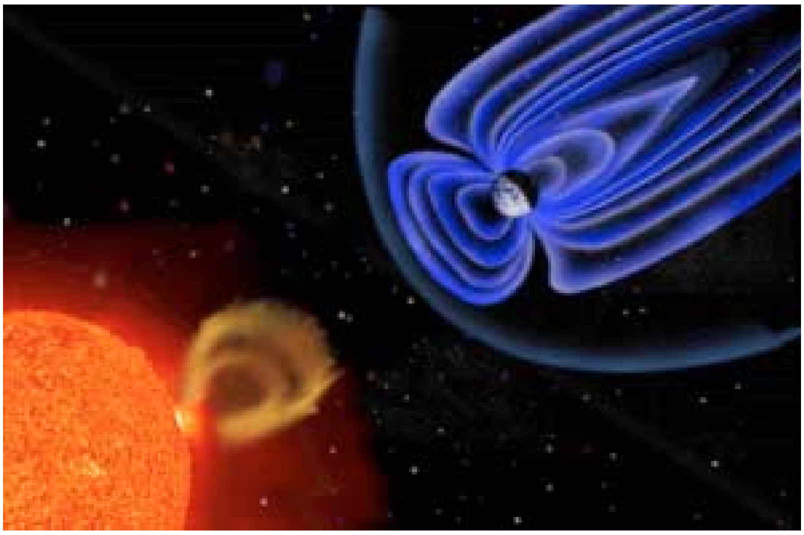
2002:316 CIV

LULEÅ UNIVERSITY OF TECHNOLOGY

MASTER'S THESIS

Solar Wind Interaction with the Terrestrial Magnetosphere



LARS G WESTERBERG

MASTER OF SCIENCE PROGRAMME

Department of Applied Physics and Mechanical Engineering
Division of Fluid Mechanics

2002:316 CIV • ISSN: 1402 - 1617 • ISRN: LTU - EX - - 02/316 - - SE

Preface

This thesis work is done at the Division of Fluid Mechanics, Luleå University of Technology. I would like to thank my supervisor Dr. Hans O. Åkerstedt for his great support and valuable assistance during this project. I would also like to thank all people who have helped and encouraged me throughout the years.

Sammanfattning

En studie av strömningen av solvinden kring jordens magnetosfär är gjord. Som förenklad modell används en plan magnetopaus i två dimensioner med stagnationspunkt i ekvatorsplanet. Den teoretiska analysen är gjord genom ordinär störningsräkning. Aktuella samband är härledda ur Navier-Stokes ekvation och MHD ekvationer. En process av speciellt intresse är magnetisk återkoppling. Genom att använda störningsmodellen kan vi göra en analys av gränsskiktet i närhet till den punkt där återkoppling sker.

Förutom teoretisk analys används numerisk analys för att lösa partiella differentialekvationer beskrivande strömningen av solvinden kring magnetopausen. Metoden som används är Chebyshev's kollokationsmetod.

Det visar sig att gränsskiktet norr om återkopplingspunkten är tunnare än söder om densamma. Förutsättningen för att stationär återkoppling skall kunna äga rum, är att hastigheten i återkopplingspunkten är mindre än Alfvénhastigheten. Detta får till följd att området kring magnetopausen där vi kan ha stationär återkoppling är begränsat. Hastigheten uppnår Alfvénhastigheten på ett avstånd av ungefär tio jordradier längs magnetopausen.

Abstract

A study of the flow of the solar wind past the terrestrial magnetosphere is done, using a simplified model in two dimensions describing a planar magnetopause. The theoretical analysis is made through ordinary perturbation technique, using Navier-Stokes and MHD equations. Process of certain interest is magnetic reconnection. By using the perturbation expansion we can get a description of the boundary layer near a reconnection point.

Besides theoretical analysis, numerical analysis is applied on partial differential equations (PDE) describing the flow of the solar wind past the magnetopause. Chebyshev Collocation Method is the current algorithm used.

It is shown that the boundary layer north of the reconnection point is thinner than the layer to the south. In order for steady reconnection to occur, the velocity at the reconnection point must be less than the Alfvén velocity. This means that the area where steady reconnection can occur is limited to the region stretching from the sub-solar point to a distance of about ten earth radii along the magnetopause.

Table of Content

1 Introduction

What has been will bee again,
What has been done will be done again,
There is nothing new under the Sun.
Ecclesiastes 1:9

The magnetosphere plays an important role for our planet. Fact is that without the magnetosphere, the environment would be so hostile that life would have been impossible to exist.

Earth's magnetic field can be considered as a dipole magnet. The direction of the field from south to north tells that the magnetic poles are vice versa to the geographical ones. This due to the definition of the magnetic south pole as the point where the field line enter the body.
The space environment is highly dynamical. The sun throws out enormous clouds consisting of ionised gas - plasma. This plasma flowing from the sun together with the sun's magnetic field and earth's magnetosphere constitute a system that gives great benefits of trying to understand.

Our future is in space. The number of technical systems that depends on the conditions of the space environment is increasing. This does not only yield the systems working in space, but also technical systems on ground or operating in the air.

The application of space physics and how it affects the technical environment goes under the name of Space Weather. Weather is a word connected to dynamical changes, thereby its appearance in this case.

Earth's magnetosphere stops the solar wind from blowing into the surface of the earth. Despite this, layers of trapped particles exist in different shells around the earth. For instance the high-energy particles gives Aurora when they enter the atmosphere. The electrons collide with different kinds of atoms like oxygen and nitrogen, causing excitation. When going back to the original energy state light is emitted and is seen as Aurora. Extreme activity on the sun can cause serious trouble to power transformers and the electricity net. The statement that the particles pass outside the earth thus needs a slight modification. Somehow particles manage to make entry into the magnetosphere. In this thesis one mechanism for the penetration of solar wind plasma into the magnetosphere is considered.

A theoretical concept introduced and accepted in space physics is called magnetic reconnection. Reconnection is a result of a breakdown of certain ideal properties of a plasma. The magnetic reconnection process is described in the next section. The magnetopause is a boundary layer with structure ready to be explored. To do so well known MHD equations and Navier-Stokes equation are used. This is done in chapter 3.

The numerical method used in this project is Chebyshev Collocation Method. It is used to investigate the velocity profile of the plasma flowing along the magnetopause, described by partial differential equations. Results are treated in chapter 4, while appendix A gives a thorough explanation of the numerical procedure. In appendix B the code for actual programs are presented.

2 Magnetic Reconnection

In this section we introduce the concept of magnetic reconnection. The basic theory is introduced, and the physical properties of the process are discussed. The theory is based on a fluid description of the plasma, called Magnetohydrodynamics (MHD). A particle point of view of reconnection exists, but is not covered in this thesis.

Magnetic Reconnection was in the beginning almost a mythical concept introduced by Parker. Some believed in it, some did not. Today it is no doubt whether this process exists or not. The question nowadays in certain cases is if reconnection is the actual process in an observed concrete phenomenon.

Describing reconnection a couple of properties of the plasma have to be considered: A plasma that has been squeezed from the sun is a highly conducting medium. It follows the sun's magnetic field as it travels towards earth. A central property is that the magnetic field is frozen into the plasma. This means that diffusion of the fieldlines through the plasma is impossible. The magnetic field of the earth can be considered as a dipole field, which at the magnetopause points upward, to the geographical north. When the magnetic field originating from the sun, the Inter Planetary Field (IMF), points southward something interesting may happen. When this is the case the two fields have opposite polarity and the process of magnetic reconnection occurs.

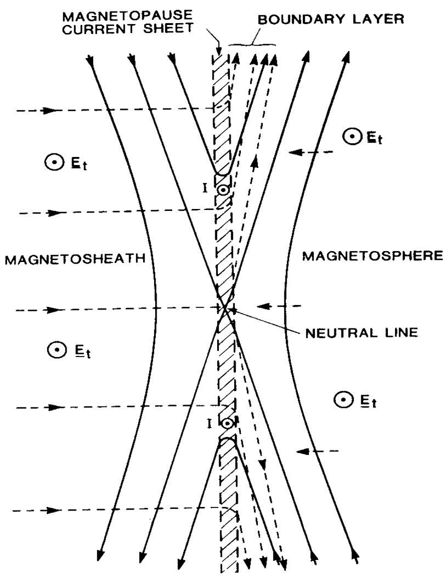


Figure 1: Field line merging.

In figure 1 reconnection at the sub-solar point is shown. The hatched region in the figure represent the magnetopause current sheet(rotational discontinuity), which separates the magnetosheath and magnetospheric fields and the dashed lines representing the plasma stream lines. The effect of magnetic reconnection is to transfer magnetic energy of the solar wind into plasma jets along the reconnected field lines away from the neutral line on either side. The speed of these jets are about twice the Alfvén speed in the magnetosheath.

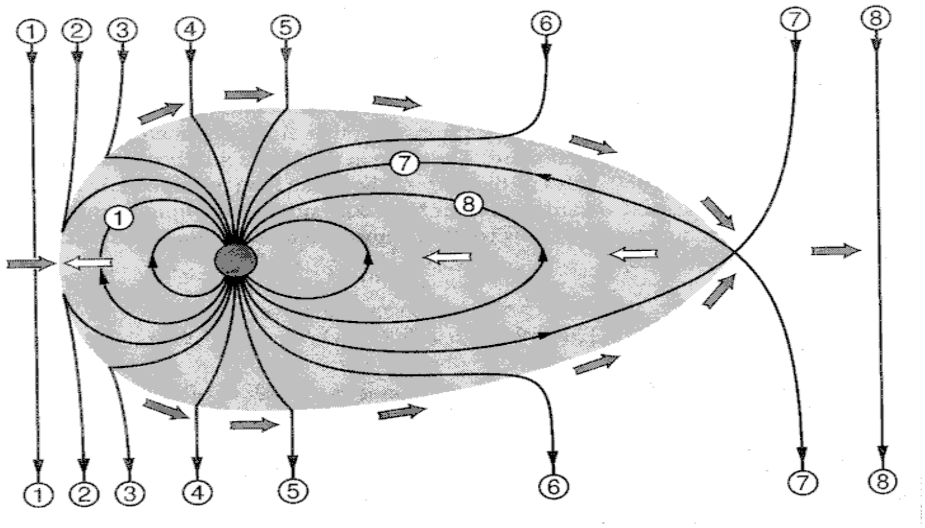


Figure 2: Field line merging and reconnection at the magnetopause

Figure 2 shows a model of how solar wind plasma can enter into the magnetosphere by the mechanism of magnetic reconnection at the sub-solar point, the equator plane. Due to the dynamical pressure from the solar wind the field lines are pushed back to the magnetotail. The field lines are then reconnected on the tailside of the magnetosphere, and due to magnetic tension causing transportation of plasma into the polar regions. Here on ground the result is seen as the Aurora.

On the dayside of the magnetosphere merging keeps peeling off the outermost layer of the magnetopause. Particles can thereby enter the magnetosphere. Magnetic reconnection transforms magnetic energy to kinetic energy. It is the main reason behind the large amounts of energy that are distributed inside the magnetosphere. The reconnection process is however not restricted to the equatorial plane of the magnetosphere. This is only the case near the dayside stagnation point – the subsolar point. Merging can exist anywhere along the magnetopause where the Interplanetary magnetic field have a southward component. One effect of magnetic reconnection at latitudes higher than the position of the polar cusp is a displacement of the cusp

What happens when reconnection occurs is that the frozen in condition breaks down. The field lines can diffuse inside the plasma and the process is on.

The magnetic field variation in a plasma can be represented by the Induction equation

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) + \frac{1}{\mu_0 \sigma_0} \nabla^2 \mathbf{B} \tag{2.1}$$

which is similar to the equation describing the evolution of vorticity in an ordinary fluid.

Using (2.1) in the limit of infinite conductivity it can be shown that the total magnetic induction encircled by a closed loop is constant even if each point on this loop has different local velocity. This frozen in concept implies that all particles and magnetic fields must remain inside the flux tube at all times independent of any motion and change in form of the flux tube.
A flux tube can be visualised as a certain kind of cylinder that at a given time contains a constant amount of magnetic flux.

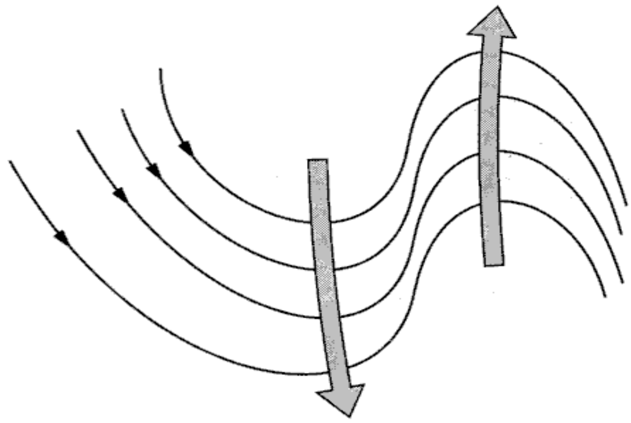


Figure 3: Magnetic field lines moving with the plasma

The definition of the frozen-in concept can be developed. Writing the diffusion equation (2.1) in simple dimensional form gives

$$\frac{B}{\tau} = \frac{VB}{L_B} + \frac{B}{\tau_d} \tag{2.2}$$

where B is the average magnetic field strength and V the average plasma velocity perpendicular to the field. L denotes the characteristic length over which the field varies and τ the characteristic time of magnetic field variations. The first term in RHS of (2.2) describe the convective derivative of the field with respect to the velocity, while the second is the diffusion term. Taking the ratio of the first and second term gives the magnetic Reynolds number

$$R_m = \mu_0 \sigma_0 L_B V \tag{2.3}$$

It is useful to use this to determine whether a medium is flow or diffusion dominated. When the magnetic Reynolds number is large the diffusion term can be neglected and the frozen-in condition is valid. If it is close to unity then diffusion rules the medium.
The magnetic Reynolds number depends on both the conductivity and velocity. A decrease in any of these gives domination to diffusion processes.

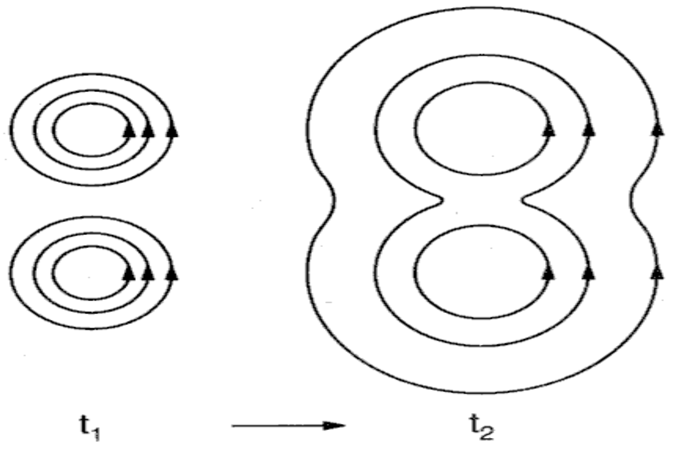


Figure 4: Diffusion of magnetic field lines

The application of reconnection in space plasma physics was first suggested by Dungey in 1961 as the principal magnetopause coupling mechanism. Since then a lot of work has been done on this topic. The basic theory has mainly been used to describe the reconnection process at the sub-solar point. However, as discussed earlier, reconnection is not limited to this region. The purpose of the present master thesis is to analyse the combination of steady magnetic reconnection in the presence of an outer magnetosheath plasma flow. We use a stagnation point model of the magnetosheath flow, Spreiter and Stahara[9]. This is a quite usual description and is for example used by Cowley and Owen[5]. In the stagnation point the velocity is zero. In our analysis we consider the reconnection process to occur at different places along the magnetopause, where we have a flow speed. We want to study the boundary layer north and south of the reconnection point. This gives us the possibilities to find limitations when steady reconnection is possible.

3 Boundary Layer Analysis

The magnetosphere is an obstacle for the solar wind as it flows through space, where the magnetopause is its outermost layer. When the super-sonic solar wind runs into the magnetosphere its velocity is first decreased to sub-sonic velocity in a bow shock. The region between the bow shock and the magnetopause is called the magnetosheath. The magnetopause is the boundary separating the magnetosheath plasma and the plasma inside the magnetosphere. The magnetopause there is actually a transition region – a current sheet(rotational discontinuity). The global length scales are so large that the local scale in the boundary layer perhaps seems a little bit strange comparing to 'ordinary' flow in connection to plates and walls. The thickness of the layer adjacent to the magnetopause is of order 100 km.

In this section a perturbation technique is used to solve the well known equations in MHD.

3.1 Equations and the boundary layer approximation

We consider the plasma motion to be governed by the Navier-Stokes equation

$$\rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \mu\nabla^2\mathbf{u} - \nabla P + \frac{1}{\mu_0}\cdot(\mathbf{B}\cdot\nabla)\mathbf{B} \tag{3.1a}$$

where P is the total pressure

$$\mathrm{p} + \frac{\mathrm{B}^2}{2\mu_0}. \tag{3.1b}$$

The magnetic field is described by the Induction equation

$$\frac{\partial\mathbf{B}}{\partial t} = \nabla\times(\mathbf{u}\times\mathbf{B}) + \frac{1}{\mu\sigma}\nabla^2\mathbf{B}. \tag{3.2}$$

Working with two dimensions, we let

$$\mathbf{u} = (\mathrm{u}^*, \mathrm{v}) \tag{3.3}$$

$$\mathbf{B} = (\mathrm{B}_x, \mathrm{B}_y) \tag{3.4}$$

where we divide the absolute velocity into a de Hoffmann-Teller velocity and u the velocity with respect to the de Hoffmann-Teller frame of reference

$$u^* = u + U_{HT} \tag{3.5}$$

In this frame of reference the electric field is zero and the flow is approximately parallel to the magnetic field.

We assume that the transition separating the magnetosheath plasma from the magnetosphere plasma is thin. We introduce δ as a characteristic thickness.

The boundary layer approximation means that the length scale in the y direction - δ, is considered much smaller than the length scale L in the x direction. Therefore we assume

$$\frac{\partial}{\partial y} = O\left(\frac{1}{\delta^*}\right) \tag{3.6}$$

$$\frac{\partial}{\partial x} = O\left(L^{-1}\right) \tag{3.7}$$

where

$$\delta = \frac{\delta^*}{\mathrm{L}}.(\mathrm{L} = \mathrm{R_E} = \mathrm{Earth\ \ radii}) \tag{3.8}$$

Here the x coordinate is directed from south to north, while the y coordinate points towards the sun.

An order of magnitude estimate shows that as in ordinary boundary layer theory the thickness scales as

$$\delta = R^{-\frac{1}{2}} \tag{3.9}$$

where

$$R = \frac{\mathrm{v_A} \cdot \mathrm{R}_E}{\nu} \tag{3.10a}$$

$$\mathrm{v} = O(\delta) \tag{3.10b}$$

$$\frac{B_x}{B_0} \rightarrow B_x \tag{3.10c}$$

$$\frac{B_y}{B_0} \rightarrow B_y \tag{3.10d}$$

$$\frac{E_z}{\mathrm{v}_A B_0} \rightarrow E_z \tag{3.10e}$$

$$\frac{y}{\delta^*} \rightarrow y \tag{3.10f}$$

$$\frac{x}{L} \rightarrow x \tag{3.10g}$$

and R is the Reynolds number.

Using this the governing dimensionless equations are

$$-\frac{1}{R_m}\left\{\frac{\partial B_x}{\partial y} - \frac{\partial B_y}{\partial x}\right\} = E_z + B_y u + B_y U_{HT} - \mathrm{vB}_x \tag{3.11}$$

$$(u + U_{HT})\left\{\frac{\partial u}{\partial x} + \frac{\partial U_{HT}}{\partial x}\right\} + \mathrm{v} \cdot R^{\frac{1}{2}} \frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} + B_x \frac{\partial B_x}{\partial x} + R^{\frac{1}{2}} \cdot B_y \frac{\partial B_x}{\partial y} + \frac{\partial^2 u}{\partial y^2} \tag{3.12}$$

$$(u + U_{HT})\frac{\partial \mathrm{v}}{\partial x} + \mathrm{v} \cdot R^{1/2} \frac{\partial \mathrm{v}}{\partial y} = -R^{\frac{1}{2}} \frac{\partial P}{\partial y} + B_x \frac{\partial B_y}{\partial x} + R^{\frac{1}{2}} \cdot B_y \frac{\partial B_y}{\partial y} + \frac{\partial^2 \mathrm{v}}{\partial y^2} \tag{3.13}$$

Two other equations coming in handy are one of the Maxwell equations, and the equation describing the property of incompressibility, i.e.

$$\nabla \cdot \overline{B} = 0 \Rightarrow \frac{\partial B_x}{\partial x} + R^{\frac{1}{2}} \frac{\partial B_y}{\partial y} = 0 \tag{3.14}$$

$$\nabla \cdot \overline{u} = 0 \Rightarrow \frac{\partial u}{\partial x} + \frac{\partial U_{HT}}{\partial x} + R^{\frac{1}{2}} \frac{\partial \mathrm{v}}{\partial y} = 0 \tag{3.15}$$

3.2 The perturbation expansion

For the solution of 3.11-3.15 we assume the following perturbation expansion

$$u = U_{HT} + u^{(0)} + R^{-\frac{1}{2}} u^{(1)} + \ldots \tag{3.16}$$

$$\mathrm{v} = \mathrm{v}^{(0)} + R^{-\frac{1}{2}} \mathrm{v}^{(1)} + \ldots \tag{3.17}$$

$$B_x = B_x^{(0)} + R^{-\frac{1}{2}} B_x^{(1)} + \ldots \tag{3.18}$$

$$B_y = B_y^{(0)} + R^{-\frac{1}{2}} B_y^{(1)} + \ldots \quad . \tag{3.19}$$

The magnetic Reynolds number is assumed to be of the same order as the ordinary Reynolds number.

Next step is to plug in the perturbation expansion into the equations (3.11) – (3.15).

This gives to the lowest order for (3.12)

$$O\left(R^{\frac{1}{2}}\right): \ \mathrm{v}^{(0)} \frac{\partial u^{(0)}}{\partial y} = B_y^{(0)} \frac{\partial B_x^{(0)}}{\partial y} \tag{3.20}$$

From the Induction equation we get to the lowest order

$$B_x^{(0)} = \frac{B_y^{(0)} \cdot u^{(0)}}{\mathrm{v}^{(0)}} \tag{3.21}$$

(3.14) and (3.15) gives

$$\begin{aligned} B_y^{(0)} &= B_y^{(0)}(x) \\ \mathrm{v}^{(0)} &= \mathrm{v}^{(0)}(x) \end{aligned} \tag{3.22}$$

Inserted into (3.20) leads to following relation

$$\frac{\partial u^{(0)}}{\partial y} \left\{ \mathrm{v}^{(0)} - \frac{B_y^{(0)^2}}{\mathrm{v}^{(0)}} \right\} = 0 \, . \tag{3.23}$$

The derivative must be different from zero, which leads to the fact that the parenthesis must be equal to zero. This leads to two possibilities that we refer to as Northern and Southern reconnection.

$$B_x^{(0)} = +u^{(0)} \qquad B_y^{(0)} = +\mathrm{v}^{(0)} \qquad \boxed{\text{(3.24) Northern Reconnection}}$$

$$B_x^{(0)} = -u^{(0)} \qquad B_y^{(0)} = -\mathrm{v}^{(0)} \qquad \boxed{\text{(3.25) Southern Reconnection}}$$

The terminology Northern and Southern is referred to when the process takes place north or south of the reconnection point, see figure 1.

By studying (3.13) to lowest order one of the important properties of a boundary layer is verified

$$O\left(R^{\frac{1}{2}}\right): \frac{\partial P}{\partial y} = 0 \tag{3.26}$$

i.e. the total pressure P is constant through the boundary layer.

3.3 Northern Reconnection

Next step is to analyse the two different cases; northern and southern reconnection, as was derived in the previous section. As earlier the basis of the analysis are the expressions obtained by the perturbation model.

To the next order the x-component of the Navier-Stokes equation becomes

$$O\left(R^0\right):\ u^{(0)}\frac{\partial u}{\partial x}^{(0)}+u^{(0)}\frac{dU_{HT}}{dx}+U_{HT}\frac{\partial u}{\partial x}^{(0)}+U_{HT}\frac{dU_{HT}}{dx}+\mathrm{v}^{(1)}\frac{\partial u}{\partial y}^{(0)}+\mathrm{v}^{(0)}\frac{\partial u}{\partial y}^{(1)}=$$

$$=-\frac{\partial P}{\partial x}+B_x^{(0)}\frac{\partial B_x^{(0)}}{\partial x}+B_y^{(0)}\frac{\partial B_x^{(1)}}{\partial y}+B_y^{(1)}\frac{\partial B_x^{(0)}}{\partial y}+\frac{\partial^2 u}{\partial y^2}^{(0)} \tag{3.27}$$

Differentiation of the Induction equation to the next order with respect to y, we get

$$O\left(R^{-\frac{1}{2}}\right):-\frac{R}{R_m}\frac{\partial^2 B_x^{(0)}}{\partial y^2}=\frac{\partial B_y^{(1)}}{\partial y}u^{(0)}+B_y^{(1)}\frac{\partial u}{\partial y}^{(0)}+\frac{\partial B_y^{(1)}}{\partial y}U_{HT}-\mathrm{v}^{(0)}\frac{\partial B_x^{(1)}}{\partial y}-\frac{\partial \mathrm{v}^{(1)}}{\partial y}B_x^{(0)}-$$

$$-\mathrm{v}^{(1)}\frac{\partial B_x^{(0)}}{\partial y}+B_y^{(0)}\frac{\partial u}{\partial y}^{(1)} \tag{3.28}$$

By using (3.14) and (3.15) an expression for the derivative of the de Hoffmann-Teller velocity is obtained. Combining the lowest order of (3.14) and (3.15) results in

$$\mathrm{v}^{(1)}-B_y^{(1)}=-\frac{dU_{HT}}{dx}\cdot y+f(x) \tag{3.29}$$

where f(x) is an arbitrary function of x.

Analysing possible boundary conditions in the limit when y goes to infinity, we choose the following

$$v^{(1)}\rightarrow-\frac{dU_{HT}}{dx}\cdot y \tag{3.30}$$

$$B_y^{(1)}\rightarrow 0 \tag{3.31}$$

implying that f(x) is equal to zero.

Combining (3.27) and (3.28) finally gives the equation for the Northern Reconnection case

$$\left(1+\frac{R}{R_m}\right)\frac{\partial^2 u}{\partial y^2}^{(0)} + 2y\frac{dU_{HT}}{dx}\frac{\partial u}{\partial y}^{(0)} - 2U_{HT}\frac{\partial u}{\partial x}^{(0)} = \frac{\partial P}{\partial x} + U_{HT}\frac{dU_{HT}}{dx} \tag{3.32}$$

3.4 Southern Reconnection

The process for this case is the same as for the previous one. During the derivation from the Navier-Stokes equation and the Induction equation a couple of sign changes are experienced compared with the northern reconnection. In the end it turns out that the equation describing southern reconnection is the same as (3.32).

However, although the equations are identical, the de Hoffmann-Teller velocity is different depending on which case that is taken to consideration

3.5 Solution

Starting with the equation

$$\alpha\frac{\partial^2 u^{(0)}}{\partial y^2}+2y\frac{dU_{HT}}{dx}\frac{\partial u^{(0)}}{\partial y}-2U_{HT}\frac{\partial u^{(0)}}{\partial x}=\frac{\partial P}{\partial x}+U_{HT}\frac{dU_{HT}}{dx} \qquad (3.33)$$

where

$$\alpha=\left(1+\frac{R}{R_m}\right) \qquad (3.34)$$

we seek to find a self-similar solution, so we put

$$\eta=y\cdot g(x) \qquad (3.35)$$

$$\mathrm{X}=x.$$

The function g is the inverse of the thickness (δ) of the boundary layer.

This gives the partial derivatives

y=η/g

$$\frac{\partial}{\partial x}=\frac{\partial}{\partial \mathrm{X}}\frac{\partial \mathrm{X}}{\partial x}+\frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial x}=\frac{\partial}{\partial \mathrm{X}}+\frac{\partial}{\partial \eta}y\cdot g'(x) \qquad (3.36)$$

$$\frac{\partial}{\partial y}=\frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial y}=g(x)\frac{\partial}{\partial \eta}\Rightarrow\frac{\partial^2}{\partial y^2}=g^2\frac{\partial^2}{\partial \eta^2}. \qquad (3.37)$$

Chain Rule

The transformed equation becomes

$$\alpha\cdot g^2\frac{\partial^2 u^{(0)}}{\partial \eta^2}+2\eta\frac{dU_{HT}}{d\mathrm{X}}\frac{\partial u^{(0)}}{\partial \eta}-2U_{HT}\left\{\frac{\partial u^{(0)}}{\partial \mathrm{X}}+\frac{\partial u^{(0)}}{\partial \eta}\frac{\eta}{g}\cdot g'\right\}=$$
$$=\frac{\partial P}{\partial x}+U_{HT}\frac{dU_{HT}}{dx} \qquad (3.38)$$

The homogeneous solution is given by the equation

$$u^{(0)}=f(\eta)\Rightarrow f''+\underbrace{\left(\frac{1}{\alpha g^2}\frac{dU_{HT}}{d\mathrm{X}}-U_{HT}\frac{1}{\alpha g^3}g'\right)}_{=1}\cdot 2\eta f'=0 \qquad (3.39)$$

The parenthesis expression has to be a constant since f only depends on η. This gives the differential equation

$$g' = \frac{1}{U_{HT}} \frac{dU_{HT}}{dx} \cdot g - \frac{\alpha}{U_{HT}} g \tag{3.40}$$

with two possible solutions

$$g = \pm \frac{U_{HT}}{\sqrt{2\alpha \int_0^{X} U_{HT} \, d\chi}} \tag{3.41}$$

giving

$$\delta = \pm \frac{\sqrt{2\alpha \int_0^{X} U_{HT} \, d\chi}}{U_{HT}} \tag{3.42}$$

where the boundary layer thickness is assumed zero at X=0 - the reconnection point. The plus or minus sign depends on whether we have northern or southern reconnection. The thickness will be discussed further on as the de Hoffmann-Teller velocity at X=0 is derived.

The homogeneous equation (3.39) then becomes

$$f'' + 2\eta \cdot f' = 0 \tag{3.43}$$

with solution

$$f = C_1 \cdot erf\left(\eta\right) + C_2 . \tag{3.44}$$

Next step is to find the particular solution. The right hand side in (3.38) depends only on X, so the solution must be independent of X.

$$\tilde{u}^{(0)} = \tilde{u}^{(0)}(\mathrm{X}) \tag{3.45}$$

Plugging in this expression in (3.38) gives the particular solution

$$\tilde{u}^{(0)} = -\frac{1}{2}\int_0^{X} \frac{1}{U_{HT}} \frac{\partial P}{\partial \chi} d\chi - \frac{1}{2} U_{HT} . \tag{3.46}$$

Total velocity

$$u^{(0)} + U_{HT} = C_1 \cdot erf(\eta) - \frac{1}{2}\int_0^{X} \frac{1}{U_{HT}} \frac{\partial P}{\partial \chi} d\chi + \frac{1}{2} U_{HT} + C_3 . \tag{3.47}$$

The constants must be determined from the boundary conditions.

For X=0 and i) $\eta \to \infty$ we choose $B_x^{(0)} = -1$
ii) $\eta \to -\infty$ $B_x^{(0)} = +1$ (3.48)

This gives

$$C_{1N} = -1,\ C_{1S} = +1$$
$$C_3 = \frac{1}{2} U_{HT}(0) \tag{3.49}$$

using (3.47). C_{1N} and C_{1S} corresponds to northern and southern reconnection. $u^{(0)} = -1$ for northern and $u^{(0)} = +1$ for southern. C_3 is the same for both cases.

The final solution then becomes

$$u^* = \pm erf(\eta) - \frac{1}{2}\int_0^{\mathrm{X}} \frac{1}{U_{HT}} \frac{\partial P}{\partial \chi} d\chi + \frac{1}{2} U_{HT} + \frac{1}{2} U_{HT}(0). \tag{3.50}$$

Next step is to match this solution with the outer solution corresponding to the magnetosheath flow. Here the flow is described around the magnetopause as a stagnation point flow Spreiter and Stahara[9], i.e.

$$U_0 + Q \cdot \mathrm{X} \tag{3.51}$$

where U_0 is the free stream velocity at the reconnection point. Q is the gradient with typical value of 0.1 per earth radii(corresponding to 25 km/s per earth radii).

Matching the asymptotic value ($\eta \to \infty$) of the boundary layer solution, with (3.51) we get

$$\lim_{\eta \to \infty} u^* = \pm 1 - \frac{1}{2}\int_0^{\mathrm{X}} \frac{1}{U_{HT}} \frac{\partial P}{\partial \chi} d\chi + \frac{1}{2} U_{HT}(\mathrm{X}) + \frac{1}{2} U_{HT}(0) = U_0 + Q \cdot \mathrm{X}. \tag{3.52}$$

$$\boxed{U_{HT}(\mathrm{X}) = U_{HT}(0) + Q_{HT} \cdot \mathrm{X}}$$

Identifying terms gives an expression for the integral

$$-\frac{1}{2}\int_0^{\mathrm{X}} \frac{1}{U_{HT}} \frac{\partial P}{\partial \mathrm{X}} d\mathrm{X} = Q \cdot \mathrm{X} - \frac{1}{2} Q_{HT} \cdot \mathrm{X} \tag{3.53}$$

This gives the final expression for the total velocity

$$u^* = \pm erf(\eta) + Q \cdot \mathrm{X} + U_{HT}(0) \tag{3.54}$$

3.6 Results and Conclusions

From a further look into the interpretation of the de Hoffmann-Teller velocity and its value at X=0 we get

$$u^{(0)} + U_{HT} = u^* = U_0 + Q \cdot \mathrm{X} \Rightarrow u^{(0)}(0) + U_{HT}(0) = U_0. \tag{3.55}$$

For the northern case: $u^{(0)}(0) = B_x^{(0)}(0) = -1$, pointing southward.
For southern reconnection: $u^{(0)}(0) = -B_x^{(0)}(0) = +1$.

This gives two different values of the de Hoffmann-Teller velocity

$$\begin{aligned} U_{HT}(0) &= U_0 + 1 && \text{Northern Reconnection} \\ U_{HT}(0) &= U_0 - 1 && \text{Southern Reconnection} \end{aligned} \tag{3.56}$$

There is actually nothing saying the magnetic field is ±1. The values are just examples normalised to the Alfvén velocity, with the important property that they have opposite sign since this is highly significant for the reconnection process.

The expressions in (3.56) can now be coupled with the solution of the thickness of the boundary layer (3.42). Taylor expansion of the de Hoffmann-Teller velocity gives

$$\delta = \pm \frac{\sqrt{2\alpha(U_{HT}(0) \cdot \mathrm{X} + Q \cdot \mathrm{X}^2}}{U_{HT}(0) + Q \cdot \mathrm{X}} \tag{3.57}$$

For the southern case the de Hoffmann-Teller velocity at X=0 is negative. Likewise, the X-coordinate is negative. This gives a positive growth of the boundary layer for both cases, due to the ± sign for the square root expression. The negative root is thus the one describing the southern case.

Due to the different de Hoffmann-Teller velocities, the boundary layer will have different structure on the northern and southern side. The higher velocity in the northern case will cause a thinner boundary layer on that side, see figure 5.

$$\delta_{Northern} = \frac{\sqrt{2\alpha((U_0 + 1) \cdot \mathrm{X}) + Q \cdot \mathrm{X}^2}}{1 + U_0 + Q \cdot \mathrm{X}} \tag{3.58}$$

$$\delta_{Southern} = \frac{\sqrt{2\alpha((U_0 - 1) \cdot \mathrm{X}) + Q \cdot \mathrm{X}^2}}{1 - U_0 - Q \cdot \mathrm{X}} \tag{3.59}$$

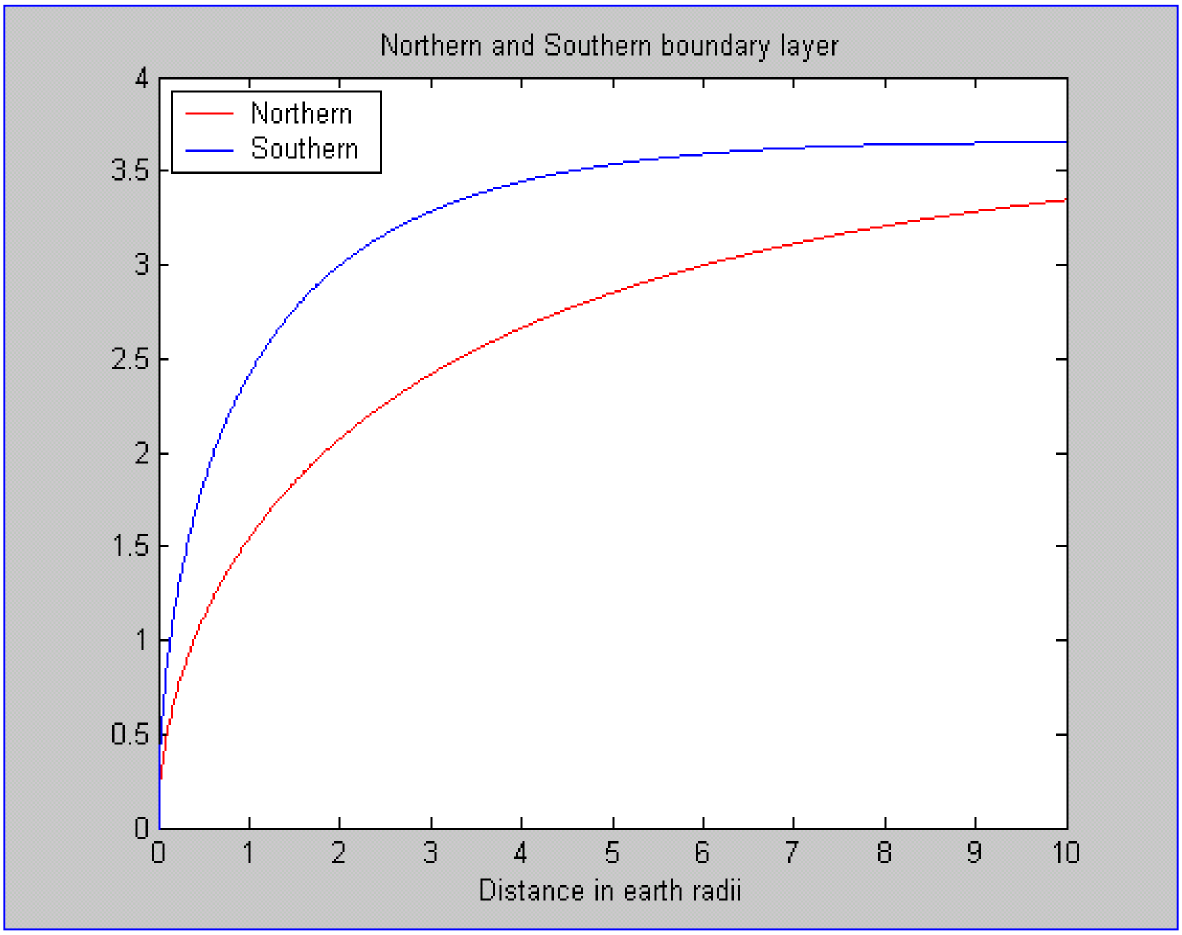


Figure 5: Boundary layer for Northern and Southern case

(3.57) and (3.58) also gives a limitation for the possibility of steady reconnection. In order to have the southern boundary layer positive this limitation becomes

$$U_0 < 1.$$

This yields that the free stream velocity at the reconnection point must be sub-Alfvénic. The result is also supported by Cowley and Owen[5].

The flow past the magnetosphere is, as discussed earlier, a stagnation point flow with origo at the sub-solar point. The solar wind velocity will increase when heading further away from the stagnation point. This means that we can only have steady reconnection in the region where the velocity is sub-Alfvénic. The solar wind reaches the Alfvén velocity at a distance of about ten earth radii, meaning that the zone at the dayside magnetopause where reconnection can occur is limited, see figure 12.

Figure 6 – 11 shows plots of the vector field for the total velocity with different velocities at the reconnection point. 6 and 7 shows the flow as it is at the sub-solar point with the characteristic pattern of the flow around a stagnation point. As the free stream velocity at the reconnection point increases the structure of the reconnection process clearly can be seen.

A further development of this theory is to improve the model for the magnetosheath flow. Today there only exists the gasdynamic model by Spreiter and Stahara[9] which does not properly include the effect of the magnetic field.

Other improvements of the present theory are

i) the generalisation of this model to take care of the magnetosphere curvature and draped magnetic field lines.

ii) to generalise the model to 3D.

iii) To consider other non-ideal effects such as for instance the inclusion of Hall-effects and gyroviscosity.

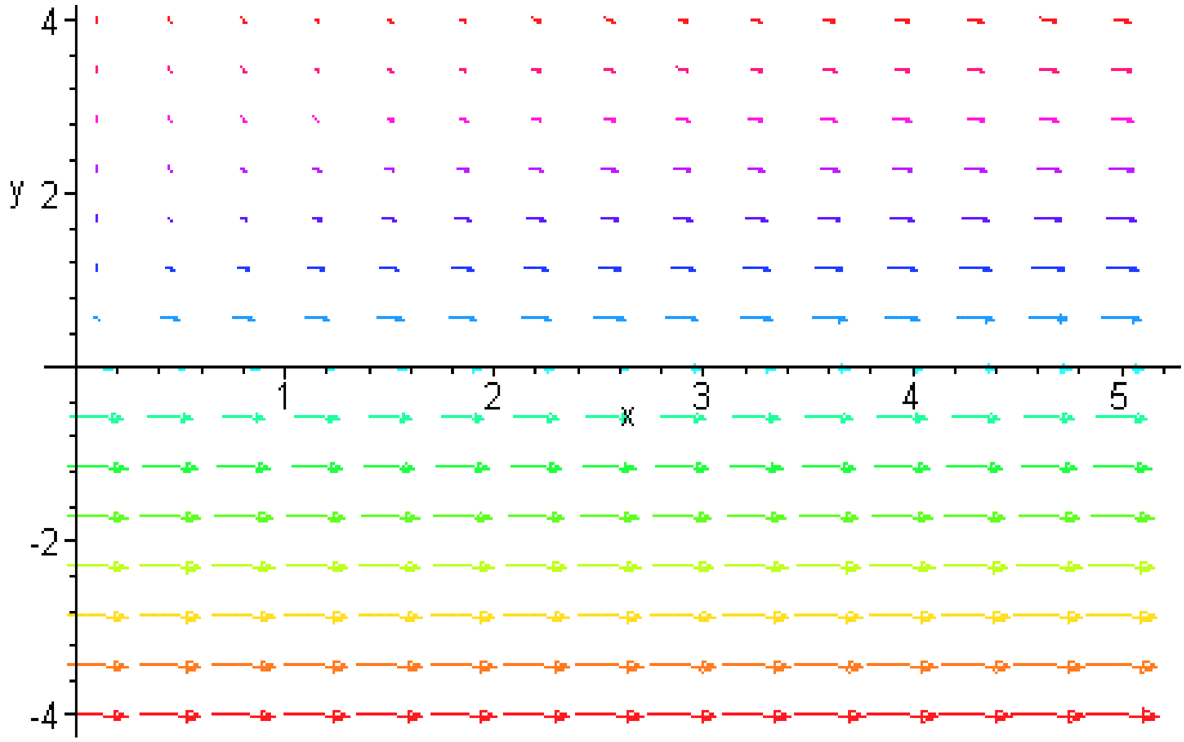


Figure 6 : Vector field for $U_0 = 0$ – Northern case

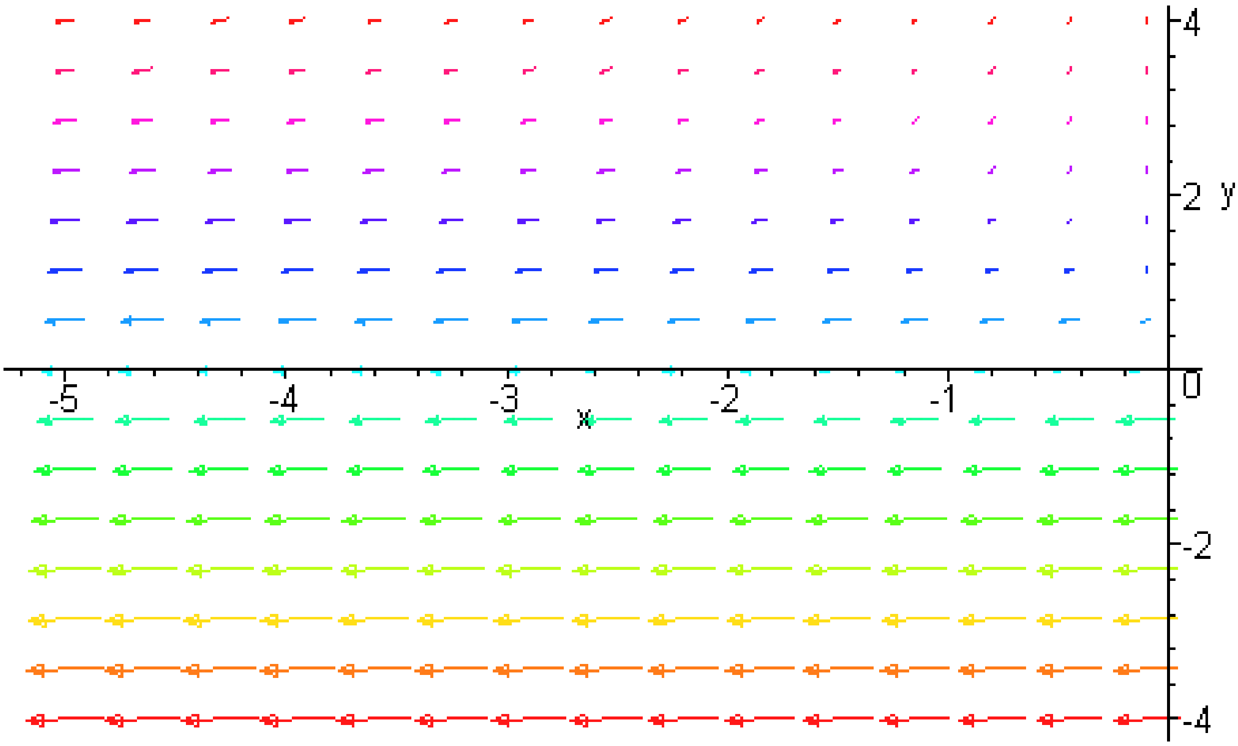


Figure 7 : Vector field for $U_0 = 0$ – Southern case

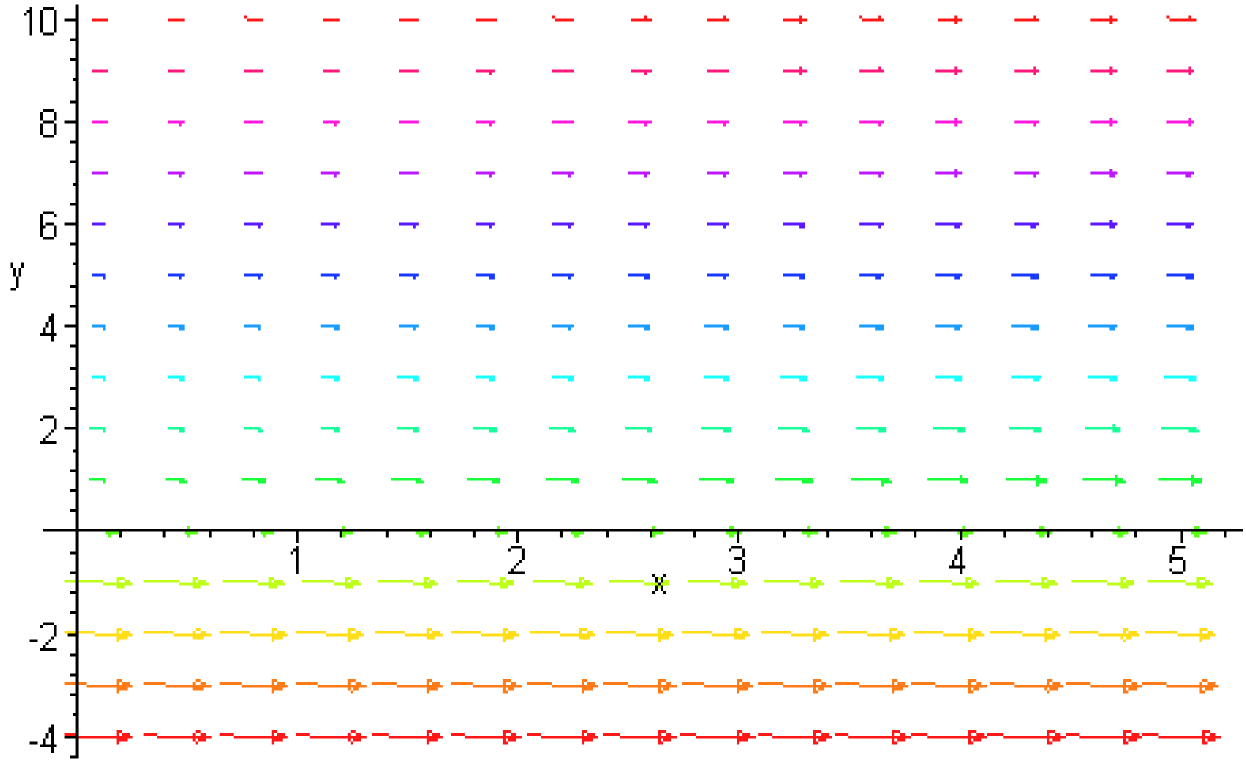


Figure 8 : Vector field for $U_0 = 0.5$ – Northern case

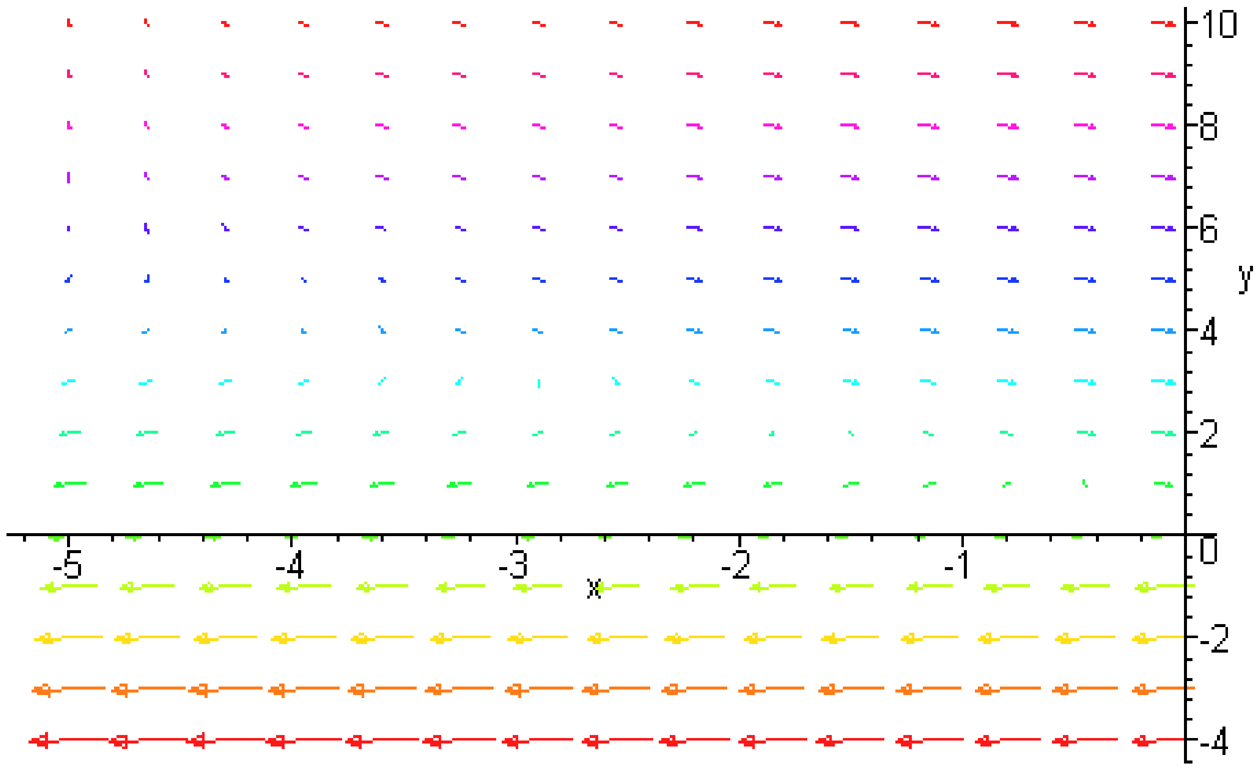


Figure 9 : Vector field for $U_0 = 0.5$ – Southern case

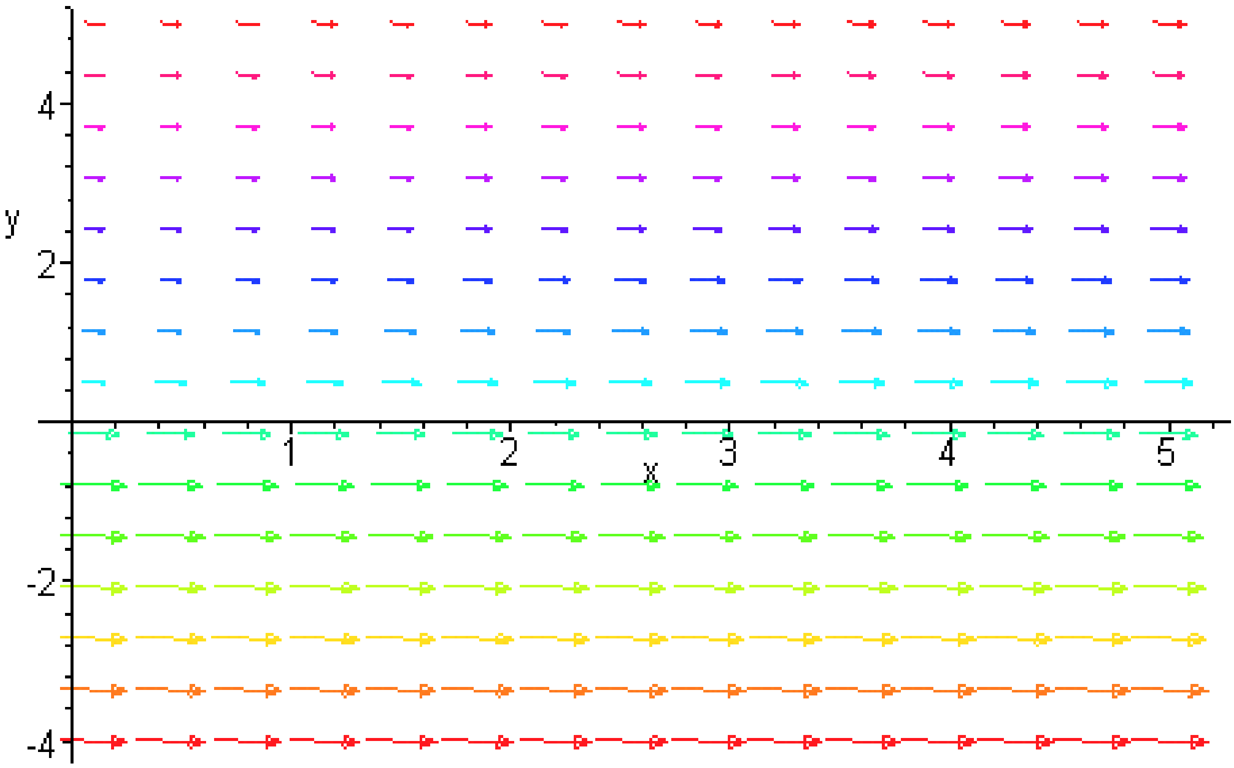


Figure 10 : Vector field for $U_0 = 0.9$ – Northern case

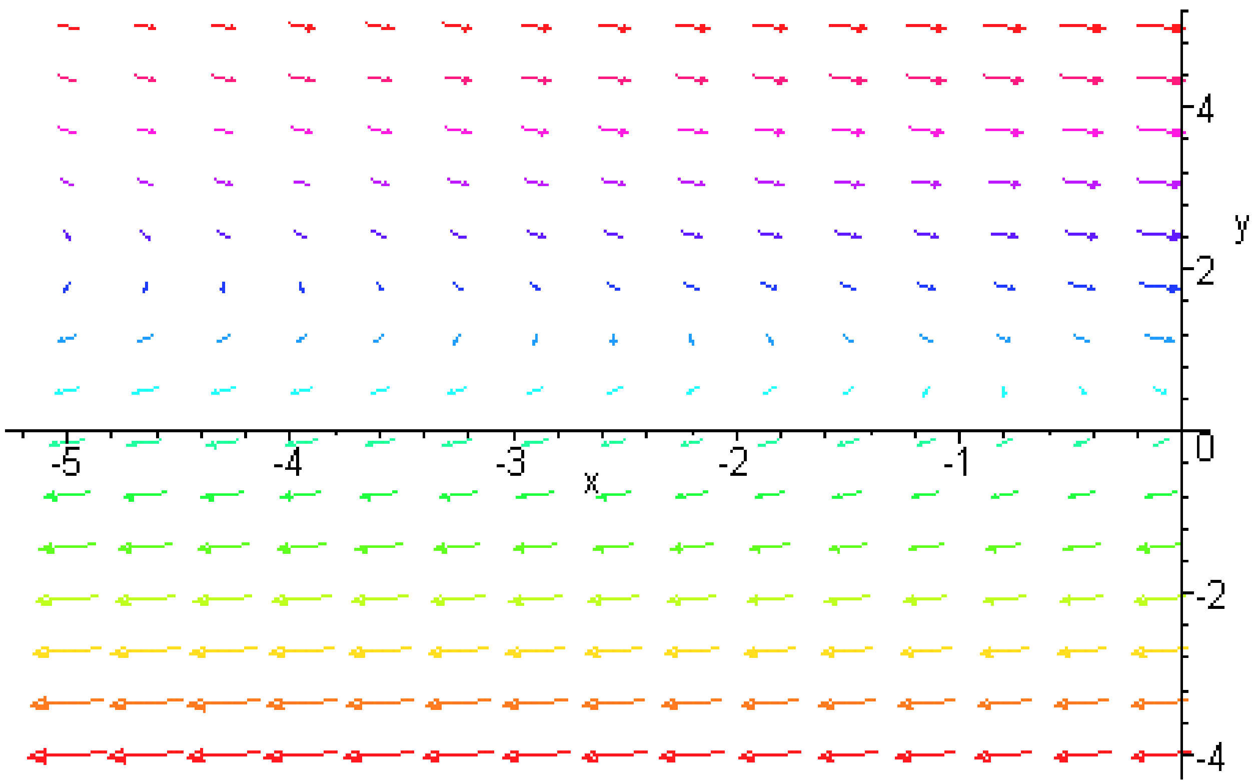


Figure 11 : Vector field for $U_0 = 0.9$ – Southern case

4 Numerical Solutions of MHD Equations

This chapter deals with the solution of MHD-equations appearing when studying the flow of the solar wind past the magnetopause. The method used is the Chebyshev Collocation Method. Appendix A describes in detail how the procedure works, while the actual MATLAB code is in appendix B. The goal is that the numerical calculations will serve as a cornerstone for further approaches in solving equations containing (all) relevant physical quantities such as velocity, magnetic field, electric field and pressure gradient.

Future work will include solving the complete non-linear equations. In chapter 3 the perturbation expansion gave a linear equation describing the total velocity. In this section we focus on a somewhat simpler case concerning the flow around the magnetopause without reconnection. That is, we solve the case treating an ordinary tangential discontinuity, having no normal component of the magnetic field. In the case of magnetic reconnection the magnetopause looses its property as a tangential discontinuity and becomes locally a rotational discontinuity with a non-vanishing normal component of the magnetic field.

As a tangential discontinuity the magnetopause is a surface of total pressure equilibrium between the solar wind-magnetosheath plasma and the geomagnetic field confined in the magnetosphere.

The magnetosheath flow past the magnetopause is somewhat similar to flow around a cylinder. The flow lines are closer together at the magnetopause, caused by the magnetosphere as it bends the flow lines into an azimuthal direction. According to the nozzle effect of the magnetosheath, the flow is once again forced to make the transition from sub-sonic to super-sonic flow. The magnetosheath plasma is compressed in a region close to the stagnation point at the nose of the magnetosphere.

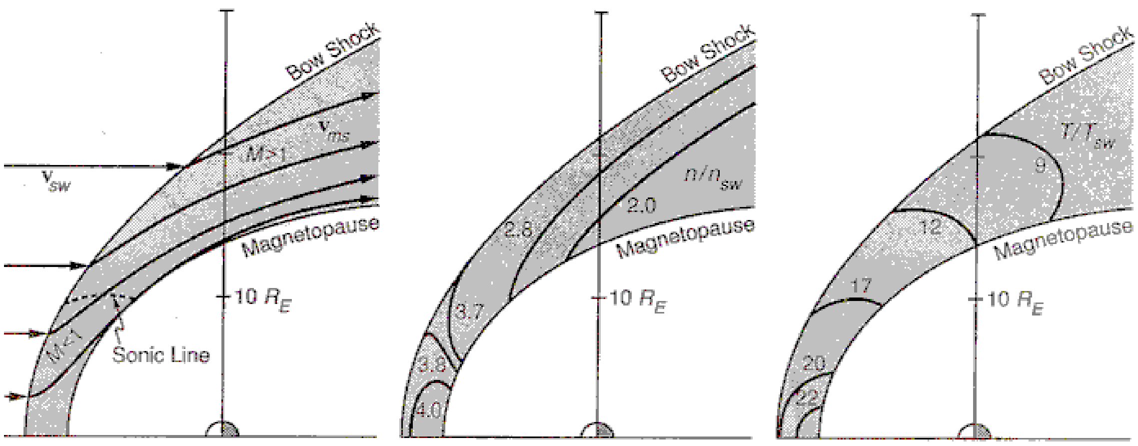


Figure 12: Magnetosheath stream lines and density and temperature isocontours.

The coupled equations describing the flow are

$$u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial y}=A^2B_x\frac{\partial B_x}{\partial x}+A^2B_y\frac{\partial B_x}{\partial y}+\frac{1}{R}\frac{\partial^2 u}{\partial y^2} \tag{4.1}$$

$$\frac{1}{R_m}\frac{\partial^2 B_x}{\partial y^2}=u\frac{\partial B_x}{\partial x}+\mathrm{v}\frac{\partial B_x}{\partial y}-B_x\frac{\partial u}{\partial x}-B_y\frac{\partial u}{\partial y} \tag{4.2}$$

The main approach here is to give example of the importance of the Alfvén number A. If A<1 the wave velocity is higher than the plasma velocity. For A<1 the plasma travels faster than the Alfvén waves and downstream obstacles cannot influence the flow upstream. For A>1 the information carried by the Alfvén waves travels faster than the plasma and obstacles downstream may have influence on the upstream flow.

These physical properties of the flow have some dramatic effects on the numerical solution of the system (4.1) and (4.2). For A<1 the boundary layer equations may be solved by a numerical method using marching in the flow direction. For this case there are no problems obtaining a numerical solution. On the other hand A>1, the numerical method becomes unstable and no numerical solution is obtained.

Further work is to come up with a routine that can handle this situation without giving unstable solutions.

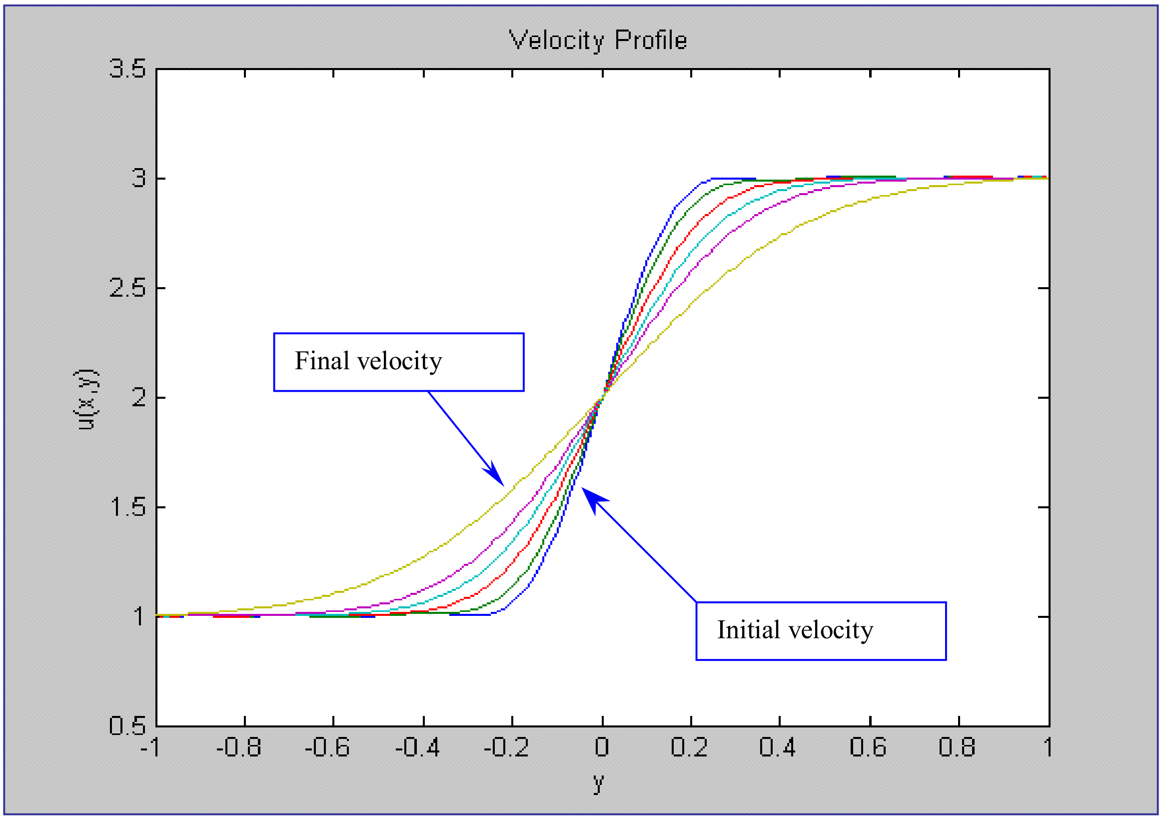


Figure 13: Velocity profile for sub-Alfvén velocity

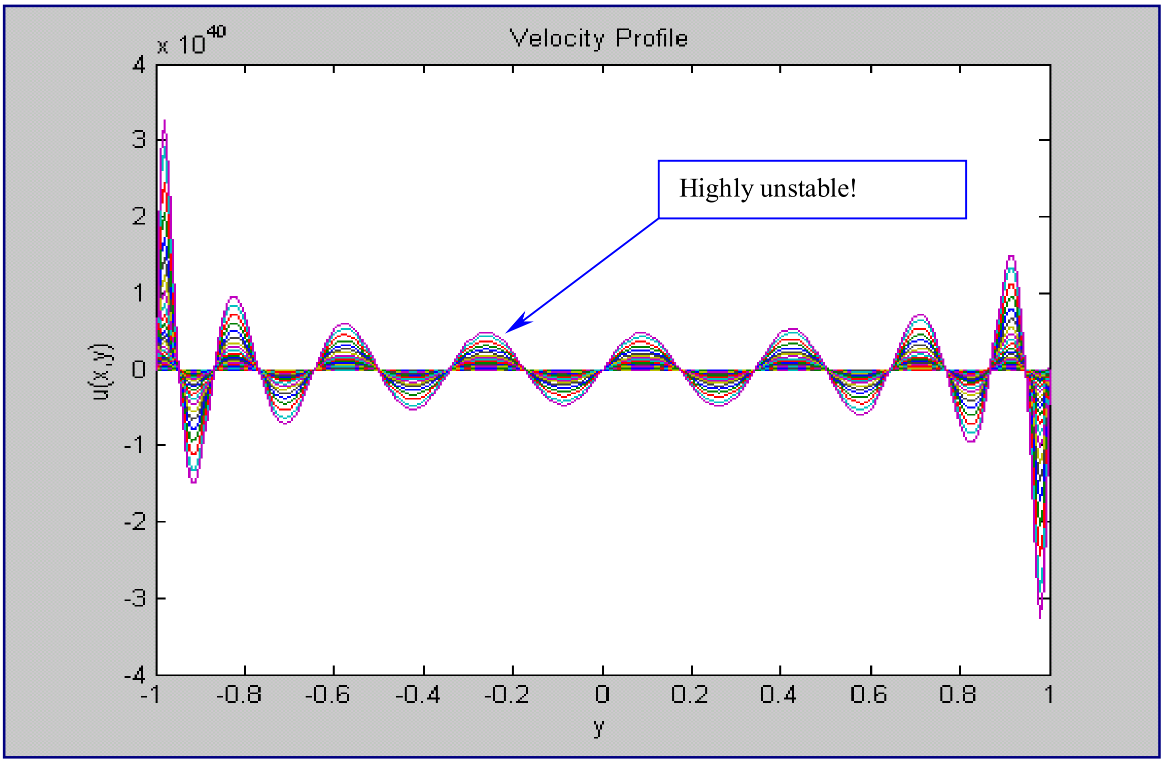


Figure 14: Velocity profile for super-Alfvén velocity

5 References

(1) Akerstedt H.O, Lövgren H.B, Fluid Dynamics Research 23 (1998) 1-25

(2) Baumjohann W, Treumann R.A, Advanced Space Plasma Physics, Imperial College Press 1997, ISBN 1-86094-026-9

(3) Baumjohann W, Treumann R.A, Basic Space Plasma Physics, Imperial College Press 1997, ISBN 1-86094-079-X

(4) Canuto C, Hussaini M.Y, Quarteroni A, Zang T.A, Spectral Methods in Fluid Dynamics, Springer-Verlag 1988, ISBN 0-387-52205-0

(5) Cowley S.W.H, Owen C.J, Pianet. Space Science, Vol. 37, No. 11, pp 1461-1475, 1989

(6) Kivelsson M.G, Russell C.T, Introduction to Space Physics, Camebridge University Press 1995, ISBN 0-521-45714-9

(7) Nayfeh A.H, Introduction to Perturbation Technique, Wiley 1993, ISBN 0-471-31013-1

(8) Morau R, Magnetohydrodynamics, Kluwer Academic Publishers 1990, ISBN 0-7923-0937-5

(9) Spreiter, J.R., Stahara S.S, A new predictive model for determining solar wind-terrestrial planet interactions, J. Geophys. Res., 85, 6769, 1980.

6 Appendix A

Chebyshev Collocation Method

In this chapter the actual numerical method is presented. A description of its advantages and how it works in practice is done.
In a time when computer power and prestanda constantly increases, the number of possibilities on how to solve a physical problem also increases. In Fluid Mechanics and Solid Mechanics there exist a couple of softwares designed to do calculations on surfaces where analytical solutions due to sinister geometries are very difficult to describe.
Here the geometry is somewhat easy to deal with, which gives great possibilities to use a spectral method. All code has been written in MATLAB from Math Works Inc.

A.1 General Properties

The chosen method in solving the upcoming equations is Chebyshev Collocation Method. The basic idea with this model is to transform the partial differential equation into a system of ordinary differential equations for the coefficients given below, which can be solved by using propitiate functions in MATLAB.
In using Chebyshev Collocation method the solution to the problem is set to be Canuto et.al.[4]

$$u(x,t) = \sum_{k=0}^{N} a_k(t) \cdot T_k(x) \tag{A.1}$$

where

$$T_k(x) = \cos(k \cdot a\cos(x)) \tag{A.2}$$

is the so-called trial function and u is like in the example with the Diffusion equation further down, the temperature function. Instead of, as in usual forms of Chebyshev polynomials, take every point inside the actual interval to consideration, the collocation method has its focus at certain points in the interval. This gives a highly accurate result at the same time, as it is economical with respect to the number of necessary computations.
A good choice of points are the ones named after Gauss, Chebyshev and Lobatto. That is,

$$x_j = \cos\left(\frac{\pi j}{N}\right) \qquad 1 \le \mathrm{j} \le \mathrm{N-1} \tag{A.3}$$

as actual points when solving the equations. N-1 are the total number of collocation points within the interior. In order to get a god result about ten collocation points are needed.

To further illustrate how the method works a well known equation is considered.

A.2 The Linear Diffusion Equation

Consider the Diffusion equation in one spatial coordinate

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 \qquad x \in [-1,1] \tag{A.4}$$

with homogeneous Dirichlet boundary conditions

$$u(-1,t) = u(1,t) = 0 \tag{A.5}$$

We write the solution as

$$u(x,t) = \sum_{k=0}^{N} a_k(t) \cdot T_k(x) \qquad k = 0,1 \ldots N \tag{A.6}$$

where N+1 are the total number of points inside the domain, including all collocation points and boundary conditions, and

$$T_k(x) = \cos(k \cdot a\cos(x)) \tag{A.7}$$

is the trial function.

When solving the problem, the number of collocation points must be decided. Theoretically it would give a higher accuracy in the solution the more points that are taken into consideration. There is one bother though; since each term in the original equation after transformation to Chebyshev space will be represented by a matrix with size proportional to the number of points inside the domain, the computational time increases rather dramatically with increasing number of collocation points. Usually it is sufficient to use eight to twelve collocation points. This gives a highly accurate solution.

Looking at respective term of actual equation and using the approximate solution (A.1)

$$\frac{\partial u}{\partial t} = \sum_{k=0}^{N} a_k^{'}(t) \cdot T_k(x) \tag{A.8}$$

and

$$\frac{\partial^2 u}{\partial x^2} = \sum_{k=0}^{N} a_k^{(2)}(t) \cdot T_k(x) \tag{A.9}$$

where the suffix (2) in (A.9) denotes the coefficients for the second derivative and the prime in (A.8) represents the derivative of a with respect to time.

In order to get the coefficients for respective derivative the following recurrence formula is used

$$a_{N+1}^{(1)} = a_{N+1}^{(2)} = 0 \quad \text{(A.10)}$$
$$a_N^{(1)} = a_N^{(2)} = 0 \quad \text{(A.11)}$$
$$c_k a_k^{(1)} = a_{k+2}^{(1)} + 2(k+1)a_{k+1} \quad \text{(A.12)}$$
$$c_k a_k^{(2)} = a_{k+2}^{(2)} + 2(k+1)a_{k+1}^{(1)} \quad \text{(A.13)}$$

$$k = N-1, N-2, \ldots 0 \quad \text{(A.14)}$$

$$k = 0 \vee \mathrm{N} \Rightarrow c_k = 2 \quad \text{(A.15)}$$
$$\textit{otherwise}$$
$$c_k = 1 \quad \text{(A.16)}$$

Replacing (A.8) and (A.9) in (A.4) gives an expression of the following kind

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \times \begin{bmatrix} a_0^{'} \\ \vdots \\ a_N^{'} \end{bmatrix} = \begin{bmatrix} & & \\ & B & \\ & & \end{bmatrix} \times \begin{bmatrix} a_0 \\ \vdots \\ a_N \end{bmatrix}. \quad \text{(A.17)}$$

Every matrix is square with N+1 elements. Top and bottom row of the matrix corresponds to the boundary conditions given in the problem, while the rows between those correspond to the collocation points. To solve this the operation inv(A)*B is done in every step. This gives a system of ordinary differential equations, which is solved with MATLAB's ode-solver. When calling the solver the actual interval has to be specified; in this case [0,1] gives a good result.. MATLAB begin in 0, solves the system of ODE's and take a step forward. This procedure continues until the endpoint is reached. Then the functions describing every coefficient are calculated. To get the solution to the Diffusion equation the coefficients are put in (A.6) and the summation is made. The result is as follows for two different start functions

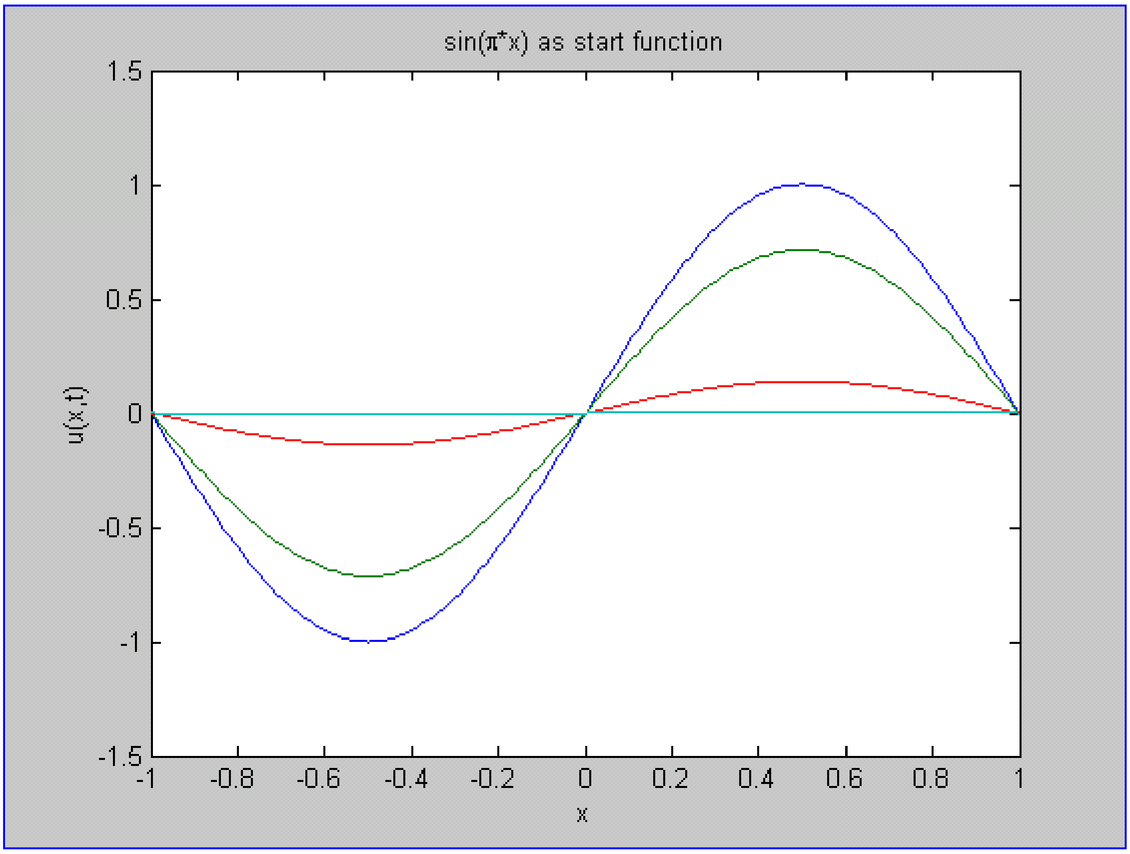


Figure 15: Diffusion for a sine function

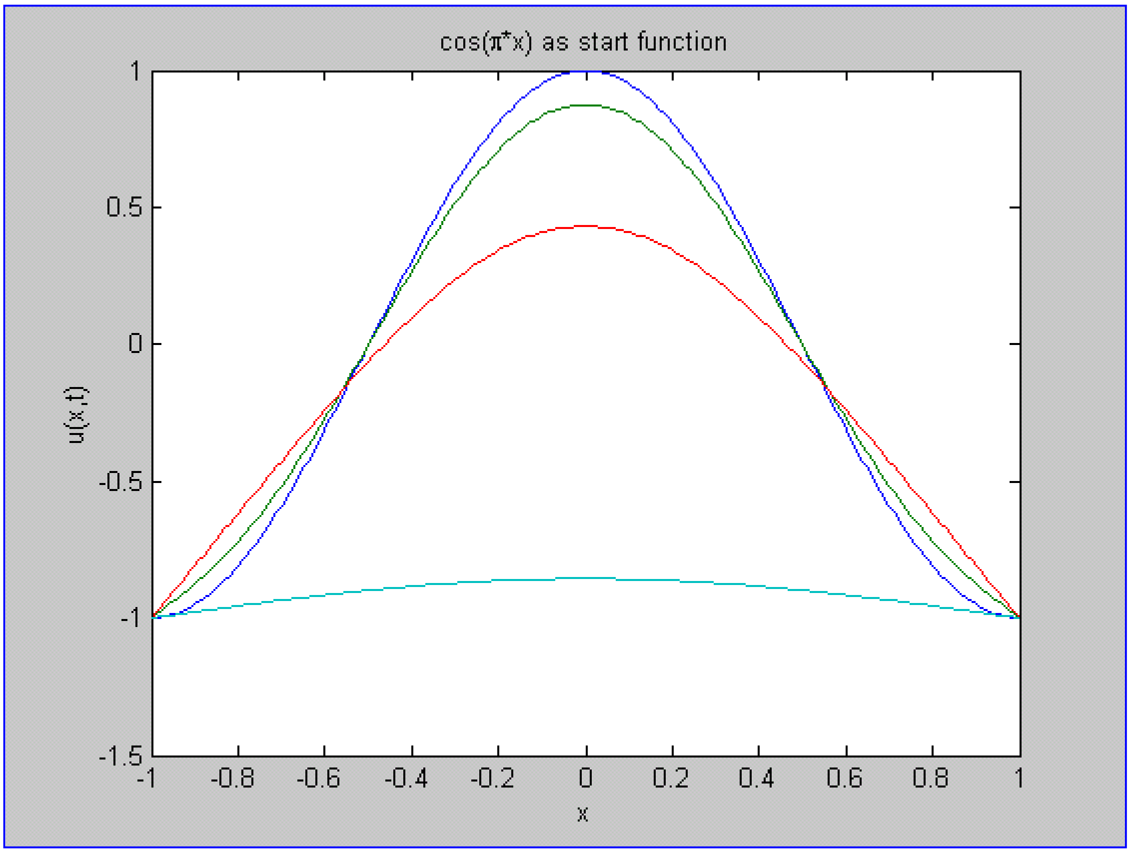


Figure 16: Diffusion for a cosine function

A.2.1 Boundary Conditions and Start Values

There are a total of N+1 points in the system. N-1 of them are collocation points. There are therefore sufficient with two more equations for a complete solution of the system – the boundary conditions. To get these the physical condition that has to be satisfied must be taken to consideration. In this case the fact that the temperature at the boundaries is constant is used, that is the time derivative of u equals zero. It gives that the elements in top and bottom row of matrix B are zero. In A (A.8) is used and put in x=-1 and x=+1 to get right values in the rows corresponding to these points

$$\left.\frac{\partial u}{\partial t}\right|_{x=\mp 1} = \sum_{k=0}^{N} a_k^{(1)}(t)\cdot T_k(x=\mp 1). \qquad \text{(A.18)}$$

This gives

$$[1 \quad -1 \quad 1 \quad -1 \quad . \quad . \quad . \quad .] \quad \text{Top row.} \qquad \text{(A.19)}$$

$$[1 \quad 1 \quad 1 \quad 1 \quad . \quad . \quad . \quad .] \quad \text{Bottom row.} \qquad \text{(A.20)}$$

In order to get correct start point for the actual function, the start function is taken into consideration. Following relation transforms this function to Chebyshev space

$$a_k(t) = \frac{2}{N\cdot c_k}\sum_{j=0}^{N} c_j^{-1} u_j(t)\cos\left(\frac{\pi jk}{N}\right) \qquad \text{(A.21)}$$

$$c_j = \begin{matrix} 2, & j=0 \text{ or } N \\ 1, & 1\le j\le N-1 \end{matrix} \qquad \text{(A.22)}$$

where u(t) is the start distribution and u(t=0) the value at the boundaries. These values are always the same as long as the boundary conditions are constant. This gives the start values for the coefficients. They are used when the ode-solver is called.

A.3 Non-Linear Equations

In the boundary layer analysis section the two reconnection cases were covered. The equation describing the total velocity is linear. The reality is however not always this smooth. The model is a slight idealisation of the real world; two dimensional and incompressible.
The truth and unavoidable fact is that we live in a non-linear world. In the study of the flow without reconnection this becomes reality which can cause a little vexation. This section treats equations that deal with one and two unknown variables. In the diffusion equation there is only the temperature which is the unknown variable. As shall be seen, it is quite straightforward to expand the system from one to two (or more) unknowns.

A.3.1 Velocity profile without magnetic field

Consider following equation

$$u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial y}=\frac{1}{R}\frac{\partial^2 u}{\partial y^2}. \tag{A.23}$$

There are two unknown quantities; u and v. By using the continuity equation the v-term can be expressed in u.

Continuity equation

$$\nabla\cdot\mathbf{u}=0\Leftrightarrow\frac{\partial u}{\partial x}+\frac{\partial \mathrm{v}}{\partial \mathrm{y}}=0. \tag{A.24}$$

Integration gives

Notice change of index!

$$\mathrm{v}=-\int_{-1}^{y_i}\frac{\partial u}{\partial x}dy=\{A.8\}=-\int_{-1}^{y_i}\sum_{k=0}^{N+1}\frac{d}{dx}(a_l(x))\cdot T_k(y)dy=-\sum_{k=0}^{N+1}\frac{d}{dx}(a_l(x))\cdot\int_{-1}^{y_i}T_k(y)dy \tag{A.25}$$

(A.25), (A.9) and (A.10) in (A.23) gives:

$$\underbrace{\sum_{k=0}^{N+1}a_k(x)\cdot T_k(y)\cdot\sum_{l=0}^{N+1}\frac{d}{dx}(a_l(x))\cdot T_l(y)-\sum_{k=0}^{N+1}\frac{d}{dx}(a_l(x))\cdot\int_{-1}^{y_i}T_k(y)dy\cdot\sum_{l=0}^{N+1}a_l^{(1)}(x)\cdot T_l(y)}=$$

$$=\frac{1}{R}\sum_{k=0}^{N+1}a_k^{(2)}(x)\cdot T_k(y) \tag{A.26}$$

Gives the matrix A in (A.17)

When transforming a partial differential equation, linear or non-linear, to Chebyshev space with collocation points the final goal is to obtain the structure of (A.17). In this case, the B matrix will be the term on the RHS. The procedure of solving the system is as earlier: Call the ode-solver in MATLAB and construct the solution according to (A.1).

When the u-component was the only quantity the index k went from zero to N. Now the maximum index value is N+1. This depends on the additional boundary conditions that have to be needed.

A.3.2 Boundary conditions

Having the v-component of the total velocity, two more boundary conditions are needed – one in –1 and one in +1. According to (A.25)

$$\mathrm{v} = -\sum_{k=0}^{N+1} \frac{d}{dx}\left(a_k(x)\right)\cdot \alpha_k \tag{A.27}$$

$$\alpha_k = \int_{-1}^{y} T_k(y)dy \tag{A.28}$$

In order to be able to integrate the test function the recurrence formula that gives the expression for the function at every k must be used

$$T_{k+1}(x) = 2x\cdot T_k(x) - T_{k-1}(x) \tag{A.29}$$

$$T_0(x) = 1 \text{ and } T_1(x) = x \tag{A.30}$$

First and last row in matrix A in (A.17) contains the coefficients for the boundary values for u as described for the diffusion example. Second and second last row is corresponding coefficients for the v-component. However, in (A.28) y=-1 gives a trivial case.

With one physical quantity there were a total of N+1 unknown coefficients. Now with the additional v-component, the number of unknowns should be N+3 since there is sufficient with two more rows representing the boundary conditions, but at y=-1 the values of the coefficients are zero. A row of zeros in matrix A cause trouble when the inverse operation is applied. Due to this, this row is not considered, and therefore end up with a total of N+2 unknown coefficients implying k=0 to k=N+1 as actual index.

For the boundary condition at y=+1 the coefficients in the second last row in A is calculated with (A.28). Corresponding row in matrix B (see (A.17)) is determined from what value the v-component is given at this point.

A.3.3 Velocity plus magnetic field

Adding a magnetic field to the equation treated in the previous example

$$u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial y}=A^2B_x\frac{\partial B_x}{\partial x}+A^2B_y\frac{\partial B_x}{\partial y}+\frac{1}{R}\frac{\partial^2 u}{\partial y^2} \tag{A.31}$$

Two quantities yield that two equations are needed. (A.27) is coupled with the Induction equation:

$$\frac{1}{R_m}\frac{\partial^2 B_x}{\partial y^2}=u\frac{\partial B_x}{\partial x}+\mathrm{v}\frac{\partial B_x}{\partial y}-B_x\frac{\partial u}{\partial x}-B_y\frac{\partial u}{\partial y} \tag{A.32}$$

First thing to do is to find linear combinations of the two equations. This procedure has two purposes: Before the combination is done, the two equations do not know that the other exist. The combinations therefore link them together. During the combination it is also sufficient to eliminate terms so the expressions, preferably, become easier to handle. The best thing would be if this gives one expression that has only the derivative of the velocity with respect to x on the LHS, and the other expression having the x-derivative of the magnetic field on the LHS.

$$\begin{aligned}&\frac{\partial u}{\partial x}=........\\&\frac{\partial B_x}{\partial x}=.......\end{aligned} \tag{A.33}$$

The use of a linearization model can do this

$$\begin{aligned}u&=U_0+u(x,y)\\B_x&=B_0+b(x,y)\end{aligned} \tag{A.34}$$

Small perturbation

A non-linearized combination of (A.31) and (A.32) is

$$\left(u^2-A^2B_x^2\right)\frac{\partial u}{\partial x}+\mathrm{v}\left\{u\frac{\partial u}{\partial y}+A^2B_x\frac{\partial B_x}{\partial y}\right\}-B_y\left\{A^2B_x\frac{\partial u}{\partial y}+uA^2\frac{\partial B_x}{\partial y}\right\}=$$

$$=\frac{A^2B_x}{R_m}\frac{\partial^2 B_x}{\partial y^2}+\frac{u}{R}\frac{\partial^2 u}{\partial y^2} \tag{A.35}$$

$$\left(u^2 - A^2 B_x^2\right)\frac{\partial B_x}{\partial x} + \mathrm{v}\left\{B_x \frac{\partial u}{\partial y} + u\frac{\partial B_x}{\partial y}\right\} - B_y\left\{A^2 B_x \frac{\partial B_x}{\partial y} + u\frac{\partial u}{\partial y}\right\} =$$

$$= \frac{u}{R_m}\frac{\partial^2 B_x}{\partial y^2} + \frac{B_x}{R}\frac{\partial^2 u}{\partial y^2} \tag{A.36}$$

In order to solve this problem the definition of the velocity components is set up as earlier. The solution for the magnetic field is set up in the same way, but with a change of coefficients; a represents the velocity and b will represent the magnetic field

$$B_x = \sum_{k=0}^{N+1} b(x,y)\cdot T_k(y). \tag{A.37}$$

The y-component is derived from Maxwell's equation ∇·**B**=0, giving the same expression as for the v-component of the velocity

$$B_y = -\sum_{k=0}^{N+1} \frac{d}{dx}(b(x))\cdot \int_{-1}^{y_i} T_k(y)dy. \tag{A.38}$$

The final system will be as in (A.17) with some slight changes in the matrices and the row vectors. There will be a total of 2*N+4 points. The number of collocation points are N-1 for both the velocity and the magnetic field. Since the system contains the x and y-component of the magnetic field, there should be a total of 2*N+6 points, but the boundary values at –1 will vanish for the y-component with the same reason as for v.

The new row vector will be

$$\left[a_0 \quad .. \quad a_N \quad a_{N+1} \quad b_0 \quad .. \quad b_N \quad b_{N+1}\right] \tag{A.39}$$

The matrices can be divided into four regions

$$\begin{bmatrix} 1 & & 2 \\ & A & \\ 3 & & 4 \end{bmatrix}\begin{bmatrix} a_0' \\ : \\ : \\ : \\ b_{N+1}' \end{bmatrix} = \begin{bmatrix} 1 & & 2 \\ & B & \\ 3 & & 4 \end{bmatrix}\begin{bmatrix} b_0 \\ : \\ : \\ : \\ b_{N+1} \end{bmatrix} \tag{A.40}$$

A1: Represents the x-derivative term of the velocity for the first equation (A.35).
A2: Same as A1, but for the magnetic field.
A3: As for A1, but for the second equation.
A4: Same as A2, but for the second equation.

Corresponding system for the B matrix with the difference that it contains the y-derivative of the terms.

Example: The first term in (A.36)

$$(u^2 - A^2 B_x^2)\frac{\partial B_x}{\partial x}$$

can be expressed as

$$\begin{bmatrix} & & \\ & Z1 & \\ & & \end{bmatrix} \begin{bmatrix} b_0' \\ : \\ b_{N+1}' \end{bmatrix} \tag{A.41}$$

The matrix Z1 will take place A4 in (A.40). If there would have been a set of equations like in (A.33), the matrix A in (A.40) would have values in the first and fourth quadrant while the other elements would be zero.

Since the matrix size for this expanded system is twice the one without the magnetic field, the computational time will be much longer.

All code in this thesis is written in MATLAB. A future development of this project would include implementation of the algorithm in C. By doing so-called mex-files there is possible to run C-code in MATLAB. This will increase the calculation speed dramatically.

7 Appendix B

MATLAB code for the programs solving actual equations

This section contains the code used to solve the actual partial differential equations. Two scripts – the ‘run’ script and the script that returns the system of ordinary differential equations used by the ode-solver, build every solution.

B.1 The run script for the Diffusion equation

```
% Program som tillsammans med funktionen 'diffusionode',
% löser diffusionsekvationen med Chebyshev Collocation Method.
clear all; help korning_diffusion;

% Vi skall ta fram begynnelsevärdena för koefficienterna; a0(t=0) ..
% aN(t=0).
% Dessa används när ode-lösaren anropas.

% Värdena fås genom att använda begynnelsefunktionen. I detta exempel
% är denna sin(pi*x). Värdet för uj i respektive
% kollokationspunkt erhålls då genom uj=sin(pi*xj). uj i sin tur, används
% för att kalkylera ak(t=0) genom (1.2.36) i Quarteroni et al.

% Tar nu fram uj.
clear all;
global N;
N=input('Ange antalet kollokationspunkter: ');
uj=zeros(1,N+1);

a_beg=zeros(1,N+1);

for k=0:N
   for j=0:N
      xj=cos(pi*j/N);
      uj=sin(pi*xj); % Initialfördelningen. Testa att ändra initialfördelning
                     % till en cos(pi*xj) t.ex.
      if (j==0 | j==N)
         a_beg(k+1)=a_beg(k+1)+0.5*uj*cos(pi*j*k/N);
      else
         a_beg(k+1)=a_beg(k+1)+uj*cos(pi*j*k/N);
      end
   end
end
```

```
for k=0:N
   if(k==0 | k==N)
      a_beg(k+1)=a_beg(k+1)*1/N;
   else
      a_beg(k+1)=a_beg(k+1)*2/N;
   end
end

% Nu är vi redo att dunka in problemet i ode-lösaren. Detta ger oss
% koefficienterna a0,...,aN.

[t,kalle]=ode15s('diffusionode',[0 1],a_beg);

% Sista steget är att ta fram den approximativa lösningen.
% Se (1.2.23) i Quarteroni et al.
x=[-1:0.01:1];
u0=zeros(1,201);
u=zeros(length(kalle),201);
u_vektor=u;

for k=1:N+1
   Tk(k,:)=cos((k-1)*acos(x));
   u_vektor(k,:)=kalle(1,k)*Tk(k,:);
   u0=u0+u_vektor(k,:);
end

for i=1:length(kalle) % Lösning för alla tidpunkter.
   for k=1:N+1
      Tk(k,:)=cos((k-1)*acos(x));
      u_vektor(k,:)=kalle(i,k)*Tk(k,:);
      u(i,:)=u(i,:)+u_vektor(k,:);
   end

end

uplot=zeros(4,201); % Plockar ut fyra tidpunkter. Ändrar man initialfunktionen
                    % får man ta med fler kurvor för finare plot. Alt. kan man
                    % ta med alla, dvs. u ist.f. uplot på rad 82.

uplot(1,:)=u(1,:);
uplot(2,:)=u(10,:);
uplot(3,:)=u(20,:);
uplot(4,:)=u(length(t),:);

 % Plottar lösningen.
 figure(1); clf;
 plot(x,uplot);
 xlabel('x'); ylabel('u(x,t)');
 title('sin(\pi*x) as start function');
```

B.1.1 Corresponding m-file

```
function aprim=diffusionode(t,kalle)
t
% Funktionen returnerar ett system av differentialekvationer
% som används av ode-lösaren. Systemet beskriver tidsderivatan av
% koefficienterna ak(t), som funktion av de oderiverade
% koefficienterna.
% För att få fram ak(2)(t) och ak(1)(t) används rekursionsformlerna.

global N;

a_2=zeros(N+1); % Initierar koeff. för andraderivatan.
a_1=zeros(N+1); % Initierar koeff. för förstaderivatan.
a=eye(N+1);     % Initierar koeff. för 0:te derivatan.

k=N-1;
while k>=0      % Huvudloop för rekurseringen. 1:a derivatans koefficienter
   q=k+1;

   if(k==0)
      c_k=2;
      s_1=(1/c_k)*(a_1(k+2,:)+2*(k+1)*a(k+2,:));
   else
      c_k=1;
      a_1(k,:)=(1/c_k)*(a_1(k+2,:)+2*(k+1)*a(k+2,:));

   end

   k=k-1;
end

%%% OBS!!! - koefficienten a(0) finns på plats/rad 1 i matrisen. Flyttar vi
% inte ned alla rader ett steg, kommer en rad att gå förlorad.

prel=a_1;
a_1=zeros(N+1);

for i=2:N+1
   a_1(i,:)=prel(i-1,:);
end
a_1(1,:)=s_1;

% Nu har vi vår matris beskrivande koefficienterna för förstaderivatan som
% uttryck av oderiverade dito. Nästa steg är att plocka fram
% koefficienterna för andraderivatan. Detta gör vi genom att använda
% aktuell rekursionsformel (1.2.40).
```

```
k=N-1;
while k>=0 % Huvudloop för rekurseringen. 2:a derivatans koefficienter
   q=k+1;

   if(k==0)
      c_k=2;
      s_2=(1/c_k)*(a_2(k+2,:)+2*(k+1)*a_1(k+2,:));
   else
      c_k=1;
      a_2(k,:)=(1/c_k)*(a_2(k+2,:)+2*(k+1)*a_1(k+2,:));

   end

   k=k-1;
end

%%% OBS!!! - här måste vi också flytta ned alla rader ett steg och fylla på
% med raden motsvarande koefficienten då k=0, som följdaktligen hamnar på
% första plats/rad i matrisen.

prel=a_2;
a_2=zeros(N+1);

for i=2:N+1
   a_2(i,:)=prel(i-1,:);
end
a_2(1,:)=s_2;

% Vi gör en matris där varje rad innehåller uttrycket för d2u/dx2 vid varje
% kollokationspunkt, j. Obeservera att vi har totalt två mindre
% kollokationspunkter än koefficienter a_k. Detta gör att vi måste lägga
% till två rader i matrisen d2u/dx2, för att de skall bli "square".
% Raderna, som är de första och sista, motsvarar randvillkoren.
% Temperaturen skall vara oberoende av tiden på ränderna dvs.
% du/dt=0.

del_mat=zeros(N+1);
B=zeros(N+1);

for j=1:N-1
   for k=1:N+1
      cosinus(j,k)=cos(pi*j*(k-1)/N);
      del_mat(k,:)=cosinus(j,k)*a_2(k,:);
   end

   summa=zeros(1,N+1);
```

```
    for s=1:N+1
        summa=summa+del_mat(s,:);
    end
    B(j,:)=summa;
    del_mat=zeros(N+1);

end

% Lägger till aktuella två rader.
prel=zeros(N+1);
for i=1:N+1
    if(i==1)

        prel(i,:)=zeros; % x=-1

    elseif(i==(N+1))

        prel(i,:)=zeros; % x=+1

    else
        prel(i,:)=B(i-1,:);
    end
end
B=prel; % Kopierar över innehållet till B.

% Nu är d2u/dx2 som funktion av originalkoefficienterna för varje
% kollokationspunkt.

% Tar fram matrisen som representerar dudt.
A=zeros(N-1,N+1);
for j=1:N-1
    for k=1:N+1
        A(j,k)=cos(pi*j*(k-1)/N);
    end
end

% Lägger till raderna för randvillkoren.
prel=zeros(N+1);
for i=1:N+1
    if(i==1)
        for k=1:N+1
            prel(1,k)=cos((k-1)*pi);
        end
    elseif(i==(N+1))
        prel(N+1,:)=ones;
    else
        prel(i,:)=A(i-1,:);
    end
end
```

```
A=prel; % Kopierar över innehållet till A.
% Hela systemet är nu klart i enlighet med det överst på sid.35.
% A och B är motsvarande matriser. I nästa steg tar vi fram (**).

C=inv(A)*B;

aprim=C*kalle;
return;

% Härmed är systemet returnerat.
```

B.2 The run script for the system (A.35) (A.36)

```
% Program som tillsammans med funktionen 'diffusionode',
% löser hastighetsekvationen med megnetfält med Chebyshev
% Collocation Method.
clear all; help korning_m_Bfalt;

% Vi skall ta fram begynnelsevärdena för koefficienterna; a0(t=0) ..
% aN(t=0).
% Dessa används när ode-lösaren anropas.

% Värdena fås genom att använda begynnelsefunktionen. I detta exempel
% är denna tanh(8*x). Värdet för uj i respektive
% kollokationspunkt erhålls då genom uj=sin(pi*xj). uj i sin tur, används
% för att kalkylera ak(t=0) genom (1.2.36) i Quarteroni et al.

% Tar nu fram uj.
q=input('Ange antalet kollokationspunkter: ');
global N;
global T;
N=q+1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Integralmatrisen till v och By.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

T=zeros(N-1,N+2);
for j=1:N-1
   e_j=cos(pi*j/N);
   for g=1:N+2
      if(g==1)
         T(j,g)=e_j-1; % Kommer ihåg att vi har en bestämd integral från -1 till e_j.
      elseif(g==2)     % Därav de konstanta termerna.
         T(j,g)=e_j^2/2-0.5;
      elseif(g==3)
         T(j,g)=2*e_j^3/3-e_j+1/3;
      elseif(g==4)
         T(j,g)=e_j^4-3*e_j^2/2+0.5;
      elseif(g==5)
         T(j,g)=8*e_j^5/5-8*e_j^3/3+e_j+1/15;
      elseif(g==6)
         T(j,g)=8*e_j^6/3-5*e_j^4+5*e_j^2/2-1/6;
      elseif(g==7)
         T(j,g)=32*e_j^7/7-48*e_j^5/5+6*e_j^3-e_j+1/35;
      elseif(g==8)
         T(j,g)=8*e_j^8-56*e_j^6/3+14*e_j^4-7*e_j^2/2+1/6;
      elseif(g==9)
         T(j,g)=e_j-256*e_j^7/7+32*e_j^5+128*e_j^9/9-32*e_j^3/3+1/63;
      elseif(g==10)
```

```
        T(j,g)=72*e_j^6+9*e_j^2/2-30*e_j^4-72*e_j^8+128*e_j^10/5-1/10;
      elseif(g==11)
        T(j,g)=-e_j+160*e_j^7-80*e_j^5-1280*e_j^9/9+512*e_j^11/11+50*e_j^3/3+1/99;
      elseif(g==12)
        T(j,g)=352*e_j^8-11*e_j^2/2-1408*e_j^10/5+55*e_j^4+256*e_j^12/3-…
              616*e_j^6/3+1/10;
      elseif(g==13)
        T(j,g)=e_j-6144*e_j^11/11+2048*e_j^13/13+168*e_j^5-24*e_j^3-…
              512*e_j^7+768*e_j^9+1/143;
      elseif(g==14)
        T(j,g)=2048*e_j^14/7+13*e_j^2/2+1664*e_j^10-91*e_j^4-1248*e_j^8-…
              3328*e_j^12/3+1456*e_j^6/3-1/14;
      elseif(g==15)
        T(j,g)=-e_j+8192*e_j^15/15+3584*e_j^11-28672*e_j^13/13-1568*e_j^5/5-...
              8960*e_j^9/3+98*e_j^3/3+1344*e_j^7+1/195;
      elseif(g==16)
        T(j,g)=1024*e_j^16+3600*e_j^8-1008*e_j^6+140*e_j^4+7680*e_j^12-15*e_j^2/2-...
              30720*e_j^14/7-7040*e_j^10+1/14;
      elseif(g==17)
        T(j,g)=e_j+2688*e_j^5/5+16384*e_j^13-3072*e_j^7-131072*e_j^15/15+...
              +28160*e_j^9/3+32768*e_j^17/17-128*e_j^3/3-16384*e_j^11+1/255;
      elseif(g==18)
        T(j,g)=-17408*e_j^16-8976*e_j^8+32768*e_j^18/9-204*e_j^4-…
              -113152*e_j^12/3+1904*e_j^6+34816*e_j^14+ …
              17*e_j^2/2+23936*e_j^10-1/18;
      elseif(g==19)
        T(j,g)=-e_j+54*e_j^3-86016*e_j^13-25344*e_j^9+73728*e_j^15-864*e_j^5-…
              -589824*e_j^17/17+131072*e_j^19/19+59904*e_j^11+6336*e_j^7+1/323;
      elseif(g==20)
        T(j,g)=-347776*e_j^10/5+20064*e_j^8-194560*e_j^14+285*e_j^4- …
              19*e_j^2/2+155648*e_j^16-3344*e_j^6+65536*e_j^20/5- …
              622592*e_j^18/9+442624*e_j^12+1/18;
      elseif(g==21)
        T(j,g)=358400*e_j^13-200*e_j^3/3+e_j-186368*e_j^11- …
              621440*e_j^19/19+1320*e_j^5+524288*e_j^21/21-84480*e_j^7/7+ …
              +327680*e_j^17+183040*e_j^9/3-1310720*e_j^15/3+1/399;

      end
    end
end

% Begynnelsevärdena: koefficientvektorn är 2N+2 lång. Första N+1 värdena
% är för hastigheten, medan resterande är för magnetfältet.

% Koefficienter framräknade i Maple.
a_beg=[2.00000000000000 1.26476816617406 0 -.399897893621021 0 .216461678159257 0
-.133143486391627 0 .852562705353561e-1 0 -.546225427316594e-1 0 …
336465622072798e-1 0 -.182446517638214e-1 0 .577567236185339e-2 0];
```

```
b_beg=[1.500000000 1.269931422 0 -.4144581769 0 .2379506476 0 -.1582811726 0
.1106837245 0 -.07739955200 0 .05148506453 0 -.02955887822 0 .009646921022 0];

beg_koeff(1:N+2)=a_beg;
beg_koeff(N+3:2*N+4)=b_beg;

[x,kalle]=ode23tb('u_Bode',[0 10],beg_koeff);
% Möjliga ode-lösare: 45,23,113,23t,23s,15s,23tb.

% Sista steget är att ta fram den approximativa lösningen.
% Se (1.2.23) i Quarteroni et al.
epsilon=[-1:0.01:1];
u=zeros(length(kalle),201);
b=u;
u_vektor=zeros(N+2,201);
b_vektor=u_vektor;

qp=size(kalle);
kalle_b=zeros(qp(1),N+2);
kalle_b=kalle(1:qp(1),N+3:2*N+4);

for i=1:qp(1) % Lösning för u för alla tidpunkter.
   for k=1:N+2
      Tk(k,:)=cos((k-1)*acos(epsilon));
      u_vektor(k,:)=kalle(i,k)*Tk(k,:);
      b_vektor(k,:)=kalle_b(i,k)*Tk(k,:);
      u(i,:)=u(i,:)+u_vektor(k,:);
      b(i,:)=b(i,:)+b_vektor(k,:);
   end

end

 % Plottar lösningen.
 figure(1); clf;
 plot(epsilon,u);
 xlabel('\epsilon'); ylabel('u(x,\epsilon)');

 figure(2); clf;
 plot(epsilon,b);
 xlabel('\epsilon'); ylabel('b(x,\epsilon)');
```

B.2.1 The m-file called by the ode-solver

```
function aprim=u_Bode(x,kalle)
x

global N;
global T;

kalle_u=zeros(1,N+2);
kalle_b=zeros(1,N+2);
kalle_u=kalle(1:N+2);
kalle_b=kalle(N+3:2*N+4);

% Koefficienterna för första och andra derivatan.

a_2=zeros(N+2); % Initierar koeff. för andraderivatan.
a_1=zeros(N+2); % Initierar koeff. för förstaderivatan.

for n=0:N+1
   k=n+1;
   if(n==0)
      Cn=2;
   else
      Cn=1;
   end

   while(k<=N+1)
      a_1(n+1,k+1)=2*k/Cn;
      k=k+2;
   end
end

for n=0:N+1
   k=n+2;
   if(n==0)
      Cn=2;
   else
      Cn=1;
   end

   while(k<=N+1)
      a_2(n+1,k+1)=k*(k^2-n^2)/Cn;
      k=k+2;
   end
end
```

```
A_koeff=0.25; % Alfvén talet
Rm=1000; % Magnetiska Reynoldstalet.
R=1000; % Reynoldstalet.
deriv_matrix=zeros(2*N-2,2*N+4);  % Den stora matrisen som beskriver det deriverade
                                  % systemet.
oderiv_matrix=zeros(2*N-2,2*N+4); % Oderiverade systemet.

% Gör termer som återkommer i ekvationerna.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Börjar med u och b.

u=zeros(N-1,1);
u_matris=zeros(N-1,N+2);
b=zeros(1,N-1);
b_matris=u_matris;

for j=1:N-1
   for k=1:N+2
      cosinus(j,k)=cos(pi*j*(k-1)/(N+1));
      u_matris(j,k)=cosinus(j,k);
      b_matris(j,k)=cosinus(j,k);
   end

end

u=u_matris*kalle_u;
b=b_matris*kalle_b;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Derivatorna m.a.p. y.

del_mat1=zeros(N+2);
del_mat2=del_mat1;
dudy=zeros(N-1,N+2);
dbdy=dudy;
d2udy2=dudy;
d2bdy2=d2udy2;

for j=1:N-1
   for k=1:N+2
      cosinus(j,k)=cos(pi*j*(k-1)/(N+1));
      del_mat1(k,:)=cosinus(j,k)*a_1(k,:);
      del_mat2(k,:)=cosinus(j,k)*a_2(k,:);

   end

   summa1=zeros(1,N+2);
   summa2=summa1;
```

```
    for s=1:N+2
        summa1=summa1+del_mat1(s,:);
        summa2=summa2+del_mat2(s,:);
    end

    dudy(j,:)=summa1;
    dbdy(j,:)=summa1;
    d2udy2(j,:)=summa2;
    d2bdy2(j,:)=summa2;
    del_mat1=zeros(N+2);
    del_mat2=del_mat1;

end

dudy=dudy*kalle_u;
dbdy=dbdy*kalle_b;
d2udy2=d2udy2*kalle_u;
d2bdy2=d2bdy2*kalle_b;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Gör termerna i ekvationen.

A=zeros(N-1,N+2);
B=A;
C=A;
D1=A;
D2=A;
E=A;
F=A;
G=A;
H1=A;
H2=A;
A1=zeros(N-1,1);
B1=zeros(N-1,N+2); C1=B1; D1=B1; E1=B1; F1=B1; G1=B1; H1=B1;

for j=1:N-1
    A1(j)=u(j)^2-A_koeff^2*(b(j)^2);
    for k=1:N+2
        A(j,k)=cos((k-1)*j*pi/(N+1))*A1(j);
    end
end
E=A;
```

```
% De konstanta termerna för varje j i det deriverade systemet.

for j=1:N-1
   for k=1:N+2
      B1(j,k)=cos((k-1)*j*pi/(N+1))*dudy(j);
      B2(j,k)=cos((k-1)*j*pi/(N+1))*dbdy(j)*A_koeff^2;
      C1(j,k)=cos((k-1)*j*pi/(N+1))*dudy(j)*A_koeff^2;
      F1(j,k)=cos((k-1)*j*pi/(N+1))*dudy(j);
      F2(j,k)=cos((k-1)*j*pi/(N+1))*dbdy(j);
   end
end

C2=B2;
G1=B2;
G2=B1;
B1=B1*kalle_u;
B2=B2*kalle_b;
C1=C1*kalle_b;
C2=C2*kalle_u;
F1=F1*kalle_b;
F2=F2*kalle_u;
G1=G1*kalle_b;
G2=G2*kalle_u;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Matriserna motsvarande respektive term.

for j=1:N-1
   for k=1:N+2
      B(j,k)=-T(j,k)*(B1(j)+B2(j));
      C(j,k)=-T(j,k)*(C1(j)+C2(j));
      F(j,k)=-T(j,k)*(F1(j)+F2(j));
      G(j,k)=-T(j,k)*(G1(j)+G2(j));
   end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dags för högerleden i ekvationerna.

for j=1:N-1
   for k=1:N+2
      D1(j,k)=(A_koeff^2/Rm)*cos(pi*j*(k-1)/(N+1))*d2bdy2(j);
      D2(j,k)=(1/R)*cos(pi*j*(k-1)/(N+1))*d2udy2(j);
      H1(j,k)=(1/Rm)*cos(pi*j*(k-1)/(N+1))*d2bdy2(j);
   end
end
H2=D2;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sätter in delmatriserna i de stora matriserna.
deriv_matrix(1:N-1,1:N+2)=A+B;
deriv_matrix(1:N-1,N+3:2*N+4)=-C;
deriv_matrix(N:2*N-2,N+3:2*N+4)=E-G;
deriv_matrix(N:2*N-2,1:N+2)=F;
oderiv_matrix(1:N-1,N+3:2*N+4)=D1;
oderiv_matrix(1:N-1,1:N+2)=D2;
oderiv_matrix(N:2*N-2,1:N+2)=H1;
oderiv_matrix(N:2*N-2,N+3:2*N+4)=H2;

% Lägger in randvillkoren.
deriv_nymat=zeros(2*N+4);
oderiv_nymat=zeros(2*N+4);

oderiv_nymat(5:2*N+2,1:2*N+4)=oderiv_matrix;
deriv_nymat(5:2*N+2,1:2*N+4)=deriv_matrix;

% Randvillkoren.
for k=1:N+2
   deriv_nymat(1,k)=(-1)^(k-1); % u(-1)
   deriv_nymat(2,k+N+2)=(-1)^(k-1); % Bx(-1)
end

a_koeff=[2 0 -2/3 0 -2/15 0 -2/35 0 -2/63 0 -2/99 0 -2/143 0 -2/195 0 -2/255 0 -2/323];
% Plockas fram med Maple.

deriv_nymat(3,1:N+2)=a_koeff; % v(-1)
deriv_nymat(4,N+3:2*N+4)=a_koeff; % By(-1)
deriv_nymat(2*N+4,N+3:2*N+4)=ones; % Bx(+1)
deriv_nymat(2*N+3,1:N+2)=ones; % u(+1)

LG=zeros(2*N+4);
q=1;
oderiv_nymat=q^2*oderiv_nymat;
LG=inv(deriv_nymat)*oderiv_nymat;


aprim=LG*kalle;

return;
```