$

with the coefficients α_{i1} and b_1 modified to reflect this change as

$$\tilde{\alpha}_{i1} = \alpha_{i1} + \sum_{m \geq p} \alpha_{im}, \tag{3.107}$$

$$\tilde{b}_1 = b_1 + \sum_{m \geq p} b_m, \tag{3.108}$$

From now on we will omit the tilde and just define by α_{i1}, b_1 and k_{pl+i} the shifted coefficients and vectors in (3.107), (3.108) and (3.106).

These simplifications significantly reduce the complexity of the method to make it efficient. Since γ is chosen to be the inverse of an integer $1/p$, according to the Theorem 3.3.5 and discussion thereafter, it can be made A-stable with imposing only a few conditions on coefficients b_j's.

3.3.4 Fourth-order exponential propagation Runge-Kutta methods

Two specific methods of order four were derived in [40]. The first one is constructed by setting $\gamma = 1/2$. The method is three-stage and uses two function evaluations per step. The parameters in the method are given by the following matrices

$$\beta = \begin{bmatrix} 0 & 0 & 0 \\ 1/4 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.109}$$

and

$$\alpha = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3/8 & 3/8 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} -16/27 \\ 1 \\ 16/27 \end{bmatrix}. \tag{3.110}$$

The resulting reduced method reformulated using the d_i vectors will be

$$\begin{aligned}
k_1 &= \phi(\frac{1}{2}hA)F(\mathbf{U}^n), \\
k_2 &= \phi(hA)F(\mathbf{U}^n), \\
w_3 &= \frac{3}{8}(k_1 + k_2), \\
u_3 &= \mathbf{U}^n + hw_3, \\
d_3 &= F(u_3) - F(\mathbf{U}^n) - hAw_3, \\
k_3 &= \phi(\frac{1}{2}hA)d_3, \\
\mathbf{U}^{n+1} &= \mathbf{U}^n + h(k_2 + \frac{16}{27}k_3).
\end{aligned} \tag{3.111}$$

By Theorem 3.3.5 this method is exact for linear differential equations and therefore is A-stable. It requires only two applications of the Arnoldi algorithm to calculate k_1 and k_3, while k_2 can be calculated using the recurrence (3.78). It is impossible, however, to derive a so-called embedded method of order three, i.e. a method which will have the same matrices β and α but different coefficients b_j. This fact will be important when we discuss an automatic error control mechanism in the next section.

Another, more complex seven stage method of order four can be constructed with $\gamma = 1/3$ and coefficient matrix

$$\beta = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/6 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/9 & 2/9 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1/6 & 0 & 0 & 1/6 & 0 & 0 & 0 \\
-1/3 & 0 & 0 & 1/9 & 2/9 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}. \tag{3.112}$$

The reduced version of this scheme is given by

$$
\begin{aligned}
k_1 &= \phi(\frac{1}{3}hA)F(\mathbf{U}^n),\\
k_2 &= \phi(\frac{2}{3}hA)F(\mathbf{U}^n),\\
k_3 &= \phi(hA)F(\mathbf{U}^n),\\
w_4 &= -\frac{7}{300}k_1 + \frac{97}{150}k_2 - \frac{37}{300}k_3,\\
u_4 &= \mathbf{U}^n + hw_4,\\
d_4 &= F(u_4) - F(\mathbf{U}^n) - hAw_4,\\
k_4 &= \phi(\frac{1}{3}hA)d_4,\\
k_5 &= \phi(\frac{2}{3}hA)d_4,\\
k_6 &= \phi(hA)d_4,\\
w_7 &= \frac{59}{300}k_1 - \frac{7}{15}k_2 + \frac{269}{300}k_3 + \frac{2}{3}(k_4 + k_5 + k + 6),\\
u_7 &= \mathbf{U}^n + hw_7,\\
d_7 &= F(u_7) - F(\mathbf{U}^n) - hAw_7,\\
k_7 &= \phi(\frac{1}{3}hA)d_7,\\
\mathbf{U}^{n+1} &= \mathbf{U}^n + h(k_3 + k_4 - \frac{4}{3}k_5 + k_6 + \frac{1}{6}k_7).
\end{aligned}
\tag{3.113}
$$

This method is also A-stable since it is exact for linear differential equations. Even though method (3.111) is more economical than the scheme (3.113) we will use the latter in our future calculations. The reason for our choice is the fact that it is possible to derive an embedded method of order three for the scheme (3.113). In the next section we describe this point in detail and another advantage of the Runge-Kutta exponential propagation schemes - the robust automatic error control mechanism which can be constructed for these methods.

3.4 Automatic error control - Gustafson's approach

As discussed above, error estimation is a crucial point in constructing a good exponential propagation method. We have already described how the generalized residuals can be used

to assess the error of the Arnoldi algorithm in approximation of $\phi(\gamma hA)v$. Now we also need a mechanism to estimate the error of the new approximation to the solution at the next time step $\mathbf{U}^{n+1}$ so that we can adjust the time step size based on this calculation. An approach proposed first by Gustafson for the regular Runge-Kutta methods can be naturally extended to the Runge-Kutta exponential propagation methods.

The key observation, which helps construction of an efficient and robust automatic error control mechanism for either ordinary or exponential Runge-Kutta schemes, is the fact that for given matrices of coefficients α_{ij}, β_{ij} and γ_{ij} and parameter γ there can be several methods derived which differ only by the coefficients b_j's. Since the parameters b_j are only used to compute the final approximation

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \sum_{j=1}^{s} b_j k_j \tag{3.114}$$

once all the vectors k_j are computed, it is cheap to calculate two approximations to $\mathbf{U}^{n+1}$ at the cost of $s-1$ extra vector additions. This leads to the following error control mechanism.

Supposed we derived an exponential Runge-Kutta method of order q with certain coefficients γ, α_{ij}, β_{ij}, γ_{ij} and b_j, with which we then compute an approximate solution $\mathbf{U}^{n+1}$ to the system of differential equations (3.1). Assume also that we found an embedded exponential Runge-Kutta method of order $\widehat{q}$ with coefficients γ, α_{ij}, β_{ij}, γ_{ij} and $\widehat{b}_j$, which gives us another approximation to the solution $\widehat{\mathbf{U}}^{n+1}$. We want the error between approximations $\mathbf{U}^{n+1}$ and $\widehat{\mathbf{U}}^{n+1}$ to be within a predefined tolerance componentwise, i.e.

$$|\mathbf{U}_i^{n+1} - \widehat{\mathbf{U}}_i^{n+1}| \leq sc_i, \qquad sc_i = Atol_i + \max(|\mathbf{U}_i^n|, |\mathbf{U}_i^{n+1}|) \cdot Rtol_i, \tag{3.115}$$

where $Atol_i$ and $Rtol_i$ are the desired absolute and relative tolerances per component prescribed by the user. The measure of the total error ε_{exrk} is defined as

$$\varepsilon_{exrk} = \|(\mathbf{U}_i^{n+1} - \widehat{\mathbf{U}}_i^{n+1})/sc\|, \tag{3.116}$$

where the norm $\|\cdot\|$ is taken to be either the normalized 2-norm

$$\|\mathbf{U}\| = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \mathbf{U}_i^2}, \tag{3.117}$$

or the maximum norm

$$\|\mathbf{U}\| = \max_{1 \le i \le N} (\mathbf{U}_i) \tag{3.118}$$

depending on which of these provide a better error estimate for a particular problem.

Since the numerical methods we used are of orders q and $\widehat{q}$ we expect the error to behave as

$$\varepsilon_{exrk} \approx C \cdot h^r, \qquad r = \min(\widehat{q}, q). \tag{3.119}$$

The optimal step size h_{opt} would ensure that

$$C \cdot h_{opt}^r \approx 1. \tag{3.120}$$

From the last two equations we can compute the optimal step size as

$$h_{opt} = h \cdot \left(\frac{1}{\varepsilon_{exrk}} \right)^{1/(r+1)}. \tag{3.121}$$

In practice in order to avoid decreasing or increasing the time step too much using the last equation and in this way ensure that the approximation at the next time step will be accurate we can introduce safety factors in the formula (3.121). That is, we define constant growth and reduction factors fac, $facmin$ and $facmax$ and modify the last formula to compute the next time step as

$$h_{new} = h \cdot \min(facmax, \max(facmin, fac \cdot \left(\frac{1}{\varepsilon_{exrk}} \right)^{1/(r+1)})). \tag{3.122}$$

The error control mechanism therefore, proceeds as follows. First, we set some starting value for the time step h and compute the approximations to the solution $\mathbf{U}^{n+1}$ and $\widehat{\mathbf{U}}^{n+1}$. Then we calculate the error ε_{exprk}. If the error is larger than 1 we reduce the size of the time step and compute the solution approximations again with the new time step. Otherwise the solution is advanced with $\mathbf{U}^{n+1}$ and the new time step size is calculated according to the formula (3.122).

Coupling the procedure described above with the error control mechanism for the Arnol-

di algorithm in the general scheme we estimate the new time step size h_{kry} based on the residual in the Krylov subspace projection and the time step h_{exprk} from (3.122) and pick the minimum of the two to be the new time step size

$$h_{new} = \min(h_{kry}, h_{exprk}). \tag{3.123}$$

With this new value for the time step size we proceed with integrating the system of equations in time.

Let us discuss now the application of this error control algorithm to the fourth-order methods presented in section 3.3.4. As we mentioned before, it is impossible to derive an embedded method of order 3 for the scheme (3.111). An alternative is to use the second order scheme (3.11) as an embedded method. The error estimate in this case might be too pessimistic and consequently the time step taken will be too small and the overall calculation much slower. For the seven stage method (3.113), however, there is an embedded method of order three which differs from (3.113) only by the last formula

$$\mathbf{U}^{n+1} = \mathbf{U}^n + h\left(k_3 - \frac{1}{2}k_4 + \frac{2}{3}k_5 + \frac{1}{2}k_6 + \frac{1}{2}k_7\right). \tag{3.124}$$

So since it is important to us to make the time step as large as possible given the accuracy requirements we choose to use method (3.113) with an embedded method (3.124) for most of our calculations in this work.

3.5 Practical implementation of the exponential Runge-Kutta methods and numerical example

3.5.1 Memory requirements and parallelization of the method

Since we hope to use the exponential Runge-Kutta methods to solve large systems of stiff nonlinear partial differential equations we should consider the issues of memory storage and efficiency for the practical implementation of the algorithm. First of all note that the method is very memory intensive. Suppose, for example, that it takes N_u megabytes to store one vector of unknowns $\mathbf{U}^n$ and that the maximum number of the Krylov vectors needed for any of the stages in the scheme is m. Then assuming that we reuse the space allocated for

the Krylov vectors at each stage, including one vector to store intermediate values, we will need at least $(m+2)N_u$ megabytes of memory on our computer to implement the method. Even if m is a small number of the order of 10-15 vectors, if the number of nodes on our grid times the number of unknown variables, which constitutes the length of $\mathbf{U}^n$, is large, the increase in memory requirement on the order of 10 is very significant compared to the simple explicit scheme. For example, if $\mathbf{U}^n$ is of length $N = n_{vars} \times n_{grid} = 8 \times 100^3 = 8,000,000$ elements, then $N_u = N \times \text{sizeof}(double) \approx 63$ megabytes and if $m = 15$, we require a computer to have at least 1 Gigabyte of memory. In other words the memory requirement increases by at least an order of magnitude compared to an explicit scheme. It will also be larger than the memory requirement for implicit schemes, since implementation of methods like conjugate gradient or GMRES do not require the user to store all the Krylov vectors until the end of the iteration and simply overwrite each computed vector with a new one as the Arnoldi iteration progresses.

The size and the stiffness of the initial system also impose an efficiency requirement on the method. We need to be able to run the code fast if we want to integrate the system over long periods of time on a large grid. These requirements lead us to the conclusion that the method should be parallelizable so that we could create an efficient implementation which would run on a distributed memory parallel computer with large total memory size. Fortunately, the iterative nature of the exponential Runge-Kutta methods make achieving this goal possible. Note that the major part of the computations in these schemes involve functional evaluations $F(\mathbf{U})$ and matrix-vector multiplications Av. Recall that function $F(\mathbf{U})$ for the systems of PDEs is just a discretized version of the spatial operators and A is a Jacobian matrix of $F(\mathbf{U})$. Therefore, effective parallelization can be achieved in the same way as for the explicit and iterative implicit methods, i.e. the development of an effective and scalable parallel implementation of the action of the spatial differential operators on a grid will ensure that the general exponential Runge-Kutta code will be efficient and scalable as well. For example, suppose we discretize our system on a grid and use finite differences to approximate the spatial operators. Then we can use the standard approach to parallelize the algorithm by splitting the grid into patches and distributing the task of computing $F(\mathbf{U})$ and Au on a single patch to a different processor. The only communication that will have to be performed during this computations is passing the ghost cell values between the processors, which is an ordinary parallelization issue considered in any parallel software or

framework.

The parallel properties of the exponential Runge-Kutta method make the algorithm very suitable for a vector machine as well. The largest numerical computations in this work are performed on a vector Cray J90 supercomputer. Our numerical experiments confirmed that the amount of parallelization in the algorithm depends mostly on the efficiency of the parallel implementation of $F(\mathbf{U})$ and Au with the former being more important since the most time in the code is spent in this routine. The second part of the code which affected greatly the efficiency was the implementation of the calculation of the vector norms. An effective vectorized implementation of these routines greatly reduced the overall computing time and the splitting of the loops involved in calculating the norms significantly increased the scalability of the code. The time spent on diagonalization of the Krylov projection matrix H_m proved to be insignificant compared to the overall computation time even if it is performed in serial.

3.5.2 Implementation

Our current code is based on the package *exp4* mentioned in [40]. *exp4* constitutes the time integrator part of the code. We have optimized this package for vector supercomputers and for large grid sizes ran it on the Cray J90 vector machine. The calculations for problems of smaller sizes were performed on a Pentium II 233 MHz Linux workstation. The specifics for each problem are implemented in a separate package interfaced with *exp4* through routines performing the functional evaluation $F(\mathbf{U})$ and multiplication of a vector by a Jacobian matrix. Note that the Jacobian matrix does not have to be stored explicitly, we just have to write a program which takes a vector as an input and outputs the resulting matrix-vector product. The code is written in the C programming language with all the low level linear algebra operations accomplished by calls to Fortran BLAS routines. When run on the Cray J90 native vector routines of the BLAS optimized for Cray are used. In the future we plan to implement the method on a distributed memory massively parallel machine in the C programming language using MPI.

If we are solving a system of partial differential equations we should also carefully consider the question of spatial discretization appropriate for this problem. Unless otherwise indicated we use central finite differences to approximate the spatial differential operators in this work. The advantage of this approach is the high order of accuracy in space and fast

computation of the spatial derivatives. Most of the problems we solve are of the convection-diffusion equation type and for these systems such spatial discretization proved to work well. For systems with hyperbolic behavior, however, upwinding should be used when discretizing the convective terms. We will return to this issue in our discussion of the future work in the last chapter.

3.5.3 Numerical example - Burgers equation

Since most of the systems of interest in our work are of the convection-diffusion type, we will demonstrate the advantages of the exponential Runge-Kutta method on a simple one-dimensional Burgers equation:

$$u_t + uu_x = bu_{xx}, \tag{3.125}$$

on an interval $x_{min} \leq x \leq x_{max}$. This equation has an exact solution

$$u(x,t) = a - c\tanh\left[\frac{c}{2b}(x - at)\right], \tag{3.126}$$

which represents a front moving with speed a. We choose $a = 1/2$, $c = 1/2$ and integrate (3.125) on an interval $t \in [t_0 = 0, t_{final} = 1.15]$ for different values of the diffusion coefficient b. Figure 3.1 shows the exact solution (3.126) for the specified values of the parameters and $b = 0.01$ along with its approximation by the exponential Runge-Kutta method at times $t = 0.01$ and $t_{final} = 1.15$ on an interval $-1 \leq x \leq 1$.

To demonstrate the advantages of the exponential Runge-Kutta method we will compare method (3.113) with the fourth-order explicit Runge-Kutta method as formulated by

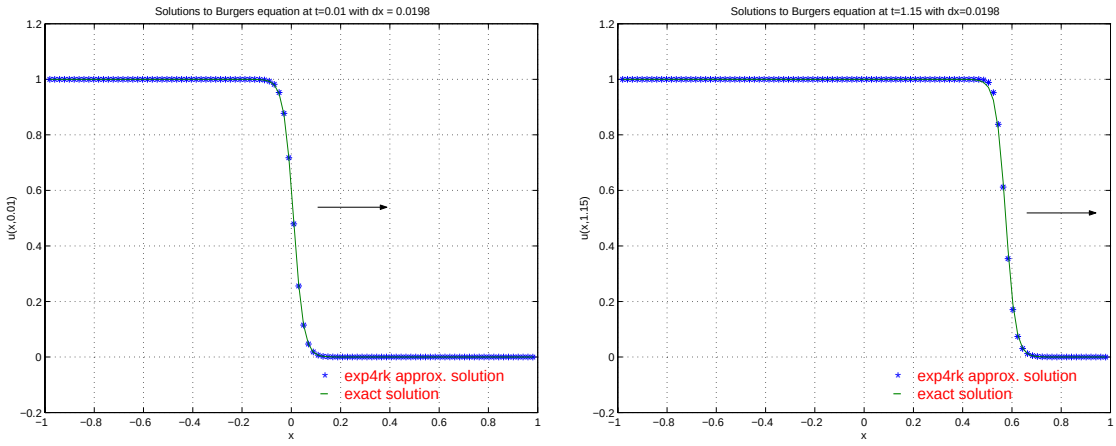


Figure 3.1: Exact and numerical solution (as calculated by the fourth order exponential Runge-Kutta method (3.113)) to the Burgers equation on an interval $-1 \leq x \leq 1$ with $a = 1/2$, $c = 1/2$ and $b = 0.01$ at times $t = 0.01$ (left) and $t_{final} = 1.15$(right).

Zonneveld:

$$k_1 = F(\mathbf{U}^n), \tag{3.127}$$

$$k_2 = F\left(\mathbf{U}^n + h\frac{1}{2}k_1\right), \tag{3.128}$$

$$k_3 = F\left(\mathbf{U}^n + h\frac{1}{2}k_2\right), \tag{3.129}$$

$$k_4 = F\left(\mathbf{U}^n + hk_3\right), \tag{3.130}$$

$$k_5 = F\left(\mathbf{U}^n + h(\frac{5}{32}k_1 + \frac{7}{32}k_2 + \frac{13}{32}k_3 - \frac{1}{32}k_4)\right), \tag{3.131}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right).. \tag{3.132}$$

We use exactly the same error control mechanism (due to Gustafson) for both methods. The formula (3.124) is used as an embedded method for the exponential Runge-Kutta scheme. For the explicit Runge-Kutta scheme the embedded numerical method is the following third order explicit Runge-Kutta method that results in using formulas (3.127) - (3.131) together with

$$\widehat{\mathbf{U}}^{n+1} = \widehat{\mathbf{U}}^n + h\left(-\frac{1}{2}k_1 + \frac{7}{3}k_2 + \frac{7}{3}k_3 + \frac{13}{6}k_4 - \frac{16}{3}k_5\right) \tag{3.133}$$

to obtain a second approximation to the solution. The automatic error control parameters

	Explicit Runge-Kutta-Kutta			Exponential Runge-Kutta					
Number of grid points,	*Total integration time (sec),*	*Average time step,*	*Relative error norm*	*Total integration time (sec),*	*Average time step,*	*Relative error norm*	*Max # of Krylov vectors*		
N	T_{total}	h_{ave}	$\|err\|_2$	T_{total}	h_{ave}	$\|err\|_2$	m_1	m_2	m_3
4000	5.36	1.7e-3	4.6e-4	5.9	1.4e-2	4.7e-4	8	3	4
8000	48.4	4.3e-4	1.6e-4	38.1	1.2e-2	2.1e-4	15	8	8
16000	502.8	1.1e-4	1.1e-5	204.1	6.2e-3	1.1e-4	20	11	11
32000	4652.0	2.7e-5	1.1e-5	1100.2	2.9e-3	1.9e-4	20	6	11

Table 3.2: Performance comparison of the exponential and explicit Runge-Kutta methods for the diffusion coefficient $b = 0.1$

used in the formula (3.122) are set at the same values for both methods, specifically relative and absolute tolerances are $Atol_i = Rtol_i = 10^{-4}$ and the factors are given values $fac = 0.9$, $facmin = 0.2$, $facmax = 5.0$. The maximum allowable number of Krylov vectors for each stage of the exponential propagation method is set to be 36.

We are interested in comparing the efficiency of the methods for integrating the Burgers equation over a time interval $[t_0 = 0, t_{final} = 1.15]$ as we increase the spatial resolution. It is well known [77] that for the explicit Runge-Kutta method if we increase the number of points on the spatial grid by a factor C we should expect the time step to be reduced by C^2 due to the stability constraint. We do not expect such a drastic reduction of the time step for the exponential Runge-Kutta method and hope that as the number of grid points increases the efficiency of the exponential Runge-Kutta method compared to the explicit scheme will also increase. We expect the exponential Runge-Kutta methods to make a difference in performance for large problems, so we start with $N = 4000$ points on the spatial interval $[-10, 10]$, then increase N and compute the total integration time, the average time step and final relative error for both numerical methods. Tables 3.3 and 3.2 summarize our results for two different values of the diffusion coefficient $b = 0.1$ and $b = 0.01$.

As we can see from the data in both tables as the number of grid points N is doubled the automatic error control mechanism reduced the time step for the explicit Runge-Kutta (RK) method by a factor of 4 as expected from the stability condition for the second-order differential equation (3.125). This is also evident from the fact that even though our tolerances are set at 10^{-4}, which is exactly the order of the error maintained by the exponential RK method, the explicit method computes the solution with smaller error. For the exponential propagation method the time step is reduced much less and in all

	ExplicitRunge-Kutta			Exponential Runge-Kutta					
Number of grid points,	*Total integration time (sec),*	*Average time step,*	*Relative error norm*	*Total integration time (sec),*	*Average time step,*	*Relative error norm*	*Max # of Krylov vectors*		
N	T_{total}	h_{ave}	$\|err\|_2$	T_{total}	h_{ave}	$\|err\|_2$	m_1	m_2	m_3
4000	40.9	1.7e-4	4.9e-6	42.5	8.2e-3	6.2e-5	20	15	11
8000	474.1	4.4e-5	5.0e-6	190.5	3.3e-3	1.5e-4	20	11	11
16000	4621.1	1.1e-5	4.1e-6	741.0	1.6e-3	4.4e-4	20	6	11
32000	45057.1	2.7e-6	4.0e-6	6696.0	1.1e-3	1.7e-4	36	8	15

Table 3.3: Performance comparison of the exponential and explicit Runge-Kutta methods for the diffusion coefficient $b = 0.1$

of the calculations in both tables was restricted by accuracy requirements of the Krylov approximation to the exponential operator, i.e.

$$h_{new} = \min(h_{kry}, h_{exprk}) = h_{kry}. \tag{3.134}$$

As the size of the problem increases the extra time required to complete each time iteration for the exponential RK method compared to the explicit scheme is offset by the larger allowed time step. Consequently the exponential RK method integrates the equations faster than the explicit scheme. The ratio of the overall integration times by the explicit method and the exponential RK scheme grows as the problem gets stiffer. For equation (3.125) the larger the diffusion coefficient b, the bigger is the magnitude of the ratio between the largest and the smallest eigenvalues of the Jacobian matrix dominated by the discretized Laplacian part of the equation. Therefore, $b = 0.1$ constitutes a stiffer problem than a calculation with $b = 0.01$. As we can see from the tables the difference between efficiency of the exponential RK versus explicit RK becomes more dramatic as N increases. Note that given this improvement, the number of Krylov vectors computed for each stage stays relatively small, i.e. the largest H_{m_i} matrix for which $e^{H_{m_i}} v$ has to be computed directly is only of order 36×36. Given a computer with a larger memory and faster processor we could increase the number of Krylov vectors even further which would result in greater savings in the size of the time step and overall computation time.

The computations presented in the latter chapters involve yet larger grids and greater number of variables. Those problems will demonstrate even larger savings offered by the exponential propagation methods, e.g. a time step 100 times larger than the explicit limit and an order of magnitude reduced overall time of integration. The data for the example

calculation in this section, however, gives an idea of what savings we can expect for the large problems and shows how the performance of the method changes as we increase the size of the system.

Chapter 4 Magnetohydrodynamics calculations for solar coronal applications using exponential propagation methods

A set of applications which can benefit from the properties of the exponential propagation methods lies in the realm of solar magnetohydrodynamics (MHD). In the beginning of this chapter we briefly discuss the solar phenomena of interest. Then we introduce the theory of MHD and explain how it is used to study certain processes in the solar corona. We concentrate on two models widely used to describe the evolution of certain configurations of plasma in the solar corona. We present and discuss the results of numerical modeling of these problems using the exponential propagation methods of the previous chapter.

4.1 Challenges of the solar corona

Three regions of plasma with different physical properties comprise the visible part of the solar atmosphere (Fig. 4.1) [67]. A thin ($\sim$ 550 km thick) bottom layer is responsible for most of the emitted solar radiation and is called the photosphere. Just outside the photosphere lies the almost completely transparent 2500 km wide chromospheric region. Finally, the outermost layer of the atmosphere extends from 1 solar radius ($1R_{\odot} = 696,000$ km) to millions of kilometers into space. This region can be seen directly during eclipse or by blocking the photosphere with a special telescope, the coronagraph (Fig. 4.2). Such images of the region gave it the name corona, which is derived from the Latin word "crown". The plasma properties change drastically with height going the photosphere to the outer corona. The temperature increases from about 6000 Kelvin in the photosphere to an average temperature of 1 million Kelvin in the corona. The density on the other hand falls rapidly from 10^{23} m^{-3} in the photosphere to 10^{15} m^{-3} in the outer chromosphere and 10^{12} m^{-3} at a height of 1R$_{\odot}$. A feature which plays a very important role in the dynamics of all the regions in the solar atmosphere is presence of a strong, topologically complex magnetic field. Due to the drastic change in physical parameters the plasma dynamics differs in the three regions. In the high density photospheric plasma the interplay between hydrodynamic pressure and magnetic forces determines the dynamics. In contrast, dynamics in the highly

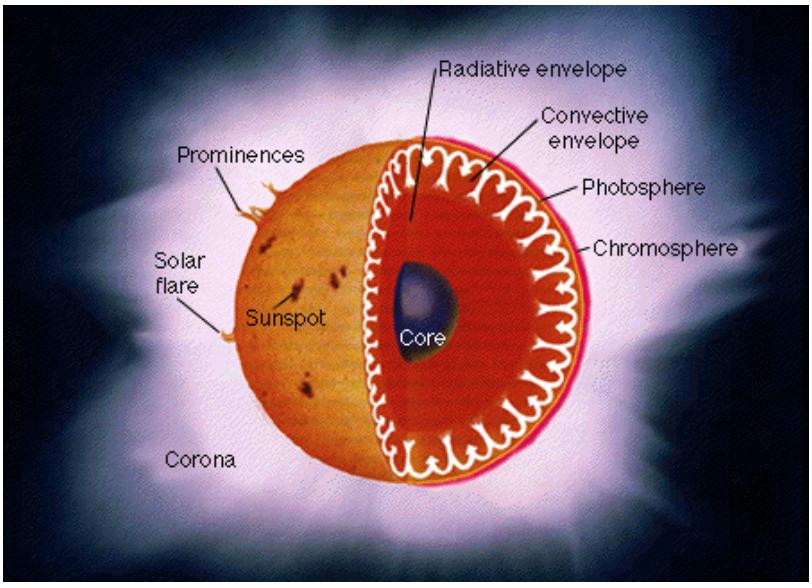


Figure 4.1: The structure of the Sun (image credit: Kaler, James B. Stars . New York: Scientific American Library, 1992).

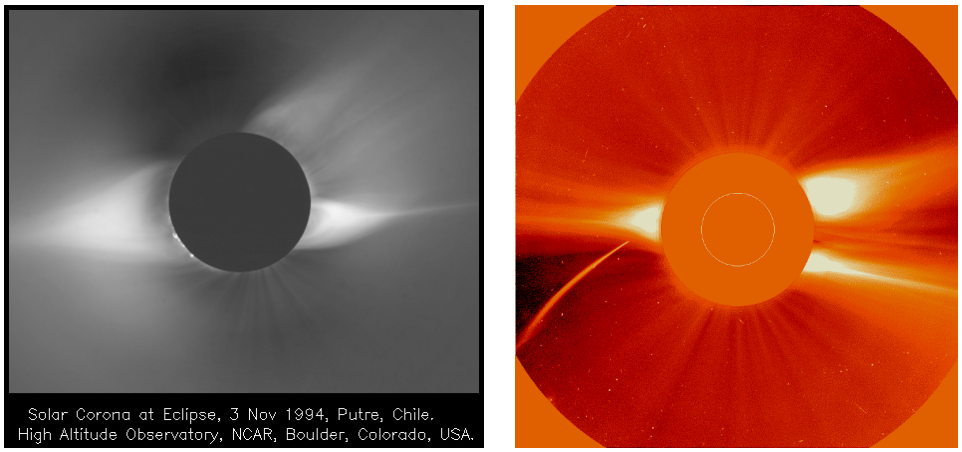


Figure 4.2: The solar corona as seen on an eclipse image taken by a team from the High Altitude Observatory (left) and an image from LASCO coronagraph(right).

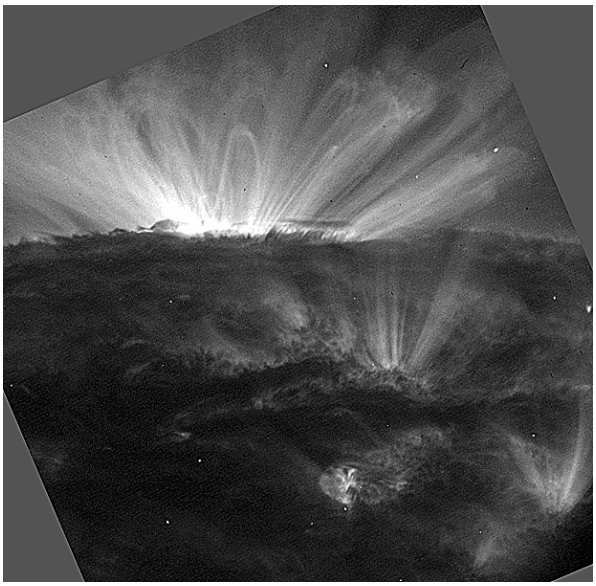


Figure 4.3: An image of the coronal loops obtained by the TRACE telescope

rarefied corona is dominated by magnetic forces only. If P is the hydrodynamic pressure and $B^2/(2\mu_0)$ is the plasma magnetic pressure (μ_0 is the magnetic permeability) then the parameter comparing the importance of these quantities is defined as

$$\beta = \frac{2\mu_0 P}{B^2}. \tag{4.1}$$

Typically, for the active regions of the corona $\beta \approx 3 \times 10^{-3}$. The behavior of such low β plasma can be determined by examining the plasma response to its internal and external magnetic fields.

Under the influence of the strong photospheric magnetic field coronal plasma becomes organized into elaborately interleaved bundles of large-scale twisted magnetic loops and arcades (Fig. 4.3). The stages in the evolution of these configurations range from long periods of stable existence in equilibrium to violent explosions which eject billions of tons of plasma material into interplanetary space. Coronal activity is usually separated into three distinct categories of eruptive events: solar flares, prominence eruptions and coronal mass ejections [68]. A *solar flare* is a violent eruptions of a cool filament embedded in an arcade of hot loops. Flares cause massive releases of stored magnetic energy and raise the temperature of surrounding plasma regions by tens of millions of degrees. *Prominences* are generally less violent than solar flares. They are giant elongated structures that consist of

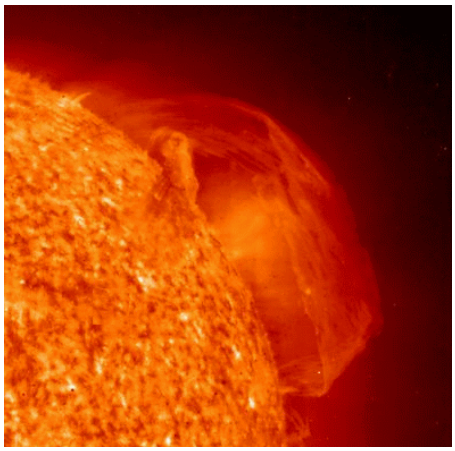


Figure 4.4: Solar prominence

material hundreds of times cooler and denser than the surrounding coronal plasma (Fig. 4.4). Prominences can be stable for weeks or months at a time and then either slowly dissolve or suddenly erupt. Most large flares and prominence eruptions are associated with consequent large-scale ejections of mass and magnetic flux from the lower corona into the solar wind. Such events are called *coronal mass ejections* (CMEs)(Fig. 4.5). An average CME carries roughly 10^{13} kg of plasma and 10^{15} Wb of magnetic flux into the interplanetary space. The frequency of CMEs ranges from several events a day to a few times a week depending on where the Sun is in its 11 year activity cycle. The scale and frequent occurrence of these events make them some of the most important contributors to space weather. It takes 2 to 5 days for the CME to reach Earth and, when this happens, the environment near and on Earth is affected in many different ways. Solar storms affect Earth's ionosphere causing disruption of short-wave radio communications, navigational and radar systems. They can also cause electric power blackouts and corrosion pipelines. The large surges of charged particles from the Sun can damage spacecrafts and increase radiation to levels dangerous for astronauts in orbit about the Earth. These effects demonstrate the importance of understanding the mechanism of the solar eruptive events and developing an ability to predict CMEs and prevent possible damage to equipment or hazards to astronauts.

As more and better observational data has been collected it has become evident that all of these events are likely to be different manifestations of the same physical process, namely loss of equilibrium of large-scale coronal loops and arcades due to the evolution

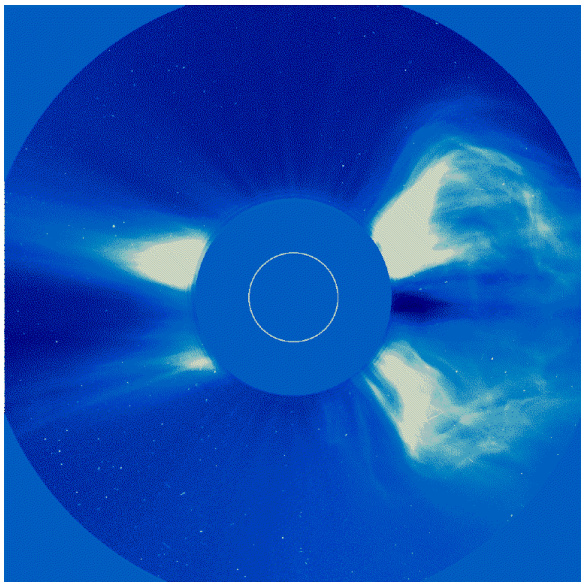


Figure 4.5: Large coronal mass ejection (CME) from 6 November 1997 as recorded by the LASCO C2 coronagraph at 12:36 UT.

of their magnetic and velocity fields. Numerous different numerical and analytical models have been proposed to explain this process. All of these theories presume eruption is a loss of equilibrium of the large-scale coronal configurations. However, the timing (gradual versus sudden) and the cause of the instability (mass loading versus magnetic energy release) differ greatly among the models. For the review of current theories see, for example, [68, 46]. None of these theories, however, can fully explain the phenomena or predict all of its properties. Since the observations also have many limitations and do not reveal the true three-dimensional topology and evolution of the coronal structures it is very hard to definitively refute or prove any of the models. The possibility that no single mechanism exists and different models can account for the dynamics of different events also cannot be ruled out due the variability of the activity.

Although the physical processes of the corona are rooted in the photosphere, it is possible to study the regions separately and model the connection between the coronal activity and the flow motions of the much more massive photosphere as boundary conditions. This is the approach we will use. In this work we will focus on numerical modeling of the evolution of the magnetic solar arcades in response to the motion of their footpoints being dragged by the photospheric flow. In the following sections we will describe our mathematical model and present some test computations. Finally in Chapter 5 we will present a three-dimensional

model of the dynamics of the solar arcades and then discuss the connection between solar observations and our simulations.

4.2 The theory of magnetohydrodynamics for solar applications

In order to model the dynamics of the large-scale plasma configurations arising in the solar corona we invoke the theory of magnetohydrodynamics (MHD). The underlying justification of MHD theory can be found in numerous references, see for example [67, 61, 74]. Therefore we provide here only a brief discussion of the MHD equations and describe the assumptions used to derive the models for the coronal plasma.

MHD describes plasma as an electrically conducting fluid, i.e. as a continuum. The dynamics of this fluid results from an interplay between the motion of plasma and the magnetic fields and is completely defined by specifying the magnetic field $\mathbf{B}$, the fluid velocity $\mathbf{V}$, density ρ, pressure P and temperature T. Typically MHD plasma is assumed to be a perfect gas, so that the equation of state

$$P = k_b \rho T, \qquad (k_b \text{ is the Boltzmann constant}) \tag{4.2}$$

can be used to eliminate temperature T from the evolution equations. All plasma properties are defined as averages over elements large compared with the microscopic structure of plasma but small with respect to the macroscopic phenomena of interest. This approximation is valid in the collision-dominated situation, i.e. the situation where the mean free path is small compared to the region of interest. In particular, it is a valid assumption for studying the large-scale coronal plasma configurations. Sometimes MHD also gives good results even in relatively collision-free situations due to the tendency of plasma particles to gyrate around the magnetic field lines [74]. One has to be careful, however, in applying MHD to such problems and understand the limitations of this description. Typically, a magnetohydrodynamic description is used to investigate the stability properties of large-scale plasma configurations, which is exactly our interest in studying solar corona.

To derive the MHD equations we need to combine the standard conservation of mass and momentum equations for a fluid, Maxwell equations that take into account magnetic

properties of the plasma, and develop an energy conservation equation to describe the evolution of the pressure. The form of the latter equation for solar coronal applications is still under much debate, since it is not clear how radiative transfer and thermal heating sources should be modeled. In most numerical studies of the corona though, the plasma is assumed to be thermally isolated from its surroundings and its motion adiabatic, i.e. if γ is the polytropic index, the energy equation is taken to be

$$\frac{DP}{Dt} - \frac{\gamma P}{\rho}\frac{D\rho}{Dt} = 0, \qquad \left(\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{V}\cdot\nabla\right) \tag{4.3}$$

which means that the quantity P/ρ^{γ} remains constant in a moving fluid element. In our models the issue of the correct form of the energy conservation equation evolution will not be important since we assume the plasma parameter β defined in (4.1) to be zero. We will return to this point and discuss it in more detail later in the section.

The forces acting on the plasma are the Lorentz force, pressure gradients, gravitational $\mathbf{F}_g$ and viscous forces $\mathbf{F}_\nu = \rho\nu\nabla^2\mathbf{V}$, which gives us the conservation of momentum equation

$$\rho\frac{D\mathbf{V}}{Dt} = \mathbf{J}\times\mathbf{B} - \nabla P + \mathbf{F}_g + \rho\nu\nabla^2\mathbf{V}, \tag{4.4}$$

where $\mathbf{J}$ is the current density and ν is the coefficient of kinematic viscosity (assumed uniform). Using Ampere's law

$$\nabla\times\mathbf{B} = \mu_0\mathbf{J} + \frac{1}{c^2}\frac{\partial\mathbf{E}}{\partial t}, \tag{4.5}$$

where $\mathbf{E}$ is the electric field, μ_0 is the magnetic permeability ($= 4\pi\times10^{-7}\mathrm{Hm}^{-1}$ in vacuum) and c is the speed of light. One of the major assumptions of magnetohydrodynamics is that displacement current term can be neglected because all characteristic velocities are non-relativistic, i.e. $V \ll c$, so

$$\mathbf{J} = \frac{1}{\mu_0}\nabla\times\mathbf{B}. \tag{4.6}$$

so that (4.4) becomes

$$\rho\frac{D\mathbf{V}}{Dt} = \frac{1}{\mu_0}(\nabla\times\mathbf{B})\times\mathbf{B} - \nabla P + \mathbf{F}_g + \rho\nu\nabla^2\mathbf{V}. \tag{4.7}$$

The induction equation is derived from Faraday's law

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}, \tag{4.8}$$

and the assumption that the electric and magnetic fields obey Ohm's law

$$\mathbf{E} = -\mathbf{V} \times \mathbf{B} + \mathbf{J}/\sigma \qquad (\sigma \text{ is the electric conductivity}). \tag{4.9}$$

Combining these equations with (4.6) we obtain

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{V} \times \mathbf{B}) - \nabla \times (\eta \nabla \times \mathbf{B}), \tag{4.10}$$

where $\eta = 1/(\mu_0 \sigma)$ is the magnetic diffusivity. Adding to this the conservation of mass equation, assuming that η is uniform and using the vector identity

$$\nabla^2 \mathbf{B} = \nabla(\nabla \cdot \mathbf{B}) - \nabla \times (\nabla \times \mathbf{B}), \tag{4.11}$$

along with the fact that the magnetic field is divergence free $\nabla \cdot \mathbf{B} = 0$ we obtain the following closed system of equations

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{V} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B}, \tag{4.12}$$

$$\frac{\partial \mathbf{V}}{\partial t} = -\mathbf{V} \cdot \nabla \mathbf{V} + \frac{1}{\rho}\left(\frac{1}{\mu_0}(\nabla \times \mathbf{B}) \times \mathbf{B} - \nabla P + \mathbf{F}_g\right) + \nu \nabla^2 \mathbf{V}, \tag{4.13}$$

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{V}), \tag{4.14}$$

$$\frac{\partial P}{\partial t} = -\mathbf{V} \cdot \nabla P - \gamma P \nabla \cdot \mathbf{V}. \tag{4.15}$$

If $\eta = 0$ the equations above describe the processes of the *ideal* magnetohydrodynamics; otherwise this is the system of equations for *resistive* MHD. The physical parameters of a particular plasma phenomenon determine which theory better describes the dynamics. Either of the systems, ideal or resistive, include the descriptions of a variety of complex physical processes, which makes solving the equations a difficult task. To model the coronal plasma we choose the resistive MHD description and use some properties of the corona to simplify the equations. First, we recall that the dynamics of the very rarefied coronal

plasma is dominated by the strong magnetic forces. This implies that we can discard pressure gradients and the gravitational force in our model of the coronal arcades. Since we use zero β plasma approximation in our calculations, only the Lorentz force will remain in the induction equation and we can ignore the pressure equation altogether. Another approximation we make is that the density is uniform. This assumption is harder to justify on physical grounds. We argue that the overall magnetic field topology of the plasma configurations will be dominated by the dynamics between the magnetic and velocity fields and since our primary interest is the changes in the magnetic field topology, as a first approximation we can assume density to be constant and uniform. Additional support for this approximation comes from a number of numerical studies [55, 5, 8] that investigated zero β plasmas and found that giving the density various nonuniform profiles did not significantly alter the dynamics or topology of the magnetic field. In the future we plan to enhance our code by including equations (4.14) and (4.15) and investigate how this change affects the dynamics. Given all these assumptions, the system of equations we are interested in solving is

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{V} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B}, \tag{4.16}$$

$$\frac{\partial \mathbf{V}}{\partial t} = -\mathbf{V} \cdot \nabla \mathbf{V} + \frac{1}{\rho \mu} (\nabla \times \mathbf{B}) \times \mathbf{B} + \nu \nabla^2 \mathbf{V}, \tag{4.17}$$

$$\rho = \text{constant}. \tag{4.18}$$

Let us now concentrate on these equations and nondimensionalize all of the variables to simplify our calculations. Denote the nominal value of the velocity magnitude and the strength of the magnetic field as V_0 and B_0, and let the typical length and time scales be l_0 and t_0. We want to perform our computations in terms of the nondimensional variables $\widehat{\mathbf{x}} = \mathbf{x}/l_0$, $\hat{t} = t/t_0$, $\widehat{\mathbf{V}} = \mathbf{V}/V_0$, $\widehat{\mathbf{B}} = \mathbf{B}/B_0$. After expressing all variables in (4.16) and (4.17) in terms of these nondimensional quantities and performing some simple algebra we obtain the non-dimensional system

$$\frac{\partial \widehat{\mathbf{B}}}{\partial t} = \frac{V_0 t_0}{l_0} \widehat{\nabla} \times (\widehat{\mathbf{V}} \times \widehat{\mathbf{B}}) + \frac{t_0 \eta}{l_0^2} \widehat{\nabla}^2 \widehat{\mathbf{B}}, \qquad (\widehat{\nabla} = \frac{1}{l_0} \widehat{\nabla}) \tag{4.19}$$

$$\frac{\partial \widehat{\mathbf{V}}}{\partial t} = -\frac{V_0 t_0}{l_0} \widehat{\mathbf{V}} \cdot \widehat{\nabla} \widehat{\mathbf{V}} + \frac{B_0^2 t_0}{\rho \mu V_0 l_0} (\widehat{\nabla} \times \widehat{\mathbf{B}}) \times \widehat{\mathbf{B}} + \frac{t_0 \nu}{l_0^2} \widehat{\nabla}^2 \widehat{\mathbf{V}}. \tag{4.20}$$

A fundamental MHD electromagnetic waves phenomena called Alfven waves are described by solutions of the linearized equation (4.16). These waves can be categorized as *shear* Alfven waves, that propagate along the magnetic field $\mathbf{B}$, and *compressional* Alfven waves that travel at arbitrary directions with respect to $\mathbf{B}$ (for detailed description of Alfven waves see [67, 61]). The propagation speed in both cases is called the Alfven velocity

$$V_A = \frac{B_0}{(\mu\rho)^{1/2}}. \tag{4.21}$$

In the parameter regime of interest these waves are the fastest modes of the system (4.16) - (4.17). We choose our normalization velocity V_0 to be

$$V_0 = V_A = \frac{B_0}{(\mu\rho)^{1/2}}, \tag{4.22}$$

and normalize time by the time it takes Alfven wave to propagate the distance l_0, i.e.

$$t_0 = t_A = \frac{l_0}{V_A}. \tag{4.23}$$

Using the values from (4.22) and (4.23) in (4.19) - (4.20) and dropping the hats we reduce these equations to

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{V} \times \mathbf{B}) + \frac{1}{S}\nabla^2 \mathbf{B}, \tag{4.24}$$

$$\frac{\partial \mathbf{V}}{\partial t} = -\mathbf{V} \cdot \nabla \mathbf{V} + (\nabla \times \mathbf{B}) \times \mathbf{B} + \frac{1}{R}\nabla^2 \mathbf{V}. \tag{4.25}$$

The nondimensional parameters in this system are the Reynolds number

$$R = \frac{V_A l_0}{\nu}, \tag{4.26}$$

which gives the ratio of the size of the inertial term to the viscous term in the momentum equation and the Lundquist number

$$S = \frac{l_0^2/\eta}{t_A}, \tag{4.27}$$

which is the ratio of the resistive to the Alfven time scales in the system. The latter parameter is very important in the theory of MHD. Although the value of the Lundquist

number in the corona is very large ($\sim 10^{10}$) the resistive term provides a very important dissipative mechanism by which magnetic field lines diffuse across plasma and so drastically change the topology of the overall magnetic field. This process is called reconnection [68] and it is believed to be at the heart of the eruptive processes in the corona. Using realistic coronal values of the Lundquist number in numerical simulations is beyond current computational resources, and so S is normally chosen as large as the numerical methods and the computing tools allow, typically on the order of $10^4 - 10^5$. While the importance of the Lundquist number is usually emphasized in the numerical magnetohydrodynamics literature, the magnitude of the Reynolds number is not given much attention. Even though for the corona R should be larger than the Lundquist number it is normally chosen to be on the order 10^2 mostly out of numerical rather than physical considerations. Our numerical experiments showed that the Reynolds number is important and its magnitude affects the dynamics of the plasma by changing the rate at which evolution is occurring. We will present calculations supporting this conclusion later in this work.

To summarize, we are interested in numerically modeling the evolution of the coronal magnetic arcades using the simplified system of magnetohydrodynamic equations (4.24)-(4.25) and studying the dependence of the dynamics on the parameters R and S. In the next section we will discuss the computational difficulties this system presents and the advantages that exponential propagation methods offer compared to the traditional numerical techniques used to solve these equations.

4.3 Numerical magnetohydrodynamics

The magnetohydrodynamic description is used for a variety of astrophysical and laboratory plasma applications. Due to the complexity of the equations the only opportunity to obtain a solution for many problems is through a numerical approach. Depending on the plasma regime different numerical techniques should be used to solve the MHD equations [58]. For example, for problems where MHD shocks arise (e.g. in modeling the global structure of the Earth's magnetosphere) the equations have to be formulated in a conservative form and Godunov-type explicit schemes have to be used to solve the system. Many such schemes have been developed and used for different plasma applications, see for example [66, 82, 24, 22].

We are interested in solving the *resistive* MHD equations in a parameter regime where no

shocks develop. The major numerical difficulty arising in this type of problems is existence of widely separated time scales in the system. For the coronal applications the Ludquist number $S = \tau_R/\tau_A$ represents the ratio of the slowest or resistive time to the fastest or Alfven time scales and is very large. This implies that the system (4.24)-(4.25) is *stiff*. As we mentioned before, coronal plasma configurations can exist stably for very long periods of time. Thus, it is the long term evolution that we want to obtain by solving the resistive MHD equations. The time scale of such evolution is expected to be much larger than the Alfven time scale but smaller than the resistive time scale. Using the explicit techniques suggested in the references above, however, would require the time step to conform to the CFL condition defined by the fast Alfven scale.

The use of explicit schemes is even more inappropriate when we take into account yet another numerically challenging aspect of our problem, namely, that the spatial scales of coronal arcade configurations and their evolution are also quite spread out. The reconnection process will be happening in a very narrow region having size determined by the inverse of the Lundquist number, while the overall scale of the coronal arcade is many orders of magnitude larger. Also, since the computational domain is finite but the real arcades are not confined to any limited space, we would like to extend the boundaries of our computational domain far enough so that the solution is not polluted by non-physical boundary effects. These considerations lead to the conclusion that the spatial grid required for this three-dimensional problem with six unknown variables should be very large; this makes the CFL restriction on the time step even more severe. In fact, the use of explicit techniques might make numerical integration of the resistive MHD equations over the time period of interest infeasible even for the fastest current supercomputer.

The alternative approach is to turn to implicit techniques to solve the equations of interest. To develop a fully implicit scheme (i.e. a scheme where all the terms on the right hand side of the equations are handled implicitly) for the resistive MHD equations is a very complicated task because the matrices arising from the spatial discretization of the full implicit operator are very large and complex to invert. It is also not always clear how much accuracy was sacrificed and how much physics filtered out in such implicit computations in order to get savings over the explicit schemes by taking a large time step. Due to these complications there have been fewer attempts to develop implicit methods for the resistive MHD system compared to the number of explicit schemes proposed. The main

approaches that have been taken in this direction try to approximate implicitly only some terms in the equations, while treating the rest of the spatial operators explicitly. Probably the most notable of such methods are the fluid-implicit-particle (FLIP) method developed by Brackbill [14] and the semi-implicit methods first suggested by Harned and Kerner [36], then further developed by Harned and Schnack [37] and finally generalized by Lerbinger and Luciani in [52]. As the name indicates the FLIP method does not solve the MHD equations in the form (4.12)-(4.15), but complements the continuum plasma model with the particle description. In fact this method is mostly used for kinetic plasma simulations and is hard to apply to the large-scale coronal phenomena. The semi-implicit methods, on the other hand, have been developed and used on stability problems for the coronal arcades. We will briefly describe this numerical technique and explain what limitations of this method made us choose an exponential propagation method instead.

Semi-implicit methods are based on modifying the momentum equation by introducing an extra artificial term and then evolving this term in an implicit fashion. Specifically, if the right-hand-side of the momentum equation is denoted by $\mathbf{F}_v$, i.e. if the original momentum equation (4.25) can be written as

$$\frac{\partial \mathbf{V}}{\partial t} = \mathbf{F}_v, \tag{4.28}$$

then some linear operator $\mathbf{G}$ is chosen and the new momentum equation is introduced as

$$\frac{\partial \mathbf{V}}{\partial t} = \mathbf{F}_v + \Delta t \mathbf{G} \cdot \frac{\partial \mathbf{V}}{\partial t}, \tag{4.29}$$

where Δt is the time step. The time derivatives in (4.29) are discretized to get

$$\frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t} = \mathbf{F}_v + \mathbf{G} \cdot (\mathbf{V}^{n+1} - \mathbf{V}^n). \tag{4.30}$$

Now in order to advance the velocity field in time, the following matrix system has to be solved:

$$(I - \mathbf{G}\Delta t)\mathbf{V}^{n+1} = \mathbf{V}^n + (\mathbf{F}_v - \mathbf{G}\mathbf{V}^n)\Delta t, \tag{4.31}$$

where I is the identity operator. The operator $\mathbf{G}$ is chosen in such a way as to attempt

to include the fast Alfven modes of the system, so that they can be solved implicitly and the severe CFL time step restriction can be relaxed somewhat. In the original semi-implicit method from [36, 37, 72] two forms of the operator $\mathbf{G}$ have been proposed. The first one is derived from the linearization of the equations (4.24) and (4.25) around some equilibrium magnetic field $\mathbf{B}_0$ and zero velocity field to get the linear Alfven wave equation

$$\frac{\partial^2 \mathbf{V}}{\partial^2 t} = [\nabla \times \nabla \times (\mathbf{V} \times \mathbf{B}_0)] \times \mathbf{B}_0. \tag{4.32}$$

Then $\mathbf{G}$ is set to be

$$\mathbf{G} = \alpha[\nabla \times \nabla \times (\mathbf{V} \times \mathbf{B}_0)] \times \mathbf{B}_0, \tag{4.33}$$

where α is a constant. Since for such $\mathbf{G}$ the inversion of $(I - \Delta t \mathbf{G})$ is still a complicated problem, another much simpler form of the this operator has been proposed

$$\mathbf{G} = \alpha \nabla^2 \mathbf{V}. \tag{4.34}$$

For both of the operators above the constant α is chosen in such a way as to ensure nonlinear numerical stability for the method.

In the version of the method proposed by Lerbinger and Luciani in [52, 7] the Laplacian plus the full linearized MHD operator is used as an implicit term, i.e.

$$\mathbf{G} = \left(c_1 \nabla^2 + c_2 \Delta t^2 / \rho \left[\nabla \times \mathbf{B}_0 \times \nabla \times (\mathbf{V} \times \mathbf{B}_0) + \nabla \times [\nabla \times (\mathbf{V} \times \mathbf{B}_0)] \times \mathbf{B}_0 \right] \right) \mathbf{V}, \tag{4.35}$$

with some constants c_1 and c_2.

The advantage of using a simple Laplacian operator as $\mathbf{G}$ is, of course, the fact that a simple conjugate gradient (CG) method can then be used to solve the matrix equation (4.31), while a much more complicated preconditioner had to be developed [7] to solve this system with $\mathbf{G}$ defined by (4.35).

The semi-implicit method has several limitations which we would like to point out. First, even though the method does remove the restriction on the time step, no general stability condition has been derived for any form of the semi-implicit operator, so there

is no automatic procedure or conditions from which constants α in (4.31) or c_1 and c_2 in (4.35) yielding a stable method can be calculated. Second, even if the scheme is proved to be stable it is not clear how modifying the momentum equation affects the accuracy of the solution. Even though the added term vanishes as $\Delta t \to 0$, the reason for using the semi-implicit method in the first place is that it allows taking a large time step and in this case the extra term will be large since it is proportional to Δt. Third, both versions of the method are not easily generalized to arbitrary spatial grids. Due to the widely separated spatial scales we might want to use an adaptive nonuniform grids for this problem. Such grids, however, drastically change the matrix $(I - \Delta t\mathbf{G})$, so that the symmetric positive-definite property for (4.34) required for using the conjugate gradient method to invert this matrix or the specific matrix structure of [7] which the preconditioner in [7] is built upon, might not be preserved. Finally, for the general nonlinear MHD problem the equilibrium magnetic field $\mathbf{B}_0$ used to define $\mathbf{G}$ in (4.35) is not necessarily known in advance and while the magnetic field calculated at a previous time could be used, it is not clear whether it will preserve the properties of the system. The semi-implicit method has been shown to perform well for many problems, for examples see [56, 53, 57, 6, 5], however, we feel that, from a purely numerical point of view, there is no rigid theoretical analysis which would provide error bounds and help determine the appropriate parameters for the method in the course of a simulation. The only way to check whether the solution obtained by the semi-implicit method is accurate is to perform an a posteriori convergence analysis. But as we discussed before, to resolve the spatial scales of interest for *three-dimensional* MHD problems we need extremely large grid sizes which push computer memory and speed requirements to the limit, so increasing grid size by several factors in order to study convergence might not be feasible.

Another important issue to be considered before developing an MHD code is the need to satisfy the zero-divergence condition implicitly imposed on a magnetic field, i.e.

$$\nabla \cdot \mathbf{B} = 0. \tag{4.36}$$

Although analytic forms of the induction equation (4.24) preserve this condition if the initial magnetic field is divergence free, when the equations are discretized and evolved, numerical errors can cause an accumulation of non-zero divergence. There are a number of different

approaches proposed in the literature to deal with this problem. For explicit Godunov-type schemes the equations have to be reformulated and modified in a special way [66] to meet this requirement. For semi-implicit methods the zero-divergence condition is satisfied by using second order discretization on a staggered grid for the spatial operators or by expressing the magnetic field in terms of the magnetic vector potential $\mathbf{A}$ ($\nabla \times \mathbf{A} = \mathbf{B}$). The disadvantage of the latter method is the difficulty in deriving the boundary conditions for $\mathbf{A}$ from the given boundary conditions for $\mathbf{B}$. Brackbill and Barnes [15] proposed to remove the nonsolenoidal part of the magnetic field by solving the Poisson equation

$$\nabla^2 \phi = -\nabla \cdot \mathbf{B} \tag{4.37}$$

for the potential ϕ after advancing the solution over a time step, then calculating the divergence free field from

$$\mathbf{B}_{df} = \mathbf{B} - \nabla \phi \tag{4.38}$$

and using this new field to compute the solution over the next time interval. It is also possible to avoid performing this projection at each time step and only use it once in a while during the computation to get rid of the accumulated numerical divergence in $\mathbf{B}$.

We want to emphasize two issues in regard to keeping the magnetic field divergence free that will be important for our numerical calculations. In [15] it was shown that this problem has a severe effect on the numerical solution when the MHD equations are formulated in a conservative form, whereas the nonconservative formulation (4.12) - (4.15) does not lead to significant problems with the solution. Zachary *et al.* [82] found that the numerical solution did not change much whether or not the divergence free projection was used. Based on these results, since we use the nonconservative form of the equations, we do not expect the solution to be significantly influenced by the numerical errors in the divergence of the magnetic field.

Another important issue is the formulation of the boundary conditions. Obviously, the imposed boundary conditions will also impact the divergence of $\mathbf{B}$. This is a difficult problem and different boundary conditions have been proposed for MHD modeling, see discussions in [8, 81, 59, 79]. We will describe the particular approach we take later in our discussion.

4.4 Relaxation theory and the numerical coronal models

The main goal of this section is to show that exponential propagation methods offer significant advantages for MHD problems compared to other numerical techniques. We will now introduce relaxation theory which is used to explain a variety of plasma phenomena, and describe how it is applied to the coronal problems. We then focus on two models of the dynamics of the solar arcades and demonstrate that the results of our numerical simulations using exponential propagation methods agree with previously published numerical studies.

4.4.1 Introduction

The question of which physical constraints and processes govern coronal plasma dynamics has long been a subject of scientific debate. An argument based on a combination of the Taylor relaxation theory and reconnection theory has probably been most frequently used to describe the evolution of the coronal magnetic fields. We now present a brief account of the history and conclusions of such theories.

Since as we mentioned before, large-scale plasma configurations in the corona remain in a stable equilibrium for long periods of time, the forces acting on this structures must be in balance. Plasma β in the corona is quite low ($\beta \leq 10^{-3}$) and so magnetic forces dominate plasma dynamics. Therefore we expect the magnetic field in equilibrium plasma configurations to be nearly force-free, i.e. the Lorentz force must vanish so that

$$\mathbf{J} \times \mathbf{B} = 0. \tag{4.39}$$

Equation (4.39) implies that the magnetic field is aligned with the electric current density vector. Taking into account (4.6), Equation (4.39) can be written as

$$\nabla \times \mathbf{B} = \alpha(\mathbf{r})\mathbf{B}, \tag{4.40}$$

where $\alpha(\mathbf{r})$ is some scalar function of a position. Taking the divergence of Equation (4.40) and using the fact that the magnetic field is always divergence free we obtain

$$(\mathbf{B} \cdot \nabla)\alpha = 0. \tag{4.41}$$

This implies that α is constant along each of the magnetic field lines. The configurations of the magnetic field that satisfy equation (4.40) are called the *force-free* fields. If $\alpha = 0$ equation (4.40) describes a potential magnetic field, which can be easily calculated from the Laplace's equation. To obtain the force-free field for the non-constant α case Equation (4.40) has to be solved. Depending on whether α is uniform or non-uniform in space, the corresponding solutions to (4.40) are called the *linear* (uniform-α) or *nonlinear* (non-uniform-α) force-free states respectively. For certain geometries it is possible to compute the solutions to (4.40) analytically. For instance, the axisymmetric magnetic field in cylindrical geometry can be calculated exactly from the Grad-Shafranov force-free equation [67, 65]. For the case of a uniform α the problem is relatively easy to solve numerically as we will demonstrate later in this chapter. In a general case of non-zero α, however, to find a solution of (4.40) is a difficult task because of an implicit nonlinearity of this equation. It is nevertheless important to solve these problems and compare the solutions with the actual coronal plasma configurations for two reasons. First, we want to check whether the stable plasma structures in the corona are indeed in force-free equilibrium states. Second, force-free magnetic fields are important in relaxation theory which has been successfully used to explain the dynamics of the laboratory plasmas and is conjectured to be applicable to the processes in solar corona.

Before discussing relaxation theory we need to define another important quantity that characterizes a state of plasma. If $\mathbf{A}$ is a magnetic vector potential corresponding to a magnetic field $\mathbf{B}$ so that

$$\nabla \times \mathbf{A} = \mathbf{B}, \tag{4.42}$$

and plasma is contained in a singly connected volume V, then the following integral

$$K = \int_V \mathbf{A} \cdot \mathbf{B} dV \tag{4.43}$$

measures the amount of twist and linkage of the magnetic field lines and is called the *magnetic helicity* of the plasma. Since the magnetic vector potential is a gauge dependent quantity, so is the magnetic helicity K. The only exception to this is the case when no magnetic field lines penetrate the boundary ∂V of the volume V, then it can be shown [11]

that K is gauge independent. In the case when the volume does not completely enclose all of the magnetic field lines and some of them intersect the boundary the classical definition of magnetic helicity (4.43) becomes ambiguous and a so-called *relative* magnetic helicity has to be defined. Finn and Antonsen [25] introduced a gauge-independent relative helicity by introducing a potential magnetic field $\mathbf{B}_{vac}$, defined such that its normal component on the boundary ∂V is equal to the normal component of $\mathbf{B}$. Such a potential magnetic field is clearly unique and if the corresponding vector potential is denoted as $\mathbf{A}_{vac}$ the relative helicity is given by

$$K_{rel} = \int_V (\mathbf{A} + \mathbf{A}_{vac}) \cdot (\mathbf{B} - \mathbf{B}_{vac}) dV. \tag{4.44}$$

This is a gauge invariant quantity which reflects the topological properties of the magnetic field, namely, it quantifies how twisted and linked the magnetic field lines are.

Now using the above definitions we will give a brief introduction to relaxation theory. In 1958 Woltjer [80] proved that, for a perfectly conducting plasma (i.e. infinite Lundquist number) in a closed volume, the magnetic helicity K defined in (4.43) is conserved and if the plasma relaxes to a minimum energy state, such state is a linear force-free field. Taylor extended Woltjer's theory to explain why a reversed toroidal field was developing near the wall in the laboratory reversed field pinch experiments, in which plasma was contained in a torus with rigid perfectly conducting walls and allowed to relax. It was also conjectured that for a slightly dissipative plasma, i.e. plasma with the small non-zero resistivity, during the relaxation process the resulting changes in topology are accompanied by small changes in $\mathbf{B}$ so that the magnetic helicity $K = \int_V \mathbf{A} \cdot \mathbf{B} dV$ stays nearly constant compared to the changes in the magnetic energy. By Woltjer's theorem this means that plasma relaxes to the minimum energy state - the linear force-free field. This theory cannot be directly applied to explain the behavior of the coronal plasma configurations since the magnetic field lines of these structures are rooted in the photosphere which has different physical properties. However, because of the success of the relaxation theory in understanding the behavior of laboratory plasmas with the use of the notion of relative helicity the conjecture has been proposed to explain the dynamics of the coronal magnetic arcades. While there is still much variation in the details of different theories, much of the coronal modeling based on relaxation theory concentrates on solving the following problem. The dynamics of the

solar coronal arcades is interpreted as the evolution of the low-beta plasma through a series of force-free configurations in response to motion of the footpoints of the magnetic field lines that are carried along by the flow of the much more massive, compared to the corona, photosphere. As a result the coronal magnetic field lines get sheared and twisted and the magnetic energy gets stored in the stressed magnetic field. This evolution is assumed to be constrained by the requirement that the total relative helicity is preserved. One of the key problems in modeling the evolution of the coronal arcades is explaining the final eruptive stage during which a sudden massive release of energy occurs. There are two most widely debated theories that attempt to describe the eruptions. In the first one it is argued that during the slow evolution of plasma through a series force-free states an unstable equilibrium or an instability are reached and this causes the eruption. There is no reconnection process changing the magnetic field topology in this case. The alternative theory uses the fact that the resistivity is non-zero and suggests that once a certain magnetic configuration has developed the reconnection process acts locally to drastically change the topology of the magnetic field and cause the release of energy.

A number of analytical and numerical studies have been undertaken in an attempt to understand the dynamics of the coronal arcades. The complexity of the resistive magnetohydrodynamic equations, the complicated three-dimensional structure of the magnetic configurations and the limitations of the observational data make confirming or disproving a theory a very difficult task. An important step towards this goal has been investigations of sheared magnetic arcades in two-dimensions via numerical solution of the equations of resistive MHD. In the next section we briefly summarize previous studies that addressed this problem, then present a specific problem and discuss the results of using exponential propagation methods for this problem.

4.4.2 Two-dimensional simulations of periodic solar arcades using the fourth-order exponential propagation method

The problem of the evolution of an axisymmetric magnetic arcade under imposed boundary shear has been the subject of many previous analytical and numerical studies. These studies have characterized the evolution as a quasi-static ideal MHD problem [26, 48, 64] or as a dynamical system [56, 12, 43, 18]. For a short review of this research area see [4]. The problem is typically prescribed by assuming an initial dipole magnetic field, and

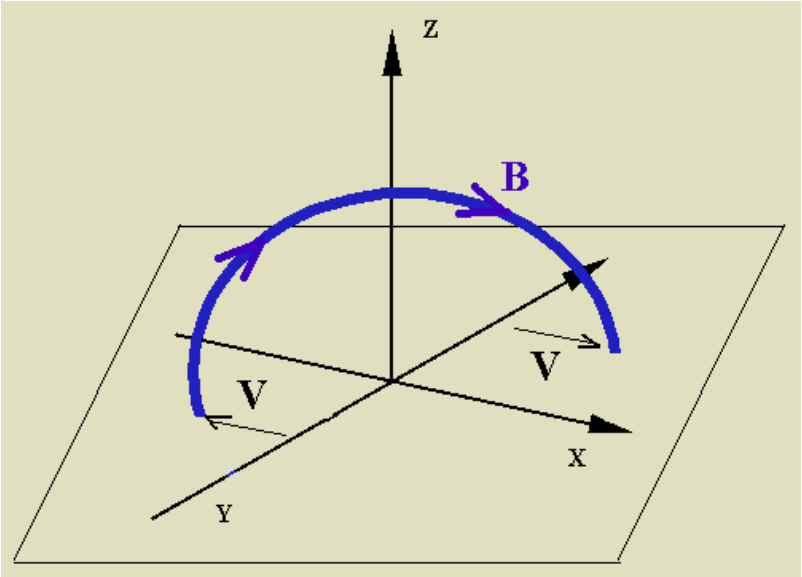


Figure 4.6: Sheared two-dimensional magnetic field line, the shearing velocity is indicated by the arrows labeled **V**

then imposing slow shearing motion of the footpoints as boundary conditions (Fig. 4.6) at a base plane corresponding to the photospheric surface. The equations are then evolved using either a full or a reduced set of the magnetohydrodynamic equations (4.13) - (4.15). Frequently several such potential magnetic arcades are positioned next to each other and used as the initial condition for the simulation. The calculation is usually carried out for several hundreds of Alfven times. While different parameters, initial configuration and imposed boundary conditions have been used in various studies, a certain dynamical behavior of the arcade in the course of evolution have been identified in many simulations. In particular, it is typically found that for some time after the onset of the shearing motion the arcade undergoes slow quasi-static evolution through a sequence of force-free states. During this quasi-static phase current concentrations develop which eventually lead to a drastic change of magnetic field topology via the reconnection process. In the latter stages of the evolution a plasmoid is formed and is ejected at approximately the Alfven speed. We now present a specific example exhibiting such behavior.

Since our primary objective here is to demonstrate that exponential propagation methods can be successfully applied to resistive MHD problems, we pick a typical two-dimensional coronal model described by Mikić *et al.* in [56] and check whether the dynamics we obtain with our code are similar to the solutions presented by Mikić *et al.*.

We solve the resistive MHD system (4.24)-(4.25) on a rectangular domain

$$\Omega = \{(y, z) : 0 \leq y \leq L_y, 0 \leq z \leq L_z\} \tag{4.45}$$

using the fourth-order exponential propagation method (3.113) described in Chapter 3. The boundary $z = 0$ corresponds to the photospheric base plane. The initial magnetic field is assumed to be vacuum field(i.e. current-free $\nabla \times \mathbf{B} = 0$) $\mathbf{B} = (B_x, B_y, B_z)$ that consists of two magnetic arcades and can be expressed analytically as

$$B_y(y, z, 0) = B_0 e^{-k_1 z} \cos(k_1 y), \tag{4.46}$$

$$B_z(y, z, 0) = -B_0 e^{-k_1 z} \sin(k_1 y), \tag{4.47}$$

$$B_x(y, z, 0) = 0, \tag{4.48}$$

where $k_1 = 2\pi/L_y$. The problem is assumed to be invariant in x-direction, so that $\partial/\partial x = 0$ in Equations (4.24)-(4.25). The photospheric flow is modeled by imposing the line-tied boundary condition at the base plane $z = 0$, specifically, the velocity is prescribed to satisfy

$$V_x(y, 0, t) = -V_0 f(t) \sin(k_1 y), \tag{4.49}$$

$$V_y(y, 0, t) = 0, \tag{4.50}$$

$$V_z(y, 0, t) = 0, \tag{4.51}$$

where the time profile of the flow is assumed to be a linear ramp up of the velocity, i.e.

$$f(t) = t/t_R, \qquad \text{for } 0 \leq t \leq t_R, \tag{4.52}$$

$$f(t) = 1, \qquad \text{for } t \geq t_R. \tag{4.53}$$

The configuration is assumed to be periodic in the y-direction. The top boundary $z = L_z$ is removed as far as possible so that it does not influence the solution and a zero-derivative condition is imposed on all the variables at this boundary. At the base boundary B_z is fixed to be

$$B_z(y, 0, t) = B_z(y, 0, t = 0) = -B_0 e^{-k_1 z} \sin(k_1 y). \tag{4.54}$$

The conditions for B_x and B_y at $z = 0$ are derived from the constraint imposed on the tangential electric field and Ohm's law. It is assumed that

$$E_x(y, 0, t) = 0, \tag{4.55}$$

$$E_y(y, 0, t) = -V_x(y, 0, t)B_z(y, 0, t). \tag{4.56}$$

Since by Ohm's law

$$E_x = -(V_y B_z - V_z B_y) + \eta(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z}) \tag{4.57}$$

$$E_y = -(V_z B_x - V_x B_z) + \eta(\frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x}), \tag{4.58}$$

imposing conditions (4.55) and (4.56) on the electric field $\mathbf{E}$ is equivalent to setting the following boundary conditions for the tangential components of the magnetic field at the base plane

$$\frac{\partial B_x}{\partial z}(y, 0, t) = 0, \tag{4.59}$$

$$\frac{\partial B_y}{\partial z}(y, 0, t) = \frac{\partial B_z}{\partial y}(y, 0, t) = -B_0 k_1 \cos(k_1 y), \tag{4.60}$$

and can also be interpreted as the zero condition on the tangential component of the current density $\mathbf{J} = 1/\mu_0 \nabla \times \mathbf{B}$.

The parameter values that we use for V_0, B_0, t_R and the Lundquist and Reynolds numbers S and R are chosen to be exactly the same as used in the simulation presented in [56] in order to our exponential propagation method code with the results presented in [56]. The maximum applied photospheric flow velocity is taken to be $V_0 = 0.01 V_A$, where V_A is the Alfven velocity. For the nondimensionalized variables in the equations (4.24) - (4.25) $V_A = 1$. The flow is increased linearly up to $t_R = 200\tau_A$ according to (4.52), with τ_A being the Alfven time, which corresponds to $t = 1$ in the equations (4.24) - (4.25). The resistive and the viscous time scales are chosen as $\tau_r = 10^4 \tau_A$ and $\tau_\nu = 100\tau_A$, which gives the Lundquist number $S = 10^4$ and the Reynolds number $R = 100$. The size of the box in the nondimensionalized spatial variables is $L_y = 1$ and $L_z = 6$, the latter corresponding to removing the upper boundary as far as possible to avoid the boundary effecting the computation.

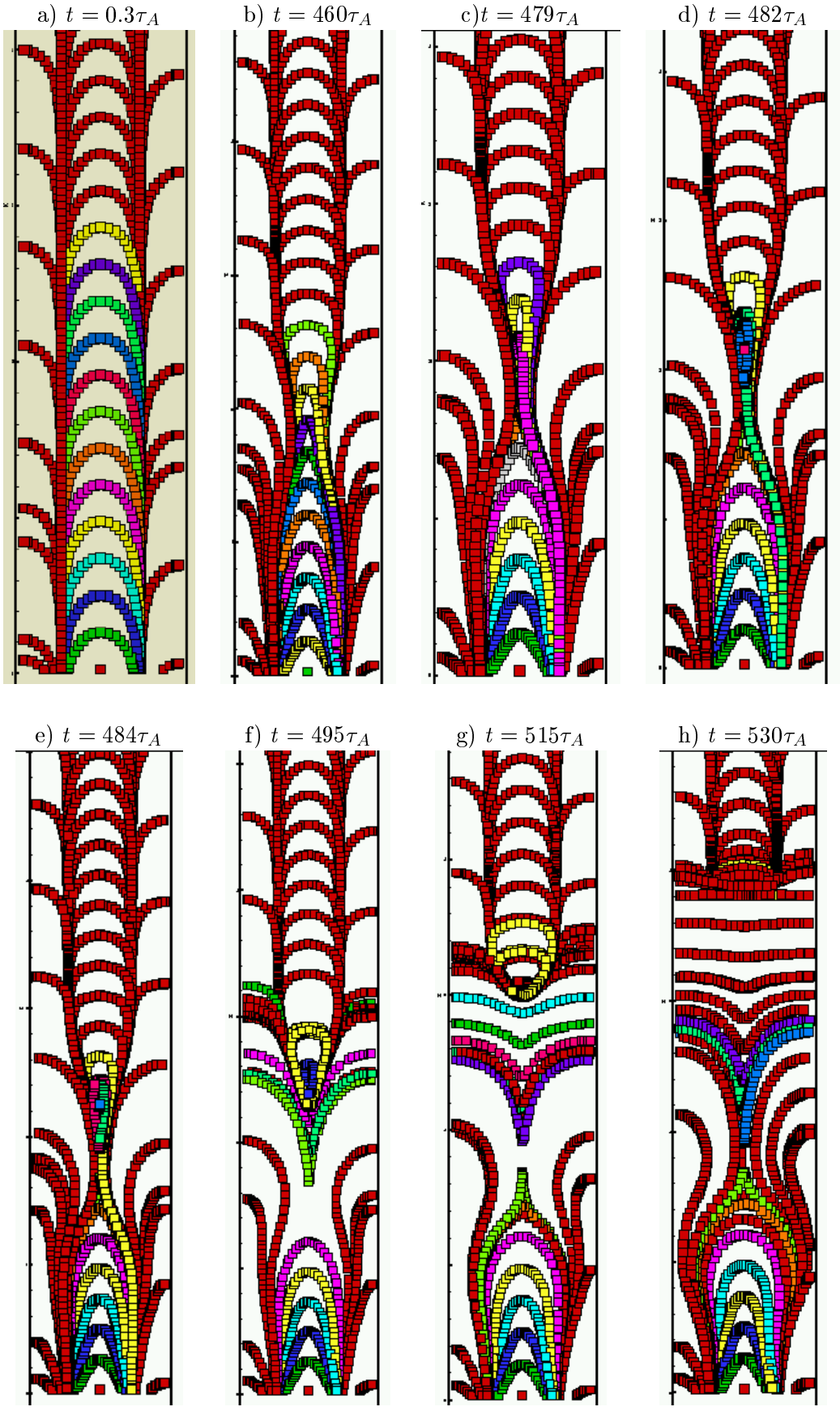


Figure 4.7: The evolution of the field lines of the magnetic field **B** projected onto yz-plane over 530 Alfven times.

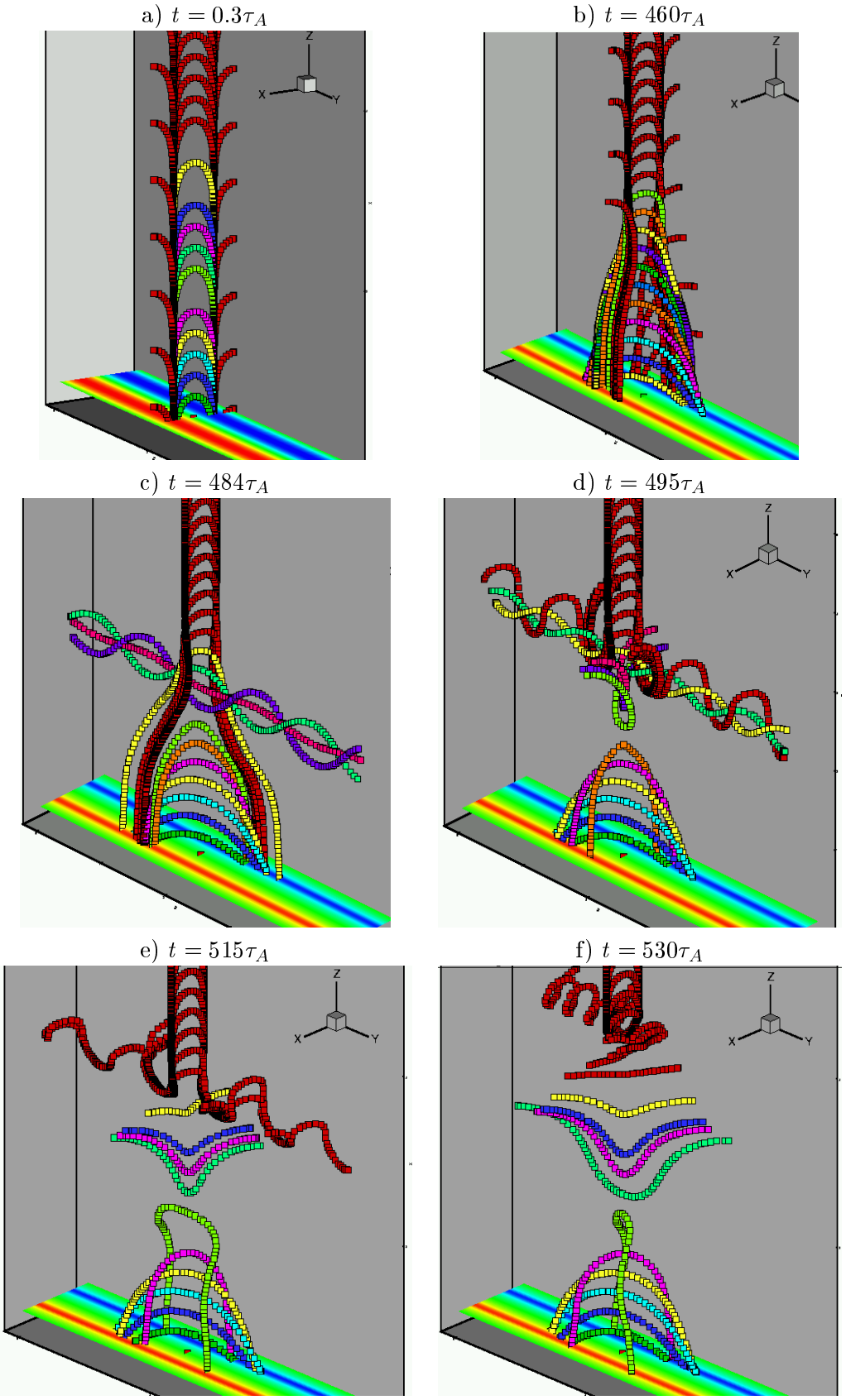


Figure 4.8: The three-dimensional side view of the magnetic field lines at different Alfven times. The contour plot of $\mathbf{B}_z$ component is displayed at the base plane.

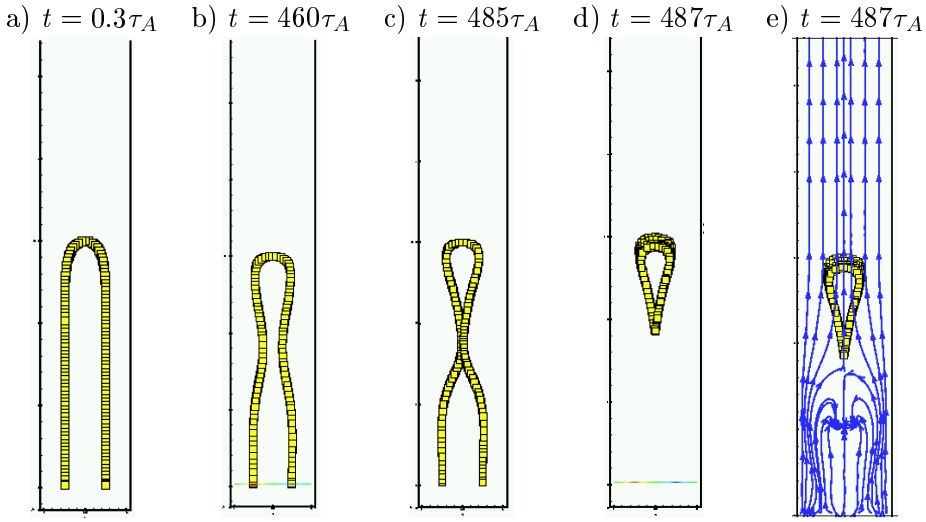
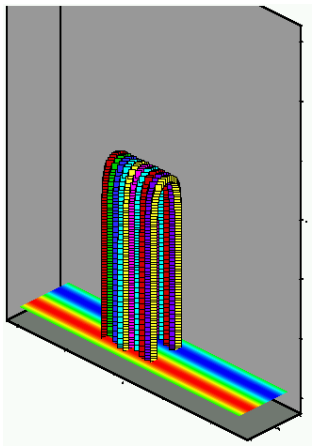


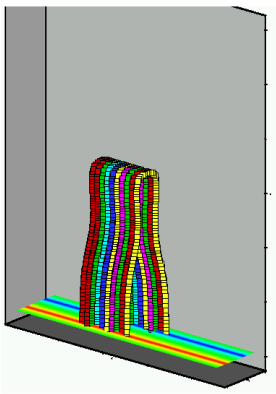
Figure 4.9: The reconnection process in the arcade of adjacent magnetic field lines in the uniform dimension projected onto yz-plane. In figure e) the velocity streamlines are shown as blue lines with arrows.

Figures 4.7, 4.8 show the temporal evolution of the magnetic field lines from $t = 0$ to $t = 530\tau_A$ (recall that t is measured in Alfven times τ_A). As we can see from these pictures the behavior is very similar to the dynamics obtained by Mikić *et al.*. As shown in Figures 4.7(a), 4.8(a) the evolution starts with the unsheared configuration of the initial potential magnetic field lines of the potential $\mathbf{B}$ given by Equations (4.48)-(4.47). As the flow velocity at the base plane is ramped up, the field lines become sheared and the configuration evolves through a series of quasi-static force-free states for a period of roughly 480 Alfven times. At this point (Fig. 4.7(c)) a current sheet forms in the central arcade and the reconnection process starts. As a result of the reconnection, the topology of the magnetic field lines changes drastically and a plasmoid like structure forms. Figures 4.9, 4.10 show a more detailed view of the reconnection process. Magnetic arcades are assumed to be uniform in the x-direction and as Figures 4.9(c), 4.10(c) demonstrate, the shear causes the field lines that have the same x coordinate to come together and eventually reconnect forming a spiral field line that outlines the surface of the plasmoid. Note that lower lying field lines are sheared more that the ones at the higher altitude. This is consistent with the observations of the coronal magnetic arcades and will also be demonstrated in the three-dimensional simulations in the next chapter. The formed plasmoid is being ejected and travels upwards as we can see from Figure 4.9(e), that shows the streamlines of the velocity relative to the position of the plasmoid. The reconnection is fast, i.e. it occurs on the Alfven time scale,

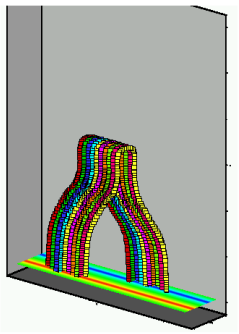
a) $t = 0.3\tau_A$



b) $t = 460\tau_A$



c) $t = 485\tau_A$



d) $t = 487\tau_A$

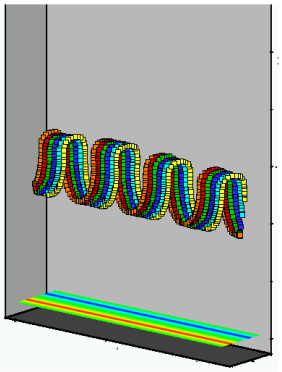


Figure 4.10: The three-dimensional view of reconnection process in the arcade of adjacent field lines in the uniform dimension

which is consistent with the result in [56].

The overall dynamics as calculated by our code closely resembles the results published in [56] and is completely consistent with the behavior expected from this model on the physical grounds. This result serves as a validation for our code. There is a difference, however, between the solutions we obtain with exponential propagation code and the ones presented in Mikić *et al.*. The timing of the distinct stages in the evolution is different, e.g. the onset of the reconnection in our calculations happens at roughly 480 Alfven times, while Mikić *et al.* notice the formation of the plasmoid at around 340 Alfven times. We attribute these differences to the numerical aspects of solving the resistive MHD equations. In particular, it is possible that the numerical boundary conditions differ somewhat in the two codes or the use of semi-implicit methods versus exponential propagation scheme has an effect on the solution. Additional numerical experiments with both codes and close comparisons of the implementations have to be made in order to determine exact causes for these differences in an undertaking we will attempt in the future. The close monitoring of the error by the automatic error control mechanism embedded in our code, however, makes us confident in our results.

Our numerical model was based on the $exp4$ code described in section 3.5.2. The spatial discretization was performed using fourth-order central finite differences on a grid and the time integration was accomplished with method (3.113) using the automatic error control mechanism as described in the previous chapter. For the figures in this section we used the data from the calculation with a 200×140 grid. We compared the efficiency of the method with the fourth-order explicit Runge-Kutta method with an identical error control scheme in the same way as was done in Chapter 3 for Burgers equation. The computations were performed on a 233 MHz Pentium II Linux workstation. The comparison study showed that in order to compute the solution for a given accuracy, the time step used by the $exp4$ code can be 100 times larger than the maximum time step size allowed by the stability constraints for the explicit Runge-Kutta scheme (Fig. 4.11). The overall integration time was reduced by a factor of 10-15 depending on the size of the grid and the final integration time used. This speed-up is expected to increase if a more refined grid is used or the system is integrated over longer periods of time. Also better results should be obtained if the parameters in the equations are changed so that the system becomes more stiff.

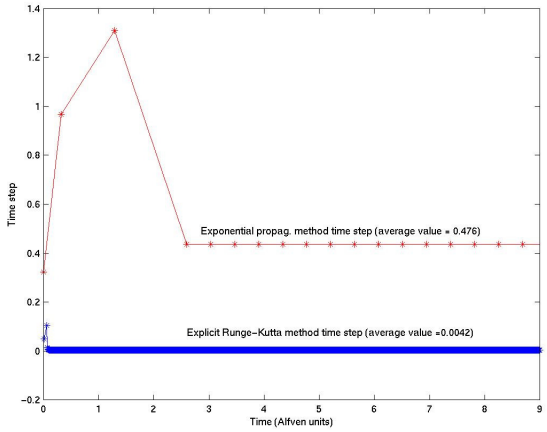


Figure 4.11: Comparison of the time step size between the 4-th order *exponential* Runge-Kutta method and the 4-th order *explicit* Runge-Kutta method

4.4.3 Computation of the three-dimensional plasma force free-states

As discussed earlier the behavior of the coronal arcades can be studied not only by solving the dynamical resistive MHD equations, but by modeling the arcades as a series of force-free plasma configurations. A clear demonstration of the connection between the dynamical simulations and the solutions to the force-free equation (4.40) has been presented in several papers [12, 28, 27]. In particular, Finn *et al.* [28] showed the similarity between the solutions to (4.40) and the results of the full MHD simulations for the axisymmetric case. Finn *et al.* demonstrated that the axisymmetric solutions to (4.40) have similar plasmoid-like structures that appear in the simulations of the type presented in the previous subsection. Finn *et al.* examined the stability of the solutions to (4.40) by computing the eigenfunctions $\delta\mathbf{B}$ and the eigenvalues λ of this equation and established that the equilibrium solutions of the force-free equation

$$\nabla \times \mathbf{B} = \alpha \mathbf{B} \tag{4.61}$$

are stable as long as $\alpha^2 \leq \lambda_1^2$, where λ_1 is the eigenvalue of

$$\nabla \times \delta\mathbf{B} = \lambda \delta\mathbf{B} \tag{4.62}$$

with the smallest magnitude. They associated loss of equilibrium of the magnetic arcade with its reaching an unstable force-free state. It was argued that the nonlinear stage of the evolution begins with the X-line being formed and the subsequent reconnection causes the ejection of the plasmoid.

Having demonstrated the connection between the slow evolution through force-free equilibria specified by the footpoint motion in the dynamical simulations and the constant α force-free states in the axisymmetric case, Finn analyzed the three-dimensional solutions to (4.40). In order to obtain force-free fields in 3D, given a certain distribution of the normal component of the magnetic field on the boundary, the following equation is solved for the magnetic vector potential $\mathbf{A}$ ($\nabla \times \mathbf{A} = \mathbf{B}$)

$$\frac{\partial \mathbf{A}}{\partial t} = \nabla^2 \mathbf{A} + \alpha \mathbf{B}. \tag{4.63}$$

The computational domain is taken to be a Cartesian box

$$\Omega = \{(x, y, z) : x_{min} \leq x \leq x_{max}, y_{min} \leq y \leq y_{max}, z_{min} \leq z \leq z_{max}\} \tag{4.64}$$

The boundary condition on the tangential component $\mathbf{A}_t$ of the vector potential is set in such a way as to be consistent with the given normal component of the magnetic field $\mathbf{B}_n = \mathbf{B} \cdot \hat{n}$, and the normal component $\mathbf{A}_n$ is specified so that the condition $\nabla \cdot \mathbf{A} = 0$ is preserved on the boundary $\partial\Omega$. Using $\nabla \times \mathbf{A} = \mathbf{B}$ and substituting the vector identity

$$\nabla^2 \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla \times \nabla \times \mathbf{A} \tag{4.65}$$

into (4.63) we obtain

$$\frac{\partial \mathbf{A}}{\partial t} = \nabla(\nabla \cdot \mathbf{A}) - \nabla \times \mathbf{B} + \alpha \mathbf{B}. \tag{4.66}$$

From the last equation we can see that since $\nabla \cdot \mathbf{B} = 0$ we have

$$\frac{\partial}{\partial t} \nabla \cdot \mathbf{A} = \nabla^2 \nabla \cdot \mathbf{A}. \tag{4.67}$$

This means that if $\nabla \cdot \mathbf{A} = 0$ is maintained on the boundary, then $\nabla \cdot \mathbf{A} \to 0$ as $t \to \infty$ and if the equation (4.63) is evolved to a steady state then from (4.66) we can see that this

steady state satisfies

$$\nabla \times \nabla \times \mathbf{A} = \alpha \nabla \times \mathbf{A}, \tag{4.68}$$

which is equivalent to the force-free equation

$$\nabla \times \mathbf{B} = \alpha \mathbf{B}. \tag{4.69}$$

In other words, if we obtained the steady state of (4.63) the curl of this solution $\mathbf{A}$ is the three-dimensional force-free magnetic field $\mathbf{B}$ with the specified normal component $\mathbf{B}_n$ on the boundary $\partial\Omega$.

To complete the numerical model the boundary conditions are given as follows. The normal component $\mathbf{B}_n$ of the magnetic field is set to be zero on the side and the top boundaries of the box Ω, while at the base plane $z = z_{min}$ it is given by

$$\mathbf{B}_n = \mathbf{B}_z = C_l \left\{ \frac{W^3}{[W^2 + x^2 + (y - y_1)^2]^{3/2}} - \frac{W^3}{[W^2 + x^2 + (y - y_1)^2]^{3/2}} \right\}, \tag{4.70}$$

where C_l, W, y_1 and y_2 are constants. The boundary conditions on the tangential component of the vector potential $\mathbf{A}_t$ are derived from the above conditions on $\mathbf{B}_n$. Finn *et al.* use the fact that expression (4.70) can be integrated and set $\mathbf{A}_t = 0$ on the side and top boundaries, $\mathbf{A}_y = 0$ and

$$\mathbf{A}_x = \frac{C_l W^3}{W^2 + x^2} \left[\frac{-(y - y_1)}{\sqrt{W^2 + x^2 + (y - y_1)^2}} + \frac{(y - y_2)}{\sqrt{W^2 + x^2 + (y - y_2)^2}} \right]. \tag{4.71}$$

The disadvantage of this approach for deriving the boundary conditions on $\mathbf{A}_t$ is that for an arbitrary $\mathbf{B}_n$ the equation $\nabla \times \mathbf{A} = \mathbf{B}$ cannot be integrated exactly, so $\mathbf{A}_t$ cannot be determined analytically at the boundary. So if, for example, $\mathbf{B}_n$ is not given by a simple analytical expression but derived from such observational data as magnetograms, the problem of deriving boundary conditions for $\mathbf{A}_t$ from such $\mathbf{B}_n$ becomes more complicated. To avoid this difficulty we propose to use a more general approach that was originally developed by Hirasaki and Hellums [38] for the problems of incompressible hydrodynamics formulated in terms of velocity and vorticity fields.

To derive the boundary conditions for $\mathbf{A}$ the following question needs to be addressed:

if a normal component of a vector field $\mathbf{B}_n$ is given on a boundary $\partial\Omega$ of some space region Ω and $\mathbf{A}$ is a vector potential field, such that

$$\mathbf{B} = \nabla \times \mathbf{A} \text{ in } \Omega, \tag{4.72}$$

$$\nabla \cdot \mathbf{A} = 0 \text{ in } \Omega, \tag{4.73}$$

$$\hat{n} \cdot \nabla \times \mathbf{A} = \hat{n} \cdot \mathbf{B} = \mathbf{B}_n \text{ on } \partial\Omega, \tag{4.74}$$

what should the conditions on $\mathbf{A}$ be at the boundary $\partial\Omega$? Hirasaki and Hellums developed a general form of the boundary conditions on $\mathbf{A}$ and proved that a vector potential field $\mathbf{A}$ that satisfies these conditions yields, in fact, a unique vector potential determined by (4.72)-(4.74). Suppose that $\partial\Omega$ consists of faces S_i with the boundary edges C_i. Note that the boundary conditions for $\mathbf{A}$ have to be consistent with the Coulomb gauge condition (4.73). According to [38] the tangential component $\mathbf{A}_t$ of the vector potential can be determined by defining a vector $\mathbf{G}$ on $\partial\Omega$ such that

$$\mathbf{A}_t = \nabla_t \times \mathbf{G}, \qquad (\nabla_t \times \mathbf{G} = -\hat{n} \times [\hat{n} \times [\nabla \times \mathbf{G}]]), \tag{4.75}$$

and the following conditions are satisfied:

(a) the component of $\nabla_t \times \mathbf{G}$ tangential to the edges on $\partial\Omega$ is continuous across each edge C_i,

(b) on each face S_i of $\partial\Omega$ the vector $\nabla_t \times \mathbf{G}$ satisfies

$$\int_{S_i} \hat{n} \cdot \mathbf{B} ds = \oint_{C_i} (\nabla_t \times \mathbf{G}) \cdot dl. \tag{4.76}$$

As proven in [38] a necessary and sufficient condition for the existence of a vector $\mathbf{G}$ that satisfies conditions (a) and (b) is

$$\int_{\partial\Omega} \mathbf{B} \cdot \hat{n} ds = 0. \tag{4.77}$$

This is automatically satisfied in our case since $\mathbf{B}$ is a divergence free magnetic field. For general tensor form of the equations that determine $\mathbf{G}$ for an arbitrary region Ω we refer

the reader to [38]. In our case of a Cartesian box domain the conditions are simplified and $\mathbf{G}$ has only a normal component $\mathbf{G}_n$, while $\mathbf{G}_t = 0$. Then we can calculate the tangential component $\mathbf{A}_t = \nabla_t \times \mathbf{G}_n$ at each of the boundary faces S_i, $i = 1, \ldots, 6$ from a set of problems

$$\nabla_t^2 \mathbf{G}_n = -\mathbf{B}_n \text{ on } S_i \tag{4.78}$$

together with the conditions (a), (b) on the edges of S_i. Equation (4.78) is derived as follows

$$\nabla \times \mathbf{A}_t = \nabla_t \times (\nabla_t \times \mathbf{G}_n) = \nabla_t \nabla_t \cdot \mathbf{G}_n - \nabla_t^2 \mathbf{G}_n = \mathbf{B}_n. \tag{4.79}$$

To ensure the zero-divergence condition for $\mathbf{A}$ at the boundary, the condition (4.75) on the tangential component $\mathbf{A}_t$ is complemented by the following condition on the normal component $\mathbf{A}_n$ of the magnetic potential

$$\frac{\partial \mathbf{A}_n}{\partial n} = 0 \tag{4.80}$$

at each face S_i. In general, the set of problems (4.78) and conditions (a), (b) has to be discretized and solved simultaneously as a large matrix system. For our calculations, however, we simplify the procedure using the fact that we can satisfy condition (a) by requiring that $\mathbf{A}_t = 0$ on the edges C_i of the Cartesian box domain Ω and since we set $\mathbf{B}_n = 0$ on all the boundaries except the base plane, where it is given by (4.70), condition (b) is automatically satisfied. Then the boundary condition on $\mathbf{A}_n$ is defined by (4.80) and the tangential component $\mathbf{A}_t$ is given by $\mathbf{A}_t = \nabla_t \times \mathbf{G}_n$, where $\mathbf{G}_n$ is calculated by solving 6 Neumann problems for each face of the box Ω:

$$\nabla_t^2 \mathbf{G}_n = -\mathbf{B}_n, \tag{4.81}$$

$$\frac{\partial \mathbf{G}_n}{\partial n} = 0. \tag{4.82}$$

The initial vector potential field is taken to be $\mathbf{A} = 0$.

Since the equation

$$\frac{\partial \mathbf{A}}{\partial t} = \nabla^2 \mathbf{A} + \alpha \nabla \times \mathbf{A}, \tag{4.83}$$

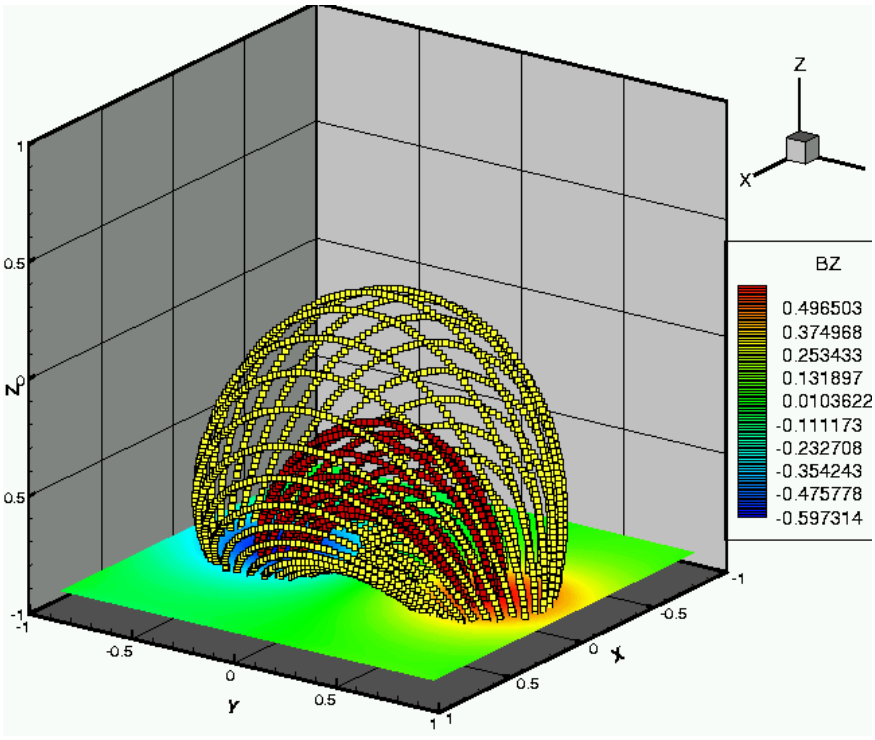
is linear we use the linear exponential propagation method described in Section 2.4 with the Jacobian matrix A derived from discretizing the spatial operator on the right hand side of (4.83). This is a steady state problem, so we are not interested in accurately calculating the evolution of $\mathbf{A}$ and only want to know the topology of the final configuration of $\mathbf{B}$ such that

$$\mathbf{J} = \nabla \times \mathbf{B} = \alpha \mathbf{B}. \tag{4.84}$$

In fact we can check whether our solution is accurate by computing $\mathbf{J}$ and checking whether (4.84) holds. Therefore, we use only first order upwind discretization in space for the term $\alpha \nabla \times \mathbf{A}$ and discretize $\nabla^2 \mathbf{A}$ using standard second-order centered finite differences. For each value of α we want to calculate the force-free field, so we evolve equation (4.83) to a steady state, compute the fields $\mathbf{B}$ and $\mathbf{J}$ and study their topology by plotting the field lines. Recall that we expect stable solutions to exist as long as $\alpha^2 \leq \lambda_1^2$, where λ_1 is the eigenvalue of the force-free equation with the smallest magnitude. Finn *et al.* studied the case when the parameters in (4.70) are given as $W = 0.4$, $C_a = 0$, $C_l = 1$, $y_1 = 0.5$, $y_2 = -0.5$, $x_{min} = y_{min} = z_{min} = -1$ and $x_{max} = y_{max} = z_{max} = 1$ for different values of α. Since we used a different form of the boundary conditions for the problem we do not expect exact agreement with the results in Finn *et al.*. Our calculations, however, resembled very closely the solutions shown in [28]. Figures 4.12-4.16 display the results of the computations with α taking the values 0, 2, 2.8, 3.1, 3.65. In all of the figures the contour plot of the z-component of the magnetic field is displayed at the base plane and the selected field lines are shown in different colors. As in [28] we also found that as α increases, the magnetic field lines corresponding to the solutions of (4.83) become more twisted and for the values of α close to 3.7 knots start to develop.

Figures 4.12(a),(b) show the side and top view of the field lines of a potential magnetic field, which is a solution to (4.83) when $\alpha = 0.0$. To demonstrate that the solutions are in fact force-free we plot the field lines for the magnetic field $\mathbf{B}$ and the current density

a)



b)

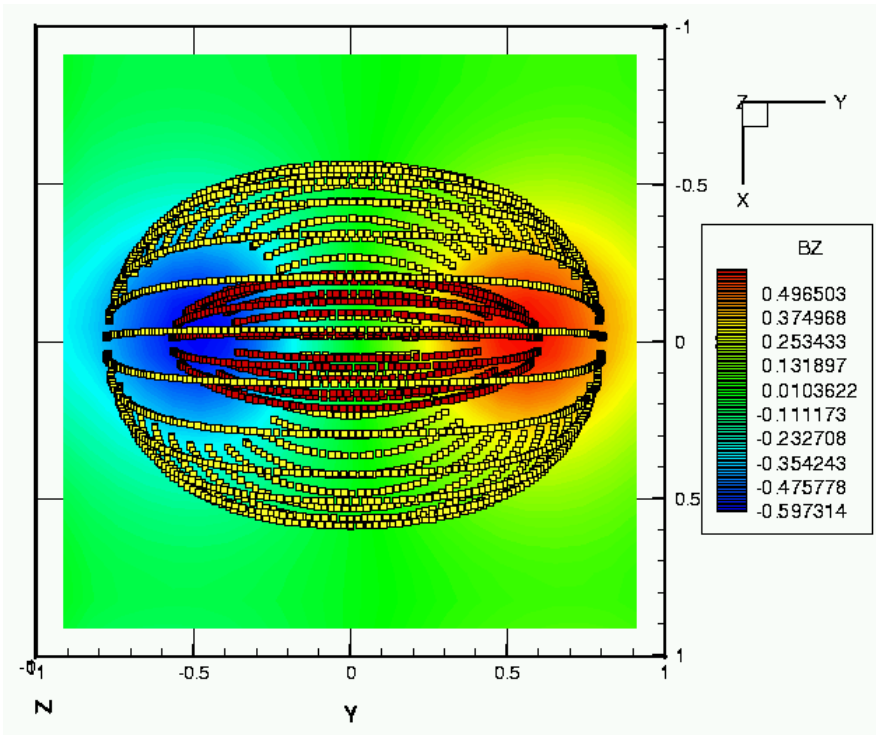


Figure 4.12: The side (a) and the top (b) view of the field lines of the force-free magnetic field **B** corresponding to $\alpha = 0.0$

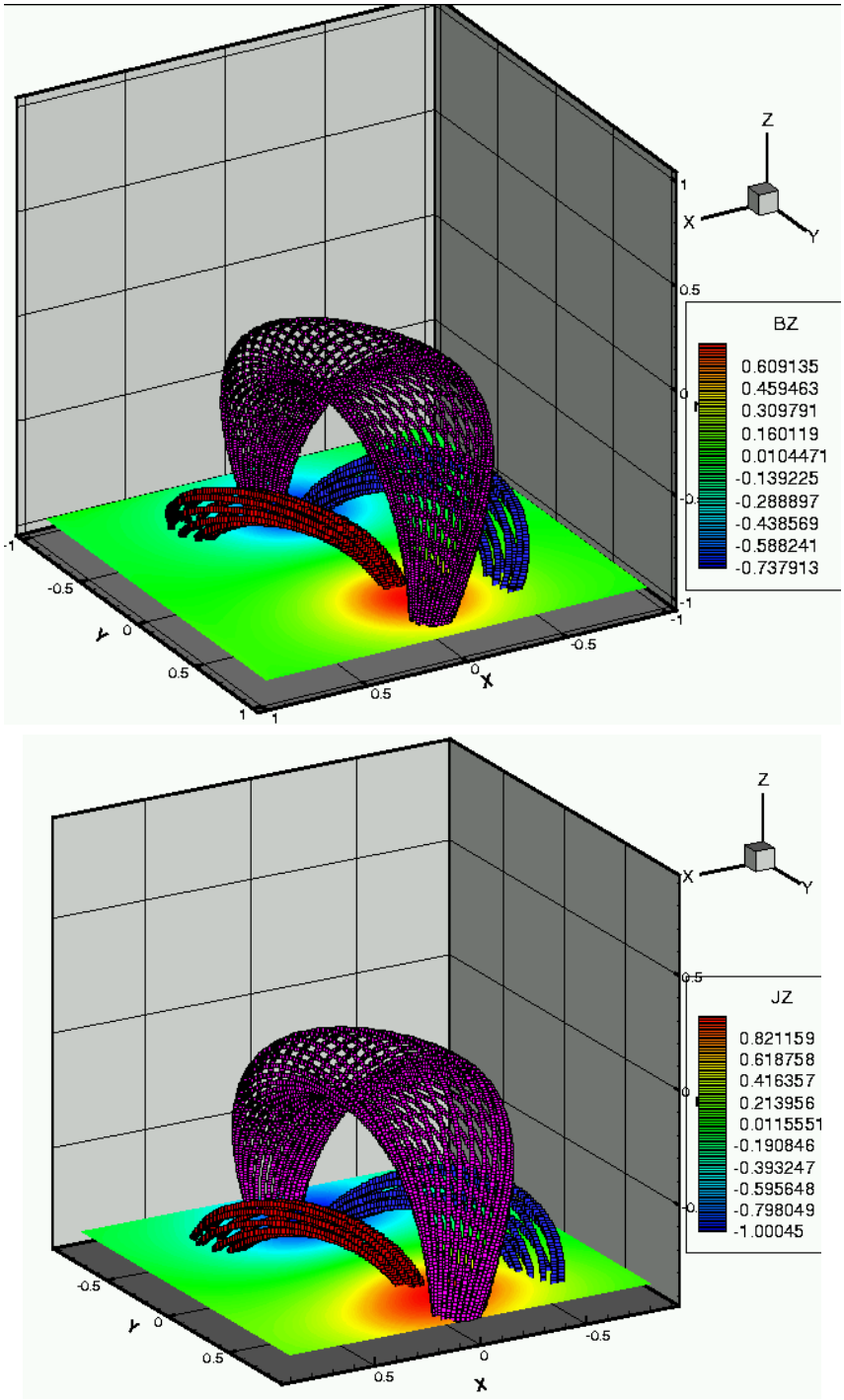
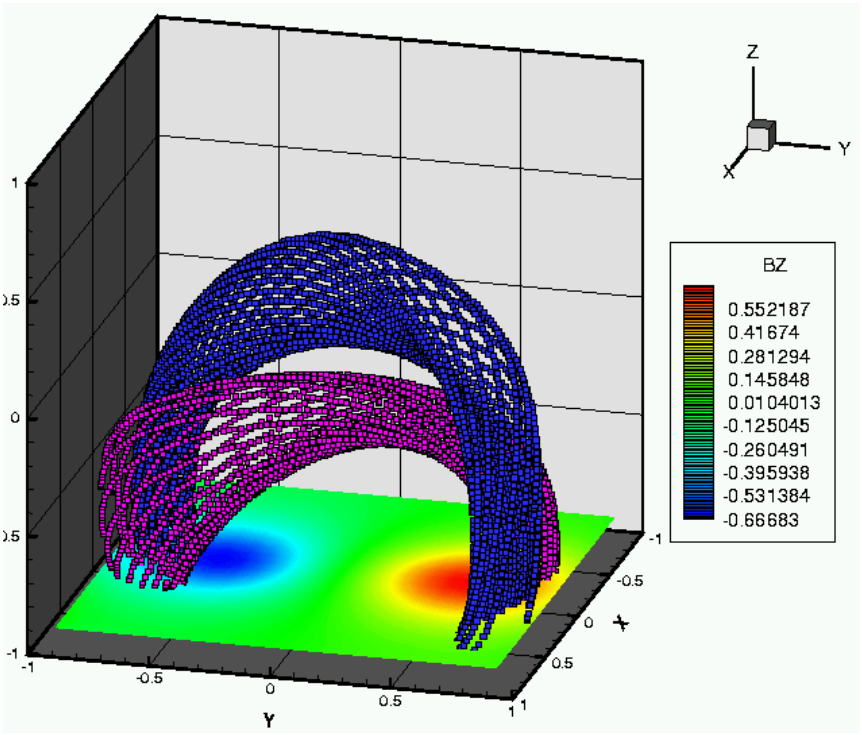


Figure 4.13: The field lines of the force-free magnetic field **B** (above) and the current density **J** (below) corresponding to $\alpha = 2.0$

a)



b)

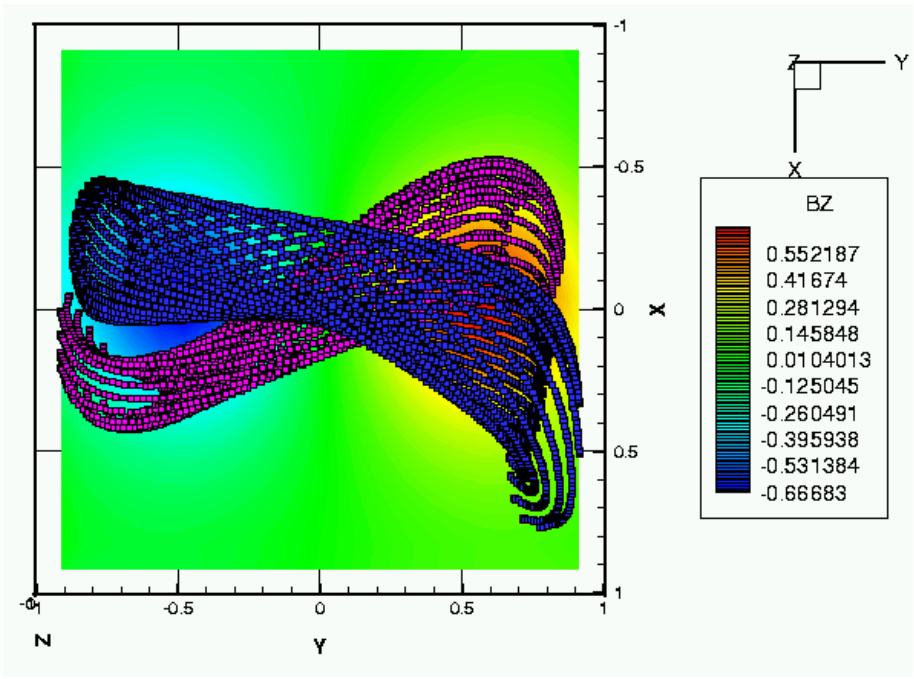
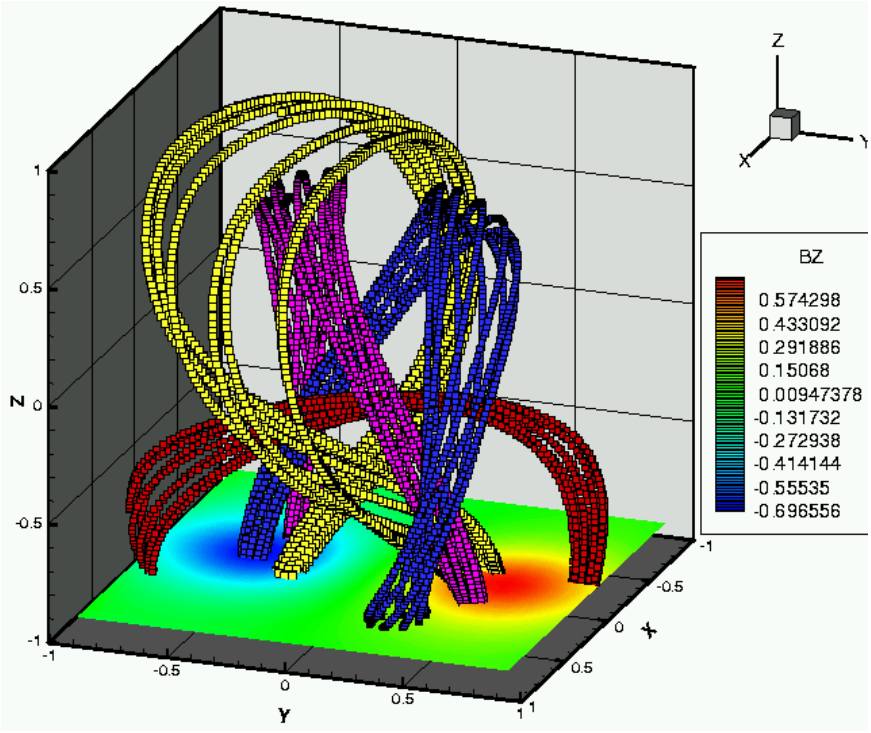


Figure 4.14: The side (a) and top (b) view of the field lines of the force-free magnetic field **B** for $\alpha = 2.8$

a)



b)

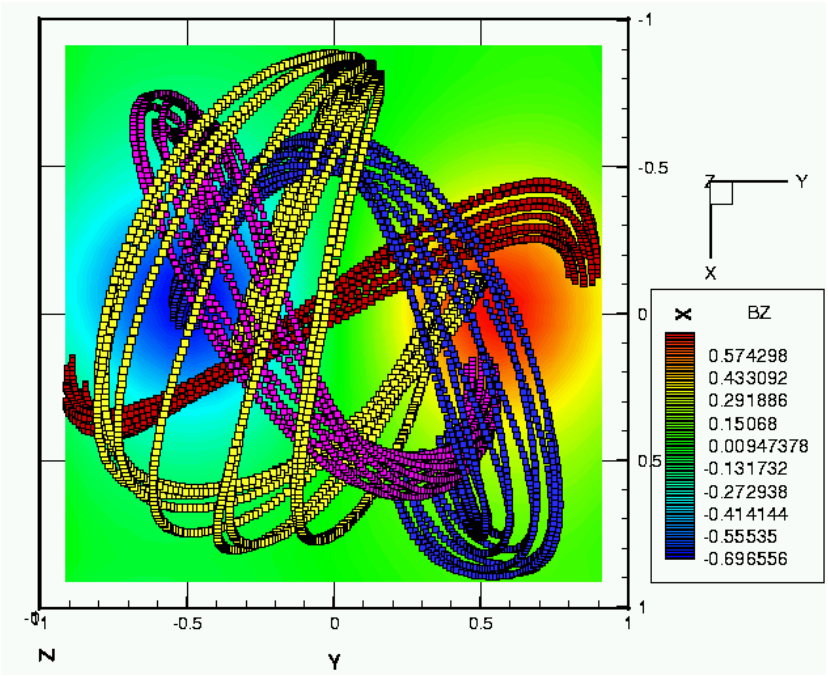
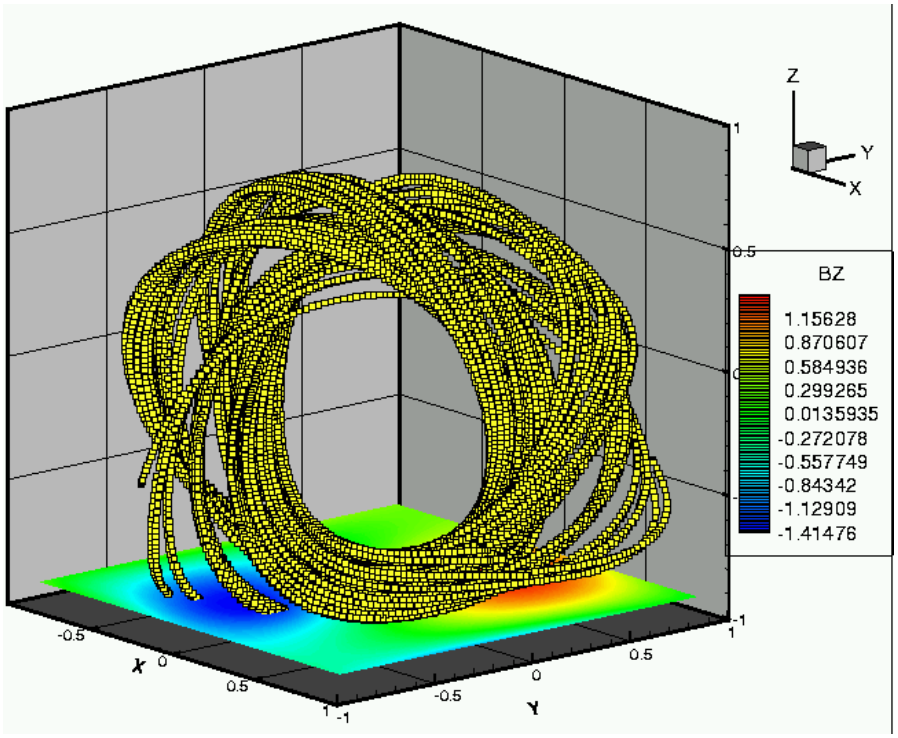


Figure 4.15: The side (a) and top (b) view of the field lines of the force-free magnetic field **B** for $\alpha = 3.1$

a)



b)

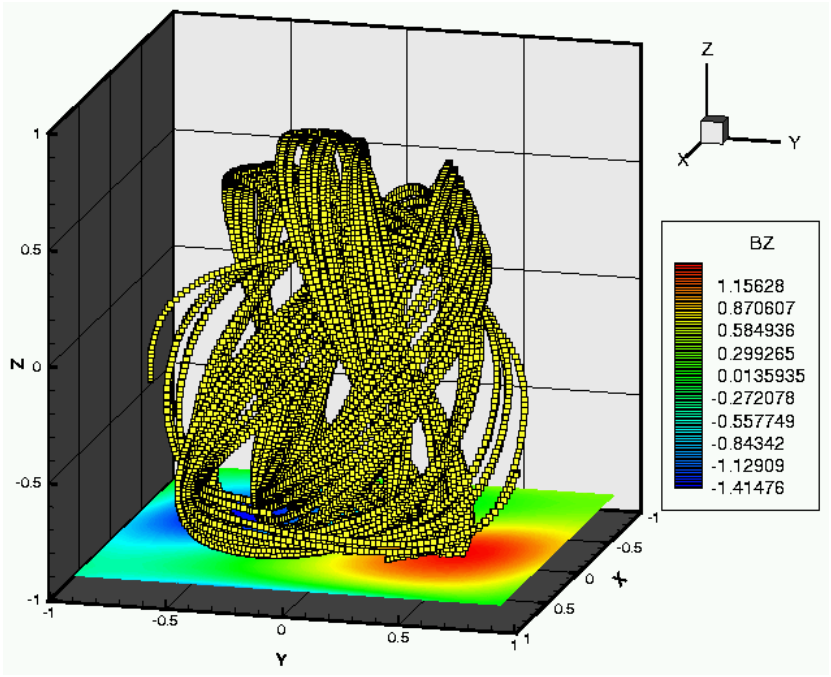
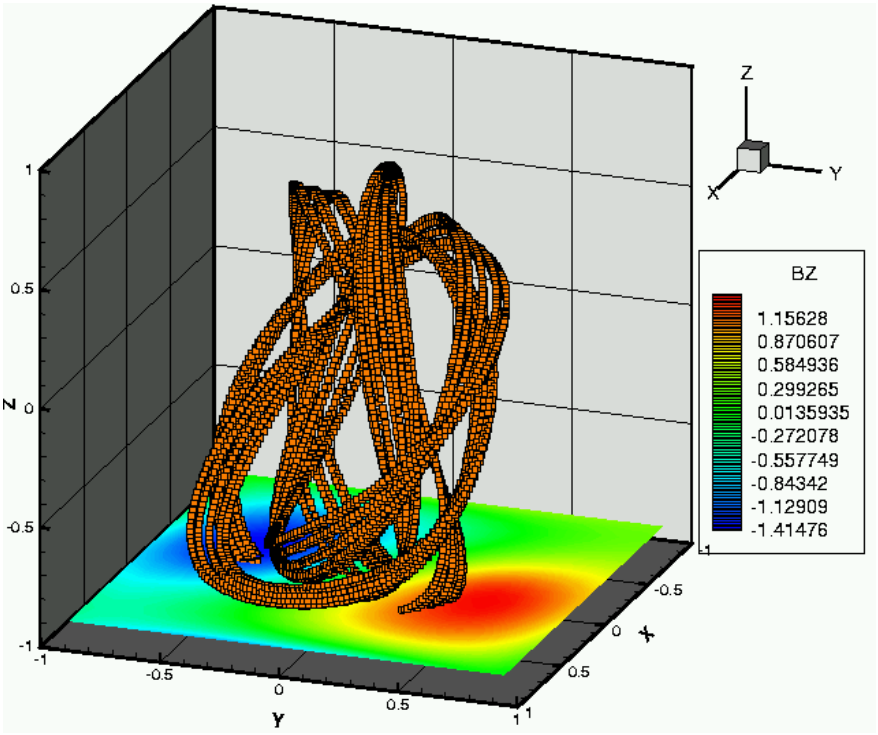


Figure 4.16: The different angle views of the field lines of the force-free magnetic field **B** for $\alpha = 3.65$

a)



b)

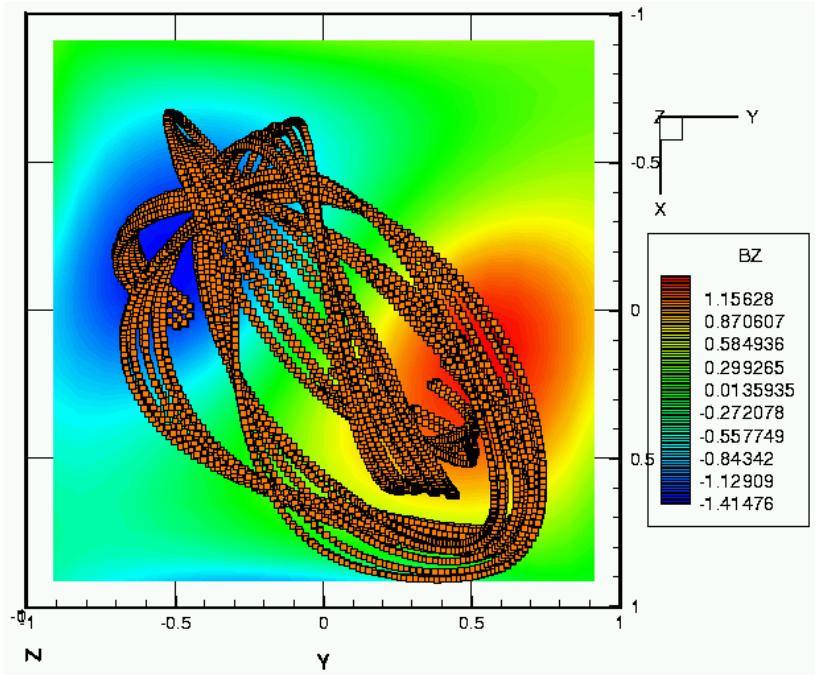


Figure 4.17: The side (a) and top (b) view of the field lines of the force-free magnetic field **B** for $\alpha = 3.65$

$\mathbf{J} = \nabla \times \mathbf{B}$ in figures 4.13(a),(b) for $\alpha = 2.0$. As evident from these figures, $\mathbf{B}$ is in fact parallel to $\mathbf{J}$, so that $\mathbf{J} \times \mathbf{B} = 0$. As α gets larger more twist is introduced into the topology of the magnetic field lines. The low lying magnetic field lines start developing a characteristic S-shape (purple flux tube in Fig. 4.14(b)), which is frequently observed on the Sun in the pre-eruption plasma configurations. Such structures are called sigmoids. Another feature appearing with the increase of α is the upwards expansion of the field lines at higher altitudes. While our calculations agree almost identically with the images obtained in [28] for $\alpha \leq 3$, for larger values of α our results differ somewhat from Finn *et al.*, which is probably due to the difference in boundary conditions of the two models. The images in [28] indicate that even for $\alpha \geq 3.1$ the magnetic field lines become knotted. In our calculations the knots do not appear for $\alpha \lesssim 3.5$. In fact the solutions for the values of α smaller than those corresponding to the knotted field lines, e.g. Fig. 4.15(a)-(b), lead to an important observation: while the low lying field lines become S-shaped (see the top view of a red flux tube on Fig. 4.15(b)), the field lines at higher altitudes form large arcades and loops that overlay the sigmoids. As seen from the top view these arcades are positioned perpendicular to the sigmoids, together forming configurations that are seen in the observations [75]. In fact we will see that topologically the force-free solutions of (4.83) for $3.0 \lesssim \alpha \lesssim 3.5$ resemble the magnetic field at certain stages of the evolution of the magnetic arcades in response to the footpoint motion when we discuss our results of the dynamical three-dimensional simulations in the next chapter. Note also that the neutral line of the $\mathbf{B}_z$ component at the base plane gets tilted, this is also a feature that matches the observational data, in particular, the results of the vector magnetograms of the magnetic field normal to the surface of the Sun.

As α gets larger than 3.7 the stable equilibrium solutions cease to exist. The twist and knottedness of the fields lines keep increasing in the course of evolution without ever settling to a steady state. From the results obtained by this model we could conjecture that if plasma evolves through a series of force-free states as it reaches the states with more twisted field lines in the presence of small resistivity a reconnection process would follow. This will prevent the knots from forming in the dissipative plasma and will cause the plasmoid like structures to be formed and ejected as in the axisymmetric case.

Let us now discuss the performance of the exponential method for the problem in this section. Due to the discretization we chose for the spatial operators the resulting Jacobian

matrix for (4.83) was symmetric-positive definite. As predicted by the theory the method proved to be unconditionally stable in this case and the time step size was limited only by accuracy requirements. Since for this problem we were only interested in the final steady state solution to the equation (4.83) and not the intermediate solutions, we took a very large time step in order to obtain the equilibrium as fast as possible. While in this subsection we focused on the problem of computing the constant-α force-free solutions of (4.40), it is possible to use a similar approach and set up pseudo-evolutionary equations to compute the nonlinear non-constant-α force-free states as, for example, was done in [47]. In general, it is important to be able to calculate the non-linear non-constant-α states from the boundary data on the magnetic field $\mathbf{B}$, since they seem to model the observed in coronal plasma structures better than the linear force-free solutions. One application of the non-linear force-free problem is reconstructing the structure of a quiet corona from the magnetograms that measure the magnetic field at its base. For a review of the current research on this problem and a description of different numerical methods used to solve it see [54, 3]. We would like to note that the stability properties of the exponential propagation methods can overcome some difficulties associated with the numerical solution of this problem and help to efficiently compute the force-free states by evolving a pseudo-evolution equation with a large time step. We plan to investigate this further in our future work.

Chapter 5 Three-dimensional numerical modeling of the evolution of magnetic arcades in solar and laboratory plasmas

5.1 Motivation for the study

As discussed in the previous chapter, coronal mass ejections (CMEs) are massive eruptive events that make a drastic impact on space weather. The effect of these sudden ejections of a billion tons of plasma from the solar atmosphere into the solar wind changes the geomagnetic environment of the Earth and can adversely affect and even damage critical systems such as communication devices, power grids, or navigational tools. While predicting these eruptive events is very important for the welfare of many public and commercial infrastructures, the present understanding of the cause and evolution of CMEs and the capabilities for predicting of these events is very limited.

The magnetic structures involved in the eruption have topologically complex three-dimensional structure. The physical processes responsible for the evolution of the pre- and post-eruptive magnetic arcades have very involved mathematical descriptions that do not yield simple solutions in most cases. Other obstacles in understanding the CMEs are the limitations of the observations. The corona is optically thin and it is very difficult to deduce the 3D pre-eruption configurations from the two-dimensional images obtained by observational instruments. The coronal regions lying below a CME magnetic arcades, are very important in the evolution of the eruptive magnetic arcades, and yet are inaccessible to observations altogether. Due to all these complications one has to dissect the problem of modeling the CMEs and study it in simpler parts with the appropriate analytical or experimental tools. While we briefly discussed some of the existing analytical models in the previous chapter, a more detailed review of such research can be found in [46].

An interesting and promising approach to modeling the eruption of the magnetic arcades has been undertaken by Bellan and Hansen [65] at Caltech. They used modified spheromak technology to create plasma configurations having the geometry similar to solar prominences

Figure 5.1: A photograph of a prominence-like configuration produced in a laboratory experiment using the modified spheromak technology (image courtesy of J.F. Hansen and P.M. Bellan)

(Fig. 4.1,4.4, 5.1). To simulate the geometry and boundary conditions of solar prominences the experiment uses a large (1.4 m in diameter) vacuum chamber. The simulated prominence is created between the magnetic poles of a horseshoe electromagnet. The plane of the pole faces (D-shaped plates in Fig. 5.1) corresponds to the ground plane of the solar surface and the horseshoe magnet vacuum magnetic field emulates the vacuum magnetic field of the Sun. A large capacitor is then connected to create a potential difference between the horseshoe magnet poles. This causes currents to flow in the plasma and to generate a self-magnetic field which twists up the background magnetic field lines. Thus the twisted prominence is simulated and its evolution is observed over the duration of the experiment. Fig. 5.2 demonstrates a typical evolution of the laboratory prominence over a time period of 4.5 μs, which in this parameter regime is equivalent to two hundred Alfven times. The evolution of plasma in the experiment has been modeled analytically as a sequence of the linear force-free solutions of the equation

$$\nabla \times \mathbf{B} = \alpha \mathbf{B}, \tag{5.1}$$

obtained using an expansion of $\mathbf{B}$ in terms of Bessel and trigonometric functions [65]. While

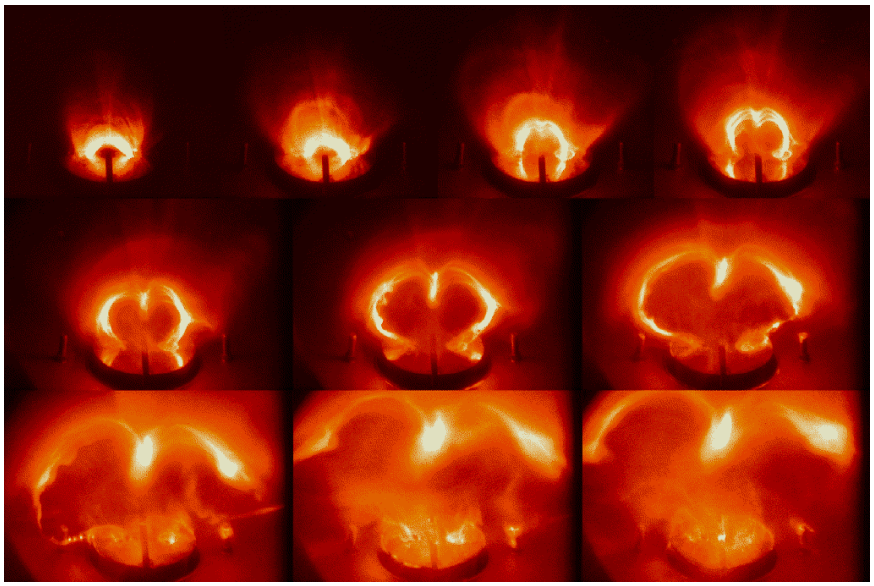


Figure 5.2: Photographs displaying a typical evolution of the laboratory prominence over 4.5 μs (image courtesy of J.F. Hansen and P.M. Bellan

these analytical solutions can capture some large scale trends in the evolution, such analysis has limitations in describing the dynamics of the plasma. For example, the boundary distribution of these analytical force-free solutions have specific structure corresponding to the base functions of the expansion, i.e. Bessel and trigonometric functions in this case. The actual boundary conditions in the experiment correspond to the dipole magnetic field created by the horseshoe magnet. Additionally, the plasma in the experiment is not ideal, and so the small resistive effects can change the topology of the magnetic field lines significantly as discussed in the previous chapter. In general, a model that better describes the plasma and captures the continuous evolution and the topology of the magnetic field lines would be more desirable. The mathematical description of such a time-dependent model is too complicated, however, to allow an analytical solution.

The above considerations lead us to the conclusion that the three-dimensional numerical models of CMEs can provide much needed and important insights into the dynamics of these events. If such models are in agreement with the observational data and experimental data they can greatly augment these with information about the small scale structure of the plasma configurations, the overall topology of the magnetic fields and more detailed information about the evolution of the coronal magnetic structures. In this chapter we will present a numerical model that models the evolution of the magnetic arcades that have

the geometry and the plasma parameter values that are close to those of the solar and experimental configurations. We study the results of the simulations and try to compare them to observations of the evolution of solar structures and laboratory experiments of prominence eruptions.

5.2 Numerical three-dimensional model description

5.2.1 Equations

Since we are interested in large-scale topology and stability of the magnetic plasma configurations, we employ once again the theory of resistive magnetohydrodynamics for our three-dimensional model. In the prominence eruption experiment as well as in the solar corona, magnetic forces dominate plasma dynamics, significantly exceeding other influences such as pressure gradients, gravity and thermal effects. The ratio of the hydrodynamic to the magnetic pressure, i.e. the plasma β defined by (4.1), is small in both the experiment ($\backsim 10^{-3} - 10^{-4}$) and the corona ($\backsim 10^{-4}$ [68]). Therefore, just as we did for the two-dimensional case of Chapter 4, we use the zero β approximation for our 3D calculations. This implies that we do not evolve hydrodynamic pressure and eliminate the pressure gradient term from the momentum equation. Another assumption we make is the uniform plasma density approximation. To justify this approximation, we repeat the argument presented in section 4.2 and argue that the dynamics of the interaction of the magnetic and velocity fields will contribute most to the overall topology of the magnetic field and a constant density profile could be used as the first approximation in the model. As was shown in several numerical studies [55, 5, 8] that investigated zero β plasmas, prescribing different nonuniform density profiles did not significantly alter the dynamics or the topology of the magnetic field. We want to emphasize that the above approximations are made to simplify the very complex system of resistive MHD and determine the basic evolution trends in the magnetic field. Future studies will enhance the system and include the effect of pressure gradients and nonuniform density.

As shown in section 4.2, the above assumptions and the nondimensionalization of vari-

ables in the resistive MHD equations leads to the following system:

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{V} \times \mathbf{B}) + \frac{1}{S}\nabla^2 \mathbf{B}, \tag{5.2}$$

$$\frac{\partial \mathbf{V}}{\partial t} = -\mathbf{V} \cdot \nabla \mathbf{V} + (\nabla \times \mathbf{B}) \times \mathbf{B} + \frac{1}{R}\nabla^2 \mathbf{V}, \tag{5.3}$$

where $\mathbf{B}$ is the magnetic field vector, $\mathbf{V}$ is the velocity, S is the Lundquist number and R is the Reynolds number. The nondimensional parameters S and R represent the ratio of the resistive to Alfven time scales and the ratio of the viscous to Alfven time scales respectively, and are defined as

$$S = \frac{l_0^2/\eta}{\tau_A}, \tag{5.4}$$

$$R = \frac{l_0^2/\nu}{\tau_A}, \tag{5.5}$$

where l_0 is a characteristic spatial scale for the configuration, η is the magnetic diffusivity, ν is the coefficient of kinematic viscosity and $\tau_A = l_0/V_A$ is the Alfven time defined using the Alfven velocity V_A, the average strength of the magnetic field B_0, the density ρ and the magnetic permeability of vacuum μ_0 as

$$V_A = \frac{B_0}{(\mu_0 \rho)^{1/2}}. \tag{5.6}$$

We numerically solve equations (5.2), (5.3) in Cartesian coordinates in a rectangular domain

$$\Omega = \{(x, y, z) : |x| \leq L_x, |y| \leq L_y, 0 \leq z \leq L_z\}, \tag{5.7}$$

with initial and boundary conditions described below.

5.2.2 Initial and boundary conditions

At the beginning of the simulation we take $\mathbf{B}$ to be a current-free, or potential, bipolar magnetic field. This corresponds to a coronal magnetic arcade in which plasma aligns itself with the strong magnetic field of the Sun. In the experimental setting this initial configuration represents the moment in the prominence formation when the gas is fully

ionized and the current starts flowing in plasma between the poles of the electrodes. From potential theory we know that a field $\mathbf{B}$ that satisfies conditions

$$\nabla \times \mathbf{B} = 0, \tag{5.8}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{5.9}$$

can be expressed as the gradient of some potential function Φ, i.e.

$$\mathbf{B} = \nabla \Phi. \tag{5.10}$$

If we specify the value of the normal component $\mathbf{B}_n$ of the magnetic field on the boundary $\partial\Omega$ then a unique field $\mathbf{B}$ can be obtained from (5.10) by solving a three-dimensional Neumann problem for the Laplace equation

$$\nabla \cdot (\nabla \Phi) = 0, \tag{5.11}$$

$$\frac{\partial \Phi}{\partial n} = \mathbf{B}_n. \tag{5.12}$$

In general an arbitrary distribution of the normal component $\mathbf{B}_n$ on $\partial\Omega$ can be prescribed as long as the total flux of $\mathbf{B}$ over this boundary vanishes. For example, the value of $\mathbf{B}_n$ might come from magnetogram data. In this case Equations (5.11)-(5.12) has to be solved numerically. These systems of equations are encountered in many scientific fields, so a large number of numerical methods exist for solving it. We could, for example, use a simple conjugate gradient scheme or a multigrid technique depending on which spatial discretization of the Laplacian operator we wish to use. We will not concentrate on this problem since the discussion of it can be found in any standard numerical analysis textbook. Since we want $\mathbf{B}$ to have a potential bipolar structure we can avoid introducing numerical errors associated with solving (5.11)-(5.12) computationally and prescribe Φ to be a potential of a dipole located below the base plane of the rectangular domain Ω. A magnetic dipole $\overrightarrow{m}$ located at $\overrightarrow{r_0}$ produces a magnetostatic potential

$$\Phi(\overrightarrow{r}) = -\frac{\overrightarrow{m} \cdot (\overrightarrow{r} - \overrightarrow{r_0})}{|\overrightarrow{r} - \overrightarrow{r_0}|^{3/2}}. \tag{5.13}$$

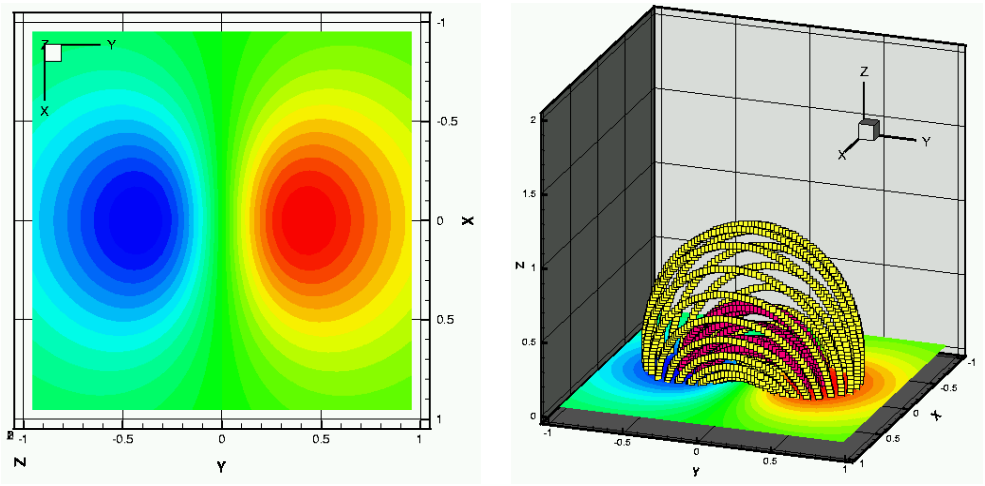


Figure 5.3: Initial magnetic field $\mathbf{B}$ resulting from a dipole placed below the base plane $z = 0$. The contour plot of the $\mathbf{B}_z$ component at $z = 0$ (left) and the magnetic field lines of the potential $\mathbf{B}$ (right).

We choose the location of a dipole to be a point $(0, 0, z_0)$ and the dipole moment to be aligned with the y-axis so that $\overrightarrow{m} = m(0, 1, 0)$. In this case the magnetic field calculated from (5.10) is

$$B_x(x, y, z) = m\frac{3xy}{[x^2 + y^2 + (z - z_0)^2]^{5/2}} \tag{5.14}$$

$$B_y(x, y, z) = m\frac{-x^2 + 2y^2 - (z - z_0)^2}{[x^2 + y^2 + (z - z_0)^2]^{5/2}} \tag{5.15}$$

$$B_z(x, y, z) = m\frac{3y(z - z_0)}{[x^2 + y^2 + (z - z_0)^2]^{5/2}}. \tag{5.16}$$

At time $t = 0$ the velocity field $\mathbf{V}$ is chosen to be zero, i.e. we start simulation with a static potential dipole magnetic field configuration. We want the maximum value of the magnitude of the nondimensionalized magnetic field $\mathbf{B}$ in the domain Ω initially to be 1, which for the dimensional variable corresponds to the value B_0. To achieve this we choose parameters in (5.14)-(5.16) to be $z_0 = -0.8$ and $m = 0.6$. This choice also defines the characteristic length scale of the system because it determines the separation of the magnetic poles in nondimensional spatial variables that in real variables corresponds to the parameter l_0. For the prominence eruption experiment this means that an electrode pole separation of 81 mm corresponds to an interval of length 0.8 in our nondimensionalized Cartesian box. Figure (5.3) shows the contour plot of the B_z component at the base plane $z = 0$ and the magnetic field lines for the initial magnetic field for the specified parameters.

There are two versions of boundary conditions on the magnetic field that we use in the simulations presented in this chapter. In the experiment, the ambient potential magnetic field is the strongest component of $\mathbf{B}$ and the vacuum chamber is much larger than the initial plasma configuration. These considerations suggest that we can just impose a Dirichlet boundary condition on the magnetic field with $\mathbf{B}(x,y,z,t)|_{\partial\Omega} = \mathbf{B}(x,y,z,0)|_{\partial\Omega}$. This condition, however, does not implicitly ensure that $\nabla \cdot \mathbf{B} = 0$. In general, the influx of the divergence from the boundaries is possible if the Dirichlet boundary conditions are set on $\mathbf{B}$. This leads us to reformulate the boundary conditions on $\mathbf{B}$ as follows. Let us prescribe the tangential component $\mathbf{B}_t$ of the magnetic field at the boundary $\partial\Omega$. Note that this condition fixes the normal component of the current $\mathbf{J}_n$ since

$$\mathbf{J}_n = \nabla \times \mathbf{B}_t. \tag{5.17}$$

If $\mathbf{B}_n$ is the normal component of $\mathbf{B}$ and $\partial/\partial n$ is a derivative normal to the boundary, the divergence-free condition can be written as

$$\frac{\partial \mathbf{B}_n}{\partial n} + \nabla \cdot \mathbf{B}_t = 0. \tag{5.18}$$

Therefore if we specify the Neumann boundary condition on the normal component as

$$\frac{\partial \mathbf{B}_n}{\partial n} = -\nabla \cdot \mathbf{B}_t, \tag{5.19}$$

it will ensure that the divergence is maintained zero at the boundary. So the second version of the boundary conditions that we use in the simulations is giving the Dirichlet boundary conditions on the tangential component $\mathbf{B}_t$ as

$$\mathbf{B}_t(x,y,z,t)|_{\partial\Omega} = \mathbf{B}_t(x,y,z,0)|_{\partial\Omega}, \tag{5.20}$$

while maintaining the Neumann condition (5.19) on the normal to the boundary component $\mathbf{B}_n$. We conducted our simulations with both versions of the boundary conditions and discovered that the overall dynamics does not change drastically. We will present these comparisons later in this chapter.

One of the major limitations of the observational data is the lack of reliable and conclu-

sive measurements of the plasma velocity at the boundary of the corona and the photosphere. Therefore, in order to formulate the boundary condition for the velocity, we rely on the set-up for the prominence eruption experiment. First, we impose the condition that there is no flow of plasma through the boundary, i.e. the normal component $\mathbf{V}_n$ of the velocity at the boundary $\partial\Omega$ is zero. Then we also assume that the side boundaries are removed far enough and the plasma at those boundaries remains relatively undisturbed at the end of the simulation, so that the tangential component $\mathbf{V}_t$ at the sides of the box Ω is also set to zero. To specify the tangential component $\mathbf{V}_t$ at the base boundary $z = 0$ we will examine the electric field at this plane and derive the condition on $\mathbf{V}_t$ using Ohm's law. First, we ignore the tangential currents along the surface of the electrode plates and assume that at the base plane the current is normal to the boundary, i.e. $\mathbf{J}(x, y, 0, t = 0) = (0, 0, J_z(x, y, 0))$. Then if $\mathbf{E}$ is the electric field, we can write Ohm's law at the base boundary for the tangential components of $\mathbf{E}$ as

$$E_x = V_z B_y - V_y B_z, \tag{5.21}$$

$$E_y = V_x B_z - V_z B_x. \tag{5.22}$$

There is no flow of plasma through the boundary so $V_z(x, y, z = 0) = 0$. Using Ohm's law (5.21)-(5.22) we can express the tangential components of $\mathbf{V}$ at $z = 0$ as

$$V_x = \frac{E_y}{B_z}, \tag{5.23}$$

$$V_y = -\frac{E_x}{B_z}. \tag{5.24}$$

Now considering the experimental setup it is natural to assume that the electric field at the base boundary is determined by some electrostatic potential ϕ created on the electrode plates by the discharge of the capacitor bank, i.e.

$$\mathbf{E} = -\nabla\phi. \tag{5.25}$$

Therefore, we can write the tangential velocity at $z = 0$ as

$$V_x = -\frac{\partial\phi/\partial y}{B_z}, \tag{5.26}$$

$$V_y = \frac{\partial\phi/\partial x}{B_z}. \tag{5.27}$$

One observation that can be made from these expressions is that the first derivatives of the electrostatic potential ϕ have to take zero value at the neutral line of the B_z component of the magnetic field (i.e. where $B_z = 0$) so that velocity does not become infinite there. Since the voltage is only measured at the poles of the electrode plates and we do not know the precise distribution of the potential ϕ throughout the base plane, we can model ϕ qualitatively using an analytical function that has properties that match the experimental setup. For example, such a function can be obtained by using the expression (5.16) for B_z and defining

$$\phi(x, y, 0) = -\phi_0(B_z(x, y, 0))^3, \tag{5.28}$$

where ϕ_0 is a constant value. This choice ensures that the electrostatic potential peaks at the magnet poles and vanishes where $B_z = 0$ in order to simulate the experimental setup. We will return to the question of setting the correct value for ϕ_0, which would correspond to the experimental strength of the electrostatic potential, a little later in the discussion and for now set $\phi_0 = 1$. Figure 5.4 shows the plot of such ϕ and as we can see such definition of the electrostatic potential has the shape that we would expect the real potential to have. For ϕ defined in such a way we can calculate the tangential components $V_x(x, y, 0)$ and $V_y(x, y, 0)$ of the velocity by substituting (5.28) into the expressions (5.26) and (5.27). Figure 5.5 displays the velocity vector at the base plane obtained in this way. As we can see from this figure the velocity field corresponds to two vortices both rotating in the same direction with centers located at the poles of the magnetic poles.

In the experiment the plasma velocity is estimated to be roughly $0.01V_A$ - $0.1V_A$. Similar values for the velocity at the base plane are usually used in numerical simulations of this type [56, 5]. Since we model ϕ phenomenologically and the exact distribution of the electrostatic potential is unknown, we might as well find an analytical expression for the tangential velocity at $z = 0$ which will qualitatively match the field shown in Figure 5.5 and scale

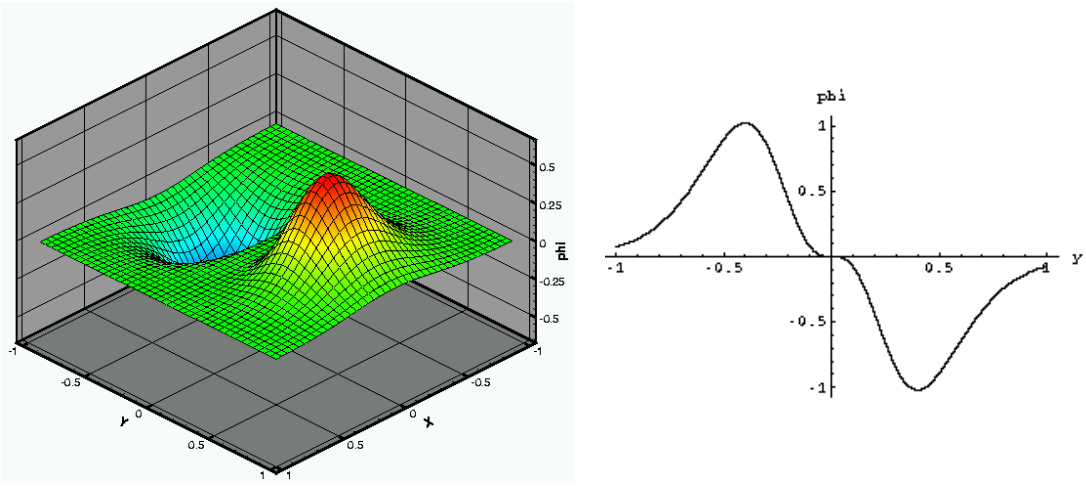


Figure 5.4: The electrostatic potential $\phi(x, y, 0) = -(B_z(x, y, 0))^3$ at the base plane $z = 0$, three-dimensional view (left) and profile $\phi(0, y, 0)$ (right).

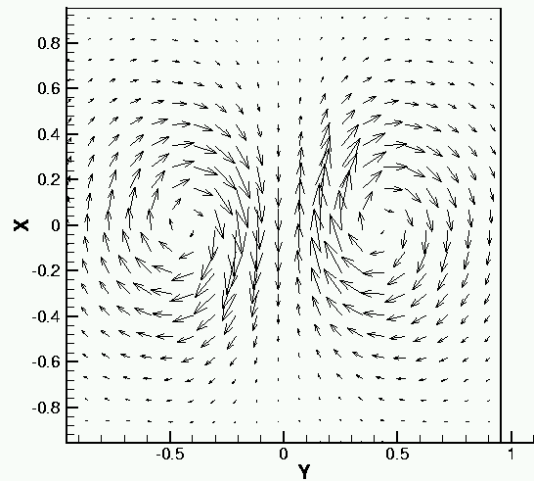


Figure 5.5: The tangential velocity field at the base plane $z = 0$ when $\phi(x, y, 0) = -(B_z(x, y, 0))^3$.

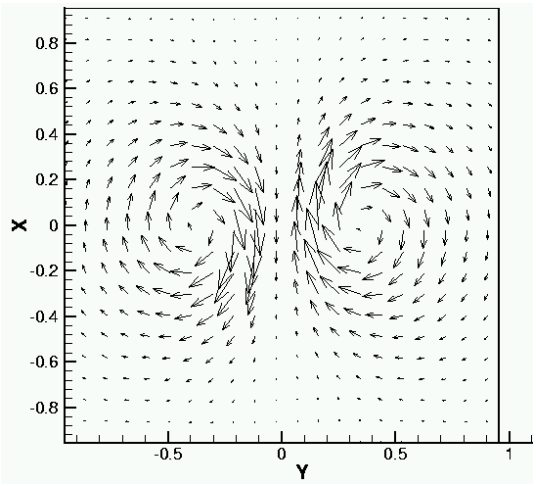


Figure 5.6: Tangential velocity field at the base plane $z = 0$ specified by expressions (5.29)-(5.31).

it according to the characteristic value of the velocity we want to set. Such an approach will simplify the parameter search to provide desirable profile of the tangential velocity and the correct magnitude. We found that if the velocity streamline function is given by the following analytical expression, these requirements are fulfilled and the correct velocity profile at the base plane is obtained as:

$$\psi(x, y) = V_0 \frac{Cy^2}{[x^2 + y^2 + C^2]^3}, \tag{5.29}$$

$$V_x(x, y, 0) = \frac{\partial \psi}{\partial y}, \tag{5.30}$$

$$V_y(x, y, 0) = -\frac{\partial \psi}{\partial x}, \tag{5.31}$$

where V_0 and C are given constants. Figure 5.6 shows the tangential velocity vector field at the base plane for $V_0 = 0.02$ and $C = 0.5$. Note that the electrostatic potential ϕ can always be recovered from a given profile of the tangential velocity by integrating equations (5.26), (5.27).

Finally, we specify the time profile of the tangential velocity at the base plane $z = 0$. In a typical simulation we ramp up the velocity linearly to its maximum value and then maintain it at that magnitude for the rest of the integration time. Using expression (5.29)

we specify the boundary conditions on the velocity at $z = 0$ as

$$V_x(x, y, 0, t) = f(t)\frac{\partial \psi}{\partial y}, \tag{5.32}$$

$$V_y(x, y, 0, t) = -f(t)\frac{\partial \psi}{\partial x}, \tag{5.33}$$

$$V_z(x, y, 0, t) = 0, \tag{5.34}$$

where

$$f(t) = \begin{cases} t/t_s & \text{if } t \leq t_s, \\ 1 & \text{if } t \geq t_s. \end{cases} \tag{5.35}$$

5.2.3 Spatial discretization, time integration scheme and implementation

The spatial discretization of the differential operators on the right hand side of equations (5.2)-(5.3) and the Dirichlet-Neumann formulation of the boundary conditions (5.19), (5.20) is accomplished using fourth-order central finite differencing on a uniform grid. The advantages of using such a discretization is the efficiency of computing the spatial differential operators compared, for example, to a fourth-order upwinding scheme, and the fact that computation of the Jacobian becomes straightforward in this case. We want to obtain the solution to high accuracy and since we are already using a fourth-order scheme to integrate the equations in time we avoid using second-order finite difference scheme in space to prevent degradation of accuracy in the numerical solution. The approach of using a uniform grid and central finite differences for spatial discretization has some limitations, however. Since we would like to prevent boundary effects from polluting the solution we extend the side and top boundaries as far as possible, and so the resolution requirements for the region of primary interest located near the magnet poles might require a grid size which is unnecessarily large. Another disadvantage is that such finite differencing does not automatically ensure that at the boundary the discretized equations preserve the discrete analog of the divergence-free condition on the magnetic field over time. While we are aware of these limitations and plan to improve this aspect of the code in the future (see the discussion in Chapter 6), the efficiency and simplicity in formulation provided by our spatial discretization scheme outweigh the disadvantages for the calculations presented in this work. In Section 5.3.4 we will assess the error from the spatial discretization and discuss the issues

of the numerical error in the divergence of the magnetic field and the effects of boundary conditions.

As in the two-dimensional case we use the fourth-order exponential Runge-Kutta method (3.113) with an automatic error control mechanism described in Section 3.4 for integrating equations (5.2)-(5.3) in time. For a detailed discussion of this time integration scheme and its implementation we refer the reader to Chapter 3. Here we will only briefly describe the code for the three-dimensional problem. Once again, we have vectorized the $exp4$ code from [40] and combined it with our implementation of the discretized resistive magnetohydrodynamic equations. The code runs on both Unix systems and the vector Cray SV1 supercomputer. The efficiency of the code results from the use of native BLAS libraries for each platform for the low order linear algebra operations. As mentioned before, the memory requirements of the exponential propagation methods are larger than those for conventional numerical techniques. In our numerical experiments we found that calculations involving integration of the equations for 1000 Alfven times on a $42 \times 42 \times 42$ grid could be performed on a 233 MHz Pentium II Linux workstation with 128 MBytes of memory, while the same computations with a grid of 84^3 nodes had to be performed on a supercomputer since they required memory larger than 128 MBytes. For example, if the boundary velocity is rapidly increased to a maximum magnitude of $0.02V_A$ over 134 Alfven times, the Lundquist number is $S = 4000$ and the Reynolds number is $R = 10$, then the integration of the equations on a 42^3 grid for 145 Alfven times required 90 MBytes of memory and ran on the 233 MHz Pentium II computer for 8 hours. A similar run but with a grid size of 84^3 required 700 MBytes of memory and ran on Cray SV1 vector supercomputer for 55 hours. Since the computational resources on the vector supercomputer are shared, the total computation time on the SV1 *cannot* be directly compared with the runs on a dedicated Pentium PC. These timings cannot be used as indicators of a parallel speed up of the code. Only a dedicated run on the vector supercomputer could make a fair comparison possible. Since we did not have an opportunity to perform such a study we can only indicate that for the SV1 runs we obtained a speedup of at least a factor of 15 compared to a serial PC. We plan to investigate this issue further when we implement the method for a distributed memory machine in future work. We emphasize though that since the exponential propagation methods are iterative and most computing time is spent in the routines that calculate the right hand side of the resistive MHD equations and its Jacobian, the speed-up should

be close to what is obtained for an explicit method or an iterative linear system solver. If the functions that compute discretized spatial derivatives of the system variables are implemented in an effective parallel way the code will have very good scalability on a parallel computer.

5.3 Three-dimensional simulations of the evolution of magnetic arcades under the influence of their footpoints motion

In this section we present the results of three-dimensional simulations using the model described above. First, we describe a particular simulation and compare the numerical results to the observational data. We show that the three-dimensional topology of the magnetic configurations provided by the numerical model replicates the two-dimensional projection images from coronagraph and that the dynamics of such structures in simulations agrees with the observational results. Secondly, we present numerical accuracy verification studies where we show that the boundary conditions do not seem to significantly affect the solution and compare the simulations at different grid resolutions. Finally, we describe how the simulation results change as we vary different parameters. In particular, we demonstrate the effect of changing the magnitude of the boundary velocity and the value of the Reynolds number on the numerical solution.

5.3.1 Numerical three-dimensional model of the topology of the magnetic field in a CME event: a proposed explanation for the classical three-part-structure of a CME

One of the key unanswered questions about the coronal mass ejections concerns the determination of their structure. As mentioned earlier the three-dimensional topology of the magnetic field lines in the pre-eruptive plasma configurations is difficult to determine from the two-dimensional images obtained by observational instruments. Certain characteristic features of the CME have been determined, however, and any theoretical model of these events should reproduce these properties. Almost every model to date of a CME suffers from some limitations, i.e. it will reproduce some characteristics of the eruptions, but

be inconsistent others. Hence, the quality of a model greatly depends on whether it can predict not one but many aspects of the CME. In this section we discuss several characteristics observed in the evolution of the magnetic configurations that produce CMEs and demonstrate how our numerical model reproduces these features. Figure 5.7 shows the time evolution of a typical CME. In the two pictures taken at 12:06 and 12:15 the characteristic three-part CME structure, consisting of a bright frontal loop, a dark cavity underneath and an embedded bright core, can be seen. Two standard models (Fig. 5.8) have been used to model the topology of the magnetic field lines in a CME [46] - the magnetic arcade and the flux rope geometries. The arcade model postulates an interaction between several stressed arcades in a multipolar magnetic field region (Fig. 5.9 (a)). The flux rope model has more adherents and interprets CME structure as an edge-on view of a flux rope. The flux rope based eruption scenarios typically involve a reconnection process during which the twisted part of the flux rope gets separated (Fig. 5.9(b)).

While both arcade and flux rope models can produce some of the observed CME features, they attempt to describe the configuration a posteriori, i.e. they do not show an evolutionary change from an unsheared and untwisted magnetic field to a topology resembling the observed images. Additionally, some observed features, for example, the bright embedded core of the CME, do not fit naturally with the proposed overall topologies. We will demonstrate here that our calculations do not have these limitations. We simply follow the evolution of the initially dipolar potential magnetic field in response to the motion of the footpoints and, as a result, obtain magnetic topologies consistent with the observed plasma configurations. Thus, our model gives a natural explanation of the structures seen in a real CME. Moreover, we show that to explain the structure of a CME at a stage captured in Figure 5.10, we do not need to invoke reconnection. In fact, this state of the configuration is nearly force-free and it is the persistent motion of the footpoints that causes the instability and forces the further dynamical changes of the geometry of the magnetic field lines.

The first simulation we present is performed using the following choice of parameters: the Lundquist number is taken to be close to its value in the laboratory experiment $S = 4000$, the Reynolds number is set at $R = 10$, and the velocity magnitude is increased linearly to a maximum value of $V_0 = 0.02V_A$ over the time interval from 35 to 135 Alfven times. For $t \geq 135\tau_A$ the velocity is maintained constant at V_0. Our computation is performed in a rectangular domain $\Omega = \{-1 \leq x \leq 1, -1 \leq y \leq 1, 0 \leq z \leq 2\}$ using a uniform grid with

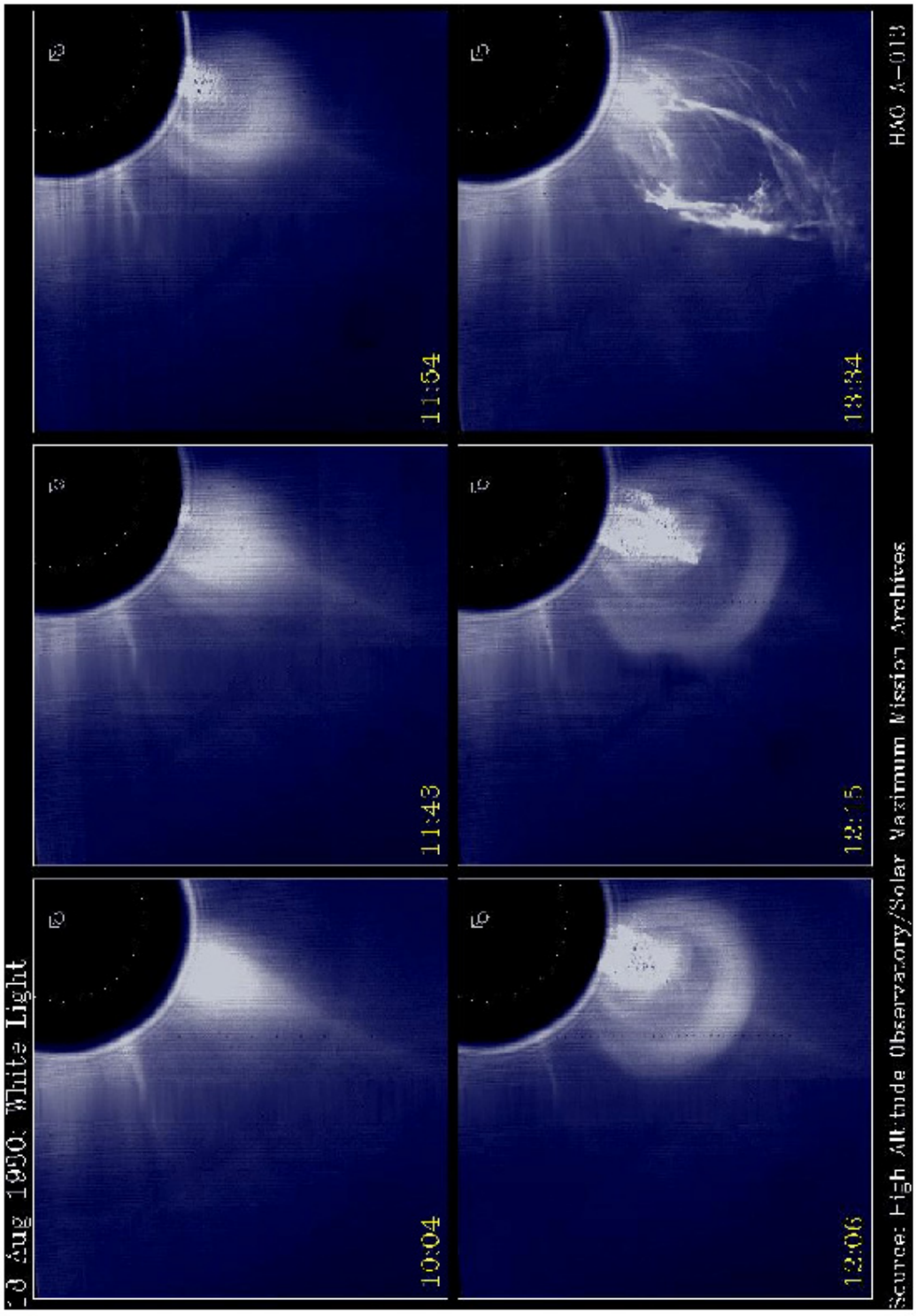


Figure 5.7: Time evolution of an observed CME (image credit: High Altitude Observatory, Boulder, Colorado)

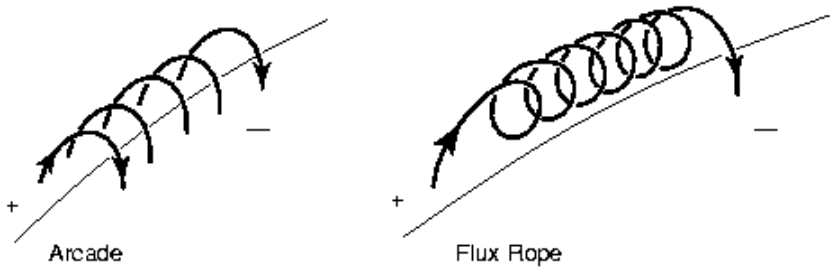


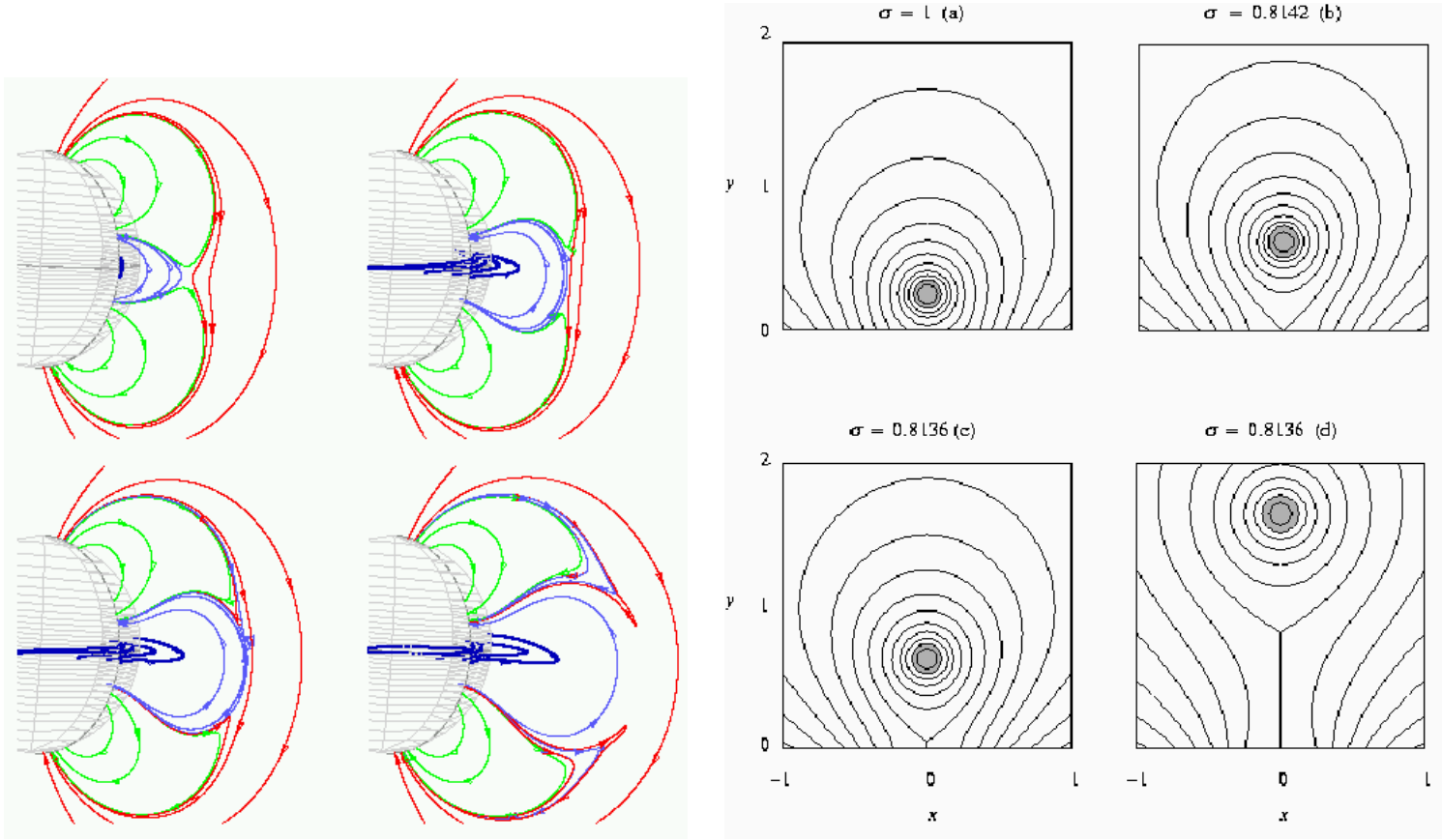
Figure 5.8: Arcade (left) and flux rope (right) magnetic topologies adopted by most CME models [46] (image courtesy of J.A. Klimchuk)

42^3 internal nodes. We will label the calculation with this set of parameters and refer to it as simulation (A) in the future. The total simulation time is 500 Alfven times. We are able to carry out this computation on a Pentium II 233 MHz computer in 18 hours because the exponential propagation methods allow us to perform calculations with a time step greatly exceeding the CFL limit. We use Dirichlet-Neumann boundary conditions for the magnetic field $\mathbf{B}$ for this run. We will justify the choice of parameters and the boundary conditions and investigate what effect they have on the solution later in this chapter. In particular, we will discuss our choice of the Reynolds number in the last section of this chapter. For now, we concentrate on understanding how magnetic field lines topologically evolve from a potential configuration to twisted and sheared structures.

Figures 5.11 (a), (b) show the time evolution of the maximum norm of the x-component of the velocity and the norm of the magnetic energy defined as

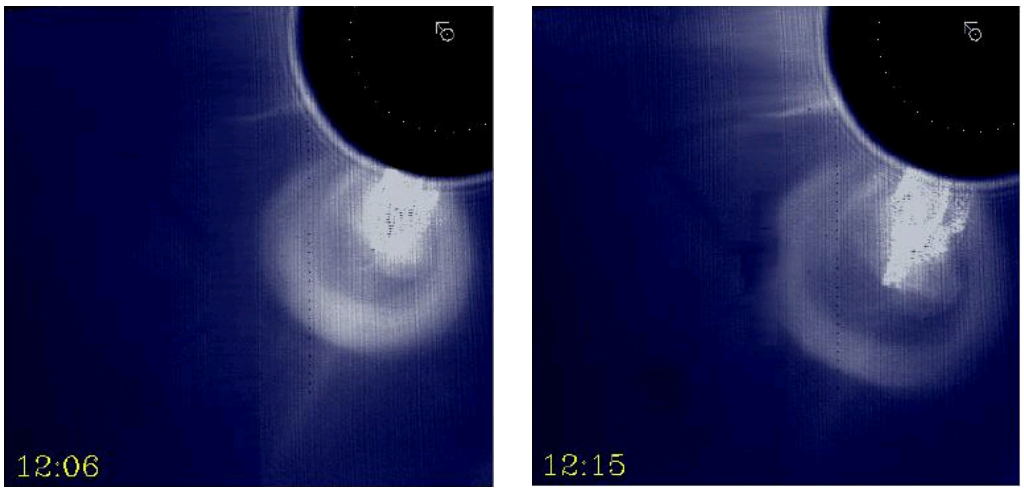
$$\|\mathbf{B}\| = \sqrt{\frac{1}{N_x N_y N_z} \sum_{1\le i\le N_x} \sum_{1\le j\le N_y} \sum_{1\le k\le N_z} [(B_x)^2_{ijk} + (B_y)^2_{ijk} + (B_z)^2_{ijk}]}. \tag{5.36}$$

First, let us examine the configuration of the magnetic field lines at $t = 491\tau_A$ right before the simulation is stopped. Figure 5.13 shows selected magnetic field lines in different colors. The magnetic configuration is viewed from different angles as the coordinate frame is rotated around the z-axis from figure (a) to (d) and tilted down. As we can see, it consists of the low lying orange colored field lines that have a conic shape if viewed as a projection on a $x-z$-plane (Fig.5.13(d)), and the overlaying arcade (purple, green, yellow and red field lines) perpendicular to the orange field lines. Figures 5.14, 5.15 show the same magnetic configuration but add the velocity streamlines to the picture. In Figure 5.14(a) the velocity



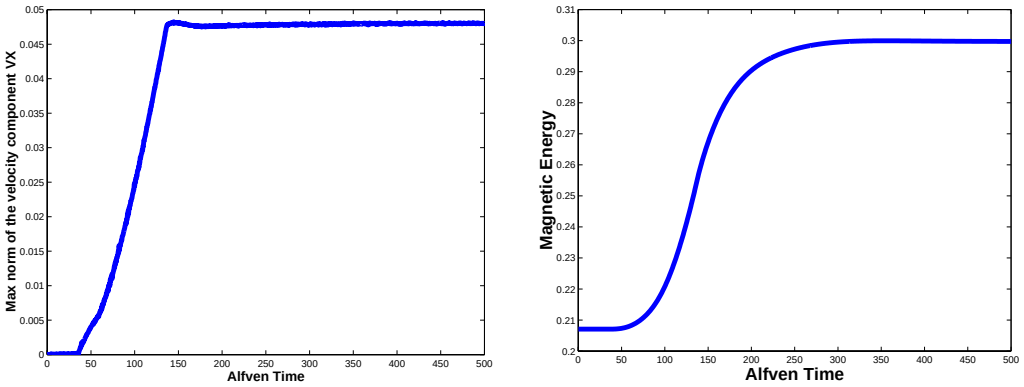
(a) (b)

Figure 5.9: The eruption scenarios for the multipolar arcade model and the flux rope configurations. a) Breakout CME model showing the evolution of a quadrupolar system in which the inner part of the central arcade are sheared by antiparallel footpoint motions near the neutral line (image from [9], courtesy of J.A. Klimchuk). b) The rise and reconnection in a flux rope, whose footpoints are subjected to a converging flow (image from [44])



(a) (b)

Figure 5.10: A characteristic three-part structure of an observed CME at two different times in the evolution



(a) (b)

Figure 5.11: a) Time profile of the maximum norm of the x-component of the velocity. b) The time evolution of the norm of the magnetic energy vector defined by (5.36).

streamlines are shown as black lines with arrows on top of the magnetic field lines and Figure 5.14(b) displays them as cyan colored lines as they are positioned with respect to the magnetic field. Finally, Figure 5.15 gives a three-dimensional perspective on where the velocity streamlines are located with respect to the magnetic configuration. As we can see from these pictures the velocity streamlines close to the base plane $z = 0$ plane reflect the imposed boundary conditions on the tangential component of $\mathbf{V}$, i.e. in the area of low lying orange magnetic field lines they form vortices rotating around the magnetic poles. As the altitude increases, however, the vortices become tilted with respect to the ground plane and finally expand upwards as a fan, indicating that the plasma material rises up from the base plane. Now let us compare the coronagraph image of an actual CME with the configuration that is obtained numerically. In order to demonstrate the similarities between the two we enhanced the image in Figure 5.10 so that the main features can be seen better. Figure 5.16 (a) displays the enhanced coronagraph image of a rising CME, while the image 5.16(b) shows the magnetic configuration and the velocity streamlines of the numerical solution to the resistive MHD model as projected onto the $x - z$ plane. It can clearly be seen that the simulation captured the main features of the image (a), i.e. the conical shape close to the photospheric surface of the CME, the heart shaped feature in the cavity and finally the bright halo-like structure which constitutes the front edge of the CME which is represented as an arcade of the magnetic field lines in figure (b). Image (a) clearly shows that the CME expands outwards. Figure 5.12 shows the contour plot of the normalized magnitude of the Lorentz force $|\mathbf{J} \times \mathbf{B}|/(|\mathbf{J}| \cdot |\mathbf{B}|)$ acting on the magnetic configuration in the simulation. This quantity decreases rapidly with height and the overall configuration is seen to be relatively magnetic force-free. Given this fact and the topology of the velocity streamlines we can see that the upwards expansion is in fact present in the numerical solution and if the model would include density and pressure this could be shown directly.

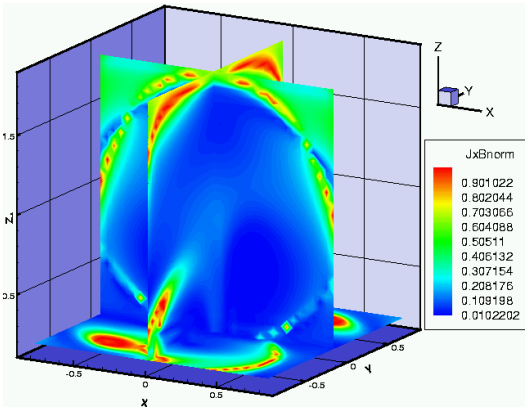
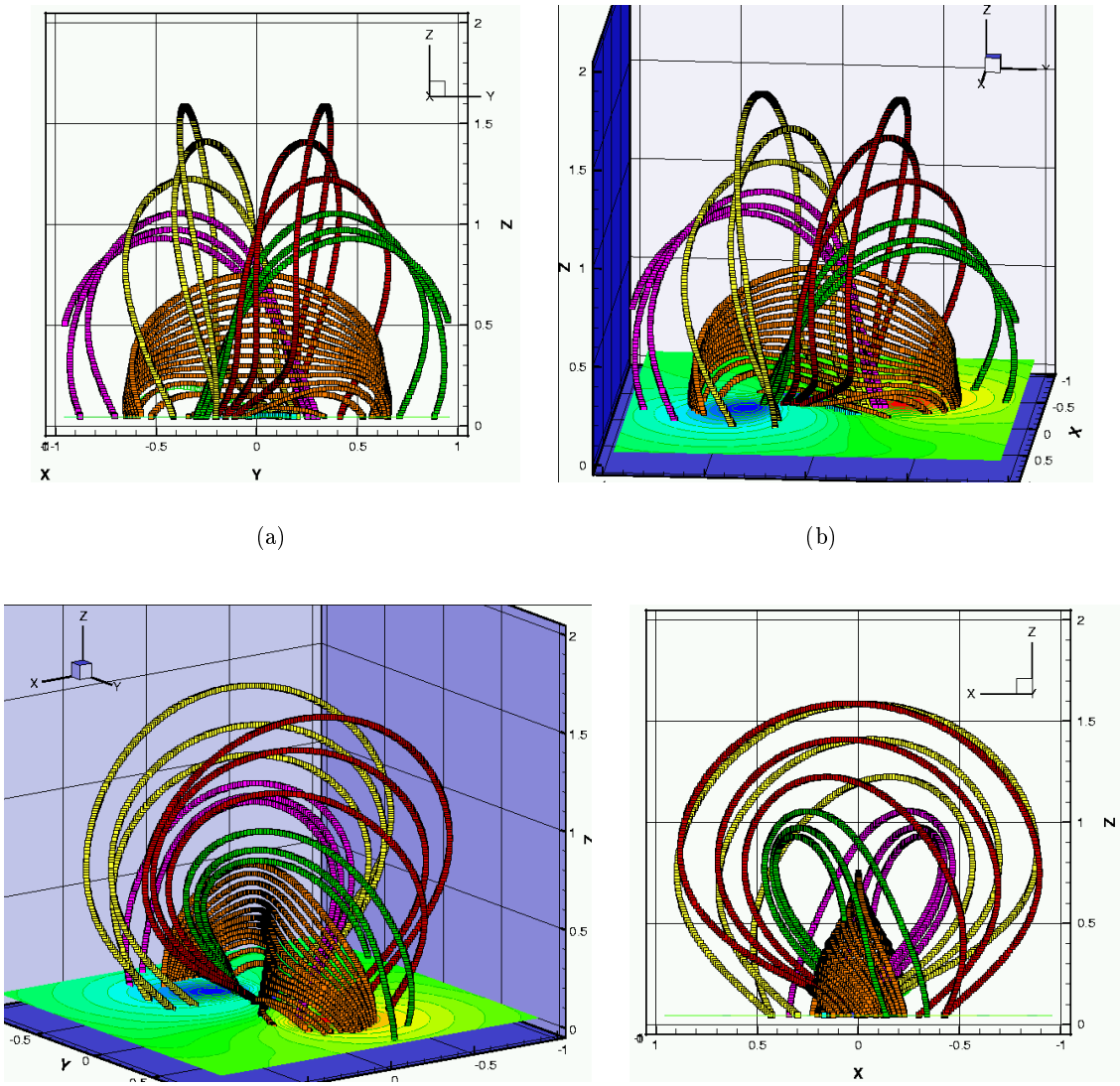


Figure 5.12: Contour plot of the normalized magnitude of the Lorentz force $|\mathbf{J}\times\mathbf{B}|/(|\mathbf{J}|\cdot|\mathbf{B}|)$.

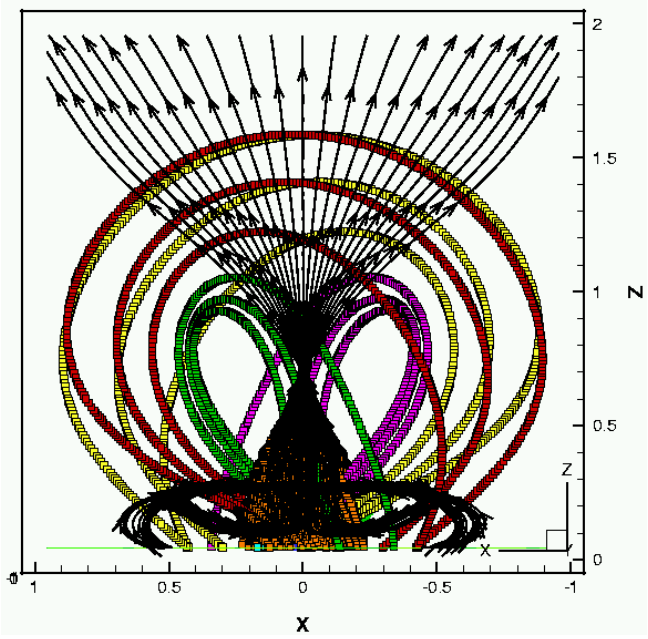
Figures 5.13-5.16 make it clear that neither flux rope, nor conic shaped magnetic lines, or multiple arcade geometries need be used to explain the classical three-part-structured CMEs (Fig. 5.10), which constitute about a third of all mass ejections observed in the corona [46]. The numerical results clearly demonstrate that it is the two-dimensional projection produced by the coronagraph that creates such CME configurations and, in fact, at this stage in the evolution of the CME its magnetic field can be relatively force-free. The motion due to residual $\mathbf{J} \times \mathbf{B}$ forces will in turn cause the expansion of the magnetic field. Since the resistivity is small the magnetic field is almost frozen into the plasma as in ideal MHD. The importance of the MHD forces is also indicated by the fact that the motion of the magnetic configuration in the simulation is tightly related to the increase of the velocity at the boundary. While the velocity magnitude at the boundary is being increased, the topology of magnetic field undergoes modifications. Once we stop increasing the velocity and maintain it at a fixed value of $t = 135\tau_A$, the magnetic configuration stays relatively unchanged and the images from $t = 134\tau_A$ look virtually the same as those in Figure 5.13. This can also be seen from the time profile of the magnetic energy (Fig. 5.11(b), which stays nearly constant for the time interval $t \in [135\tau_A, 500\tau_A]$. To understand the time evolution of the magnetic field we now discuss the observed evolution of an eruptive solar configuration and examine the time change in the topology of the magnetic field lines in the simulation.



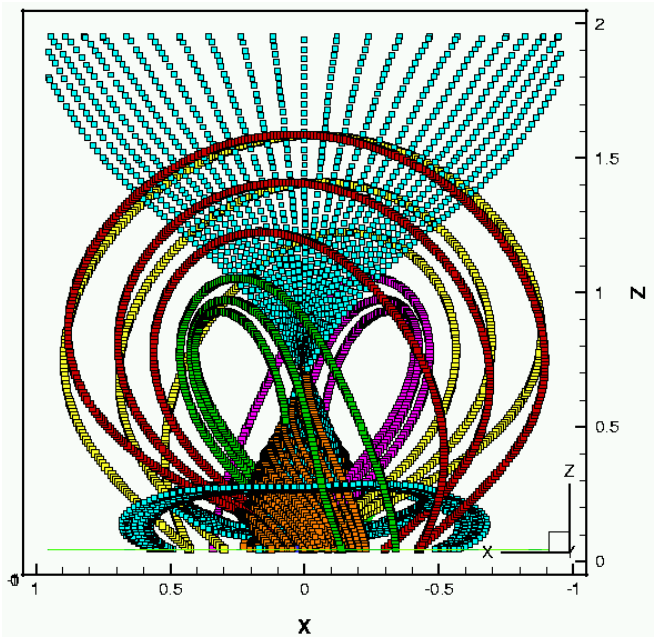
(a) (b)

(c) (d)

Figure 5.13: The topology of the magnetic field lines at $t = 491\tau_A$ as viewed from two different angles in a) and b).

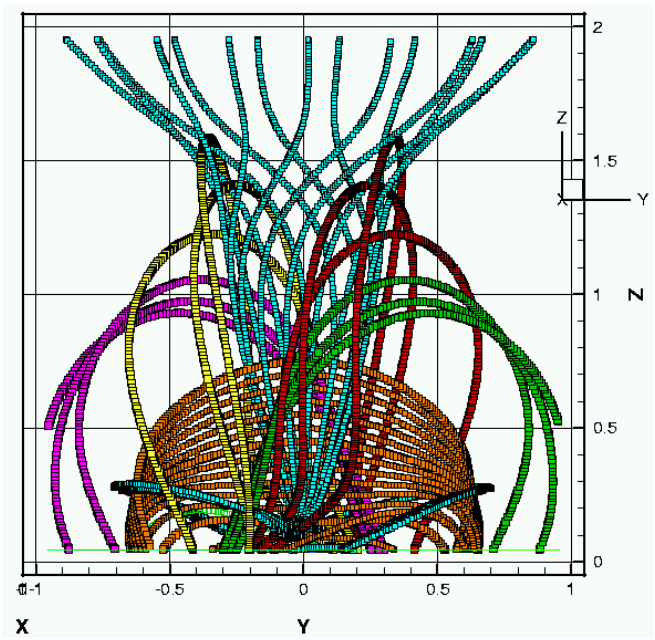


(a)

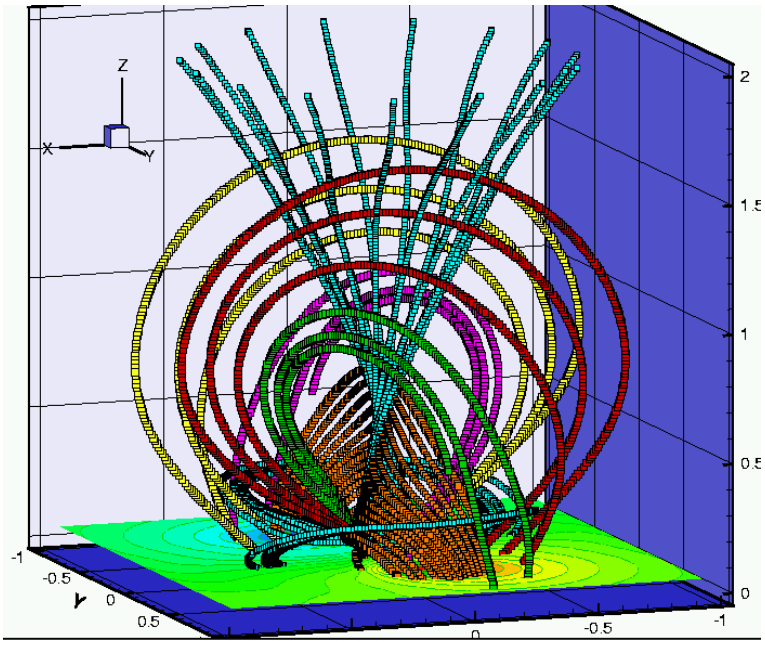


(b)

Figure 5.14: The velocity stream lines a) projected on top of the magnetic field lines shown as black lines with arrows, b) shown as positioned with respect to the magnetic field configuration as cyan colored lines.

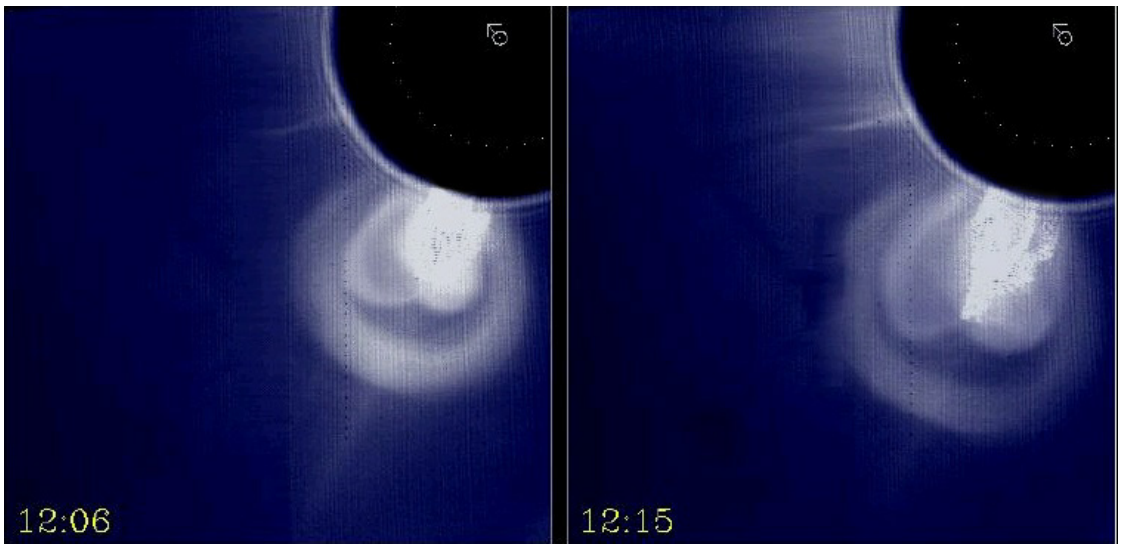


(a)

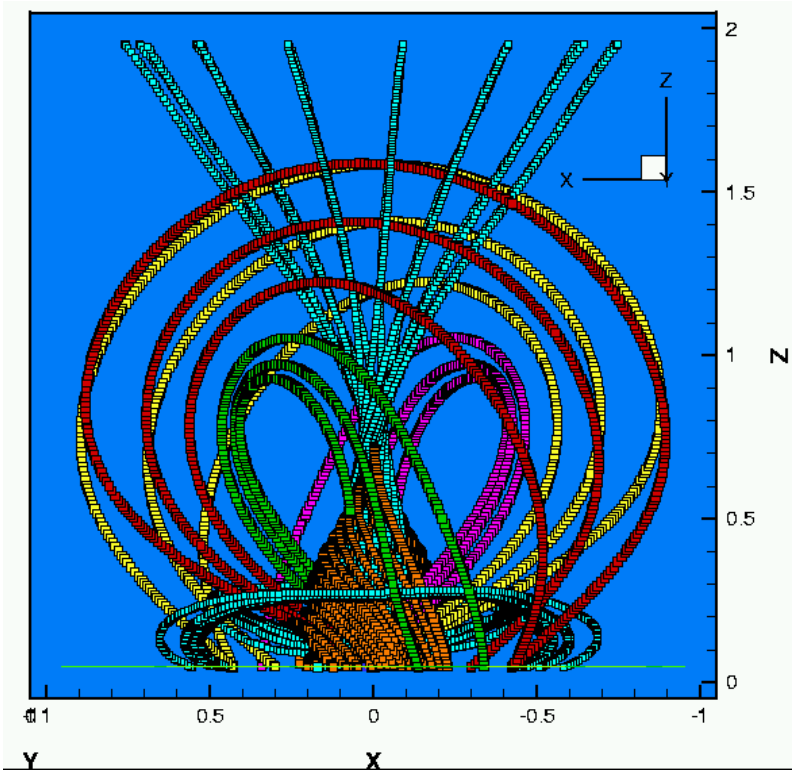


(b)

Figure 5.15: The velocity streamlines (cyan colored) with respect to the magnetic field configuration as viewed from different angles.



(a)



(b)

Figure 5.16: a) Coronagraph image of CME from figure 5.10 with enhanced brightness to emphasize a heart shaped structure in the CME cavity. b) xz-plane projection of the magnetic configuration and the velocity streamlines of the numerical model of a CME.

5.3.2 Sigmoid-to-arcade evolution of the magnetic configurations: comparison of the observational data and the numerical simulation results

Sterling *et al.*[75] published an observational study where they trace the evolution of so-called halo CMEs. This term defines the eruptions that result in a CME traveling directly along the line of sight of an observer, i.e. the images of such events capture the magnetic configuration involved as looked at from above. The authors found that the pre-eruptive formations are usually the sigmoids, which during the eruption are replaced with the magnetic arcades positioned perpendicular to S shapes of the initial sigmoid. Figure 5.18(a)-(j) shows an evolution of a typical halo CME as seen on a soft X-ray image. The picture (a) displays the initial sigmoid configuration and (j) shows the newly formed arcade that runs normal to the direction of the pre-eruption sigmoid. Figure 5.17 reprinted from [75] shows schematically the change that the magnetic configuration undergoes during the eruption.

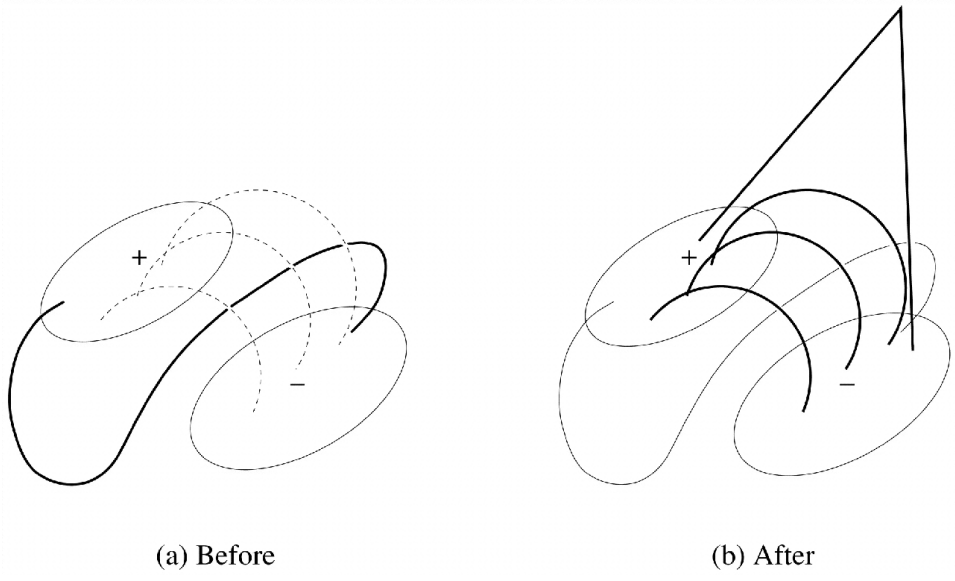


Figure 5.17: Sketch summarizing morphological properties of sigmoid-to-arcade observations: a) pre-eruption sigmoid structure, b) post-eruption arcade with the sigmoid structure dimmed or absent (reprinted from [75], courtesy of A.C. Sterling)

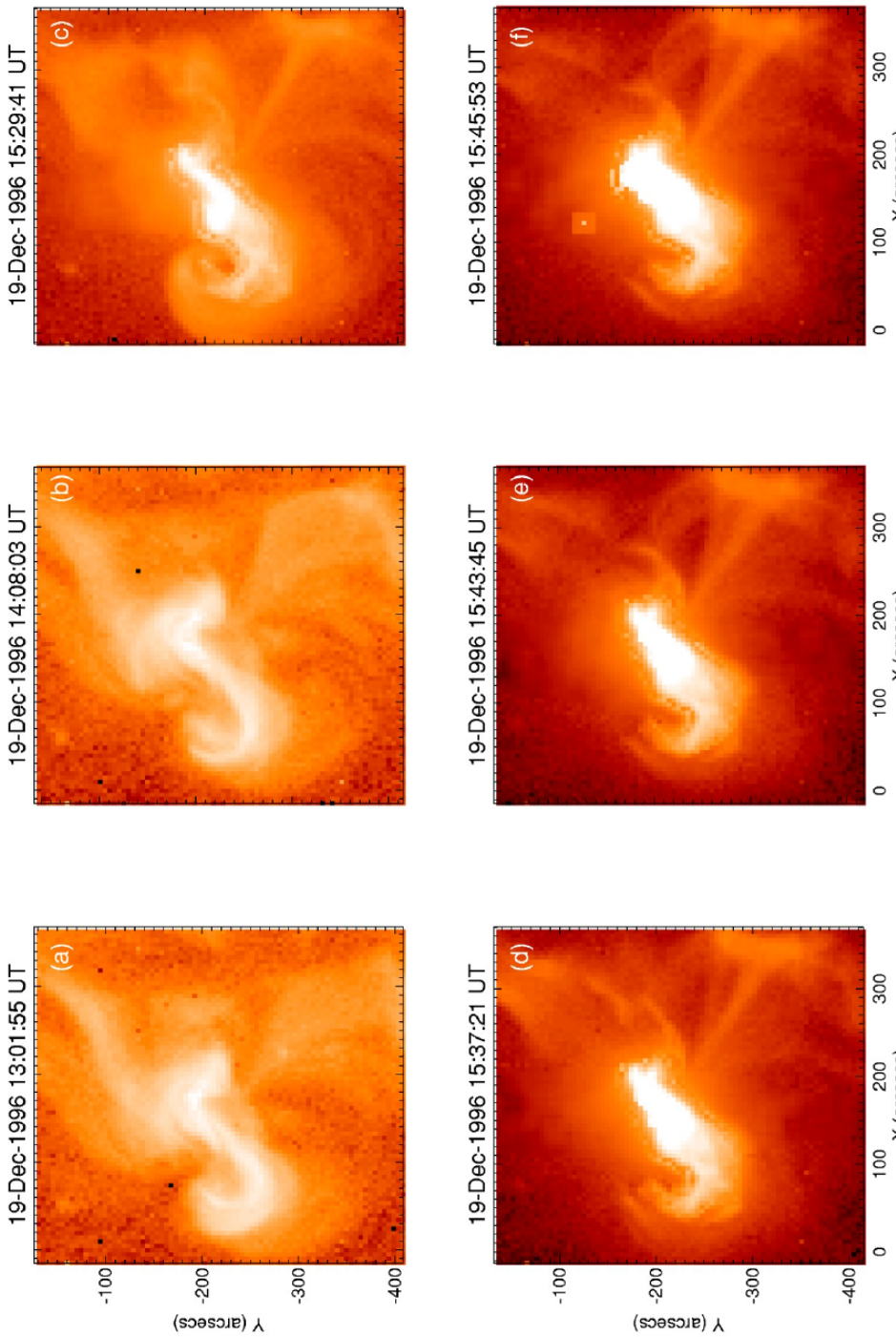


Figure 5.18: Evolution of the 1996 December 19 CME event in Soft X-Ray image (a)-(j) and in EUV (k),(l) (reprinted from [75], courtesy of A.C. Sterling)

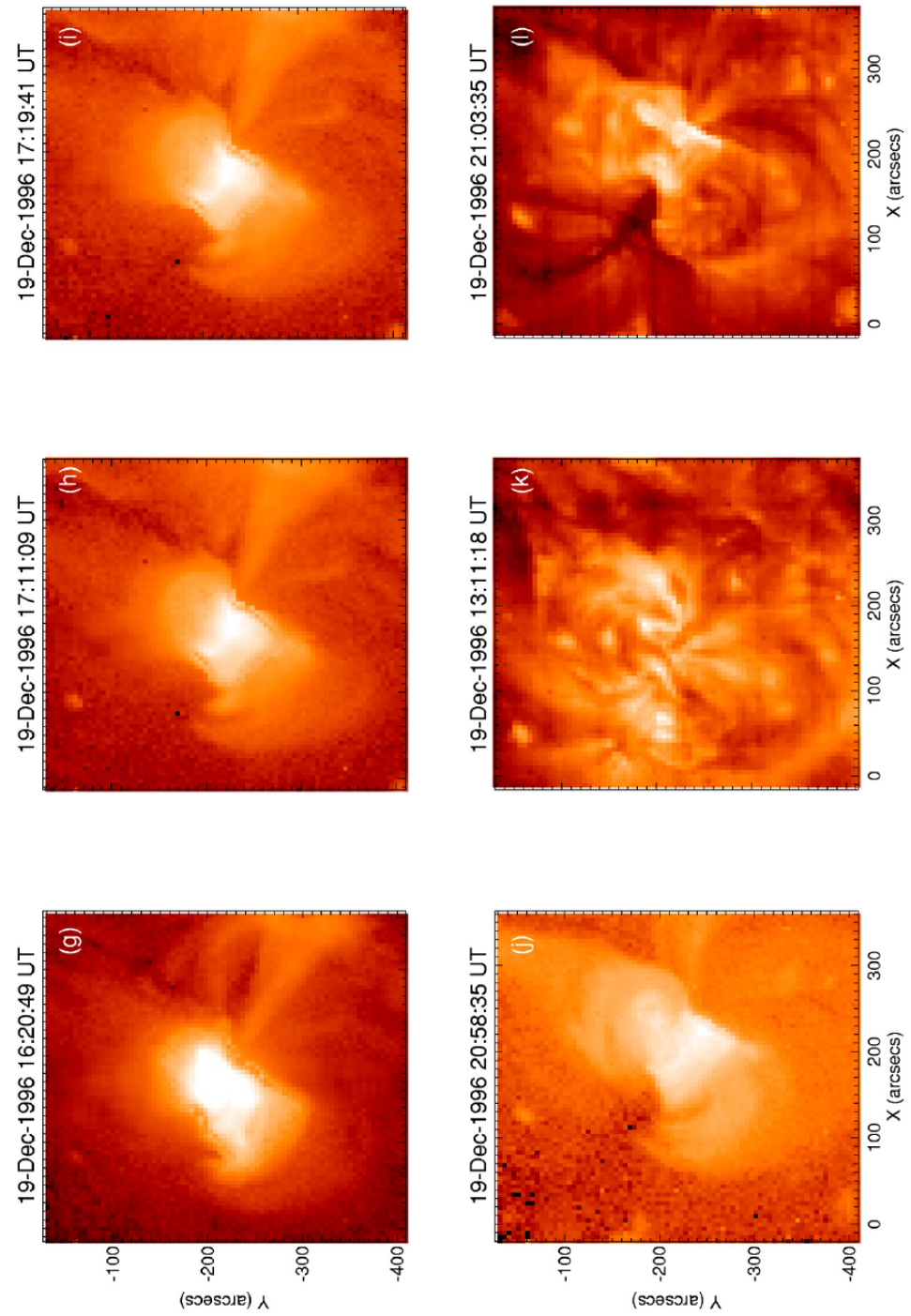
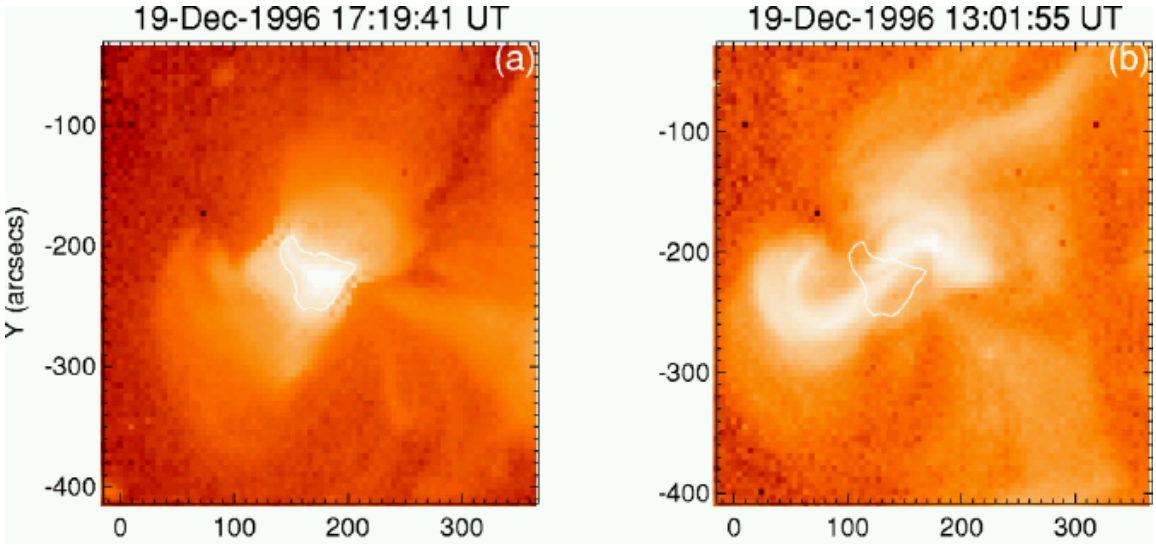
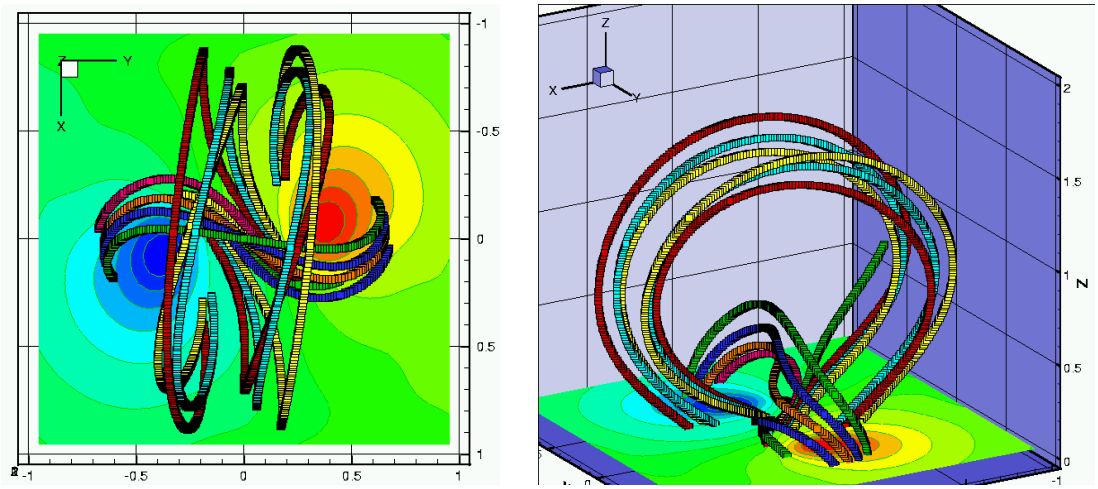


Figure 5.18: - *Continued.*



(a)



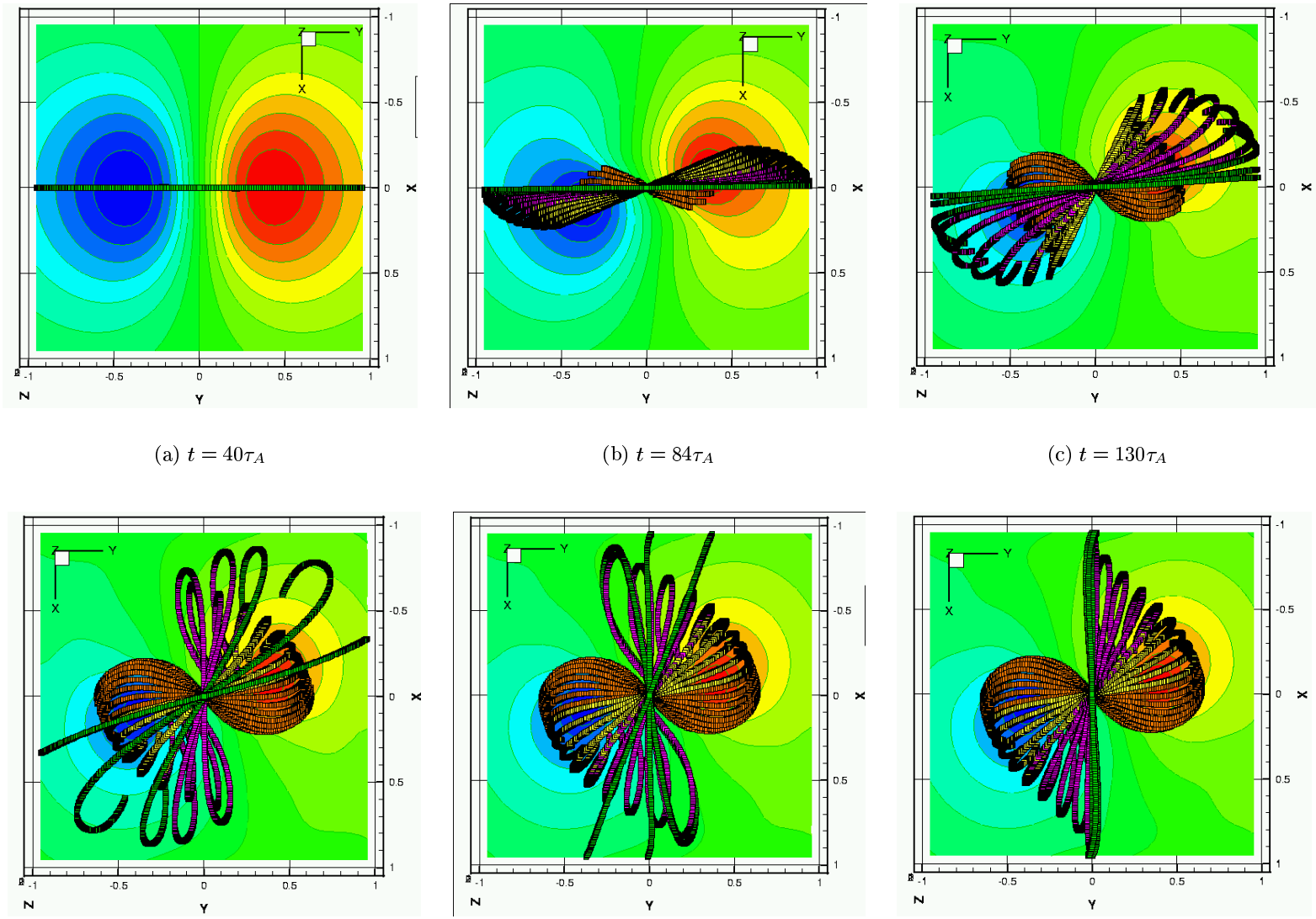
(b) (c)

Figure 5.19: a) Soft X-Ray image of the developed arcade (left) and the preexisting sigmoid with the overlaid contour of the arcade (right). (b),(c) The image of the sigmoid and the overlaying arcade in a numerical model of a CME.

Now let us once again look at the magnetic configuration that we obtained from the numerical simulation. Figures 5.19(b),(c) show selected magnetic field lines as viewed from the top and side and compares them with the sigmoid-arcade transformation in an actual CME as displayed in 5.19(a). As we can see, the positioning of the arcade relative to the sigmoid in the numerical model matches the images from the observations. To demonstrate the sigmoid-to-arcade transformation we will examine the evolution of the magnetic configuration as it changes in time during the simulation. For this purpose we plot magnetic field lines at different altitudes that cross the line $\{x = 0, y = 0, 0 \leq z \leq 2\}$ for different times in the simulation and trace the changes in their topology. The colors of the field lines indicate the altitude at which they cross the z-axis in the following order with increasing height: orange, yellow, purple and green. Figure 5.20(a)-(f) shows the evolution as seen from the top, Figure 5.21(a)-(f) examines the time changes from the side view and Figures 5.22(a)-(f) and 5.23(a)-(c) display the structure as projected onto a $x - z$ and $y - z$ planes correspondingly.

A number of features of a real CME can be seen in the numerical solution. First, the top view clearly demonstrates the sigmoid-to-arcade evolution seen in Figure 5.18. The higher lying (green) magnetic field lines rotate at a slower rate, thus forming an arcade normal to a sigmoid formed by more twisted field lines at lower altitudes (orange). Secondly, it becomes evident that the cusps observed in the solar magnetic configurations are merely an artifact of a two-dimensional projection of a three-dimensional magnetic topology. Figures 5.22(b),(c) and 5.23(b),(c) demonstrate this property and resemble closely cone-shaped structures called *helmet streamers* that can be seen routinely in the corona (Fig. 4.2) as well as the cusps described in [75] (Fig. 5.17). Thirdly, the low lying cone-like structures constituting the inner part of the CME configuration in Figure 5.10 (right) can be identified as a projections of sigmoids at a low altitude (orange magnetic field lines). Finally, Figure 5.23(d) shows another side view of the magnetic configuration, where the sigmoids are also seen as cusp-like structures, that are usually explained as the post-reconnection arcades similar to the two-dimensional models described in Chapter 4, but can clearly be simply a result of a projection effect as well. We want to emphasize that all of these features commonly observed in the corona are usually described by different models as results of distinct mechanisms and different topological configurations. The magnetic configurations produced by our simulations, however, show that all these characteristics can be united as

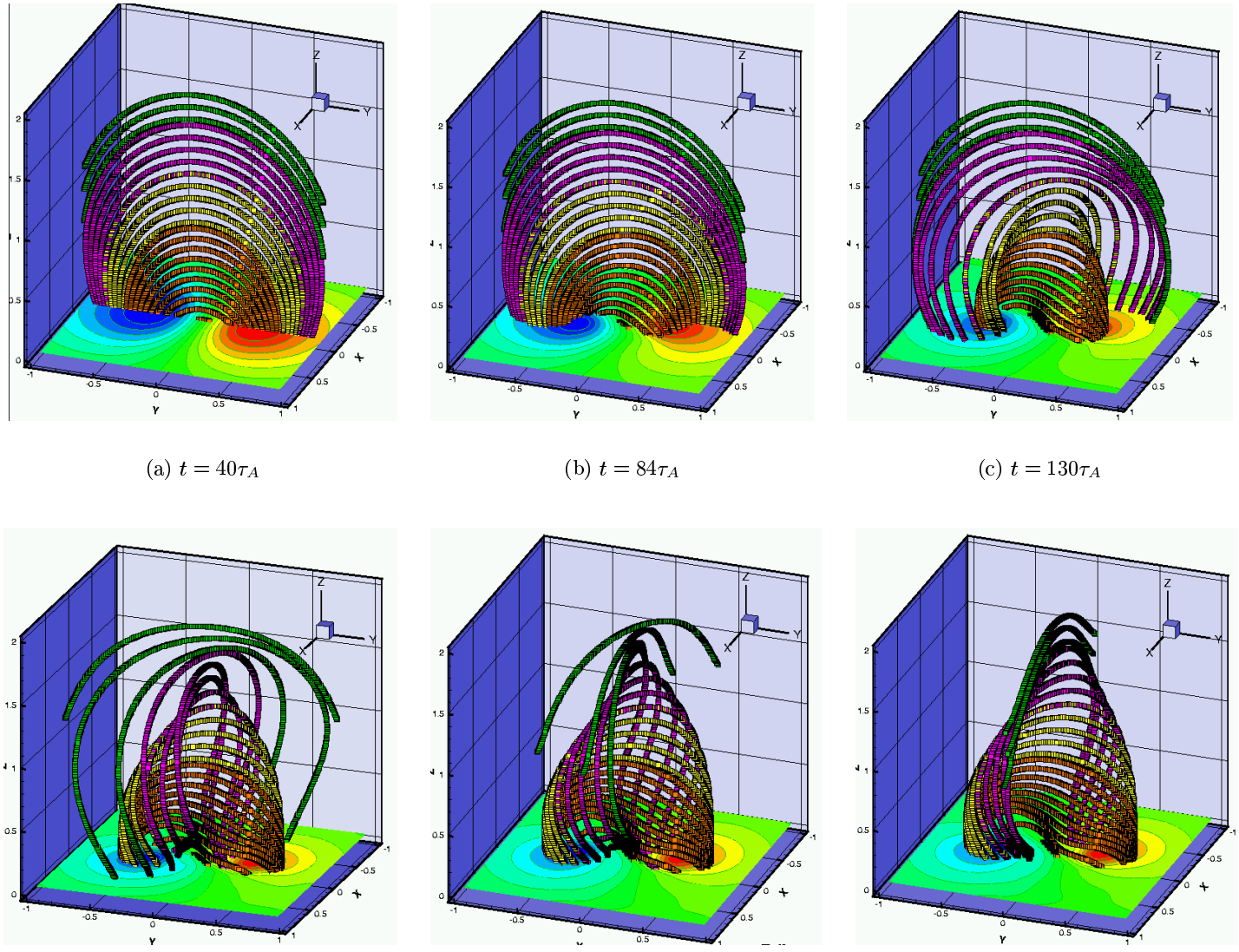
representations of the same process and topology and viewed as different projections of a single configuration. Additionally, the computations indicate how such magnetic configurations can be obtained from an untwisted and unsheared potential configuration simply by imposing a rotating velocity flow at the base plane. Thus we do not use any artificial boundary conditions that are hard to justify from the physical point of view as is done in some models.



(a) $t = 40\tau_A$ (b) $t = 84\tau_A$ (c) $t = 130\tau_A$

(d) $t = 175\tau_A$ (e) $t = 220\tau_A$ (f) $t = 310\tau_A$

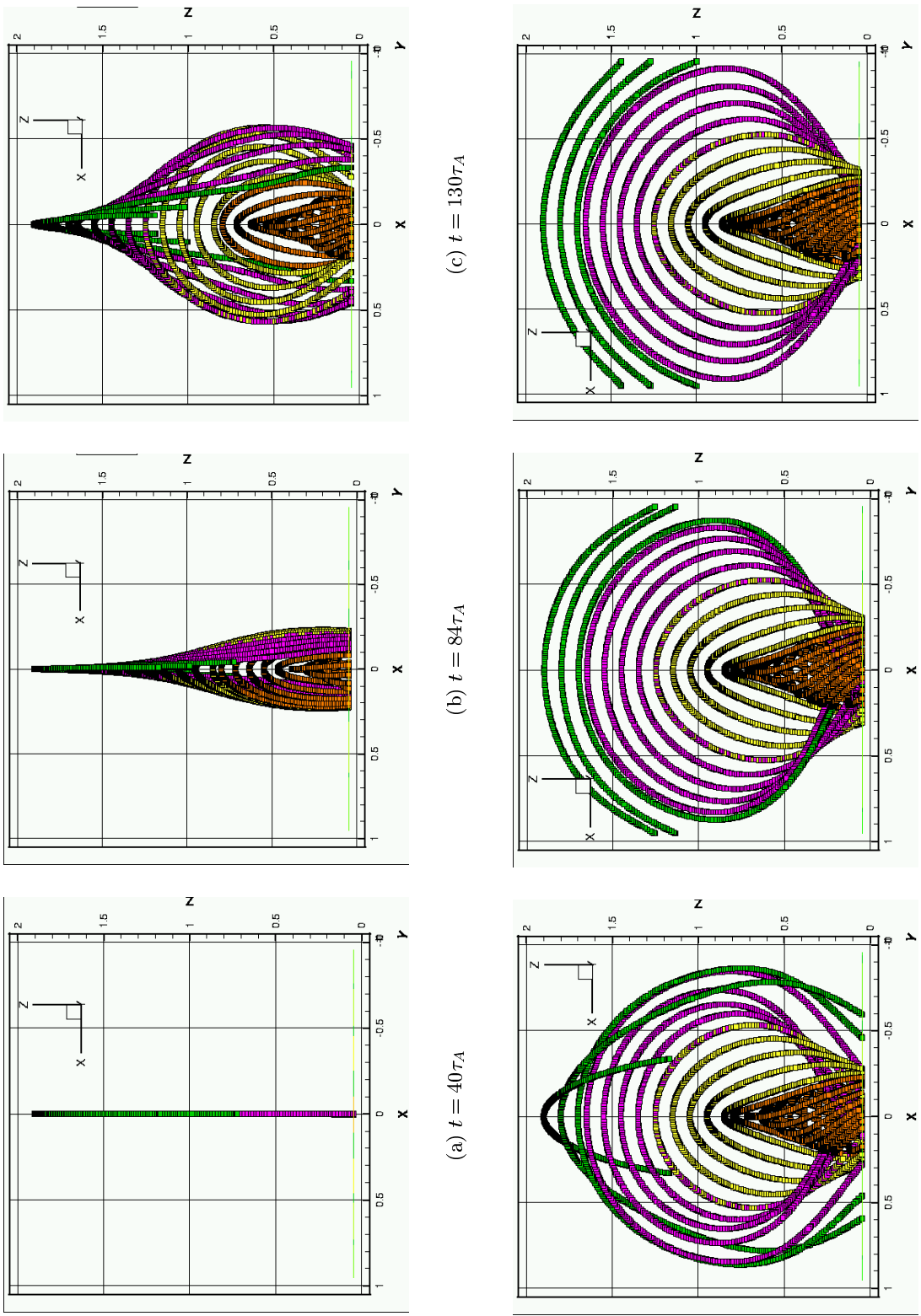
Figure 5.20: Sigmoid-to-arcade evolution in the numerical simulation as viewed from the top.



(a) $t = 40\tau_A$ (b) $t = 84\tau_A$ (c) $t = 130\tau_A$

(d) $t = 175\tau_A$ (e) $t = 220\tau_A$ (f) $t = 265\tau_A$

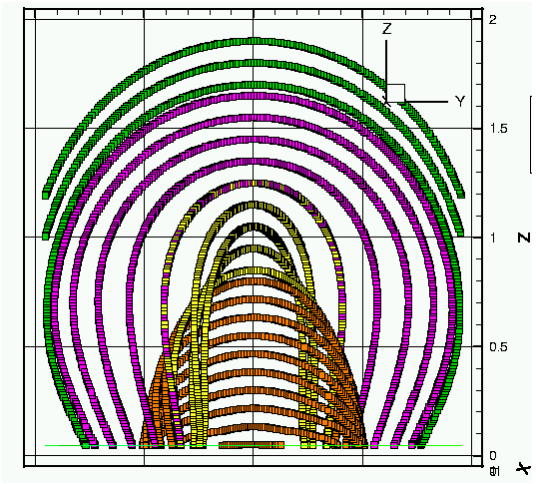
Figure 5.21: Sigmoid-to-arcade evolution in the numerical simulation as viewed from the side.



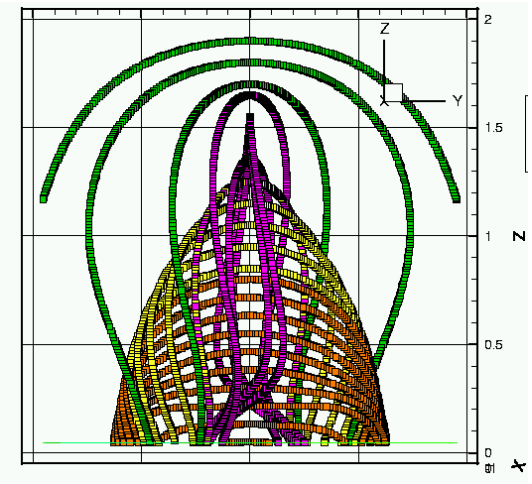
(a) $t = 40\tau_A$ (b) $t = 84\tau_A$ (c) $t = 130\tau_A$

(d) $t = 175\tau_A$ (e) $t = 220\tau_A$ (f) $t = 310\tau_A$

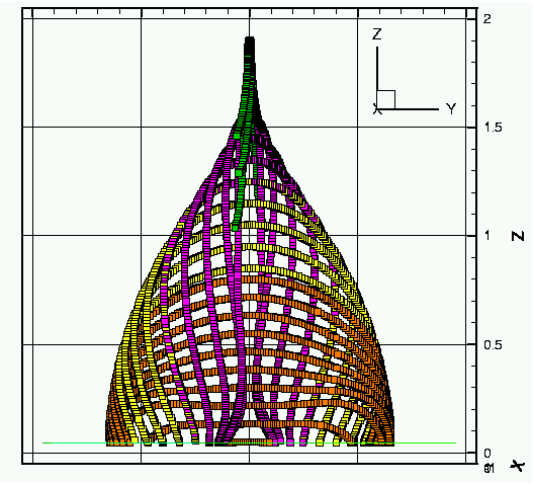
Figure 5.22: Sigmoid-to-arcade evolution in the numerical simulation as viewed projected onto the plane $x - z$.



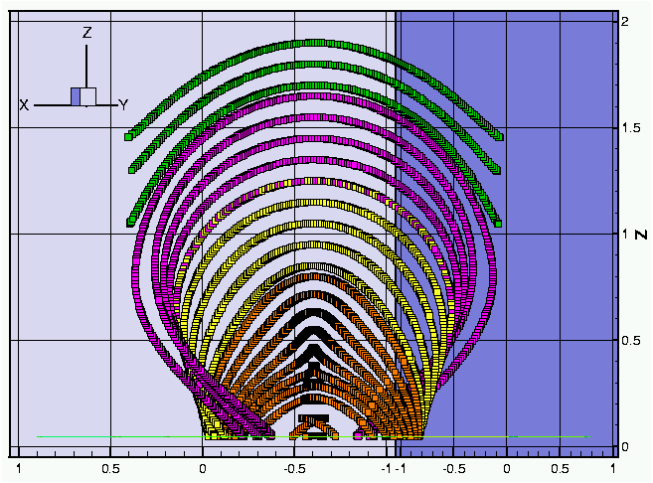
(a) $t = 130\tau_A$



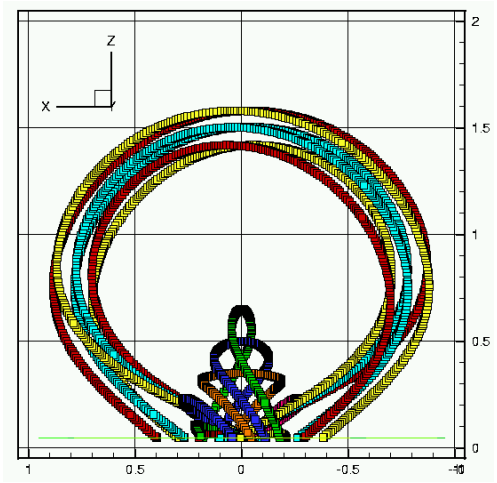
(b) $t = 175\tau_A$



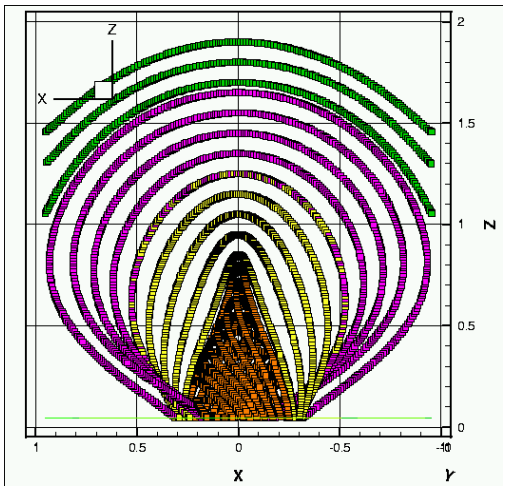
(c) $t = 354\tau_A$



(d) $t = 399\tau_A$



(e) $t = 492\tau_A$



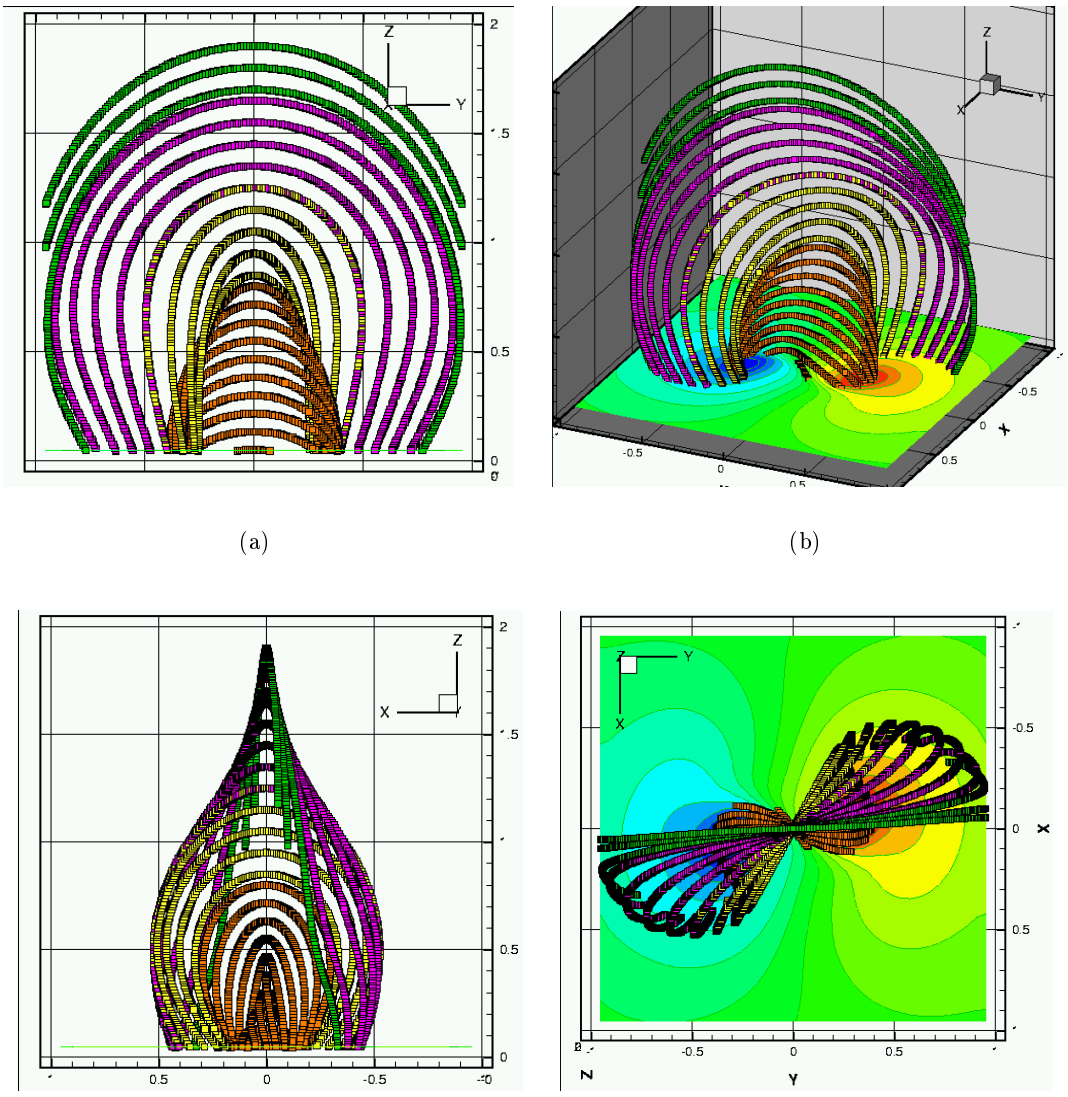
(f) $t = 492\tau_A$

Figure 5.23: Different coronal features can be identified as projections of the topology of the magnetic field lines obtained from numerical simulation: a), b), c) helmet streamers, d) low altitude cusps, e),f) sigmoid-arcade structures and classic three-part structured CMEs.

5.3.3 The effect of boundary conditions on the numerical solution

One of the major issues for any numerical study is the formulation of the correct boundary conditions. It is well known that numerical errors, resulting from incorrect values imposed on the variables at the boundaries, can pollute the solution and introduce unphysical dynamics. Therefore, we carried out a study to verify the robustness of our calculations with respect to the imposed boundary conditions and investigated how changes in the formulation of the boundary conditions affect the numerical solution.

First, we performed a computation that uses the same set of parameters as the simulation (A) of the previous section, but instead of Dirichlet-Neumann boundary conditions on the magnetic field $\mathbf{B}$, imposes purely Dirichlet conditions on $\mathbf{B}$ fixing them at the values of the initial potential $\mathbf{B}(x, y, z, t) = \mathbf{B}(x, y, z, 0)$. Both types of boundary conditions are described in detail in Section 5.2.2. Figure 5.24 shows the magnetic configuration obtained from the calculation with pure Dirichlet boundary conditions at $t = 133\tau_A$. Let us compare these images to Figures 5.23(a), 5.21(c), 5.22(c) and 5.20(c), which show the same view of the simulation (A) configuration at $t = 130\tau_A$. While the overall topology and the time dynamics of the magnetic configurations are very similar in both simulations a careful examination of these images indicates that the twisting of the magnetic field lines proceeds somewhat more slowly in the computation with pure Dirichlet boundary conditions. This is especially evident if we look at the lower lying orange colored magnetic field lines that comprise the initial sigmoid configurations. The top views in particular indicate that these field lines are less twisted and do not expand as much as in simulation with the Dirichlet-Neumann boundary formulation. This can be explained by the fact that if the magnetic field lines are fixed at the base plane at the initial values it somewhat reduces their flexibility and movement, so that the shearing process proceeds more slowly. We emphasize, however, that the overall dynamics is almost unchanged which indicates the robustness of our numerical model. Another confirmation of this conclusion can be seen from the plots showing how magnetic energy changes with time for both calculations (Fig. 5.26). The profiles are very similar. However, because the field lines twist faster in the calculation with Dirichlet-Neumann boundary conditions, the magnetic energy increases faster in this case as well.



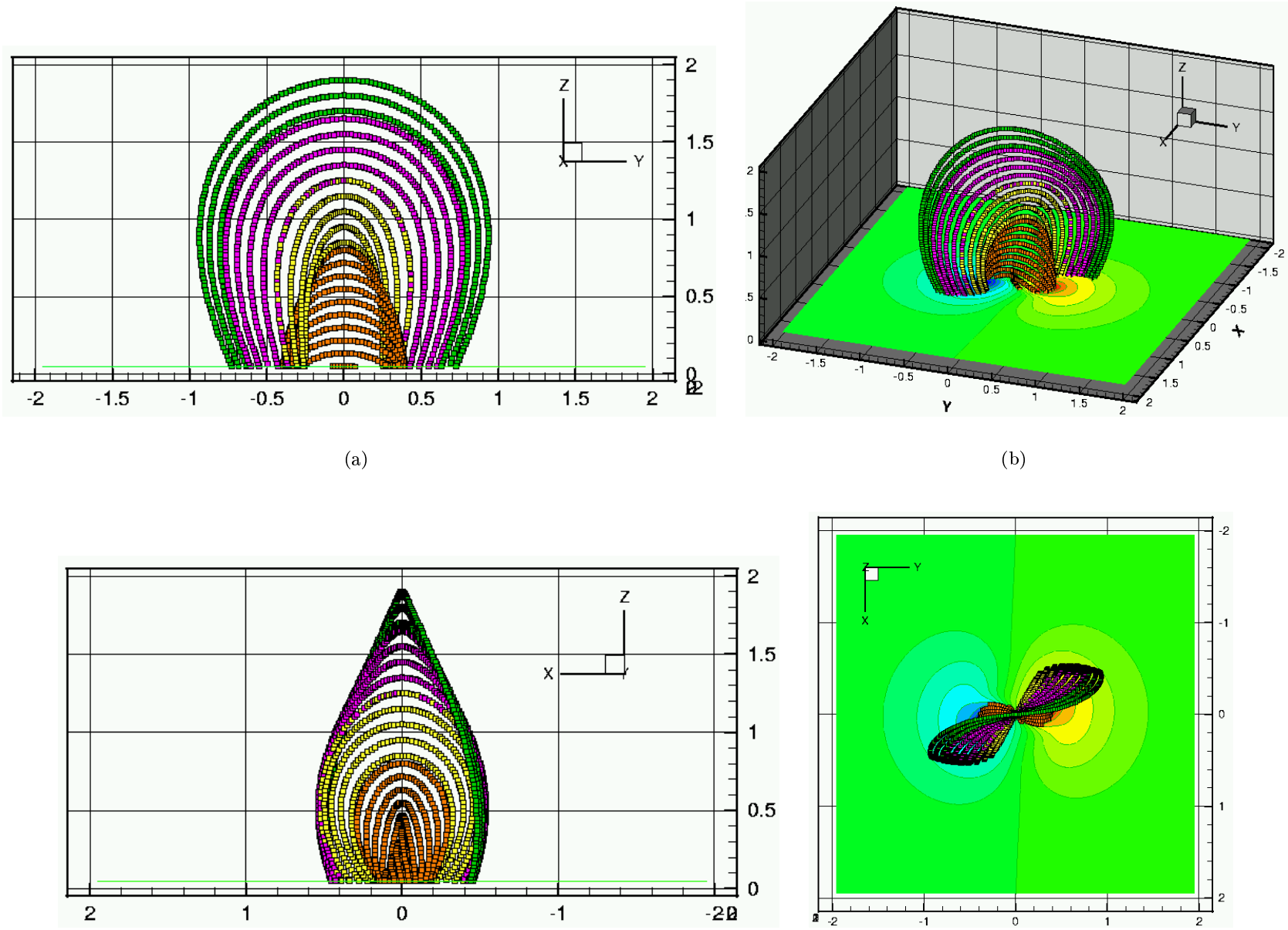
(a) (b)

(c) (d)

Figure 5.24: Magnetic configuration at $t = 133\tau_A$ computed using Dirichlet boundary conditions for all components of the magnetic field $\mathbf{B}$ in a computational box of size $2 \times 2 \times 2$ (233 MHz Pentium II calculation).

Another check that we have performed is to extend the side boundaries of the computational domain further and then examine the resulting changes in the solution. We ran the simulation again with the same set of parameters and the Dirichlet boundary conditions on $\mathbf{B}$, but in this case doubled the number of grid points and the size of the computational box in the $x-$ and y-directions. Figure 5.25 displays the resulting magnetic configuration from this computation and Figure 5.27 compares the magnetic energy and the maximum norm of the velocity component V_x for the two simulations. Comparing Figures 5.25 and 5.24 we notice that the configuration in a smaller box matches exactly the topology of the field lines in a larger domain within its boundaries. The green colored magnetic lines especially highlight this observation, since in a large computational box we see them in full length perfectly matching the dynamics of lower lying field lines. Figure 5.24 confirms this as well by indicating that the time profiles of the magnetic energy and V_x are the same. The difference between the magnetic energy curves in 5.27(a) can be easily explained from the definition of the norm $\|\mathbf{B}\|$ in (5.36). Since this norm is inversely proportional to the number of grid nodes, if the magnitude of the magnetic field at the base plane has nearly compact support, increasing the number of grid points will result in a decreased total norm. In our future simulations we plan to use better scaled norm definitions.

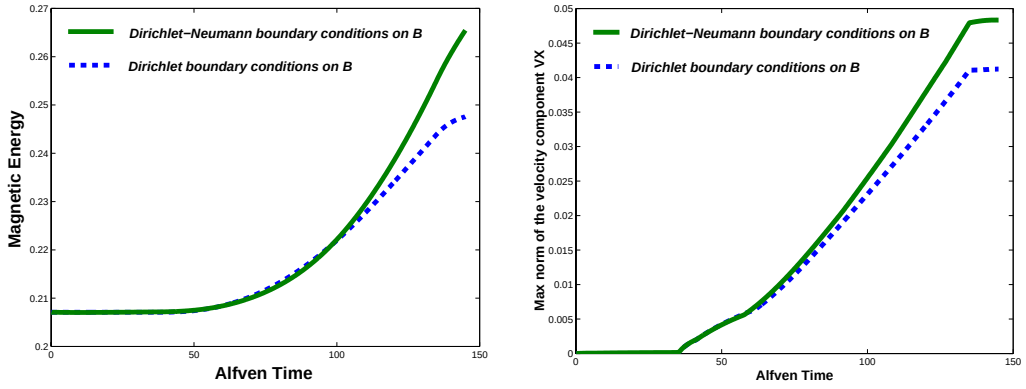
We can draw two important conclusions from this investigation of the effect of boundary conditions. First, this study shows that the overall dynamical change of the magnetic topology is nearly unaffected by the differences between the two types of boundary conditions. Second, we learned that calculations performed in smaller domains truthfully portray the topology of the resulting magnetic configuration; thus, increasing the size of the computational box does not change the magnetic configuration in the smaller volume, but simply gives extra information about the magnetic field outside the small box.



(a) (b)

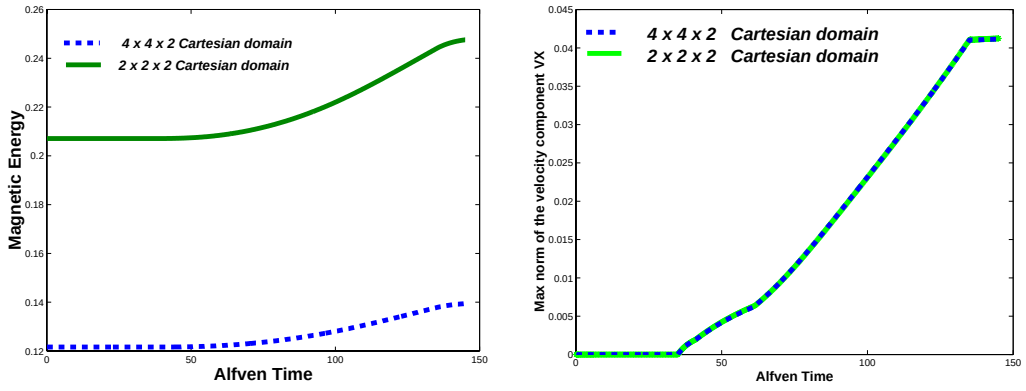
(c) (d)

Figure 5.25: Magnetic configuration at $t = 133\tau_A$ computed using Dirichlet boundary conditions for all components of the magnetic field **B** in a computational box of size $4 \times 4 \times 2$ (SV1 vector supercomputer calculation).



(a) (b)

Figure 5.26: Comparison of the calculations with Dirichlet-Neumann vs. Dirichlet boundary conditions imposed on the magnetic field **B**: a) time profile of the magnetic energy norm defined by (5.36), b) time evolution of the maximum norm of the x-component of the velocity V_x.



(a) (b)

Figure 5.27: Comparison of the calculations in a small $2\times2\times2$ vs large $4\times4\times2$ computational domain: a) time profile of the magnetic energy norm defined by (5.36), b) time evolution of the maximum norm of the x-component of the velocity V_x.

5.3.4 The assessment of the numerical error in the simulations

Another issue that needs to be addressed in any numerical study is the effect of the spatial discretization and the grid resolution on the numerical solution. Due to the complexity of the resistive MHD equations and a large number of difficulties in solving this system numerically, this issue is typically not given enough attention in numerical MHD research. The minimum grid size required for the appropriate resolution in MHD simulations sometimes is already pushing the boundaries of the computational resources and therefore, increasing the grid size to perform the resolution study can be simply impossible. Given our computational resources we were able to double the grid size and perform simulation on a Cray SV1 vector supercomputer to verify the convergence of the solution.

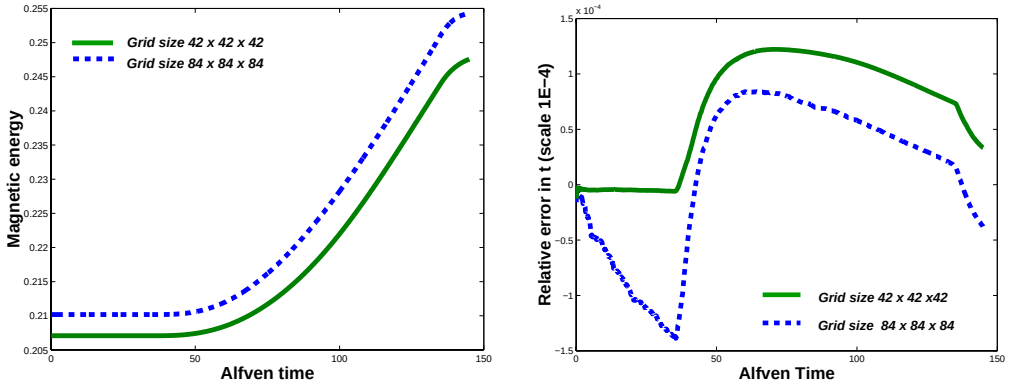
Simulation (A) of Section 5.3.1 was performed on a $42\times42\times42$ grid. We executed another computation with the formulation and the parameters identical to those of simulation (A) but with the grid size increased to 84^3 nodes. Figures 5.28(a) and 5.29 plot the norm $\|\mathbf{B}\|$ defined by (5.36) and the maximum norm of the velocity components V_x and V_z for each grid size. While the plots in Figure 5.29 confirm that the velocity undergoes essentially the same evolution in both calculations, they cannot really be used as an indicator of the convergence since the maximum norm is an inappropriate measure for this. A better perspective is provided by the norm of the magnitude of the magnetic field, which according to the plot in Fig. 5.28(a) only differ by about 2%. Taking into considerations the limitations of norm (5.36) discussed earlier we can only interpret this result as an indication that the convergence is in fact achieved, but will perform a more detailed study with better formulated norms in the future.

We wish to mention another accuracy check that complements the resolution study for our numerical model. Recall that in order to solve the MHD equations using the exponential propagation method we formulate them as an autonomous system (Section 3.1), i.e. the time variable is evolved using the equation

$$t' = 1. \tag{5.37}$$

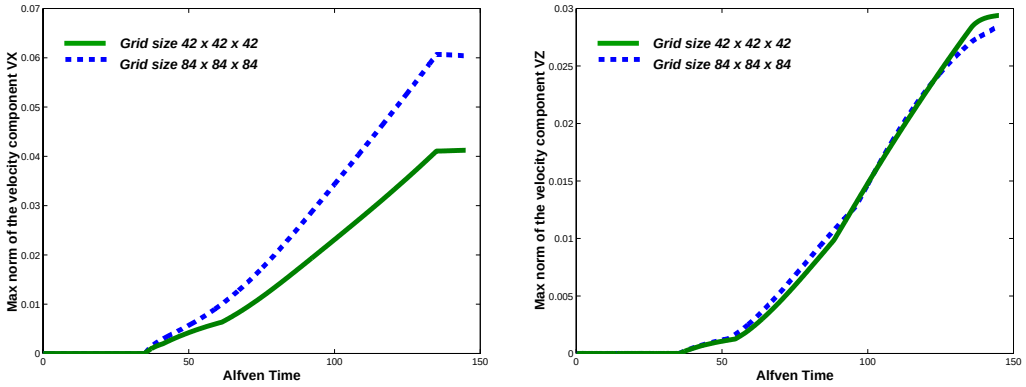
This gives us an opportunity to perform an additional check of the accuracy of the solution by comparing the computed value of the time variable with the time calculated from the time steps used by the numerical time integration scheme. We show the result of this

comparison in figure 5.28(b). The absolute and relative tolerances for the automatic error control mechanism are set at 10^{-2} and 10^{-3} respectively. As we can see from the plot of the relative error, its value is well under the tolerance required. This check reassures that the numerical solution of a discretized MHD system is accurate to within the tolerance we specified.



(a) (b)

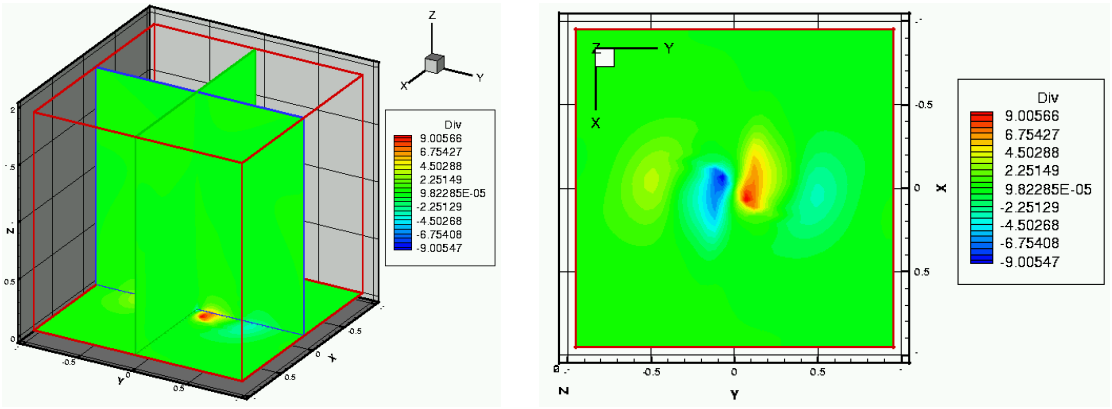
Figure 5.28: Comparison of the calculations using grid size 42^3 performed on 233 MHz Pentium II vs. 84^3 grid nodes used in a SV1 vector supercomputer simulation run: a) time profile of the magnetic energy norm defined by (5.36), b) time evolution of the relative error in the time variable evolved according to the equation $t' = 1$.



(a) (b)

Figure 5.29: Comparison of the calculations on a grid of sizes 42^3 (233 MHz Pentium II calculation) vs 84^3 (SV1 vector supercomputer calculation)

As mentioned before another important contribution to the error in the MHD computations which must be controlled is the numerical divergence of the magnetic field. The non-zero divergence of $\mathbf{B}$ results from the finite precision nature of the simulations and the fact that the discretized equations do not necessarily satisfy the discrete versions of vector identities that enforce $\nabla \cdot \mathbf{B} = 0$ in the continuum equations. Since we used a uniform grid and a straightforward fourth-order central-finite-difference discretization of the spatial operators, we do observe the divergence condition not being satisfied in our calculations to the precision we would like throughout the whole volume of the computational domain. The regions of maximum error in the divergence, however, are limited to the small patches close to the base plane where the magnetic field is the strongest (Fig. 5.30). We do not observe any instabilities in the solution or unlimited growth in the divergence if the grid resolution is increased or the parameters of a simulation changed. The error in the divergence is explained by the fact that we do not use special discretization for spatial operators at the boundaries to keep the discrete version of a divergence condition satisfied. In the future we plan to use a staggered grid and a special discretization of the spatial operators that ensures that the equation for the evolution of the divergence of the magnetic field is kept identically zero in time (see discussion in Chapter 6).



(a) (b)

Figure 5.30: Contour plot of the numerical divergence of the magnetic field at $t = 500\tau_A$.

5.3.5 Increasing the magnitude of the boundary velocity

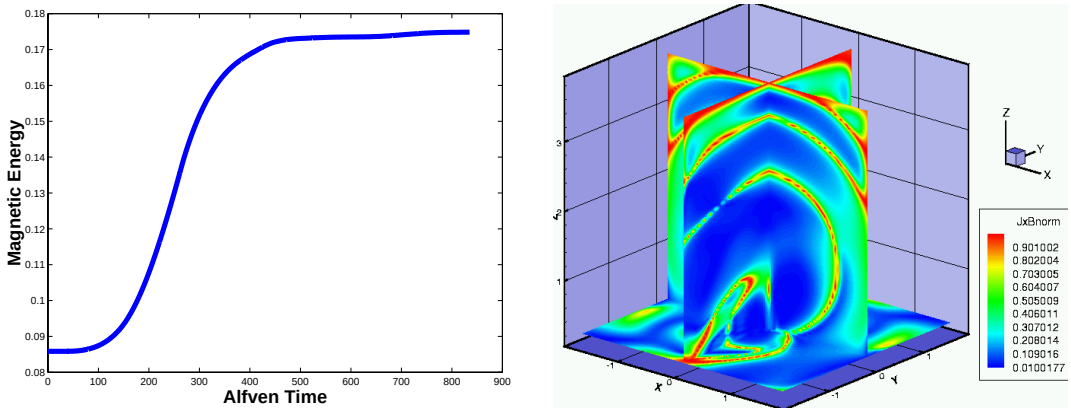
The dynamics of the numerical solution seems to be closely related to the time profile of the velocity flow imposed at the ground plane. Therefore, in this section we will investigate how the evolution the magnetic configurations changes when we increase the maximum magnitude of the boundary velocity and change the time interval over which it is ramped up. We have performed several simulations similar to (A), where we ramped up the boundary velocity to different maximum values at a different speed. For the calculations with $V_{x0} \lesssim 0.13$ the resulting dynamics of the numerical solution were very close to the evolution in the simulation (A), i.e. the initial potential magnetic field gets sheared and twisted up until the moment when the boundary velocity becomes constant in time and then settles to some stable equilibrium state. We observe no reconnection occurring in the computational domain for V_0 in this range. In cases where V_{x0} reaches values larger than 1.4 the magnetic topology of the magnetic configurations is different from the structures obtained in simulation (A). In this section we present two simulations with different values of the maximum velocity V_0 and discuss the possible connection of the evolution in these regimes with the eruptive stages in the dynamics of coronal magnetic configurations.

In the first calculation, which we will label (B), the boundary velocity is increased over 220 Alfven times in such a way that a maximum value 0.8 of V_{x0} is reached at $t = 255\tau_A$ (Fig. 5.32). The computational domain is taken to be a rectangle

$$\Omega = \{-2 \leq x \leq 2, -2 \leq y \leq 2, 0 \leq z \leq 4\}. \tag{5.38}$$

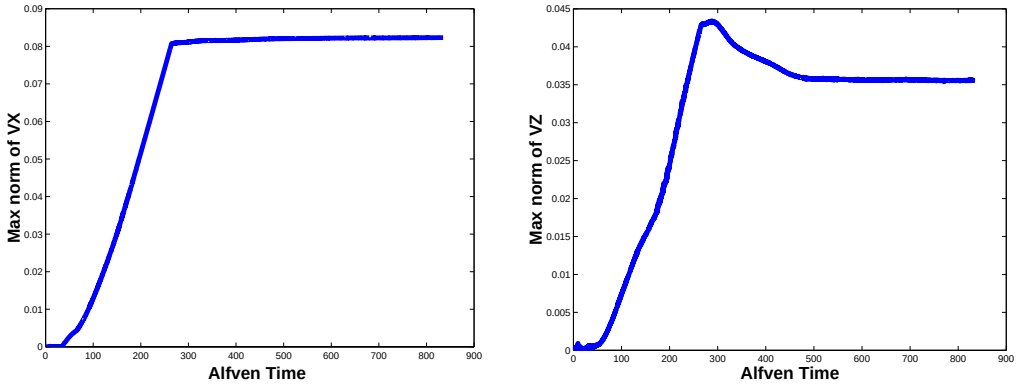
Other parameters are chosen to have the same values as in simulation (A). The shape of the magnetic energy plot in this simulation is very similar to the computations we discussed before. However, despite the fact that the velocity in simulation (B) is increased to a larger value than for calculation (A), the maximum of the magnetic energy for (B) is actually 0.17 vs 0.3 for (A). The key to explaining this result lies in examining the magnetic configuration resulting from simulation (B).

Figure 5.33 shows the magnetic field lines that cross the z-axis at different heights (color coded as before) at the end of the simulation at the point where a relatively steady state has been achieved at $t = 645\tau_A$. As we can see the major difference in the geometry of the magnetic field compared to the images obtained in simulation (A) is the presence



(a) (b)

Figure 5.31: a) Time evolution of the magnetic energy. b) The contour plot of the normalized magnitude of the Lorentz force $|\mathbf{J} \times \mathbf{B}|/(|\mathbf{J}| \cdot |\mathbf{B}|)$.

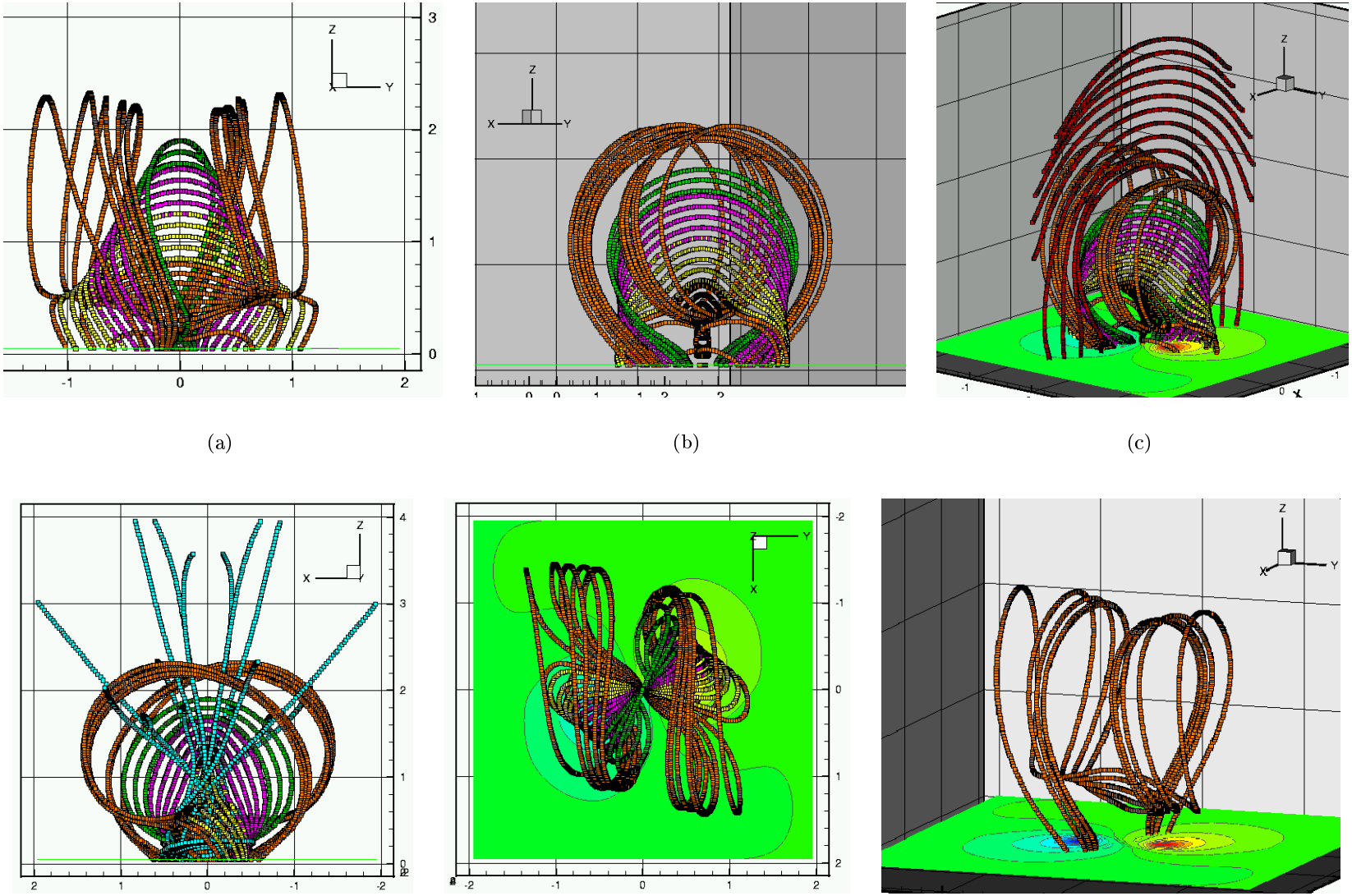


(a) (b)

Figure 5.32: Time profile of the velocity components a) V_x and b) V_z.

of topologically different flux-rope-like field lines (Fig. 5.33(f)). We examined how this configuration forms and discovered that it is the result of the field lines of the overlaying arcade coming close together at a particular point near the base plane and reconnecting. Figure 5.37(a)-(c) shows the pre-reconnection arcades and the resulting post-reconnection flux rope. It is the reconnection process that causes release of energy and explains the overall lower magnetic energy level for simulation (B) compared to the calculation (A). The reason that the releases of energy caused by the reconnection do not appear as oscillations in the plot of the magnetic energy (Fig. 5.31(a)) is the continuous injection of energy by continuous increase of the boundary velocity at the base plane. Apparently, the latter process occurs at a faster rate than the loss of energy through reconnection and therefore we do not see clear indication of the energy releases in the magnetic energy plot. In the next simulation presented in this section this will not be the case and we will be able to identify the reconnection process by looking at a time profile of the magnetic energy.

Once again projection of the magnetic topology at different angles (Fig. 5.33) resemble the coronagraph images taken at different times during a CME. Figure 5.33(b), for example, shows the loop of a flux rope and the sigmoid-like field lines projected in such away as to appear to have cusps in the middle; this could be incorrectly interpreted as a plasmoid being ejected due to the reconnection below in a fashion similar to the two-dimensional axisymmetric models.

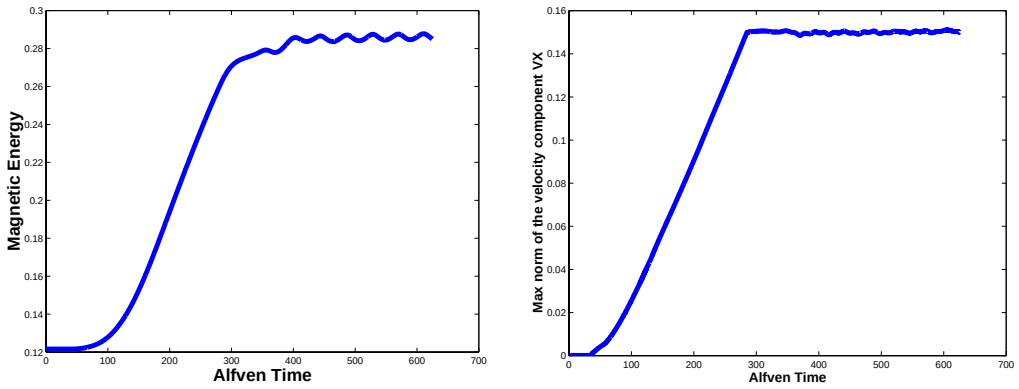


(a) (b) (c)

(d) Velocity streamlines are colored in cyan. (e) (f)

Figure 5.33: Different views of magnetic lines at $t = 645\tau_A$ for the simulation in which the magnetic energy increases as in figure 5.31.

In the next simulation we ramp up the velocity to an even higher value over 285 Alfven times. Except for the time profile of the imposed boundary velocity this simulation differs from computation (B) only in the size of the computational domain. Here $\Omega = \{-2 \leq x \leq 2, -2 \leq y \leq 2, 0 \leq z \leq 2\}$. Other parameters are kept the same as in calculations (A) and (B). We will refer to this computation by the label (C) in the future. There are two major departures from the dynamics and the magnetic topology of the previous simulations in the results from calculation (C). First we notice that the evolution of the magnetic energy at the stage where boundary velocity is kept fixed is no longer represented by a nearly horizontal line. In fact, we see oscillations (Fig. 5.34(a)) that indicate multiple releases of energy, followed by immediate replenishing by the constant velocity flow at the base plane. The magnetic topology also changes significantly (Fig. 5.36). We no longer see the clearly defined regions of morphologically different field lines forming sigmoids and overlaying arcades; instead the configuration becomes much more complex and resembles more the final stage of a CME shown in Figure 4.5. In fact, the twisted "arms" of the real CME can be identified with the spiral field lines observed in the numerical model, which we show separately in figure 5.37(d).



(a) (b)

Figure 5.34: Time profile of the a) magnetic energy and b) the maximum norm of the velocity componentV_x.

Another interesting aspect of the magnetic configuration obtained in the simulation (C) is the reconnection process that occurs between the spiral field lines that extend upwards and exit the box. Figure 5.37(d)-(f) depicts this reconnection process by showing the pre-reconnection spirals (d) and the post-reconnection magnetic field lines (e). Our study of

boundary condition effects above showed that if the computational domain is extended, the magnetic configuration in the initial small box is simply a part of a larger structure. Thus we can rightfully assume that the initial spiral "arms" exiting the domain are simply a part of a larger loop and the post-reconnection configuration shown in figure 5.37(e) is simply a lower part of a large plasmoid like plasma structure that is being ejected upwards according to the topology of the velocity streamlines. While the ejection of such a plasmoid in three-dimensional magnetic configuration has been conjectured before, to our knowledge, it has never been observed in three-dimensional dynamic MHD calculations. In order to confirm our theory we need to analyze the simulation in a larger domain which will capture the full length of the spiral field lines and determine whether the configuration in 5.37(e) is a closed curve. We were unable to perform such a simulation due to the limitation in hard disk space available on the workstation where we visualize and analyze the data. We plan to conduct such study in a near future though as soon as the appropriate resources become available.

We also observe the following trend in the dynamics of the magnetic configurations in the simulations. As the boundary velocity is driven to a larger magnitude there is a corresponding increase in the normalized magnitude of the Lorentz force $|\mathbf{J} \times \mathbf{B}|/(|\mathbf{J}| \cdot |\mathbf{B}|)$. To see this, compare the contour plots of $|\mathbf{J} \times \mathbf{B}|/(|\mathbf{J}| \cdot |\mathbf{B}|)$ for simulations (A), (B), (C) (Fig. 5.12, 5.31(b) and 5.35). This indicates that the magnetic configuration departs from a force-free state further as the velocity flow at the ground plane gets stronger. This means that the eruptive stage in the dynamics of the magnetic structures cannot be explained as evolution through a series of force-free equilibrium configurations and has to be studied as a dynamical model.

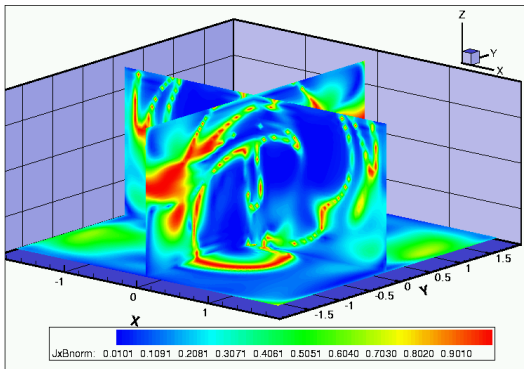
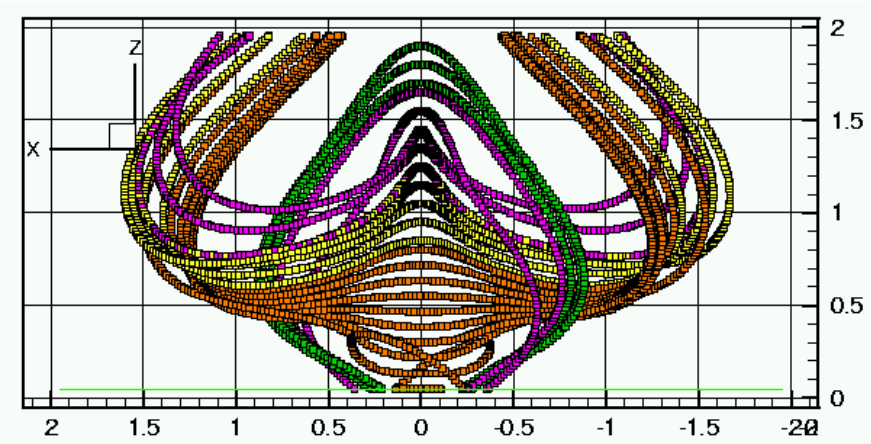


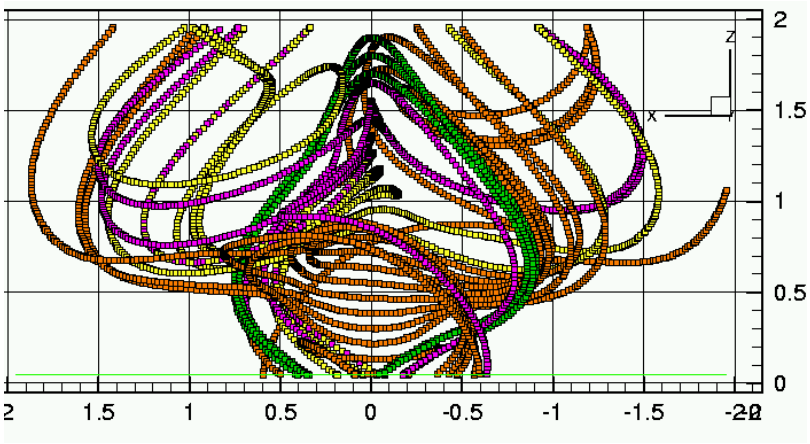
Figure 5.35: The magnitude of the Lorentz force $|\mathbf{J} \times \mathbf{B}|/(|\mathbf{J}| \cdot |\mathbf{B}|)$ at $t = 622\tau_A$.

We conclude this section by presenting two more coronagraph images of different CME events and view them as the projections of the magnetic configurations obtained by our numerical model. One of the major obstacles in understanding the structure of the CME is the difficulty in separating the events into different categories according to their topology. Images of different CME events obtained via observations, like those shown in figures 4.5, 5.38(c),(d), differ greatly and provoke researchers to use different models to describe each event. We emphasize that the important characteristic of our numerical model is that the magnetic configurations obtained via dynamical simulation match quite distinct CME images if observed from a different angle. This eliminates the need to use different artificial boundary and initial conditions to reproduce different CME images and provides a clear explanation for the diversity of structures seen on two-dimensional coronagraph projections of the coronal magnetic configurations. Figure 5.38 demonstrates another example where the coronagraph images (c),(d) can be the two-dimensional projections of the magnetic configurations in figures 5.36(a),(b) or 5.38(a),(b).

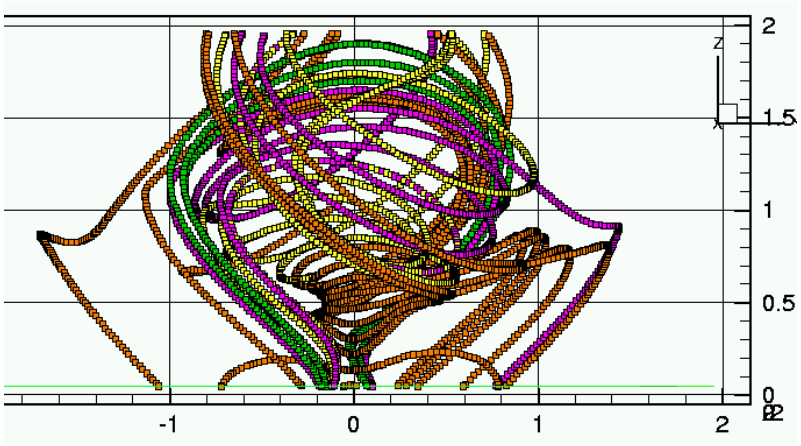
In the future we plane to collaborate with observers to verify our results and compare the numerical simulations with more images obtained by observational instruments.



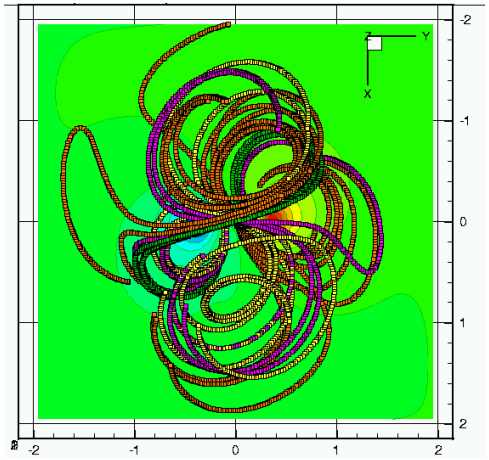
(a) $t = 415\tau_A$, xz- plane view



(b) $t = 622\tau_A$, xz plane view

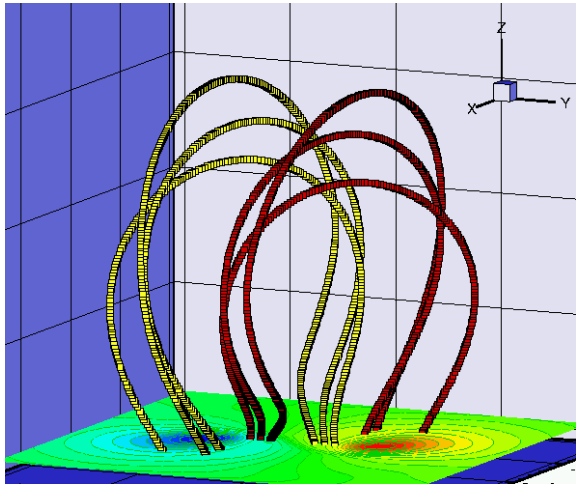


(c) $t = 622\tau_A$, yz plane view

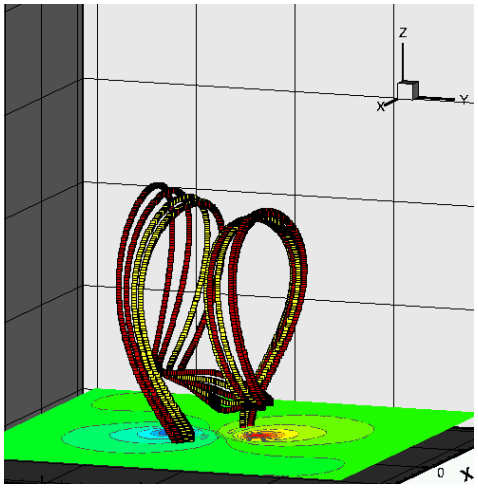


(d) $t = 622\tau_A$, top view

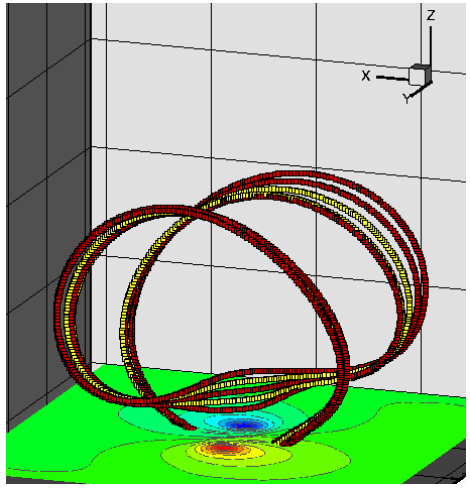
Figure 5.36: Views of the magnetic lines configuration for the simulation for which the magnetic energy increase is shown in figure 5.34 a).



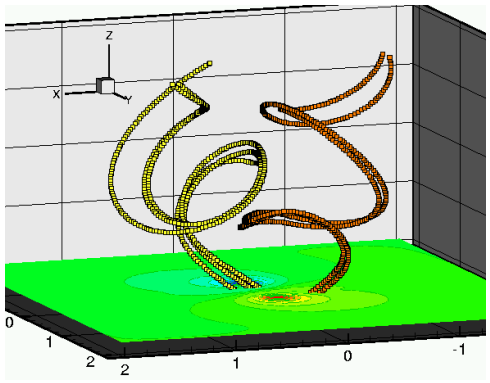
(a) Before reconnection



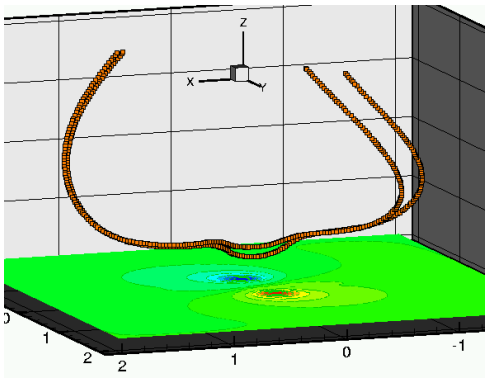
(b) After reconnection



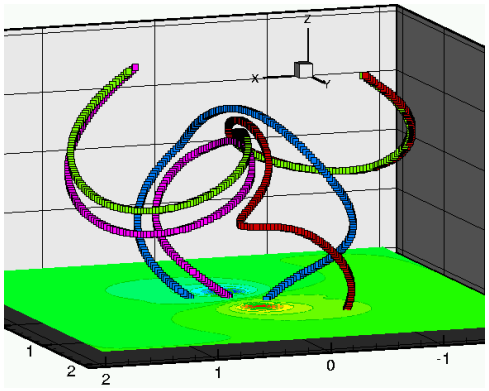
(c) After reconnection



(d) Before reconnection

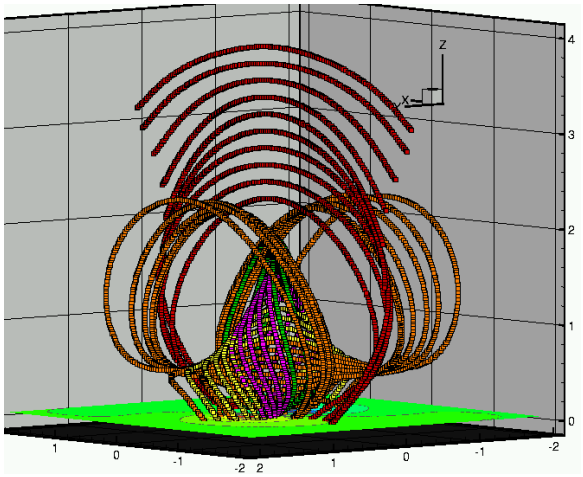


(e) After reconnection

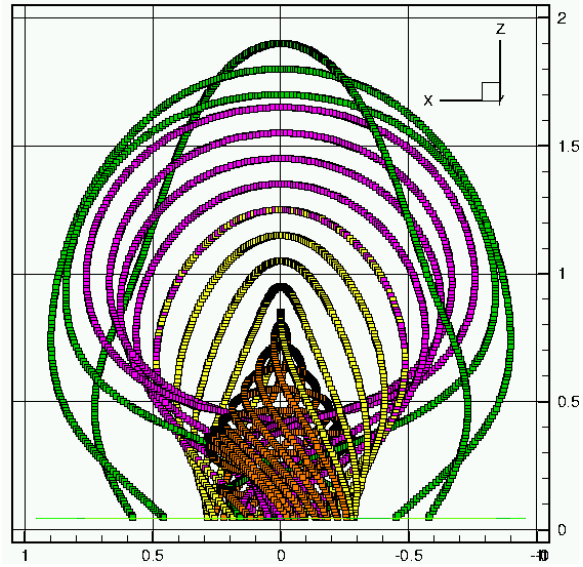


(f) Pre- and post-reconnection field lines

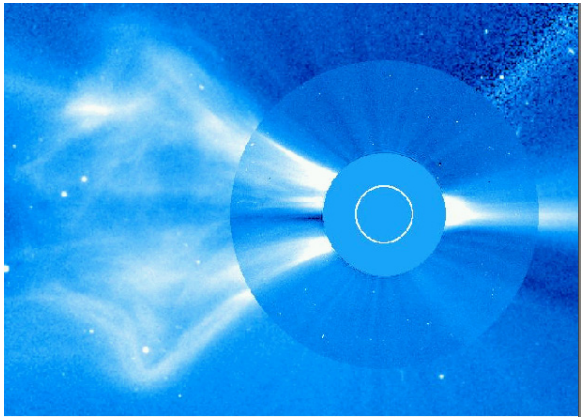
Figure 5.37: Two identified reconnection processes that cause the formation of a), b), c) a flux-rope-like configuration from simulation (B), and c), d), e) a part of a plasmoid-like structure from simulation (C).



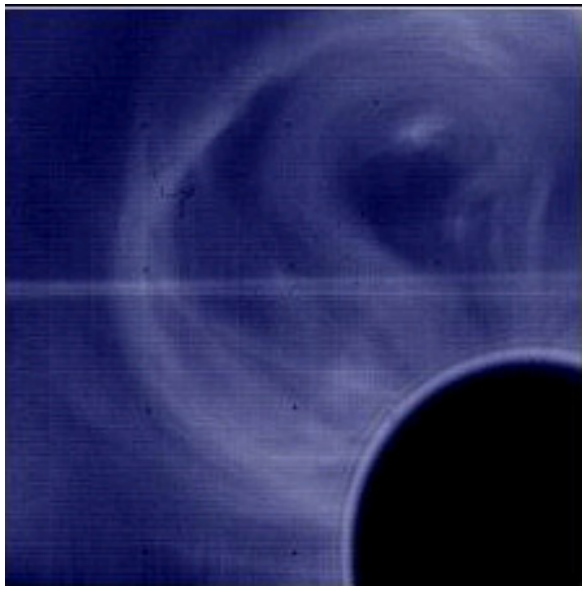
(a) $t = 645\tau_A$, simulation (B)



(b)



(c)



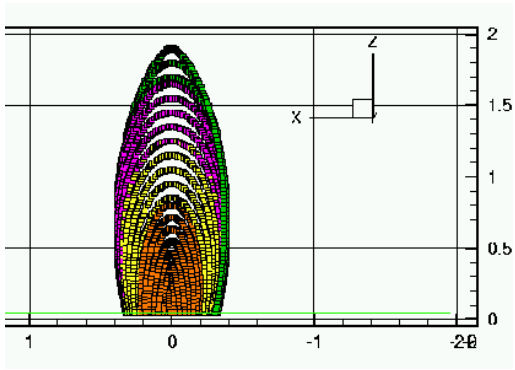
(d)

Figure 5.38: The magnetic field topology produced in a simulations, displayed in figure 5.36 a), b) here in figures a), b), could explain the configuration we see in real CME coronagraph images in c),d).

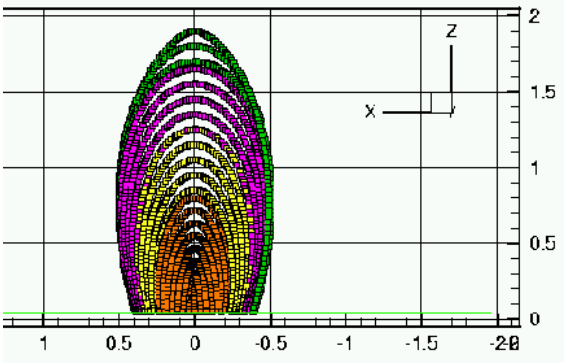
5.3.6 The importance of the Reynolds number

We conclude this chapter with a brief note on the dependence of the numerical solution on the value of the Reynolds number; this number was defined in (5.5) as a ratio of the viscous to Alfven time scales. The majority of numerical MHD studies of the coronal magnetic arcades set the viscosity time scale τ_ν to be of order of $0.01\tau_A$. This is done for numerical rather than physical considerations, because the real viscosity of the coronal plasma is extremely small. Some rough estimates indicate that R in the corona is actually larger than the Lundquist number by several orders of magnitude. While aware of this problem in this work we adopted the conventional strategy of taking a rather low value of the Reynolds number. In this section we justify our approach by studying how the dynamics of the calculation changes if the Reynolds number is increased. We performed several simulations where we kept all of the parameters identical to the simulation with the Dirichlet boundary conditions described in Section 5.3.3 except for the value of the Reynolds number, which we increased. The study showed that while the topological changes that the magnetic field undergoes in the course of the simulation did not change for the larger values of R, the time rate at which these changes occur became much slower. Consider, for example, the simulation where we set $R = 100$. Figure 5.39 shows the magnetic configuration obtained at $t = 147\tau_A$. In general, we see the same type of twisting motion in this simulation as we encountered in the simulations of section 5.3.3. If we compare Figure 5.39 to the magnetic configuration in the simulation with $R = 10$ shown in Figure 5.25 at $t = 133\tau_A$ we can see that the twisting and shearing proceeds much faster in the latter case. We have to perform a more detailed study to find a full explanation for this behavior of the numerical solution. At this moment we can only speculate about the causes for the slow down in the twisting rate of the magnetic field lines as the Reynolds number increases. Here we mention one possible explanation. Consider two open containers with fluids of different viscosities. If we introduce a perturbation in the shape of rotating vortices at the surface of each fluid, then we can expect that the rotational motion will propagate into the container better for more viscous fluid. A similar process occurs for more viscous plasma as the twist in the magnetic field lines driven by a velocity flow at the boundary is propagated faster if the viscosity is larger. We have to carry out more numerical simulations to verify that this explanation is, in fact, valid. For now, we will just indicate that the only difference in the dynamics

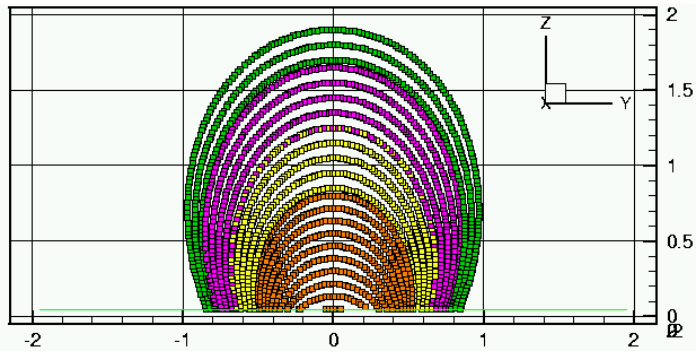
that we observed for the simulation with increased Reynolds number was the rate at which the twisting occurs. Since the real value of the viscosity in the corona is very low, it could be argued that the puzzling long term stability of the coronal magnetic arcades before the eruption could be obtained in a numerical simulation by giving the Reynolds number a very large value.



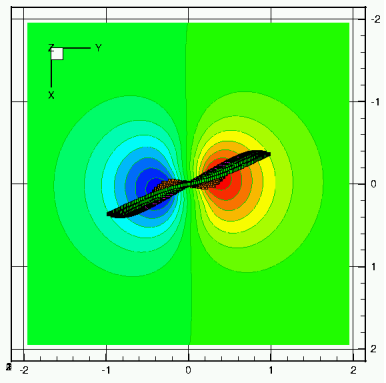
(a) $t = 147\tau_A$, xz plane view



(b) $t = 297\tau_A$, xz plane view



(c) $t = 147\tau_A$, yz plane view



(d) $t = 147\tau_A$, top view

Figure 5.39: The resulting magnetic configuration from a simulation with the Reynolds number $R = 100$ (Dirichlet boundary conditions are imposed on $\mathbf{B}$).

Chapter 6 Conclusions and future work

6.1 Summary and conclusions

In this work we have analyzed new exponential propagation methods and used them to model physical processes which determine the dynamics of the solar coronal plasma. In the first two chapters of the thesis we introduced the ideas of exponential propagation and the Krylov subspace approximation to the functions of matrices and showed how these concepts can be used to construct an efficient numerical technique for time integration of large stiff systems of differential equations. Our analysis of the proposed exponential propagation methods highlighted important issues in developing such techniques and outlined the procedure for building new methods of this type. We also demonstrated the advantages that the exponential propagation technique has compared to the explicit and implicit schemes. In particular, in a numerical example of Section 3.5.3 we presented a quantitative comparison of the exponential Runge-Kutta method and the explicit Runge-Kutta scheme and showed that the former provides an efficient and stable scheme for integration of differential equations with the time step greatly exceeding the CFL bound. We have discussed the issues of practical implementation of exponential propagation method, including its favorable parallel scalability and the computational resource requirements. In general, we demonstrated that exponential propagation methods provide a promising alternative to explicit and implicit schemes for efficient numerical integration of stiff systems of differential equations.

In Chapters 4 and 5 we applied exponential propagation methods to a particular application in solar physics and presented new theories based on the results of our numerical simulations. We have validated our numerical approach on two existing models. Our computations of the evolution of two-dimensional magnetic arcades highlighted an important topological feature of the reconnection process that, to our knowledge, was not emphasized before (Fig. 4.10). Our studies demonstrated that exponential propagation methods can be effectively used for solution of both dynamical systems of equations and steady state problems. In particular, we have computed the evolution of two-dimensional magnetic arcades and also three-dimensional linear force-free states of MHD plasma. We furthermore de-

scribed the design of a code, based on the exponential propagation techniques, that models nonlinear force-free states of plasma configurations. We discussed the advantages of the exponential propagation methods over other numerical techniques currently used in numerical magnetohydrodynamics. In particular, we demonstrated how the automatic error control mechanism available for the exponential propagation methods allows an assessment of the accuracy of the numerical solution. Comparisons with explicit methods clearly showed the advantages of the exponential propagation technique for resistive MHD systems.

Finally, we presented the results of a numerical three-dimensional model of the evolution of the coronal magnetic arcades in response to the motion of the footpoints. We proposed a novel formulation of the boundary conditions, inspired by the laboratory simulations of prominence eruptions. Our numerical simulations lead us to a theory which offeres an explanation for the topologically complex structure and evolution of the eruptive plasma configurations in the corona. We proposed that the observed structural diversity of different eruptive events can be explained as different angle projections of a unique complex three-dimensional magnetic topology. We have compared the dynamics of the magnetic arcades obtained in the simulation with the observational studies and demonstrated their agreement. Several numerical tests confirmed the accuracy of the solution and indicated how it depends on the values of the parameters in the simulations.

6.2 Future work

There are two major research directions we plan to pursue. First, we will work on developing new exponential-propagation-type methods and extend the application of exponential propagation techniques to problems in fields other than plasma physics. In particular, we will investigate the possibilities of combining a multistep-type approach and exponential propagation concepts to create new numerical schemes. We believe a theoretical study could be carried out which would help to identify the types of problems that would benefit from the use of exponential propagation methods. More specifically, we would like to determine the classes of differential operators that can be exponentially propagated to obtain an accurate numerical solution to corresponding evolution problems. We could also try to derive better error estimates for specific problems. Another interesting issue to explore is the effect of different discretization schemes for spatial differential operators on the time integration

scheme. We would also like to study the application of exponential propagation methods to hyperbolic systems. In particular, we want to investigate whether the combination of the high-order upwind-type spatial discretization and the exponential propagation in time will yield an effective numerical method for hyperbolic equations. Our experiments with Burger's equation indicate that this could be a promising approach for such problems. Since the linear exponential propagation methods of Chapter 1 are much easier to implement and require less flops per time step than the exponential methods for nonlinear problems described in Chapter 2, we would also like to explore the following idea. Suppose we need to solve a nonlinear system of equations in which each equation is linear in the variable it evolves (MHD is an example of such system). Then we could use a predictor-corrector-type approach and integrate the equations using a linear exponential propagation method consecutively, substituting the new values for the evolved variables into the next equation to be integrated. Surely, such an approach will reduce the size of a time step in the integration to maintain the accuracy of the solution; the savings obtained from a reduction of the number of flops per time iteration could, on the other hand, overweigh the decrease of the time step.

We would also like to make the following improvements to our current code. First, we plan to develop a version of the code for distributed memory parallel machine using MPI. This will allow us to run larger problems and study the scalability of the methods. Due to the structure of the exponential propagation methods we expect them to be highly parallelizable and allow computations with large spatial grids. The resistive MHD part of the code will be rewritten using different grid structure and discretization for the spatial differential operators. Using a staggered grid and natural discretizations for the divergence, gradient and curl operators developed by Hyman and Shashkov [42, 41] will allow us to eliminate errors in the divergence of the magnetic field. Such a grid could also be made nonuniform to reduce the grid size for problems and capture the small spatial scales of the system. Finally, we would like to include a multigrid solver which would compute the initial magnetic configurations from an arbitrary distribution of the normal component of the magnetic field at the boundary. This improvement would be useful for problems of reconstruction of the coronal magnetic topology from magnetogram data (see paragraph below). Further study of the boundary conditions in the problem will also be performed.

Another direction for our research lies in further investigations based on our resistive MHD model of the coronal plasma configurations. We plan to carry out more studies of

the dependence of the behavior of the numerical solution on the parameter values, e.g. the size of the magnetic configuration, the values of the Lundquist and Reynolds numbers, the maximum magnitude and the rate of change of the velocity. A further validation of our theory will benefit from a collaboration with observers and more comparisons of results from the numerical and observational studies. It would also be interesting to explore the predictive capabilities of our model by running simulations with the parameters scaled to correspond to coronal values at a particular time. Our numerical approach can also provide efficient means of solving another problem in coronal modeling - the reconstruction of the coronal magnetic topology from magnetogram data [54]. A pseudo-evolutionary approach of Klimchuk *et al.* [47] can be used together with the exponential propagation method to compute the nonlinear force-free states of the plasma configurations to accomplish this task. As our understanding of the behavior of the coronal magnetic arcades progresses, we will extend our model to include additional physical processes. We will incorporate the continuity and energy equations into the model and study how the nonuniform density profile and non-zero β assumptions change the behavior of the numerical solution. This modification of the model will allow us to generate images qualitatively similar to those obtained by observational instruments and then compare them directly. It would be interesting to incorporate the actual magnetogram data as boundary conditions in our computations and compare the results with the observed structures. We would also like to pursue the modeling of the laboratory simulations of the prominence eruptions. We plan to investigate how the simulation parameters should scale to match the experimental values and compare the images obtained from the computations and experiments.

Bibliography

[1] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff ODE's. *SIAM Journal of Numerical Analysis*, 14:1006–1021, 1977.

[2] R. Alt. *Méthodes A-stables pour l'intégration de systémes différentielles mal conditionnés.* Thése, Univ. Paris VI, 1971.

[3] T. Amari, J.J. Aly, J.F. Luciani, T.Z. Boulmezaoud, and Z. Mikić. Reconstructing the solar coronal magnetic field as a force-free magnetic field. *Solar Physics*, 174:129–149, 1997.

[4] T. Amari, J.F. Luciani, J.J. Aly, and Z. Mikic. Opening solar magnetic fields: some analytical and numerical MHD aspects. In N. Crooker, J.A. Joselyn, and J. Feyman, editors, *Coronal Mass Ejections*, volume 99 of *Geophysical Monograph*, pages 101–109. American Geophysical Union, 1997.

[5] T. Amari, J.F. Luciani, J.J. Aly, and M. Tagger. Plasmoid formation in a single sheared arcade and application to coronal mass ejections. *Astronomy and Astrophysics*, 306:913–923, 1996.

[6] T. Amari, J.F. Luciani, J.J. Aly, and M. Tagger. Very fast opening of a three-dimensional twisted magnetic flux tube. *The Astrophysical journal*, 466:L39–L42, July 1996.

[7] T. Amari, J.F. Luciani, and P. Joly. A preconditioned semi-implicit method for magnetohydrodynamics equations. *SIAM Journal of Scientific Computing*, 21(3):970–986, 1999.

[8] T. Amari, J.F. Luciani, Z. Mikic, and J. Linker. Three-dimensional solutions of magnetohydrodynamic equations for prominence magnetic support: twisted magnetic flux rope. *The Astrophysical Journal*, 518:L57–L60, 1999.

[9] S.K. Antiochos, C.R. DeVore, and J.A. Klimchuk. A model for solar coronal mass ejections. *Astrophysical Journal*, 510:485–493, 1999.

[10] W.E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[11] P.M. Bellan. *Spheromaks : a practical application of magnetohydrodynamic dynamos and plasma self-organization.* Imperial College Press, 2000.

[12] D. Biskamp and H. Welter. Magnetic arcade evolution and instability. *Solar Physics*, 120(1):49–77, 1989.

[13] D.S. Scott B.N. Parlett. The Lanczos algorithm with selective orthogonalization. *Mathematics of Computation*, 33(145):217–238, January 1979.

[14] J.U. Brackbill. FLIP MHD: A particle-in-cell method for magnetohydrodynamics. *Journal of Computational Physics*, 96:163–192, 1991.

[15] J.U. Brackbill and D.C. Barnes. The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations. *Journal of Computational Physics*, 35:426–430, 1980.

[16] J.C. Butcher. Coefficients for the study of runge-kutta integration processes. *Journal of Australian Mathematical Society*, 3:185–201, 1963.

[17] C. Van Loan C. Moler. Nineteen dubious ways to compute the exponential of a matrix. *SIAM review*, 20(4):801–836, October 1979.

[18] G.S. Choe and C.Z. Cheng. A model of solar flares and their homologous behavior. submitted to Astrophysical Journal, 2000.

[19] N. Crooker, J.A. Joselyn, and J. Feyman. *Coronal Mass Ejections.* Geophysical Monograph. American Geophysical Union, 1999.

[20] M. Crouzeix. *Sur l'approximation des équations différentielles opérationnelles linéaires par de méthodes de Runge-Kutta.* PhD thesis, Univ. Paris VI, 1975.

[21] H.A. Van der Vorst. An iterative solution method for solving $f(a)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix a. *Journal of Computational and Applied Mathematics*, 18:249–263, 1987.

[22] C.R. DeVore. Flux-corrected transport techniques for multidimensional compressible magnetohydrodynamics. *Journal of Computational Physics*, 92:142–160, 1991.

[23] W.S. Edwards, L.S. Tuckerman, R.A. Friesner, and D.C. Sorensen. Krylov methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 110:82–102, 1994.

[24] S.A.E.G. Falle, S.S. Komissarov, and P. Joarder. A multidimensional upwind scheme for magnetohydrodynamics. *Monthly notices of the royal astronomical society*, 297:265–277, 1998.

[25] J.M. Finn and T.M. Antonsen. Magnetic helicity: what is it and what is it good for? *Comments on Plasma Physics and Controlled Fusion*, 9:111–120, 1985.

[26] J.M. Finn and J. Chen. Equilibrium of solar coronal arcades. *Astrophysical Journal*, 349(1):345–361, 1990.

[27] J.M. Finn, P.N. Guzdar, and J. Chen. Fast plasmoid formation in double arcades. *Astrophysical Journal*, 393(2):800–814, 1992.

[28] John M. Finn, Parvez N. Guzdar, and Daniel Usikov. Three-dimensional force-free looplike magnetohydrodynamic equilibria. *The Astrophysical Journal*, 427:475–482, May 1994.

[29] R.A. Friesner, L.S. Tuckerman, B.C. Dornblaser, and T.V. Russo. A method for exponential propagation of large systems of stiff nonlinear differential equations. *Journal of Scientific Computing*, 4(4):327–354, 1989.

[30] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM Journal of on Scientific and Statistical Computing*, 13(5):1236–1264, September 1992.

[31] C. W. Gear. *Numerical initial value problems in ordinary differential equations.* Prentice-Hall, Englewood Cliffs, NJ, 1971.

[32] C.W. Gear and Y. Saad. Iterative solution of linear equations in ODE codes. *SIAM Journal on Scientific and Statistical Computing*, 4(4):583–601, 1983.

[33] Gene Golub and Charles F. Van Loan. *Matrix computations.* The Johns Hopkins University Press, 1989.

[34] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations*, volume II. Springer, 1996.

[35] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations*, volume I. Springer, 1996.

[36] D.S. Harned and W. Kerner. Semi-implicit method for three-dimensional compressible magnetohydrodynamic simulations. *Journal of Computational Physics*, 60:62–75, 1985.

[37] D.S. Harned and D.D. Schnack. Semi-implicit method for long time scale magnetohydrodynamic computations in three dimensions. *Journal of Computational Physics*, 65(1):57–70, 1986.

[38] G.J. Hirasaki and J.C. Hellums. A general formulation of the boundary conditions on the vector potential in three-dimensional hydrodynamics. *Quarterly Of Applied Mathematics*, XXVI(3):331–342, 1968.

[39] Marlis Hochbruck and Christian Lubich. On Krylov subspace approximations to the matrix exponential. *SIAM Journal on Numerical Analysis*, 34(5):1911–1925, October 1997.

[40] Marlis Hochbruck, Christian Lubich, and Hubert Selhofer. Exponential integrators for large systems of differential equations. *SIAM Journal on Scientific Computing*, 19(5):1552–1574, September 1998.

[41] J.M. Hyman and M. Shashkov. Adjoint operators for the natural discretizations of the divergence, gradient and curl on logically rectangular grids. *Applied Numerical Mathematics*, 25:413–442, 1997.

[42] J.M. Hyman and M. Shashkov. Natural discretizations for the divergence, gradient, and curl on logically rectangular grids. *Computers and Mathematics with Applications*, 33(4):81–104, 1997.

[43] B. Inhester, J. Birn, and M. Hesse. The evolution of line-tied coronal arcades including a converging footpoint motion. *Solar Physics*, 138(2):257–281, 1992.

[44] P.A. Isenberg, T.G. Forbes, and P. Demoulin. Catastrophic evolution of a force-free flux rope: a model for eruptive flares. *Astrophysical Journal*, 417:368–386, 1993.

[45] P. Kaps and P. Rentrop. Generalized runge-kutta methods of order four with stepsize control for stiff ordinary differential equations. *Numerical Mathematics*, 33:55–68, 1979.

[46] J.A. Klimchuk. Theory of coronal mass ejections. In P. Song, G. Siscoe, and H. Singer, editors, *Space Weather*, AGU Monograph. American Geophysical Union, 2000.

[47] J.A. Klimchuk, S.K. Antiochos, and D. Norton. Twisted coronal magnetic loops. *Astrophysical Journal*, 2000. submitted.

[48] J.A. Klimchuk and P.A. Sturrock. Force-free magnetic fields - is there a loss of equilibrium. *Astrophysical Journal*, 345(2):1034–1041, 1989.

[49] M.A. Kurdi. *Stable high order methods for time discretization of stiff differential equations.* PhD thesis, Univ. of California, 1974.

[50] John D. Lambert. *Numerical methods for ordinary differential systems: the initial value problem.* John Wiley & Sons, 1991.

[51] Peter Lancaster. *Theory of matrices.* Academic Press, Inc., 1969.

[52] K. Lerbinger and J.F. Luciani. A new semi-implicit method for MHD computations. *Journal of Computational Physics*, 97:444–459, 1991.

[53] J.A. Linker and Z. Mikić. Disruption of a helmet streamer by photospheric shear. *The Astrophysical Journal*, 438:L45–L48, 1995.

[54] A.N. Mcclymont, L. Jiao, and Z. Mikić. Problems and progress in computing three-dimensional coronal active region magnetic fields from boundary data. *Solar Physics*, 174(1-2):191–218, 1997.

[55] Z. Mikic. Magnetohydrodynamic modeling of the solar corona. *Physics of Fluids B*, 2(6):1450–1454, 1989.

[56] Z. Mikic, D.C. Barnes, and D.D. Schnack. Dynamical evolution of a solar coronal magnetic field arcade. *The Astrophysical Journal*, 328:830–847, May 1988.

[57] Z. Mikić and J.A. Linker. The initiation of coronal mass ejections by magnetic shear. In N. Crooker, J.A. Joselyn, and J. Feyman, editors, *Coronal Mass Ejections*, volume 99 of *Geophysical Monograph*, pages 57–64. American Geophysical Union, 1997.

[58] K. Murawski and T. Tanaka. Modern numerical schemes for solving magnetohydrodynamic equations. *Astrophysics and Space Science*, 254:187–210, 1997.

[59] Y. Nakagawa. Evolution of magnetic field and atmospheric responses, ii. Formulation of proper boundary equations. *The Astrophysical Journal*, 247:719–733, 1981.

[60] A. Nauts and R.E. Wyatt. New approach to many-state quantum dynamics: The recursive-residue-generation method. *Physical Review Letters*, 51(5):2238–2241, 1983.

[61] D. Nicholson. *Introduction to plasma theory.* Krieger Publishing Company, 1983.

[62] S. P. Nørsett. Semi-explicit Runge-Kutta methods. Technical Report 6/74, Dept. of Math., Univ. of Trondheim, 1974.

[63] T.J. Park and J.C. Light. Unitary quantum time evolution by iterative Lanczos reduction. *Journal of Chemical Physics*, 85:5870–5876, 1986.

[64] U. Platt and T. Neukirch. Theoretical study of onset conditions for solar eruptive processes - influence of the boundaries. *Solar Physics*, 153(1-2):287–306, 1994.

[65] P.M.Bellan and J.F.Hansen. Laboratory simulations of solar prominence eruptions. *Physics of Plasmas*, 5(5):1991–2000, May 1998.

[66] K.G. Powell. An approximate Riemann solver for magnetohydrodynamics (that works in more than one dimension). ICASE Report 194902, NASA Langley Research Center, April 1994.

[67] E.R. Priest. *Solar magnetohydrodynamics.* D.Reidel Publishing Company, 1982.

[68] E.R. Priest and T. Forbes. *Magnetic reconnection: MHD theory and applications.* Cambridge University Press, 2000.

[69] H. H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *Computer Journal*, 5:329–330, 1962/63.

[70] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential. *SIAM Journal on Numerical Analysis*, 29(1):209–228, February 1992.

[71] Yousef Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, 1996.

[72] D.D. Schnack, D.C. Barnes, Z. Mikic, D.S. Harned, and E.J. Caramana. Semi-implicit magnetohydrodynamic calculations. *Journal of Computational Physics*, 70:330–354, 1987.

[73] W.L. Seward, G. Fairweather, and R.L. Johnston. A survey of high-order methods for the numerical integration of semidiscrete parabolic problems. *IMA Journal of Numerical Analysis*, 4:375–425, 1984.

[74] J.A. Shercliff. *A textbook of magnetohydrodynamics.* Pergamon Press, 1965.

[75] A.C. Sterling, H.S. Hudson, B.J. Thompson, and D.M. Zarro. *YOHKOH* SXT and *soho* EIT observations of sigmoid-to-arcade evolution of structures associated with halo coronal mass ejections. *The Astrophysical Journal*, 532:628–647, 2000.

[76] J. Stoer and R. Bulirsch. *Introduction to numerical analysis.* Springer-Verlag, 1993.

[77] John C. Strikwerda. *Finite difference schemes and partial differential equations.* The Wadswarth & Brooks/Cole Mathematics Series, 1989.

[78] T. Strom. On logarithmic norms. *SIAM Journal of Numerical Analysis*, 12:741–753, 1975.

[79] M.T. Sun, S.T. Wu, and M. Dryer. On the time-dependent boundary conditions of magnetohydrodynamic flows. *Journal of Computational Physics*, 116:330–342, 1995.

[80] L. Woltjer. A theorem on force-free fields. *Proceedings of National Academy of Science*, 44:489–491, 1958.

[81] S.T. Wu and J.F. Wang. Numerical tests of a modified full implicit continuous Eulerian (FICE) scheme with projected normal characteristic boundary conditions for MHD flows. *Computer Methods in Applied Mechanics and Engineering*, 64:267–282, 1987.

[82] A.L. Zachary, A. Malagoli, and P. Colella. A higher-order Godunov method for multidimensional ideal magnetohydrodynamics. *SIAM Journal of Scientific Computing*, 15(2):263–284, 1994.