Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Eindhoven University of Technology

MASTER

Studying bifurcations in proposed L-H transition models using Bifurcator

Paquay, S.

Award date:
2012

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 08. Mar. 2018

Studying bifurcations in proposed L-H transition models using Bifurcator

Stefan Paquay

November 9, 2012

Abstract

The transition from low to high confinement mode in fusion plasmas has been intensively studied since its discovery, as it allows for much more efficient fusion reactors. The physical details behind the transition are not yet fully uncovered. However, the transition exhibits dynamic behaviour naturally corresponding to bifurcations (sharp transitions, dithering, hysteresis). Any physical model attempting to explain this transition should therefore also exhibit such bifurcations. In this research a numerical solver named Bifurcator was developed. It uses the method of lines using second order central differences in combination with a diagonally implicit Runge-Kutta method. Bifurcator was applied to two 1D (parabolic) transport models for the transition from low to high confinement mode. Both models exhibited the bifurcations associated with the transition. Furthermore, the locations of these bifurcations in parameter space qualitatively agreed with theoretical predictions. This agreement indicates that the numerical methods are indeed suitable to study bifurcations in 1D transport models, and thus that more complex models for which theoretical analysis is not viable can be studied numerically using Bifurcator.

Contents

Chapter 1

Introduction

In 1982 the *high confinement mode* (H-mode) was discovered at the ASDEX tokamak in Germany [1]. Thanks to the superior energy confinement associated with this mode of operation (with respect to the at that time standard mode of operation, the *low confinement mode* or L-mode), economical fusion power plants came a step closer to realisation. However, the exact details of the transition from the L-mode to H-mode are not yet fully understood. The focus of this research is on verification of proposed models for this transition, rather than the derivation of new models.

1.1 Characteristics of L-mode and H-mode

The most important differences (with respect to fusion power output) of the H-mode with respect to the L-mode are an increase in confinement time, combined with hotter and denser plasma cores. The density and temperature gradients at the edge steepen, and as a result the density and temperature in the core increase. The increase in confinement time and temperature means that the fusion reactor runs much more efficiently in H mode [2]. This reduces the cost-performance ratio of reactors significantly and brings economic nuclear fusion power plants closer to realisation. In figure 1.1 radially resolved profiles in the ASDEX-Upgrade tokamak are shown, at two different time levels. At the second time level, the profiles underwent the transition from L-mode to H-mode. The transition to H mode exhibits qualitative behaviour that can be identified with certain bifurcations (fold, Hopf, cusp). For example, a smooth increase of heating power can cause the sudden transition from L-mode to H-mode. In a very short time, the plasma shape becomes significantly different by the formation of a transport barrier, resulting in steep gradients in the density and temperature near the separatrix. With the modification of parameters that regulate the friction between the rotating plasma and the edge, these sharp transitions can be tuned out, leaving only smooth transitions from L- to H-mode.

Not only is there this sudden transition, there is also the presence of *hysteresis* in the transition: The threshold power for the backward transition is significantly lower than the threshold power for the forward transition. An experimental observation of hysteresis is shown in figure 1.2. Finally, under the right circumstances, it is possible for the plasma to go into a *dithering* behaviour: The plasma parameters continuously oscillate between L and H mode, and only after some time the plasma settles into H mode.

The combination of these three bifurcations are naturally organised by a co-dimension

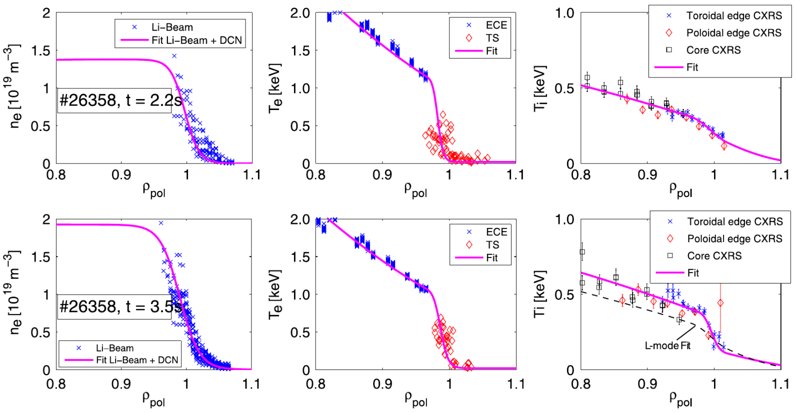


Figure 1.1: L-mode (upper) and H-mode (lower) profiles in the ASDEX-Upgrade tokamak. Taken from [3].

3 bifurcation.

1.2 Possible Mechanisms

Various mechanisms have been suggested that may explain the transition from L-mode to H-mode. The reduction in transport is usually attributed to a velocity shear. This velocity shear destroys turbulent eddies and thus suppresses turbulent radial transport. The $\mathbf{E} \times \mathbf{B}$ velocity shear is often suggested as being responsible [5]. For the origin of such a shear, many models have been proposed. These might be flows induced by a large pressure gradient, or by localized zonal flows generated by the turbulence itself. Zonal flows generated by the turbulence itself can play a role [6]. While both of these flows can explain reduction in radial transport, they do by themselves not explain how this suppression again amplifies the velocity shear, thus providing a self-enforcing state. A multitude of models have been proposed for this.
In this research, a physics-based model is studied where the transport coefficients for the density and temperature are coupled to an electric field. This electric field has its own self-consistent evolution equation, which is also coupled back to the density and temperature gradients. The generation of this radial electric field can also be due to different mechanisms. One such mechanism could be ion losses at the X point. Ion losses explain why an electric field builds up and thus an $\mathbf{E} \times \mathbf{B}$ velocity shear can build up, especially for high ion temperatures in the edge, as the H-mode threshold power depends on the direction of the ∇B-drift of the ions with respect to the divertor X-point.
Because most models consist of a system of partial differential equations which make theoretical analysis difficult, a numerical approach is necessary.

1.3 Numerical studies

The models describing this transition consist of transport equations with non-constant coefficients. These transport equations are parabolic, so they are quite dissipative. For

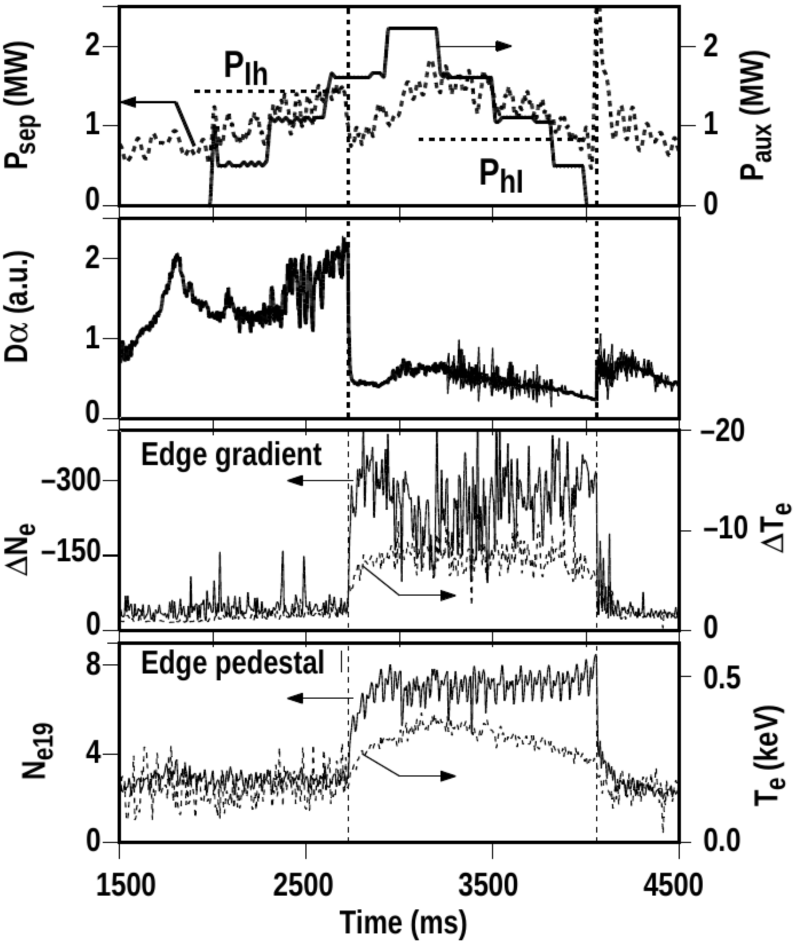


Figure 1.2: Time trace showing hysteresis in the heating power for the L-H transition. The heating power is increased and decreased in increments, and the heating power at which the transition from L-mode to H-mode occurs clearly differs from the heating power at which the transition back occurs. Taken from [4].

instance the density and temperature profiles are, in the absence of sources, expected to remain monotonic at all times. Also, the spatial domain is one dimensional, so this need not be computationally demanding. On the other hand, since the subject is bifurcations, the research will concentrate on those parts of parameter space where bifurcations, critical points/lines and sudden jumps in time and space need to be resolved. The numerical methods should be able to handle these cases routinely, if not always efficiently (i.e. the focus should be on general applicability rather than efficiently handling a single model so that most of the work can be applied to different models without much hassle).

1.4 Research questions

To test the various models proposed for the spontaneous generation of the H-mode, it will be studied if the proposed models show the typical bifurcation characteristics associated with L to H and H to L transitions. More specifically, they should exhibit sharp transitions, hysteresis, and dithering. Additionally, the various models can quantitatively influence the position of these bifurcations in parameter space. A proposed model might for example predict a threshold heating power that is totally inconsistent with thresholds known from experiments. Thus the positions of bifurcations obtained for different models in a parameter scan can also provide valuable information into the validity of the model. With this in mind, the physics research question is:
Do the proposed models exhibit qualitatively the bifurcation behaviour typical for the L-H transitions (large jumps in phase space, hysteresis, dithering)? Also, how do these models compare to each other quantitatively and qualitatively?
Since a numerical code to solve the equations was developed, there was also a numerical research question that was answered:
Which numerical techniques are required for dealing with the non-linearities of bifurcation behaviour in 1D parabolic transport models?
After all, the numerical study of bifurcations requires a focus on parameter values for which near-singular behaviour, and discontinuous dependence on parameters, is expected. The answer to the second research question is of course to provide a means to answer the first, but as the behaviour of the system is expected to be numerically not so well-behaved, it should be considered a point of research in its own right.

1.5 Outline of this report

In this thesis, theoretical models that should describe accurately the transition from L to H-mode in tokamak plasmas are studied. These models are described in chapter 2. This chapter also gives some theoretical background about bifurcations. Since the models are too complicated to study analytically without approximations, they are mostly solved numerically. For this purpose a numerical solver was developed. For more information about this program, the reader is referred to the program's user manual and source code documentation [7]. The specific numerical methods and algorithms used in this solver are discussed in chapter 3, which can safely be skipped if the reader is not interested in the details behind the numerical approach. In chapter 4 some benchmarks are performed to test the solver. The results of the numerical study of these models is also presented there. Finally, the results are discussed in chapter 5 and conclusions are drawn from these results, and chapter 6 gives suggestions for further research.

Chapter 2

Theory

2.1 Bifurcations in fusion plasmas

The dynamic behaviour of the L-H transition in fusion plasmas naturally correspond to certain bifurcations. The relevant bifurcations will be discussed in this section.

2.1.1 Sharp transitions and Hysteresis

One characteristic of the L-H transition in fusion plasmas is that the transition can happen very suddenly. With just a slight increase of heating power, all of a sudden the plasma enters H-mode, a transport barrier is formed at the edge and the confined density and temperature increase significantly [2]. This can be observed by ramping up the external heating power P when in L-mode or down when in H-mode. Then, at a critical value for P, suddenly the transport of heat and particles will decrease at the plasma edge and as a result of this, the profiles for n and T will follow this change on a transport time scale.

Mathematically, sharp transitions are the result of two *fold* bifurcations. A fold bifurcation occurs when by smoothly varying a parameter the solution to an equation fails to exist. A simple example is the following ordinary differential equation:

$$\dot{x} = a + x^2,$$

with a being a parameter. This equation has steady state solutions only when $a \leq 0$. In particular, $a < 0$ results in steady states for $x = \pm\sqrt{-a}$ and $a = 0$ results in a steady state for $x = 0$. If a is increased from $a < 0$ to $a > 0$, then no steady state solutions exist anymore. x will simply keep increasing towards ∞. It is easiest to portray this by drawing in the $a-x$-plane the curves for which $\dot{x} = 0$ (See figure 2.1a).

Not only can profiles exhibit a sharp transition when smoothly varying a parameter, the power at which the sharp transition from H-mode to L-mode occurs can also be different from the heating power required to go from L-mode to H-mode. This is known as *hysteresis*: the type of solution observed depends not just on the current parameter settings, but also on the history of the profiles.

Hysteresis also occurs when there are two fold bifurcations in the system (figure 2.1b). A simple example is the ODE

$$\dot{x} = a + bx - x^3. \tag{2.1}$$

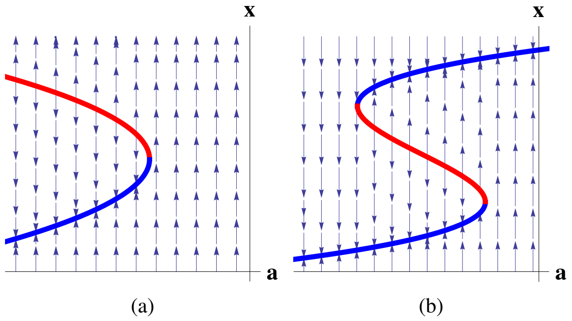


Figure 2.1: Left: a phase plot of $\dot{x} = a + x^2$ with a single fold bifurcation at $a = 0$. For $a \leq 0$ a stable (blue) and an unstable (red) branch for the differential equation which disappear when $a > 0$. Right: A phase plot of $\dot{x} = a + bx - x^3$, with two fold bifurcations at $a_{\text{crit},\pm}$ (at which $dx/da \to \pm\infty$), with two regions with one steady state branch ($a < a_{\text{crit},-}$ or $a > a_{\text{crit},+}$), and an intermediate region with three steady state branches ($a_{\text{crit},-} < a < a_{\text{crit},+}$, with the blue branches being stable and the red branch being unstable).

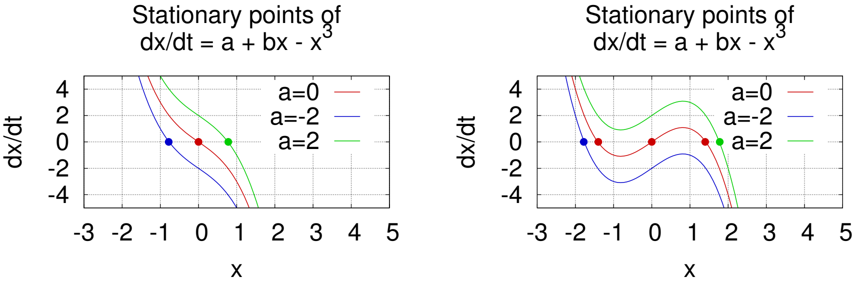
Depending on the values of a and b, the ODE can have 1, 2 or 3 stationary points. For example, for $a = 0$, both $x = 0$ and $x = \pm\sqrt{b}$ are roots of $\dot{x}$. This implies that for $b \leq 0$ only one real steady state solution is available, at $x = 0$ (figure 2.2a). If $b > 0$, there are three steady states for $a = 0$ (figure 2.2b). Furthermore, varying a can also result in changing the number of solutions, which can be seen in the other curves drawn in figures 2.2a and 2.2b.

The existence of hysteresis is best illustrated using figure 2.1b. This figure is slightly modified to illustrate what happens when dynamically changing the parameter a (figure 2.3. Hysteresis is observed because at all times x converges to the nearest steady state branch. Either a has a value that allows only one steady state value, or there are two. Since the nearest steady state branch depends on whether one comes from the top or bottom branch, the branch that is taken in the intermediate region with multiple steady state branches depends on wether x came from high ($a > a_{\text{crit},+},$) or low a ($a < a_{\text{crit},-},$). This implies hysteresis: the value of x in the intermediate region depends on the previous values of x and a.

The above picture was sketched for an ODE equivalent to equation (2.1) with $b > 0$. However, the existance of the middle branch and a regio with multiple branches also depends on b. What the parameter b basically determines is the distance between $a_{\text{crit},-}$ and $a_{\text{crit},+}$. For $b = 0$ it turns out that $a_{\text{crit},-} = a_{\text{crit},+}$, meaning that there is no hysteresis visible. Furthermore, for $b < 0$ there are no singularities (i.e. there is no a for which $dx/da = 0$), thus hysteresis is not possible. For $b > 0$ the magnitude of b determines the distance between the critical values for a, and thus also the width of the region in which hysteresis is observed.

2.1.2 Dithering

Oscillatory solutions can occur when an ODE or PDE has no steady steady state solution. In tokamaks such regimes are sometimes observed and are referred to as *dithering*.



(a)

(b)

Figure 2.2: Changing the parameters a and b can change the number of stationary points of the ordinary differential equation in (2.1). On the left, for $b < 0$ there is always just one stationary point. On the right, $b > 0$ and then, depending on the value of a, either one, two or three stationary points are available. In the case of three roots, the middle root is always unstable.

This happens most often during the ramping up of the heating power. During such a dithering phase the radially resolved profiles typically oscillate between L-mode-like and H-mode-like. To describe it in terms of bifurcations, an extra degree of freedom has to be added to the two-fold model, in the shape of the parameter c :

$$\dot{x} = -(a+bx+x^3)+c(x+y), \qquad \dot{y} = -(x+y) \tag{2.2}$$

This system of equations is actually a simplified variant of the *Fitzhugh-Nagumo* model, a well known model in bifurcation analysis. Dithering occurs when, due to the Hopf bifurcation, all stable steady state solutions have turned into unstable solutions. As a consequence of this, the variables will not converge to a final steady state value. Instead, they enter an oscillating regime, in which all variables change in a periodic fashion. For the model in equation (2.2), dithering can occur for many different parameter settings. Many combinations of parameters can lead to oscillatory behaviour. Phase plots for various values of b are shown in figure 2.4, which show the solution type changing from converging to steady states for $b > b_c$ to limit cycles for $b < b_c$. This change from steady-state to non-steady-state solutions is known as the *Hopf* bifurcation.

2.2 Models for the L-H transition

Of course, fusion plasmas are fairly complicated systems. When trying to describe them, a lot of possible effects have to be ignored just to keep the analysis of these systems viable. However, even a simple model should exhibit dithering, hysteresis and sharp jumps.

Some models are already known to contain the bifurcations observed in experiments, and as such are a good benchmark for studying suitable numerical methods for solving such equations. Of other models it is not necessarily known if they exhibit the same bifurcations. They can be used as actual “field study” cases.

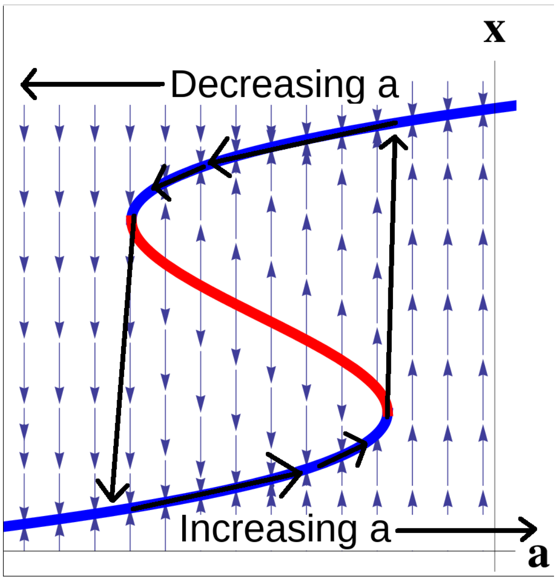


Figure 2.3: Schematic representation of the observed steady state solution for (2.1) when changing a, at constant $b > 0$. By increasing a, the steady state solution moves along the blue curve until this curve hits a singularity at $a_{\text{crit},+}$, where $dx/da \to \infty$. When a is increased beyond $a_{\text{crit},+}$, the nearest (and only) steady state solution is the top branch, so x now converges to that branch. When decreasing a when on this top branch the solution stays on the nearest branch (now the top), until $a = a_{\text{crit},-}$, where $dx/da \to -\infty$. When decreasing a below $a_{\text{crit},-}$, x again jumps to the bottom branch.

2.2.1 Zohm's model

One of the models we study definitely has all bifurcations relevant to this research located around a co-dimension 3 point. Zohm's model should serve as a proper test case for the numerical methods used, as it is known beforehand that it contains the relevant bifurcations. It was introduced and studied by Zohm in [8]. In this report this model will be denoted as "Zohm's model". The model consists of a system of three coupled partial differential equations, of which one is non-linear:

$$\begin{aligned}
\frac{\partial n}{\partial t} &= -\frac{\partial \Gamma}{\partial x}, \qquad \Gamma = -D(Z)\frac{\partial n}{\partial x} \\
\frac{\partial}{\partial t}\left(\frac{nT}{\gamma-1}\right) &= -\frac{\partial q}{\partial x}, \qquad q = -\chi(Z)n\frac{\partial T}{\partial x} + \frac{T}{\gamma-1}\Gamma \\
\varepsilon\frac{\partial Z}{\partial t} &= \mu\frac{\partial^2 Z}{\partial x^2} + c_n\frac{T}{n^2}\frac{\partial n}{\partial x} + c_T\frac{1}{n}\frac{\partial T}{\partial x} + G(Z) \\
G(Z) &:= a - b(Z - Z_S) - (Z - Z_S)^3.
\end{aligned}$$

The variables here are the density n, the temperature T, and the normalized electric field $Z := \rho_p e E_r / T_i$, with ρ_p the cyclotron radius of an electron in the poloidal direction, e the elementary charge, E_r the radial electric field and T_i the ion temperature. In this system of equations, the first equation represents conservation of particles in the

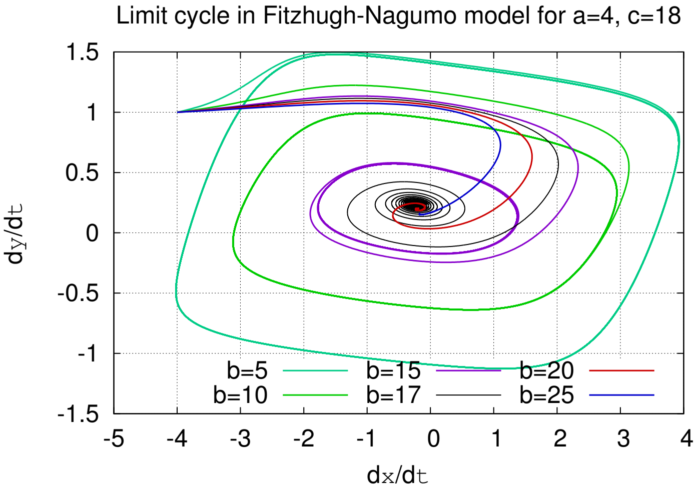


Figure 2.4: Dithering in the Fitzhugh-Nagumo model (2.2). When the parameter $b > b_c$, the system of equations has a steady state solution to which the variables x and y converge. However, if $b < b_c$, these steady state solutions are unavailable and an oscillatory solution is obtained. For these particular parameters, $15 < b_c < 17$

absence of sources. Change in density is driven purely by a diffusive flux Γ, which is coupled to the density gradient via the diffusivity D. The second equation is conservation of heat, again in the absence of sources. The thermal energy can change due to conduction driven by the temperature gradient and conductivity χ, as well as convectively via the particle flux. γ here represents the adiabatic constant. Finally, the third equation for Z represents a self-consistent evolution equation for the electric field, that can change due to gradients in n and T. The factor ε represents the typical time scale at which Z can change, which can be faster than for the density and temperature. The term μ is typically small and represents some friction the electric field has with itself, which smooths out Z. The factors c_n and c_T give weights for the contributions of the density and temperature gradients to the change in Z. Finally, the term $G(Z)$ is a polynomial that is an approximation to various other effects that can influence the electric field, e.g. friction with neutral beam heating. The parameters a, b and Z_S should therefore be determined from data of actual fusion experiments.
In Zohm's model, the diffusion and conduction coefficient are given by

$$\chi(Z) := 3D(Z), \qquad D(Z) := \frac{1}{2}(D_{\max} + D_{\min}) + \frac{1}{2}(D_{\max} - D_{\min})\tanh(Z).$$

In our study, this model is slightly generalised into $\chi(Z) := D(Z)/(\zeta(\gamma - 1))$, with $\gamma - 1 = \frac{2}{3}$.
The idea behind this model is that large negative values for Z cause a high diffusivity (associated with L-mode profiles) and large positive values cause low diffusivity (associated with H-mode profiles). A large increase in Z can be effected by large gradients in temperature and/or density, which in turn can be induced via heating power. This makes that Zohm's model can indeed show H-mode like profiles. It is a useful model as it contains the co-dimension 3 bifurcations, which are the main focus of this research.

2.2.2 Flow shear model

A more realistic model might have the diffusion and conduction coefficients D and χ coupled to the $\mathbf{E}\times\mathbf{B}$ flow shear. Since $Z\sim\mathbf{E}$ and $\mathbf{B}\approx constant$, this can be achieved by coupling D and χ to the gradient of Z. The flow shear reduced turbulent transport by tearing apart turbulent eddies that reside in the plasma edge. This in turn reduces the outward radial transport at the edge. The diffusion model based on the gradient of Z is:

$$\chi(Z)=\frac{D(Z)}{\zeta(\gamma-1)},\qquad D(Z):=D_{\min}+\frac{D_{\max}-D_{\min}}{1+a_1Z^2+a_2Z\frac{\partial Z}{\partial r}+a_3\left(\frac{\partial Z}{\partial r}\right)^2}$$

In the flow shear model, the electric field gradient now has the same functionality as the electric field magnitude had in Zohm's model: For large ∇Z the diffusion coefficient is close to the L-mode value $D_{\min}$, while for small ∇Z it is close to the L-mode value $D_{\max}$. This way, regions with high electric field gradients (which models increased flow shear) experience reduced transport.

2.2.3 Boundary conditions

No model is complete without boundary conditions. For the temperature and density, it is assumed that there is a passive scrape-off layer-like respons to particle and heat flux. This leads to exponential decay for the density and temperature:

$$\left.\frac{\partial n}{\partial x}\right|_{x=0}=\left.\frac{n}{\lambda_n}\right|_{x=0},\qquad \left.\frac{\partial T}{\partial x}\right|_{x=0}=\left.\frac{T}{\lambda_T}\right|_{x=0}$$

On the side of the plasma core, an influx of heat and particles is assumed. These influxes follow from heating power that is injected at in the plasma core, as well as fuelling of the plasma. The electric field is assumed to be flat there, at least with respect to the large jumps observed at the edge plasma in H-modes:

$$\Gamma(x\to\infty):=\Gamma_\infty\qquad q(x\to\infty):=q_\infty,\qquad \left.\frac{\partial Z}{\partial x}\right|_{x\to\infty}=0.$$

Unfortunately, no proper physical boundary condition for the electric field near the separatrix is known. Under consideration in this research are:

$$Z(0)=\text{const},\quad \left.\frac{\partial Z}{\partial x}\right|_{x=0}=0,\quad \left.\frac{\partial^2 Z}{\partial x^2}\right|_{x=0}=0,\quad \left.\frac{\partial Z}{\partial x}\right|_{x=0}=\frac{Z(0)}{\lambda_Z}$$

The first two of these conditions can be translated into physical interpretations in terms of the velocity of the plasma. After all, the plasma rotation at the edge is driven by $\mathbf{E}\times\mathbf{B}$, and, assuming $\mathbf{B}$ is almost constant, having $Z=0$ at the edge translates to $v_{\mathbf{E}\times\mathbf{B}}=0$, which is a *no-slip* boundary condition. Similarly, having $\partial Z/\partial x=0$ at the edge implies a *no-strain* BC. The third boundary condition is less intuitive, but effectively it achieves $\mu=0$ at the edge. The fourth boundary condition is a scrape-off layer-like response to the applied velocity shear, equivalent to those for the temperature and density.

Zohm proposes in his model a condition equivalent to $\partial^2Z/\partial x^2=0$ [8]. Note that there is no explicit term for the heating power in either model. Instead, the influence of plasma heating can be modelled through the heat and particle fluxes coming from the core, q_∞ and Γ_∞. This makes sense as the models here are purely edge modelling, where no heat is depositited but heat is still transfered to the edge through diffusive transport.

2.2.4 Initial conditions

For diffusive problems, initial conditions are not as important as boundary conditions, but a system of PDEs still cannot be solved numerically without initial profiles. In this research, the steady state profiles for $Z = const$ are used as initial conditions:

$$\begin{aligned} n(x) =& -\frac{\Gamma_\infty \lambda_n}{D(Z)} \left(1 + \frac{x}{\lambda_n}\right) \\ T(x) =& q_\infty \frac{\gamma - 1}{\Gamma_\infty} \left[1 - \frac{\lambda_n}{\zeta\lambda_T + \lambda_n} \left(1 + \frac{x}{\lambda_n}\right)^{-\zeta}\right]. \end{aligned}$$

The profiles for $Z = 0$ are shown in figure 2.5. The reason for this choice is that these profiles can actually be computed analytically. Since the numerical methods will include iterative methods (as the evolution equation for Z is non-linear), choosing an initial profile that is consistent with the equations makes it easier to converge to the actual steady state solution.

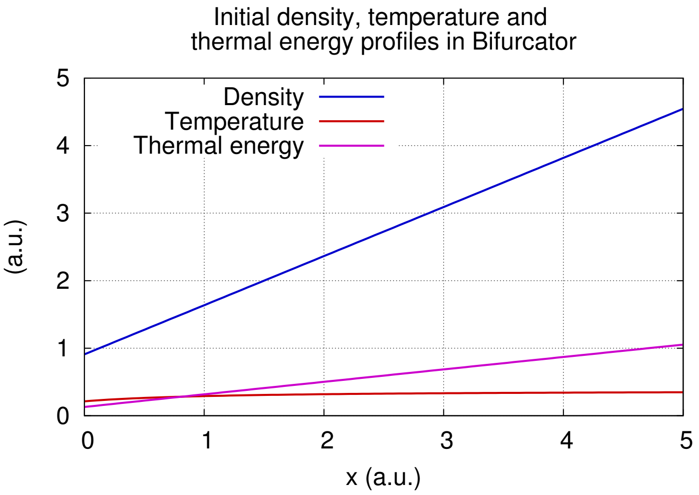


Figure 2.5: Initial profiles for n and T used in Bifurcator. They are the steady state profiles for $Z = const = 0$.

2.3 Theoretical predictions of bifurcation locations

It is possible for Zohm's model to theoretically predict where the bifurcations will be located [9]. For arbitrary functions for the diffusivity, the location of the Hopf and fold bifurcations can be computed. Based on some assumptions illustrated in [9], the following relation holds in the region where the edge transport barrier is formed:

$$D^2(Z) = -\theta G(Z) D(Z), \qquad \theta := \frac{\Gamma_\infty^2 \lambda_n^2}{q_\infty \zeta(\gamma - 1)} \frac{\lambda_n + \zeta\lambda_T}{c_n \lambda_T + c_T \lambda_n}$$

Combining this criterion with the conditions for the fold and Hopf bifurcations, which are $d(GD)/dz = 0$ and $d(G/D)/dZ = 0$ respectively, expressions for curves in parameter space can be derived that separate regions with different solution types. In

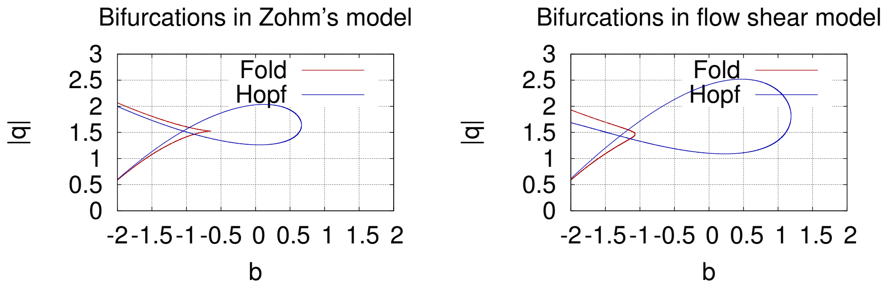


Figure 2.6: Theoretical predictions of the bifurcation landscape for variable parameters b and q, and the other parameters at values in table 2.1.

figure 2.6 these curves are shown for Zohm's model and the flow shear model. Zohm's model uses the boundary condition $(\partial^2 Z/\partial x^2)_{x=0} = 0$ while the flow shear model uses $(\partial Z/\partial x)_{x=0} = Z_0/\lambda_Z$. Clearly, for these values the bifurcation landscapes qualitatively look quite alike. Even though the boundary conditions differ, the shape of these curves suggests that both models should exhibit similar transition characteristics. This hypothesis will be tested with numerical analysis in chapter 4.

The fish-like curves in 2.6 should be interpreted as follows: In the region in between the two fold lines, there are three steady state solutions to the differential equations. Outside of the fold, there is only one. The Hopf curve is a bit more tricky, but predicts the following: Inside the loop made by the intersection of the Hopf curve with itself, a region of oscilaltory solutions is expected. Thus, increasing the heat flux for that value of b should result in a dithering transition to H-mode, which is sometimes observed in experiments.

To the left of the loop, the Hopf curve lies close to the fold curve. In between the two Hopf curves, both steady state solutions predicted by the fold are stable. This allows two solutions in this region, making hysteresis possible. By crossing the Hopf curve at the higher heating power side, the L-mode solution becomes unstable. On the other hand, crossing the Hopf curve at the low heating power side, the H-mode solution becomes unstable. Finally, to the right of the Hopf loop, there is only one steady state solution without bifurcations, so there only smooth transitions from L-mode to H-mode take place.

Table 2.1: Parameter values corresponding to the bifurcation landscapes in figure 2.6.

Γ_∞	λ_n	ζ	λ_T	λ_Z	γ	c_n	c_T
-0.8	1.25	1.1	1.5	1.25	$\frac{5}{3}$	1.1	0.9
D_{min}	D_{max}	a	Z_s	a_1	a_2	a_3	
0.8	2.0	-1.5	1.4	1	0	1	

Chapter 3

Numerical techniques

For the numerical study of bifurcations, proper choices have to be made for the numerical approach. This chapter describes the various methods that are chosen for numerically solving the problems, and why precisely these methods were chosen. The techniques discussed here are all implemented in or have at some point been implemented in *Bifurcator*, a C++ program developed in the course of this research. The actual implementation is omitted from this report for the sake of brevity, but can be found in the Bifurcator manual [7]. Bifurcator is designed to be especially suitable to solve partial differential equations containing bifurcations. Furthermore, it has some useful features for searching for bifurcations, the most useful being the ability to "home in" on the value of a single parameter for which a transition from one solution type to another occurs. Descriptions of the implementation, as well as the source code, can be found in the program manual [7]. A flow chart of Bifurcator is shown in figures 3.1 and 3.2.

Basically, Bifurcator uses the *method of lines* for solving the system of partial differential equations (PDEs): First, space is discretised by some finite difference approximation in order to transform the system of PDEs to systems of ordinary differential equations (ODEs). Then various time integration methods for systems of ODEs can be applied to the obtained system. Furthermore, Bifurcator can adaptively refine the spatial discretisation grid to guarantee a specific vertical resolution. The numerical methods and algorithms used in Bifurcator will be discussed in this chapter.

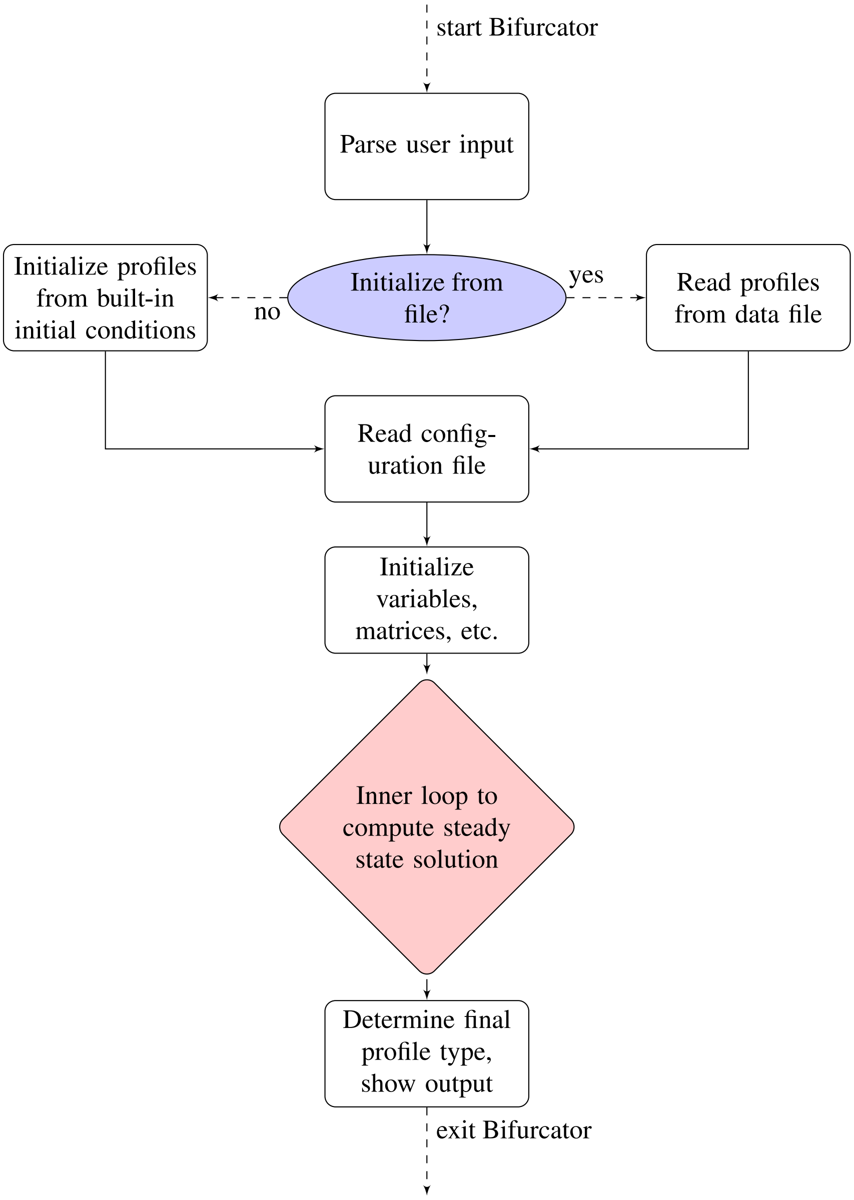

start Bifurcator
Parse user input
Initialize from file?
no
yes
Initialize profiles from built-in initial conditions
Read profiles from data file
Read configuration file
Initialize variables, matrices, etc.
Inner loop to compute steady state solution
Determine final profile type, show output
exit Bifurcator

Figure 3.1: The flow of Bifurcator for computing a steady state profile. The details for the inner loop are shown in figure 3.2.

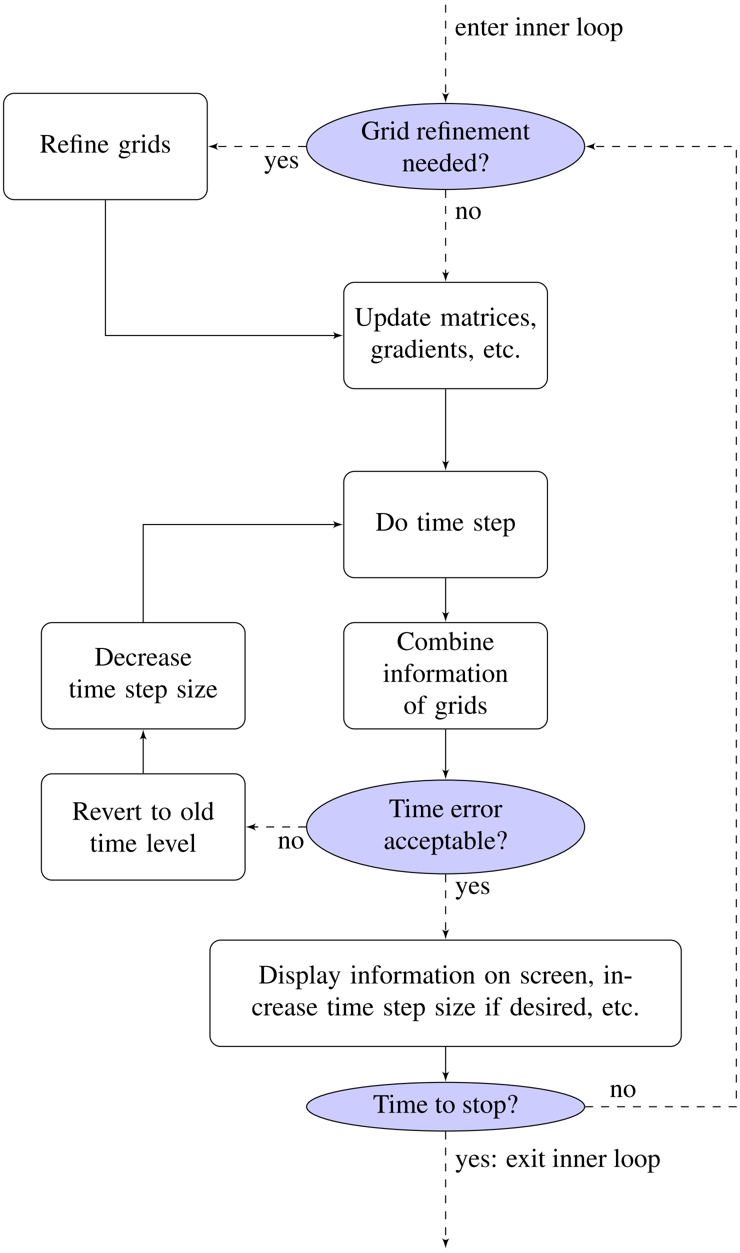


Figure 3.2: The inner loop of Bifurcator that computes the solution to the system of equations.

3.1 Discretisation of space

For the discretisation of space, three points central differences are used. First, the physical domain is reduced to a set of points, with indices i ranging from 0 to $N-1$. If the physical length of the domain is R, then this leads to a grid spacing of $\Delta x := R/(N-1)$. The variable N is the resolution of the grid. In these grid points, approximations to some unknown variable u are computed. These values are denoted as u_i, with i again being the index corresponding to the grid numbering. Using central differences, the approximation to the first and second spatial derivative of a variable become:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x},$$
$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2},$$

The accuracy of these approximations can be derived by applying a Taylor series to them:

$$u_{i\pm 1} := u(x \pm \Delta x) = u(x) \pm \Delta x u'(x) + \frac{(\Delta x)^2}{2} u''(x) \pm \frac{(\Delta x)^3}{6} u'''(x) + \frac{(\Delta x)^4}{24} u''''(x) \ldots$$

The accent here denotes derivation with respect to x. With this, the linear combination approximating the first derivative turns out to be

$$\frac{u_{i+1} - u_{i-1}}{2\Delta x} = \frac{2\Delta x u'(x) + \frac{(\Delta x)^3}{3} u'''(x) + O(\Delta x)^5}{2\Delta x} = u'(x) + O\left((\Delta x)^2\right)$$

In a similar fashion, the approximation for the second derivative becomes

$$\frac{u_{i+1} + u_{i-1} - 2u_i}{(\Delta x)^2} = \frac{(\Delta x)^2 u''(x) + \frac{(\Delta x)^4}{12} u''''(x) + O(\Delta x)^6}{(\Delta x)^2} = u''(x) + O\left((\Delta x)^2\right)$$

Thus, the central difference approximations for both the first and second derivative are accurate up to terms of order $(\Delta x)^2$. This means that doubling the grid resolution should result in results that are four times as accurate. Also, since the terms that follow the $(\Delta x)^2$ are third or higher derivatives of u, the central difference method should exactly resolve problems for which the exact solution is a polynomial of degree 2 or less. The following example will illustrate this.

3.1.1 Example of central difference exactness for special cases

Consider the following ordinary differential equation:

$$u''(x) = a, \qquad u(0) = u'(0) = 1$$

Obviously, the exact solution can be found by integrating twice and applying the boundary conditions. This grants $u(x) = \frac{a}{2}x^2 + x + 1$.
Discretising the domain $[0, 1]$ with central differences results in a grid of size N and a grid spacing Δx, with $i = 0$ corresponding to $x = 0$ and $i = N - 1$ corresponding to

$x = 1$. For the inner grid points (i.e. the points for which $i > 0$ and $i < N-1$) this results in the following difference equation:

$$\frac{u_{i+1} + u_{i-1} - 2u_i}{(\Delta x)^2} = a.$$

Such an equation can be solved by using z-transforms. The details are not that important, the solution is

$$u_i = a\frac{(i+1)(i+2)}{2}(\Delta x)^2 + i\left(\Delta x - \frac{3a}{2}(\Delta x)^2\right) + \left(1 - a(\Delta x)^2\right)$$

Substitution in the difference equation indeed results in $u_{i+1} + u_{i-1} - 2u_i = a(\Delta x)^2$, thus this solution is correct. Expanding all the terms in this equation and simplifying results in a lot of terms canceling, and the only thing left will be $ai^2(\Delta x)^2/2 + i\Delta x + 1$. Since by definition $i\Delta x = x_i$, this means that

$$u_i = \frac{a}{2}(i\Delta x)^2 + (i\Delta x) + 1 = \frac{a}{2}x_i^2 + x_i + 1 = u(x_i),$$

thus the central difference approximation is exact. The left boundary is also resolved exactly, as the boundary conditions imply $u_0 = 1$ (by the first boundary condition) and $u_{-1} = u_1 - 2\Delta x$ (by applying central differences to the second boundary condition), resulting in

$$\frac{u_1 - 2u_0 + u_{-1}}{(\Delta x)^2} = \frac{2u_1 - 2\Delta x - 2}{(\Delta x)^2} = a \to u_1 = \frac{a}{2}(\Delta x)^2 + \Delta x + 1,$$

which of course equals $u(\Delta x)$. Showing $u_0 = u(0) = 1$ is trivial, simply substituting $i = 0$ shows this.

3.1.2 Staggered grids

Some equations contain not pure second derivatives, but rather the derivative of some function multiplied by a first derivative, like for example the density:

$$\frac{\partial n}{\partial t} = -\frac{\partial}{\partial x}\Gamma = \frac{\partial}{\partial x}\left(D\frac{\partial n}{\partial x}\right). \tag{3.1}$$

It is possible to discretise these equations in such a way that they obey the physical conservation law, in the sense that "flux in = flux out" for each grid point, while still being second order consistent. This will be illustrated in this section.
The spatial derivative in equation (3.1) is best resolved by applying a central difference approximation to the first derivative twice, but halfway the grid points:[1]

$$\begin{aligned}\frac{\partial}{\partial x}\left(D\frac{\partial n}{\partial x}\right)\Big|_{x_i} &\approx \frac{\delta_{\frac{1}{2}}\left(D_i(n_{i+\frac{1}{2}} - n_{i-\frac{1}{2}})\right)}{(\Delta x)^2} \approx \frac{D_{i+\frac{1}{2}}(n_{i+1} - n_i) - D_{i-\frac{1}{2}}(n_i - n_{i-1})}{(\Delta x)^2} \\ &= \frac{1}{(\Delta x)^2}\delta_{\frac{1}{2}}(D_i\delta_{\frac{1}{2}}(n_i)),\end{aligned}$$

[1] An alternative way would be to split the derivatives using the product rule and apply the central difference schemes from section 3.1 to these derivatives. This approach is also second order consistent, but applying it to for example the diffusion equation grants $\frac{d}{dt}(n_i + n_{i+1}) \neq \Gamma_{i+\frac{3}{2}} - \Gamma_{i-\frac{1}{2}}$ with $\Gamma_{i\pm\frac{k}{2}} := \frac{1}{\Delta x}D_{i\pm\frac{k}{2}}\left(n_{i\pm\frac{k}{2}+\frac{1}{2}} - n_{i\pm\frac{k}{2}-\frac{1}{2}}\right)$, which violates conservation of particles.

with the difference operator $\delta_{\frac{1}{2}}(y_i)$ defined as $\delta_{\frac{1}{2}}(y) = y_{i+\frac{1}{2}} - y_{i-\frac{1}{2}}$. Of course the values $D_{i\pm\frac{1}{2}}$ are not known, but they can be computed via interpolation. Linear interpolation is second order consistent, just like central differences, so this does not add a significant error to the scheme. This can be easily seen by applying Taylor series to the interpolation formula halfway the grid points:

$$\begin{aligned}
D_{i+\frac{1}{2}} \approx& \frac{1}{2}(D_i + D_{i+1}) = \\
&\frac{1}{2}\left(D_{i+\frac{1}{2}} - \frac{\Delta x}{2}D'_{i+\frac{1}{2}} + \frac{(\Delta x)^2}{8}D''_{i+\frac{1}{2}} + \frac{\Delta x}{2}D'_{i+\frac{1}{2}} + \frac{(\Delta x)^2}{8}D''_{i+\frac{1}{2}}\right) + O((\Delta x)^3) \\
&= D_{i+\frac{1}{2}} + \frac{(\Delta x)^2}{8}D''_{i+\frac{1}{2}}
\end{aligned}$$

Here again the prime denotes derivation with respect to x. Shifting the index i with 1 results in the formula for $D_{i-\frac{1}{2}}$.

To demonstrate that the staggered grid approach also gives a second order consistent approximation, again apply Taylor series:

$$\begin{aligned}
\delta_{\frac{1}{2}}(D_i\delta_{\frac{1}{2}}(n_i)) =& D_{i+\frac{1}{2}}\left(n_i + \Delta x n'_i + \frac{(\Delta x)^2}{2}n''_i + \frac{(\Delta x)^3}{6}n'''_i + \frac{(\Delta x)^4}{24}n_i^{(4)} + O((\Delta x)^5) - n_i\right) \\
&- D_{i-\frac{1}{2}}\left(n_i - \left(n_i - \Delta x n'_i + \frac{(\Delta x)^2}{2}n''_i - \frac{(\Delta x)^3}{6}n'''_i + \frac{(\Delta x)^4}{24}n_i^{(4)} - O((\Delta x)^5)\right)\right) \\
=& D_{i+\frac{1}{2}}\left(\Delta x n'_i + \frac{(\Delta x)^2}{2}n''_i + \frac{(\Delta x)^3}{6}n'''_i + \frac{(\Delta x)^4}{24}n_i^{(4)} + O((\Delta x)^5)\right) \\
&+ D_{i-\frac{1}{2}}\left(-\Delta x n'_i + \frac{(\Delta x)^2}{2}n''_i - \frac{(\Delta x)^3}{6}n'''_i + \frac{(\Delta x)^4}{24}n_i^{(4)} - O((\Delta x)^5)\right) \\
=& \Delta x n'_i(D_{i+\frac{1}{2}} - D_{i-\frac{1}{2}}) + \frac{(\Delta x)^2}{2}n''_i(D_{i+\frac{1}{2}} + D_{i-\frac{1}{2}}) \\
&+ \frac{(\Delta x)^3}{6}n'''_i(D_{i+\frac{1}{2}} - D_{i-\frac{1}{2}}) + \frac{(\Delta x)^4}{24}n_i^{(4)}(D_{i+\frac{1}{2}} + D_{i-\frac{1}{2}}) + O((\Delta x)^6)
\end{aligned}$$

Applying Taylor series to the terms with $D_{i\pm\frac{1}{2}}$ leads to the following identities:

$$\begin{aligned}
D_{i+\frac{1}{2}} + D_{i-\frac{1}{2}} =& \frac{1}{2}(2D_i + D_{i+1} + D_{i-1}) = 2D_i + \frac{(\Delta x)^2}{2}D''_i + O((\Delta x)^4) \\
D_{i+\frac{1}{2}} - D_{i-\frac{1}{2}} =& \frac{1}{2}(D_{i+1} - D_{i-1}) = \Delta x D'_i + \frac{(\Delta x)^3}{6}D'''_i + O((\Delta x)^5).
\end{aligned}$$

Substituting them into the expression for the staggered grid leads to

$$\begin{aligned}
\delta_{\frac{1}{2}}(D_i\delta_{\frac{1}{2}}(n_i)) =& \Delta x n'_i\left(\Delta x D'_i + \frac{(\Delta x)^3}{6}D'''_i\right) + \frac{(\Delta x)^2}{2}n''_i\left(2D_i + \frac{(\Delta x)^2}{2}D''_i\right) \\
&+ \frac{(\Delta x)^3}{6}n'''_i\left(\Delta x D'_i + \frac{(\Delta x)^3 D'''_i}{6}\right) + \frac{(\Delta x)^4}{24}n_i^{(4)}D^{(4)} + O((\Delta x)^6)
\end{aligned}$$

Now, combing all terms with the same order of Δx together results in

$$\delta_{\frac{1}{2}}(D_i\delta_{\frac{1}{2}}(n_i)) = (\Delta x)^2\left(n'_iD'_i + D_in''_i\right) + \frac{(\Delta x)^4}{12}\left(2n'_iD'''_i + 3n''_iD''_i + 2n'''_iD'_i\right) + O((\Delta x)^6).$$

The exact derivative is of course $n_i''D_i + n_i'D_i'$ by virtue of the product rule, so the approximation $\delta_{\frac{1}{2}}(D_i\delta_{\frac{1}{2}}(n_i))/(\Delta x)^2$ is indeed second order consistent. Furthermore, adding the time derivatives in two adjacent grid points results in

$$\begin{aligned}
\frac{d}{dt}(n_i + n_{i+1}) =& \frac{1}{(\Delta x)^2}\left[D_{i+\frac{1}{2}}(n_{i+1} - n_i) - D_{i-\frac{1}{2}}(n_i - n_{i-1})\right] \\
&+ \frac{1}{(\Delta x)^2}\left[D_{i+\frac{3}{2}}(n_{i+2} - n_{i+1}) - D_{i+\frac{1}{2}}(n_{i+1} - n_i)\right] \\
=& \frac{1}{(\Delta x)^2}\left[D_{i+\frac{3}{2}}(n_{i+2} - n_{i+1}) - D_{i-\frac{1}{2}}(n_i - n_{i-1})\right]
\end{aligned}$$

The right hand side can be rewritten in terms of the particle fluxes halfway two grid points:

$$\frac{d}{dt}(n_i + n_{i+1}) = \frac{1}{\Delta x}\left[\Gamma_{i+\frac{1}{2}} - \Gamma_{i-\frac{1}{2}} + \Gamma_{i+\frac{3}{2}} - \Gamma_{i+\frac{1}{2}}\right] = \frac{1}{\Delta x}\left[\Gamma_{i+\frac{3}{2}} - \Gamma_{i-\frac{1}{2}}\right].$$

If one considers a control volume around each cell, centered on the grid points, then the edges of those cells coincide exactly with the point halfway the grid points (i.e. the positions $i \pm \frac{1}{2}$). Therefore, the fluxes $\Gamma_{i\pm\frac{1}{2}}$ can be interpreted as the fluxes flowing in and out of these control volumes at the cell interfaces. With this interpretation, the above equation translates to "flux in = flux out", implying that with this staggered grid scheme, physically conserved quantities are also conserved numerically.

Applying the staggered grid approximations to the spatial derivatives in the system of PDEs, the equations are transformed into systems of ODEs, with one equation per grid point for each of the variables:

$$\begin{aligned}
\frac{d}{dt}\mathbf{n} =& \mathbf{An} + \mathbf{f}_n \\
\frac{d}{dt}\mathbf{U} =& \mathbf{BU} + \mathbf{f}_U \\
\frac{d}{dt}\mathbf{Z} =& \mathbf{CZ} + \mathbf{f}_Z(\mathbf{n}, \mathbf{U}, \mathbf{Z})
\end{aligned}$$

The "right-hand side" functions $\mathbf{f}_n$ and $\mathbf{f}_U$ follow purely from the boundary conditions, while $\mathbf{f}_Z(\mathbf{n}, \mathbf{U}, \mathbf{Z})$ contains the boundary condition for Z as well as the contribution of the gradients of n and T and the non-linear self-interaction of Z. These systems can then be solved by one of many time integration methods. They are discussed in the next section, section 3.2.

3.2 Time integration methods

The system of ODEs obtained after spatial discretisation can be time-integrated by various time integrators. Most time integration schemes generalise quite easily from a single equation to a system of equations. The problem is that these systems can be quite stiff, and usually are for diffusion equations, especially with large grids. There is no clear mathematical definition of what stiff means [10], but a good working definition is: *A differential equation is stiff if the time step size needed to maintain stability with an explicit method is much smaller than the physical time scale of interest.* This definition also quite closely captures the essence of the problem: Because the interesting behaviour of the ODE occurs on the smallest physical time scale or larger, it would make sense accuracy-wise to choose a time step size in the order of a fraction of this physical time scale. However, the stability constraints do not allow this. To obtain proper numerical results with an explicit method on a stiff problem, the time step size has to be much smaller than the time step size dictated by accuracy. Thus, a lot of extra work is done to obtain uninteresting results.

These problems can be circumvented by using an implicit time integration scheme. While these are usually more computationally expensive as they require the solving of systems (in this research even non-linear), they can be constructed so that they are *unconditionally stable*, meaning that there is no constraint on the time step size. This allows them to beat explicit methods in computation time. Some of the implicit time integration methods will be discussed below. First, however, some basics about numerical (time) integration will be discussed.

3.2.1 Numerical integration

All numerical time integration methods deal with the following problem: Suppose an ODE or a system of ODEs has to be integrated:

$$\frac{d}{dt}y = f(t,y), \qquad y(0) = y_0 \tag{3.2}$$

with f some single-valued function of both time t and y itself. The usual approach is to integrate y from y_0 at $t = 0$ to some desired time level t_m via integration:

$$y(t_m) = y_0 + \int_{t=0}^{t=t_m} f(t,y(t))dt.$$

The range of the integral is usually divided into many time levels, and at each time level the integral is approximated with a fairly simple formula. Then the value for y at the next time level t_{n+1} is given by

$$y(t_{n+1}) = y_n + \int_{t=t_n}^{t=t_{n+1}} f(t,y(t))dt. \tag{3.3}$$

Most numerical methods to solve differential equations do so by solving equation (3.3) starting at $n = 0$, and substituting the numerical solution for $y(t_n)$ back into the scheme to compute an approximation for $y(t_{n+1})$. Since the function $f(t,y)$ is not easily integrated (else a numerical integration would not be necessary), most numerical methods solve this equation by replacing the integral with some approximation, say I_n, to obtain

$$y(t_{n+1}) \approx y_{n+1} := y_n + \Delta t I_n.$$

During this chapter, the numerical solutions to $y(t_n)$ be denoted as y_n.

For solving a system of ODEs, a problem can arise with implicit methods. After all, the system of ODEs now reads

$$\frac{d}{dt}\mathbf{y} = \mathbf{F}(t,\mathbf{y}),$$

where $\mathbf{F}$ can be nonlinear. Since implicit methods rely on values of $\mathbf{F}$ for t and $\mathbf{y}$ that are, in principle, not known, this means that we now have to solve a system of nonlinear equations. This can be done, however, using *Newton iteration*. An algorithm for Newton iteration for systems of equations is given in algorithm 1.

3.2.2 Consistency

Before going in to details about the various time integration methods, the property of *consistency* will be discussed. The *order of consistency* of a method basically tells you how accurate it is. It is easiest to demonstrate consistency using the local discretisation error. Let $\varepsilon(t_n;\Delta t)$ be the error in a solution at time $t_n := n\Delta t$, obtained by some numerical method, using some time step size Δt :

$$\varepsilon(t_n;\Delta t) := |y_{\text{numerical}}(t_n;\Delta t) - y_{\text{exact}}(t_n)|.$$

Then the numerical method is said to be consistent if

$$\lim_{\Delta t \to 0} \varepsilon(t_n;\Delta t) = 0.$$

Furthermore, the order of consistency of a numerical method is the largest value for p for which

$$\lim_{\Delta t \to 0} \frac{\varepsilon(t_n;\Delta t)}{(\Delta t)^p} = 0.$$

Intuitively, consistency determines if a numerical method will approximate the exact solution for a sufficiently large resolution, and the order of consistency determines how fast the approximation improves as the resolution increases.
As an example, consider the *explicit Euler method.* Its scheme reads

$$y_{n+1} = y_n + \Delta t f(t_n, y_n).$$

The exact solution at the next time level can also be expressed as a Taylor series approximation for the current time level:

$$y(t_{n+1}) = y_n + \Delta t f(t_n, y_n) + \frac{(\Delta t)^2}{2}\dot{f}(t_n, y_n) + \frac{(\Delta t)^3}{6}\ddot{f}(t_n, y_n) + \ldots.$$

Algorithm 1 Newton iteration to compute a root of a system $\mathbf{y} = \mathbf{F}(\mathbf{y})$

```
Let $\mathbf{y}_0$ be some initial guess, and $\varepsilon$ the accuracy tolerance
Set $k = 0$
Compute the residual $\mathbf{r}_k(\mathbf{y}_k) = \mathbf{y}_k - \mathbf{F}(\mathbf{y}_k)$
Compute the Jacobian matrix of $\mathbf{r}_k(\mathbf{y}_k)$ : $\mathbf{J}_k(\mathbf{r}(\mathbf{y}_k)) = \begin{pmatrix} \frac{\partial r_1}{\partial y_1} & \cdots & \frac{\partial r_1}{\partial y_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_N}{\partial y_1} & \cdots & \frac{\partial r_N}{\partial y_N} \end{pmatrix}$
Set $\lambda_k = \|\mathbf{r}_k(\mathbf{y}_k)\|$
while $\lambda_k > \varepsilon$ do
    Compute $\mathbf{s} = -\mathbf{J}^{-1}(\mathbf{r}(\mathbf{y}))\mathbf{r}(\mathbf{y})$
    Update approximate solution: $\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{s}$
    Set $k = k + 1$.
    Update the residual.
    Update the Jacobi matrix.
    Set $\lambda_k = \|\mathbf{r}_k(\mathbf{y}_k)\|$
end while
```

Thus, the local error is

$$\begin{aligned}
\varepsilon(t_{n+1};\Delta t) =& \left| y_n + \Delta t f(t_n,y_n) - \left(y_n + \Delta t f(t_n,y_n) + \frac{(\Delta t)^2}{2}\dot{f}(t_n,y_n) + \dots \right) \right| \\
=& |(\Delta t)^2 \dot{f}(t_n,y_n) - \frac{(\Delta t)^2}{2}\dot{f}(t_n,y_n)| + O((\Delta t)^3) \\
=& \left| (\Delta t)^2 \left(\dot{f}(t_n,y_n) - \frac{1}{2}\dot{f}(t_n,y_n) \right) \right| + O((\Delta t)^3)
\end{aligned}$$

From this expression, it is quite clear that $\varepsilon(t_{n+1};\Delta t) \to 0$ as $\Delta t \to 0$. Furthermore, the order of consistency turns out to be 1, as

$$\lim_{\Delta t \to 0} \frac{\varepsilon(t_{n+1};\Delta t)}{\Delta t} = 0$$

but

$$\lim_{\Delta t \to 0} \frac{\varepsilon(t_{n+1};\Delta t)}{(\Delta t)^2} \neq 0$$

A concept related to the local error is the *global error*. This can be though of as the total accumulation of local errors. It is usually one order lower than the local error. This is because for a local error of order p, for any time domain $[0,\tau]$ there are at most $\max_n \tau/\Delta t_n$ steps, which results in a total error of

$$\varepsilon_{\text{global}} = \sum_n \frac{\tau}{\Delta t_n} \cdot \varepsilon_{\text{local}} \leq \max_n \frac{\tau}{\Delta t_n} O((\Delta t_n)^p) = \max_n O((\Delta t_n)^{p-1}).$$

In the case of the explicit Euler method, this means that halving the time step sizes halves the error. Since halving the time step also means one has to do twice as much work, this is not very efficient. The order of consistency is typically the same as the order of the global error.

The order of consistency is one of the properties that determined if a time integration method was considered suitable or not. For each of the assessed methods, its order of consistency will be discussed at least briefly. The other most important property is *stability*. However, unlike consistency, it is convenient to discuss the stability of a method only after introducing the method itself. Therefore, first the various considered time integration methods will be discussed in sections 3.2.3-3.2.5, and after that the stability of all these methods will be discussed (section 3.2.6).

3.2.3 The Euler methods

The simplest implicit method is the implicit Euler method or backward Euler method. In the case of the implicit Euler method, the integral in equation (3.3) is approximated simply by an end-point approximation:

$$y(t_{n+1}) \approx y_{n+1} = y_n + \Delta t_n f(t_{n+1}, y_{n+1}), \tag{3.4}$$

where $\Delta t_n := t_{n+1} - t_n$. If the right hand side is evaluated at t_n, y_n instead, the explicit Euler method is obtained. It is possible to combine the implicit and explicit Euler methods into the so-called ϑ-method. This method approximates the integral by a linear combination of the implicit and explicit Euler methods, with $0 \leq \vartheta \leq 1$:

$$y(t_{n+1}) \approx y_{n+1} = y_n + \Delta t_n \left[(1-\vartheta) f(t_n, y_n) + \vartheta f(t_{n+1}, y_{n+1}) \right].$$

For $\vartheta = 0$ the explicit Euler is obtained, and for $\vartheta = 1$ the implicit Euler is obtained. Note that in general, the function f can be non-linear and thus equation (3.4) might have to be solved iteratively. Newton iteration is quite suitable for this. With only simple modifications, algorithm 1 can also be applied to the ϑ-method.

Whenever stability is required, the implicit Euler method is a very good choice. However, its accuracy leaves much to be desired. Following the procedure of section 3.2.2 and some Taylor expansions, it can be shown that the implicit Euler method is consistent only up to first order, just like the explicit Euler method. The ϑ-method with $\vartheta = \frac{1}{2}$ is second order consistent (it is the trapezoidal rule), making it a better choice for accuracy. However, with $\vartheta = \frac{1}{2}$ the method is only barely stable, and the lack of numerical damping makes that it is not well suited for solving stiff systems.

What is important is that the implicit Euler method is numerically very stable, as will be shown in section 3.2.6. Since the combination of stability and consistency is necessary and sufficient for convergence[2], the implicit Euler method is always a good choice when the accuracy of the time integration is not that important.

In this research however, accuracy could very well turn out to be important. Therefore, a time integration method that combines the stability of the implicit Euler with higher accuracy is required. In sections 3.2.4 and 3.2.5 two of such methods will be discussed and it will be assessed how suitable they are for dealing with bifurcations. The stability of all methods will be discussed in section 3.2.6.

3.2.4 Backward differentiation formula

During the initial stages of this research, it turned out that most explicit methods were not sufficiently stable to solve the system of equations. This quickly lead to the investigation of only so-called *A-stable* methods. (Note that from the pronunciation, it seems as if an A-stable method is not stable. In fact the opposite is true, A-stable methods are unconditionally stable.) This concept will be discussed in more depth in section 3.2.6. For now it is sufficient to think of A-stable methods as methods for which there is no stability constraint on the time step size.

In order to achieve a higher accuracy while retaining A-stability, there are two approaches: multistep methods and Runge-Kutta methods. Of the multistep methods, only the *backward differentiation formula* methods (BDF) are A-stable with consistency orders higher than 1 (although no A-stable BDF of order > 2 exist) [10,11]. This section will briefly discuss the BDF, its merits and why it is not chosen as the main time integration method in this research.

The BDF method can be interpreted most easily as a generalisation to the implicit Euler method. Instead of searching for a y_{n+1} so that

$$y_{n+1} - y_n = \Delta t_n f(t_{n+1}, y_{n+1}),$$

a higher order interpolation polynomial is used for the left hand side, leading to

$$y_{n+1} - \frac{4}{3}y_n + \frac{1}{3}y_{n-1} = \Delta t_n f(t_{n+1}, y_{n+1}) \tag{3.5}$$

for a second order consistent method. The large benefit of the BDF method is that it is also A-stable[3]and the extra accuracy comes with a fairly mild penalty. Extra storage

[2] Actually, this is only true for linear problems. In the case of non-linear equations, it is not really clear what exactly leads to what. It is possible for an approximation to a nonlinear solution to converge even though it was not stable. However, for reasonably well-behaved functions, this statement is probably still true.

for one time level is needed, and that is it. This means that the BDF method obtains good accuracy without a large additional cost. Furthermore, the BDF is rock solid and very capable of dealing with stiff systems. In fact, the BDF is among the first numerical methods proposed specifically for stiff differential equations [12]. The biggest problem however is that the coefficients in the formula are only constant if the time step size does not change. Since in this research we expect dithering (with periods of little dynamic activity and periods with a lot of activity), it is very convenient if the time step size can be changed dynamically to match the "activity" of the equations. However, implementing a BDF that is capable of working with such adaptive time steps is far from trivial. This is the main reason why Runge-Kutta methods were analysed.

3.2.5 Diagonally implicit Runge-Kutta methods

An alternative way to achieve a higher accuracy in a time integration method is by approximating the integral in 3.3 by a more accurate numerical quadrature formula. This is the basis of Runge-Kutta (RK) methods. The explicit RK methods are quite well-known, although they are totally unfit for solving stiff systems. Fortunately, it is possible to construct implicit RK methods that are just as accurate and that retain most of the stability properties the implicit Euler method has.

A Runge-Kutta method solves ODEs by computing an approximation to the right hand side of equation (3.3). These approximations consist of multiple evaluations and/or approximations of the integrand at various positions. The values of the integrand there are known as *stages*. The implicit Euler method can be thought of as a special Runge-Kutta method of one stage. A general formula for computing the stages is

$$k_i = f\left(t_n + \Delta t_n c_i, y_n + \Delta t \sum_j a_{ij} k_j\right), \tag{3.6}$$

where i ranges from 1 to the number of stages, say s. Then, a linear combination of the stages is added to y_n to obtain an approximation for y_{n+1} :

$$y_{n+1} = y_n + \Delta t_n \sum_{i=1}^{s} b_i k_i$$

The implicit Euler can be interpreted as a Runge-Kutta method with $s = c_1 = b_1 = a_{11} = 1$. Also, by taking $s = 1$, $c_1 = 0$, $b_1 = 1$ and $a_{11} = 0$, the explicit Euler method is obtained. The most convenient way of describing a Runge-Kutta method is by using a so-called Butcher tableau, which is a table containing the coefficients c_j, b_j and a_{ij} in a structured manner. See table 3.1 for Butcher tableaus for the explicit Euler, implicit Euler and ϑ-methods, as well as a general Butcher-tableau with the symbols b_i, c_i and a_{ij}.

Obviously, there are s weights b_i and c_i, and $s \times s$ a_{ij}-coefficients. There are certain criteria the weights and coefficients should satisfy for the method to be consistent and stable. For explicit RK methods, these coefficients can simply be derived from Taylor series. For implicit methods, it is a bit more tricky, and the details will be omitted. Suitable coefficients are chosen from the various literature on the subject [10, 11, 13].

[3] The BDF method is only A-stable up to second order. For orders 4 to 6 it is A(α)-stable, a term which will be explained in 3.2.6. Suffice to say the method is still very stable, even for orders 4 to 6. Only when the exact solution to the problem contains oscillations of high frequencies that damp out very slowly there might be problems.

Table 3.1: Butcher tableaus for the explicit Euler method (left), implicit Euler method (centre-left) and ϑ-method (centre-right). Note that the ϑ-method is actually a two-stage method. The last table (right) is a general Butcher tableau with three stages, for illustration purposes. In practice, values of 0 are usually omitted from the tableaus.

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \qquad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0 & 1 \\ \hline & 1-\vartheta & \vartheta \end{array} \qquad \begin{array}{c|ccc} c_1 & a_{11} & a_{12} & a_{13} \\ c_2 & a_{21} & a_{22} & a_{23} \\ c_3 & a_{31} & a_{32} & a_{33} \\ \hline & b_1 & b_2 & b_3 \end{array}$$

Algorithm 2 Basic flow of a general, three-stage Runge-Kutta method

```
Set $n = 0$.
Set $t_n = 0$.
Set $\mathbf{y}_n = \mathbf{y}_0$, with $y_0$ the initial condition.
Let $\Delta t_n$ be some initial time step size.
Let $\tau$ be the final time.
while $t_n < \tau$ do
    Compute $\mathbf{k}_1 = \mathbf{f}(t_n + c_1\Delta t, \mathbf{y}_n + \Delta t(a_{11}\mathbf{k}_1 + a_{12}\mathbf{k}_2 + a_{13}\mathbf{k}_3))$
    Compute $\mathbf{k}_2 = \mathbf{f}(t_n + c_2\Delta t, \mathbf{y}_n + \Delta t(a_{21}\mathbf{k}_1 + a_{22}\mathbf{k}_2 + a_{23}\mathbf{k}_3))$
    Compute $\mathbf{k}_3 = \mathbf{f}(t_n + c_3\Delta t, \mathbf{y}_n + \Delta t(a_{31}\mathbf{k}_1 + a_{32}\mathbf{k}_2 + a_{33}\mathbf{k}_3))$
    Set $\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\,[b_1\mathbf{k}_1 + b_2\mathbf{k}_2 + b_3\mathbf{k}_3]$.
    Set $t_{n+1} = t_n + \Delta t_n$.
    Set $n = n + 1$.
    (Optionally: Compute a better time step $\Delta t_n$.)
end while
```

The order of consistency of the method can be tested numerically on some test problems, which will be done in section 4.1. The stability is easier to assess, and will be done in section 3.2.6. The general flow of using a Runge-Kutta method is summed up in algorithm 2.

The system of equations considered in this research contains some non-linear systems. For those systems, the stages have to be computed iteratively (the procedure is quite similar to the one described in algorithm 1). From equation (3.6) it is obvious that each stage can depend on other stages. However, in principle, each of the stages is unknown. It is vital for the sake of the Newton iteration process that the stages can be computed sequentially, so that each non-linear equation contains only a single unknown. This immediately implies that all a_{ij} with $j > i$ should be 0. This leads to the so-called diagonally implicit Runge-Kutta (DIRK) methods, as only the diagonal and below-diagonal entries are non-zero. In this research three DIRK methods were considered. Their Butcher tableaus are shown in table 3.2. The DIRK methods have some advantages over BDF methods, and one disadvantage. First of all, they can be constructed in such a way that they are A-stable or even L-stable, independent of their order. The second advantage is that DIRK methods with s stages are typically one order more accurate than BDF methods of s steps. Of course, this price is paid by being more computationally expensive, which is the disadvantage of DIRK methods with respect to BDF methods. Finally, because DIRK methods are single step methods, the time step size can be changed dynamically each step without having to change anything in the time integration routines. This is useful, because it allows for retrying a time integration step with a modified step size when something went wrong. Most notably, it can be used to retry a time step with a reduced time step size when Newton iteration fails to converge. As an added bonus, by a technique known as *embedding*, it is possible with a DIRK method (or any Runge-Kutta method for that manner) to obtain an estimate for the error due to the discretisation of time at hardly any computational cost. This method will be explained in section 3.2.7.

3.2.6 Stability of time integration methods

The stability of a numerical method is just as important as the consistency. It is determined by considering the test equation $y' = \lambda y$. If the real part of $\lambda < 0$, then the absolute value of the exact solution is monotonically decreasing. For the numerical method to be stable, this *monotonicity condition* has to be satisfied for the numerical solution as well.

As an example, consider both the implicit and explicit Euler methods. The schemes for these methods applied to the test equation are

$$\begin{aligned} y_{n+1}^{\mathrm{EE}} &= y_n^{\mathrm{EE}} + \Delta t \lambda y_n^{\mathrm{EE}} = (1 + \lambda \Delta t) y_n^{\mathrm{EE}}, \\ y_{n+1}^{\mathrm{IE}} &= y_n^{\mathrm{IE}} + \Delta t \lambda y_{n+1}^{\mathrm{IE}} \to y_{n+1}^{\mathrm{IE}} = \frac{1}{(1 - \lambda \Delta t)} y_n^{\mathrm{IE}} \end{aligned}$$

Table 3.2: Butcher tableaus for the DIRK methods studied in this research. The left method is third order consistent and A-stable, the centre method is second order consistent and L-stable (a special case of A-stability which will be explained in section 3.2.6). The right method has two rows with weights. The idea behind this is that by using the same stages but different weights (b_i-coefficients) two different numerical solutions can be found of different orders of consistency. These two approximations can then be used to give an estimate for the local error, which can be used for time step size control (this will be discussed further in section 3.2.7). In this case, the first row of weights result in third order consistency and L-stability, while the second row of weights results in second order consistency and A-stability. Note that all L-stable methods have $b_i = a_{si}$. (The implicit Euler method from table 3.1 also satisfies this condition, and it is in fact also L-stable.) The coefficients for the last method are given by $\gamma = 1 - \frac{1}{2}\sqrt{2}(\cos\phi - \sqrt{3}\sin\phi)$, $\phi = \frac{1}{3}\arctan\frac{\sqrt{2}}{4}$, $b_2 = \frac{3(1-4\gamma+2\gamma^2)^2}{4(1-6\gamma+9\gamma^2-3\gamma^3)}$, $b_1 = 1 - b_2 - \gamma$, $a_{21} = \frac{1-4\gamma+2\gamma^2}{2b_2}$ and $b' = \frac{1-2\gamma}{2-2\gamma}$.

(a) Coefficients for A-stable, two-stage DIRK.

$$\begin{array}{c|cc} \frac{3+\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} & \\ \frac{3-\sqrt{3}}{6} & -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(b) Coefficients for L-stable two-stage DIRK.

$$\begin{array}{c|cc} \frac{2-\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} & \\ 1 & \frac{\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} \\ \hline & \frac{\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} \end{array}$$

(c) Coefficients for DIRK methods with three stages.

$$\begin{array}{c|ccc} \gamma & \gamma & & \\ \gamma + a_{21} & a_{21} & \gamma & \\ 1 & b_1 & b_2 & \gamma \\ \hline & b_1 & b_2 & \gamma \\ & 1 - b' & 0 & b' \end{array}$$

Here, EE denotes the explicit Euler method and IE the implicit Euler method. The monotonicity conditions for the schemes are

$$
\begin{aligned}
|y_{n+1}^{\mathrm{EE}}| \leq |y_n^{\mathrm{EE}}| &\rightarrow |1+\lambda\Delta t| \leq 1 \\
|y_{n+1}^{\mathrm{IE}}| \leq |y_n^{\mathrm{IE}}| &\rightarrow |1-\lambda\Delta t| \geq 1.
\end{aligned}
$$

These conditions only have to hold for $\mathrm{Re}(\lambda) < 0$, as only then is the exact solution also monotonically decreasing. For the explicit Euler method, the region for which the monotonicity condition holds is the interior of the circle of radius 1 in the complex plane, centred around $\lambda\Delta t = -1$. For the implicit Euler method however, the region of stability is the region *outside* the circle in the complex plane with radius 1 centred around $\lambda = 1$. For the explicit Euler there is thus a constraint on the time step size: it can be no larger than $1/\lambda$. For the implicit Euler there are no constraints on the time step size, as $|1-\lambda\Delta t| > 1$ for all λ with $\mathrm{Re}(\lambda) < 0$. Therefore, it is unconditionally stable. This property is also known as *A-stability*.

The above computations of the stability regions can be done for all numerical methods. The monotonicity condition is then generalised to

$$
|y_{n+1}^{M}| \leq |R(\lambda\Delta t)||y_n^{M}|
$$

with $R(\cdot)$ a rational function (i.e. a ratio of two polynomial functions). The superscript M denotes some unspecified numerical method. The method is then again stable if $|R(\lambda\Delta t)| < 1$ for $\mathrm{Re}(\lambda\Delta t) < 0$ (this implies $\mathrm{Re}(\lambda) < 0$ as Δt is a real number). The region of stability is then the area in the complex plane bounded by $|R(\lambda\Delta t)| = 1$ on the side where $|R(\lambda\Delta t)| < 1$. For more advanced methods it is difficult to derive analytic constraints on the time step size, but contour plots are a great way of showing the regions of stability. In figure 3.3 the stability regions of the explicit and implicit Euler methods are plotted, as well as the stability regions for the DIRK methods discussed in section 3.2.5. Some methods are not A-stable, but they *almost* are. The BDF methods of orders 4 to 6 are an example of this. For such methods, the concept of A(α)-stability is useful. A method is A(α)-stable if it has $|R(\lambda\Delta t)| < 1$ for all $\lambda\Delta t$ with $\{\mathrm{Re}(\lambda\Delta t) < 0, \arg(\lambda\Delta t) < \alpha\}$, with $\arg(\lambda\Delta t)$ its complex argument $\arctan(\mathrm{Im}(\lambda\Delta t)/\mathrm{Re}(\lambda\Delta t))$. This idea can be illustrated by mentally picturing a cone over the stability regions in figure 3.3 with an angle of α with the real axis. In the case of the BDF methods of orders 3 to 6, the values for α are 86.03°, 73.35°, 51.84° and 17.84° [10]. For orders 3 and 4, this loss of A-stability is only noticeable for problems where $\lambda\Delta t$ has a large imaginary part and a small negative real part, which implies a problem with fast oscillations that die out very slowly. When the problem at hand is a stiff problem without fast oscillations, a BDF method of order 3 or 4 should still be very suitable, so the fact that they are not A-stable should not be a definitive reason for dismissing them. In this research, their unwieldiness in combination with adaptive time step sizes, however, is.

A property closely related to A-stability is *L-stability*. A numerical scheme is L-stable if it is A-stable and also satisfies $|R(\lambda\Delta t)| \rightarrow 0$ for $\lambda\Delta t \rightarrow -\infty$. For DIRK methods up to order 3, L-stability adds constraints on the coefficients of the form $b_i = a_{si}$, with s the number of stages. The implicit Euler method also satisfies this condition, as it has $a_{11} = b_1 = 1$ and $s = 1$. The explicit Euler method is not A-stable, so it is also not L-stable. L-stability basically means that instantaneously damping modes of the solution are also damped out instantaneously. The stability functions of the two three-stage DIRK-methods are shown in figure 3.4, which nicely show that for the L-stable DIRK $|R(Z)|$ indeed goes to 0 while for the A-stable DIRK it does not. Since for three stages it is only possible to construct one L-stable DIRK, the embedded DIRK variant

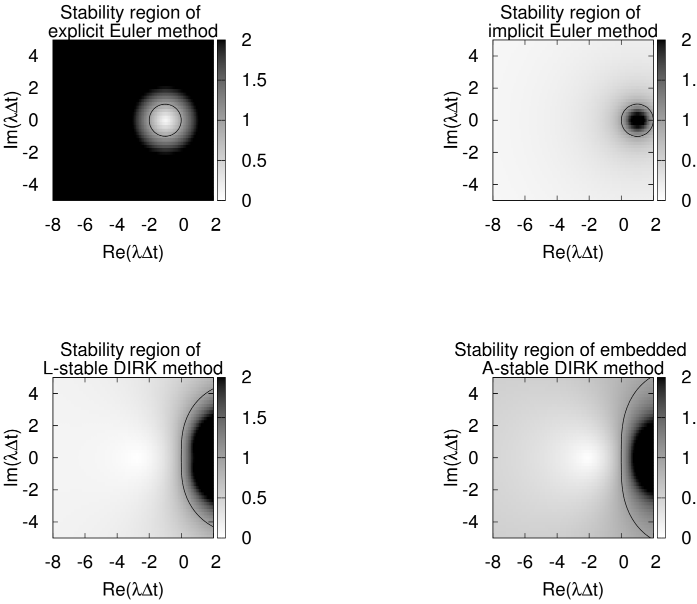


Figure 3.3: Regions of stability for various time integration methods (white, enclosed by the black contour). The explicit Euler method (top left) is conditionally stable with the condition $|1+\lambda\Delta t| < 1$. The implicit Euler method is unconditionally stable, as the region of stability encompasses the entire left half of the complex plane. The same is true for the L-stable DIRK (bottom left) and the A-stable embedded DIRK (bottom right). Both DIRK methods use the coefficients in table 3.2c, and are both implemented in Bifurcator.

was chosen to minimise the value for $R(\lambda\Delta t)$ for $\lambda\Delta t \to -\infty$ under the constraint that one of the quadrature weights b_i is 0. This way, although the embedded method is not actually L-stable as well, it does complement the L-stable DIRK that is used as the main integrator.

3.2.7 Error estimation and time step size control

One of the most useful features of DIRK methods (or many Runge-Kutta integrators in general) is the so-called "embedding" of a second RK solver with an order of accuracy different from the other. This was already hinted at in the previous sections, and will be explained here.

For some Runge-Kutta methods, it is possible to find two different sets of coefficients b_i and b'_i for the same coefficients a_{ij} and c_i, so that

$$y_{n+1}^{RK1} := y_n + \Delta t \sum_{i=1}^{s} b_i k_i = y_{n+1}^{\text{real}} + C_1(\Delta t)^p + O((\Delta t)^{p+1})$$

$$y_{n+1}^{RK2} := y_n + \Delta t \sum_{i=1}^{s} b'_i k_i = y_{n+1}^{\text{real}} + C_2(\Delta t)^{p-1} + O((\Delta t)^p)$$

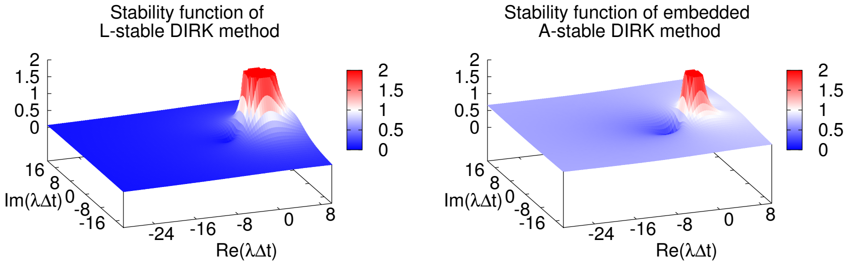


Figure 3.4: Stability functions for the embedded DIRK pair. Left is the L-stable DIRK and right is the A-stable DIRK. As expected, for the L-stable DIRK $R(\lambda\Delta t) \to 0$ for $\lambda\Delta t \to -\infty$, while for the A-stable DIRK it stagnates at around 0.7.

It is possible to form a linear combination of these two approximations to form an estimate for the error in the lower order approximation:

$$y_{n+1}^{RK2} - y_{n+1}^{RK1} = C_2(\Delta t)^{p-1} - C_1(\Delta t)^p + O((\Delta t)^p) = C_2(\Delta t)^{p-1} + O((\Delta t)^p) \tag{3.7}$$

This is of course an estimate for the error in y_{n+1}^{RK2} with respect to the exact solution, assuming there is no error in the approximation at the previous time level.
This error estimate is very useful, as the computation of all the stages gets done only once. There is therefore very little overhead in the computation of this error. As a result of this, it gives a very practical way of adaptive time step size control. This will be illustrated now. Suppose that we want the error in the simulation to remain bounded by some tolerance $\eta_{\Delta t}$. Since the actual solution is unknown, the best we can do is require that some estimation for the time stepping error is bounded by the tolerance for some time step size. The error made with the old time step is approximately

$$\varepsilon_{\text{old}} := C_2(\Delta t_{\text{old}})^{p-1},$$

while the error with a new, better time step size is estimated to be

$$\varepsilon_{\text{new}} := C_2(\Delta t_{\text{new}})^{p-1}.$$

In principle, C_2 changes in between time levels but is taken to be constant as an approximation. Combining the two equations gives an expression for C_2 :

$$C_2 = |y_{n+1}^{RK2} - y_{n+1}^{RK1}|/(\Delta t_{\text{old}})^{p-1}$$

Substituting in the expression for ε_{new} results in

$$\varepsilon_{\text{new}} := |y_{n+1}^{RK2} - y_{n+1}^{RK1}| \left(\frac{\Delta t_{\text{new}}}{\Delta t_{\text{old}}}\right)^{p-1}$$

Requiring that $\varepsilon_{\text{new}} = \eta_{\Delta t}$ then results in

$$\eta_{\Delta t} := |y_{n+1}^{RK2} - y_{n+1}^{RK1}| \left(\frac{\Delta t_{\text{new}}}{\Delta t_{\text{old}}}\right)^{p-1},$$

from which Δt_{new} can be isolated to obtain

$$\Delta t_{\text{new}} = 0.9\left(\frac{\eta_{\Delta t}}{|y_{n+1}^{RK2} - y_{n+1}^{RK1}|}\right)^{\frac{1}{p-1}} \cdot \Delta t_{\text{old}}. \tag{3.8}$$

Here a safety factor of 0.9 is added to attempt to correct for accidental underestimations of the error. In practice, the time step does not have to be changed each step to the optimal value. Usually, the time step is changed only if the error estimation exceeds the tolerance by some factor α. In Bifurcator, $\alpha = 4$. Of course, when the time step size gets increased because the error estimate was too small, there is no need to recompute the profiles at the latest time level. However, should the error estimate be too large, the obtained numerical results for t_{n+1} are too inaccurate to be trusted. They should then be discarded and instead the time step should be reduced and the computation of the profiles at later time levels should be resumed from t_n with this new optimal time step. Since the equations solved in this research are also non-linear, keeping the error bounded is not always enough for convergence. The Newton iteration used to solve the equation for Z should also converge within a reasonable amount of iterations. Newton iteration typically fails to converge if the solution to the system of equations is too far away from the starting approximation. In Bifurcator, the initial approximation is the profile from the previous time step. As a consequence, this initial profile might be too far away from the next if the time step size is too large. To accommodate for this, when the maximum number of iterations is reached, the profiles for n, T and Z are again reverted to the previous time level t_n, and the time step size is halved. The factor of $\frac{1}{2}$ might not be optimal, but in practice it works fine so there did not seem to be a reason to change it. Of course, resuming from a previous time step is quite costly, as quite a few operations have already been performed (namely the computation of n and T, as well as a significant number of Newton iterations). By modifying equation (3.8) slightly, convergence becomes a lot more regular. Let N be the number of Newton iterations used in the previous time step. Then, modifying equation (3.8) to

$$\Delta t_{\text{new}} = 0.9\left(\frac{\eta_{\Delta t}}{\sqrt{N}|y_{n+1}^{RK2} - y_{n+1}^{RK1}|}\right)^{\frac{1}{p-1}} \cdot \Delta t_{\text{old}} \tag{3.9}$$

will result in a better time stepping strategy. The reason for this is because the actual error tolerance is now reduced with a factor of $\sqrt{N}$, pushing the time step size below the time step size required purely for accuracy reasons. This reduces the possibility of having a time step size large enough for the solutions for Z between two time steps being too far away from each other, as it drives the time step size towards a value for which only one or two iterations are needed.

A second trick to allow for much more regular convergence is adding a maximum growth factor c_{max} to the time step size. Every time the time step size is changed, it becomes

$$\Delta t_{\text{new}} = \Delta t_{\text{old}} \cdot \min\left(c_{\text{max}}, 0.9\left(\frac{\eta_{\Delta t}}{\sqrt{N}|y_{n+1}^{RK2} - y_{n+1}^{RK1}|}\right)^{\frac{1}{p-1}}\right) \tag{3.10}$$

This is a second way of preventing the “distance” between the solutions at subsequent time levels becoming too large. Such a maximum growth factor tends to make the Newton iteration penalty from equation (3.9) obsolete, but for robustness it makes sense to keep both implemented.

Equation (3.10) is also the expression for the optimal time step size used in Bifurcator, and the time step reverting criteria discussed above are also implemented.

3.3 Adaptive grid refinement

One of the key features of Bifurcator is its capability to adaptively refine the grid. This ensures that all bifurcations are resolved at a sufficient resolution, no matter what the initial grid resolution is. It might also significantly reduce computation time, as the regions where little activity takes place also require few grid points, so the effective system size of a properly resolved bifurcation might actually be lower if using an adaptive grid.

Implementing an adaptive grid is not trivial. The choices for refinement and implementation will be discussed in this section. Also, the algorithms for growing and shrinking grids have to be derived, and a proper hierarchy between the grids has to be chosen. All these choices will be discussed in this section. It might be a bit technical.

3.3.1 Patching the base grid

Bifurcator starts with one or two grids, depending on the adaptive grid mode chosen. Each grid consists of an array of values for n, T and Z, as well as a radial position x. Whenever refinement is needed for some of these positions, they are flagged for refinement. Then, after time integration of the current time level has finished, a new grid is spawned on the flagged region, which is twice as dense (i.e. it has grid spacing $\Delta x/2$) as the previous grid. On this finer grid, the same equations are solved numerically on the inner grid points. Depending on the position of the patches, however, the boundary conditions might vary. After all, the physical boundary conditions are only prescribed on the physical edge of the domain. For patches lying only on the inner grid points of the base grid, another boundary condition has to be solved.

In the implementation of Bifurcator, the patches have physical boundary conditions on the left and/or right of their first and/or last grid point overlaps with the physical edge of the domain. Else, the boundary conditions have to be replaced with a *local boundary condition*. Two simple variants are possible: Local Dirichlet and Local Neumann boundary conditions. The will be briefly explained in the following sections.

Local boundary conditions of the patches

The reasoning behind this choice is that the profiles of the patches and their parent grids should at least be continuous. This makes sense in this research as it deals with conservation laws, but might not always make sense. The easiest way to achieve continuity is to make the edge values of the profiles equal to the overlapping values of the parent grid, leading to *local Dirichlet boundary conditions*. Since the first and last points of the patches actually overlap with the parent grid, this prescribes a boundary condition for the virtual points of the patch: The values in the virtual grid points are the mean of the parent values in the last coinciding point and the first point next to it of the parent. See figure 3.5 for a sketch of this. Note that in order for this scheme to work, the parent grid always has to have points to the left and right of the patch, unless the edge is physical.

Of course, sane initial values are also a requirement. In Bifurcator, the initial values of the grid points of the patch that coincide with a point of the base grid are copied, while

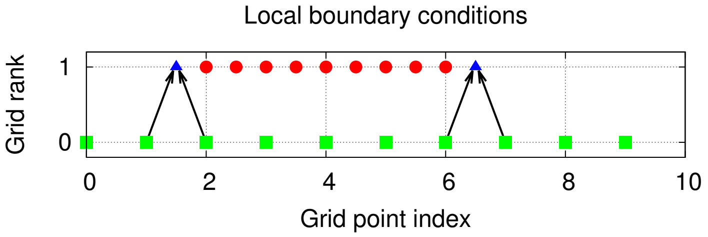


Figure 3.5: A graphical representation of local boundary conditions. The virtual grid points on the left and right (blue triangles) get values determined by the two parent grid points to the left and right of the grid point (green squares at the base of the arrows). Using these boundary conditions and the regular equations for the inner grid points, the profiles on the patch are computed (red circles).

the points in between are interpolated in the same way as the virtual grid points. After some test cases, it turns out that this assumption is actually really bad, because for different grid sizes, the numerical methods should converge to different numerical solutions. This means that the final steady state solutions on patches generally do not overlap, and thus it makes no sense to make the solutions continuous across patches. In fact, this is detrimental to the solutions because this forced continuity results in a steep gradient at the edge of the patch. This in turn results in an unrealistic value of the particle and heat fluxes at the edge of the patches.

Energy conservation across patches

A more reasonable approach is to use the fact that in the current models, particles and energy are conserved. Thus, the fluxes in left-most and right-most points of the patch should equal the corresponding fluxes of the parent. This leads to the following boundary conditions for the density and thermal energy for the left:

$$\begin{aligned}
\Gamma_{p,j} =& \Gamma_{c,0} = -D(Z_{c,0})\frac{\partial n_c}{\partial x}\Big|_0 \\
q_{p,j} =& q_{c,0} = -\left[(\gamma-1)\chi(Z_{c,0})n_{c,0}\left(\frac{\partial}{\partial x}\frac{U_c}{n_c}\right)\Big|_0 + \frac{D_{c,0}U_{c,0}}{n}\frac{\partial n}{\partial x}\right]\Big|_0 \\
&= -(\gamma-1)\chi(Z_{c,0})n_{c,0}\frac{\partial}{\partial x}\left(\frac{U_c}{n_c}\right)\Big|_0 + \frac{U_{c,0}}{n_{c,0}}\Gamma_{p,j}.
\end{aligned}$$

In the above equations, j represents the offset of the profiles of the child from the parent, so the j'th grid point of the parent is the 0'th grid point of the child. The subscript p, j thus denotes the j'th grid point of the parent. Similarly, the subscript $c, 0$ indicates the first grid point of the child grid. If the patch consists of N grid points, the local boundary condition for the right side becomes

$$\begin{aligned}
\Gamma_{p,j+(N+1)/2} =& \Gamma_{c,N-1} = -D(Z_{c,N-1})\frac{\partial n_c}{\partial x}\Big|_{N-1} \\
q_{p,j} =& q_{c,0} = -(\gamma-1)\chi(Z_{c,N-1})n_{c,N-1}\frac{\partial}{\partial x}\left(\frac{U_c}{n_c}\right)\Big|_{N-1} + \frac{U_{c,N-1}}{n_{c,N-1}}\Gamma_{p,j+(N+1)/2}.
\end{aligned}$$

These equations can be discretised using central differences to obtain expressions for the values for n and U in the virtual grid points. For Z there is no corresponding flux or quantity to be conserved, so for Z only Dirichlet boundary conditions are implemented. Discretisation leads to

$$\begin{aligned}
n_{c,l} =& n_{c,1} + \frac{2\Delta x_c}{D(Z_{c,0})}\Gamma_{p,j} \\
n_{c,r} =& n_{c,N-2} - \frac{2\Delta x_c}{D(Z_{c,N-1})}\Gamma_{p,j+(N+1)/2} \\
U_{c,l} =& n_{c,l}\left(\frac{U_{c,1}}{n_{c,1}} + \frac{2\Delta x_c}{(\gamma-1)\chi_{c,0}n_{c,0}}\left[q_{c,0} - \frac{U_{c,0}}{n_{c,0}}\Gamma_{p,j}\right]\right) \\
U_{c,r} =& n_{c,r}\left(\frac{U_{c,N-2}}{n_{c,N-2}} - \frac{2\Delta x_c}{(\gamma-1)\chi_{c,N-1}n_{c,N-1}}\left[q_{c,N-1} - \frac{U_{c,N-1}}{n_{c,N-1}}\Gamma_{p,j+(N+1)/2}\right]\right)
\end{aligned}$$

Unfortunately, there is no such conservation law associated with the electric field. The best one can do is to make the gradients of Z at the edge of the patches continuous. This should lead to continuous diffusivity and conductivity across the patches. A better approach would be to force the diffusivity to be constant and to compute Z from this, but this is not possible when the expression for the diffusivity is not invertible.

3.3.2 Growing a patch

Of course the regions of refinement can change during the course of the computation. This occurs for example in the case of the flow shear model, where the location of the jump can move around quite freely, and typically tends to grow inward from the edge. Also in the case of a very high confinement mode there are typically steep gradients that grow inwards, so the region of refinement also should grow.
To facilitate this, the possibility to grow patches also has to be included. This is quite straightforward. Whenever growth is necessary, new arrays to contain the data at all the new points of refinement are created. Then any values the patch has on the old region of refinement are copied. The rest of the points are computed through linear interpolation of the parent grid, just like when creating a new grid.

Growing parents

As mentioned earlier, in order to make sure the local Dirichlet boundary conditions can be resolved, the parent has to have points to the left and right of the patch edges. Thus, whenever a patch is grown, a check is necessary to see if the new edges of the patch coincide with those of the parent. If they do, the parent also has to be grown. Growing the parent can be done using the same approach as growing a patch. It is however necessary to add points in multiples of two, as the first and last point of each patch should coincide with a point of its parent grid.

Merging patches

When all patches are allowed to grow and their parents grow as well to accommodate for the boundary conditions, then it can happen that two patches grow against each other. This occurs for example if two patches grow to a distance of no more than $2\Delta x$ apart. In this case, it is sensible to merge the parent grids together, as this will prevent doing extra work. Merging in Bifurcator works a lot like growing: the left most patch

is grown to the size of the union of the two grids to merge. Then the information the patches already contained are copied to the new, larger patch. Finally, the children of the other patch are copied to the first patch, and the second patch is deleted.
Sometimes it might even be possible that three patches overlap in some regions. This situation can still be resolved by simply merging all patches into the first grid.

3.3.3 Refinement criteria

When starting out with a single grid, it is difficult to think of a good refinement criterion. A fairly intuitive criterion is to simply refine when the difference between two adjacent points is large. That way, a minimal vertical resolution is guaranteed. Furthermore, it seems straightforward that the errors due to spatial refinement are large in regions where the numerical solution is steep, and this criterion leads to refinement in precisely those areas. For these reasons, this criterion is also implemented in Bifurcator. Mathematically, it is the following condition

$$\text{if } \max_i \left(\max \left\{ \left| \Delta x \frac{\partial n}{\partial x} \right|_i, \left| \Delta x \frac{\partial T}{\partial x} \right|_i, \left| \Delta x \frac{\partial Z}{\partial x} \right|_i \right\} \right) > \varepsilon, \qquad \text{refine.}$$

Here, i is an index going over all the grid points currently available and ε is a pre-set tolerance. When a grid point should be refined, it is flagged. Because refining can be quite expensive, whenever refinement of a single grid point is necessary, all points in a buffer around that point are also flagged for refining. If a single point needs refinement, it is likely that its neighbours will need refining sooner or later, so by flagging them before the vertical resolution in those points is above tolerance, more points are refined in a single step. This results in only a single patch being created, and thus less overhead.
When all grid points are checked and some points should be refined, then for each of the regions up for refinement, either a new patch is created if there was none available, or an already existing patch (partially) on that domain is grown to match the entire domain.
When the variables can be of vastly different magnitudes, it might be better to scale the gradients with the variables:

$$\text{if } \max_i \left(\max \left\{ \left| \frac{\Delta x}{n_{\max}} \frac{\partial n}{\partial x} \right|_i, \left| \frac{\Delta x}{T_{\max}} \frac{\partial T}{\partial x} \right|_i, \left| \frac{\Delta x}{Z_{\max}} \frac{\partial Z}{\partial x} \right|_i \right\} \right) > \varepsilon, \qquad \text{refine.}$$

However, in the current studies, all variables where of similar magnitudes, so this rescaling was not necessary.

3.4 Summary

Based on the exploration of the various methods in the previous sections, in Bifurcator the methods of choice are DIRK methods for time integration and central differences for space discretisation. Combining the spatial discretisation with adaptive grid refinement and using adaptive time stepping via an embedded DIRK-method should result in a very robust and accurate solver, well capable of dealing with the bifurcations we expect to find in the models to study.

Chapter 4

Numerical Experiments

This chapter discussed all the numerical experiments, and their outcome. All numerical results were obtained using several different versions of Bifurcator. The default parameter settings are shown in appendix A, and for the sake of brevity only (relevant) differences from this file will be shown at the actual results. The results are all given in numerical units, which do not represent any physically tangible units. However, the units between simulations are consistent each other. If the units in the configuration file were in S.I. units, the results would be too. Since the main focus of this research was identifying the bifurcation landscape of the models, no attention was given to finding proper S.I. values for the parameters, however.

Also note that Bifurcator uses a coordinate system in which the separatrix is located at $x = 0$ while the plasma core is at $x \to +\infty$. For the quantities that have an inherent direction, this can result in negative values. More specifically, for the fluxes q_∞ and Γ it will result in negative values. These negative values simply imply particles and heat flowing from the hot, dense plasma core towards the plasma edge. Whenever the terms "increasing", "decreasing" or equivalent are used in combination with these fluxes, it is simply meant that their absolute value (i.e. magnitude) is increased or decreased.

4.1 Benchmarks

In order to verify the proper working of the program, some benchmarks were done. A very straightforward benchmark is to compute solutions to a known ODE or PDE and compare the obtained numerical result with the actual solution. This can be done for various grid sizes or time step sizes, and the error should reduce as the spatial/temporal resolution increases. Furthermore, the actual scaling of the error with the resolution gives the order of consistency of the spatial discretisation.

4.1.1 Spatial discretisation scheme

In chapter 2 the analytic solution to the system with $Z = 0$ is derived. It is plotted in figure 2.5. The benchmark consists of starting with an arbitrary initial profile, keeping $Z = 0$ and then letting time evolve until steady state is reached. If this is done with a significantly small time step (or with adaptive time stepping with very strict error tolerances) then the error in the spatial grid should dominate, not the error in the time integration. In order to verify the error scaling with the spatial grid, the above experiment is done for the grid sizes listed in table 4.1. The table also contains the observed absolute and relative errors in the thermal energy and density (denoted by $\varepsilon_{\text{abs}}(\cdot)$ and $\varepsilon_{\text{rel}}(\cdot)$). See also figure 4.1 for a visual representation of the data. From figure 4.1 it is clear that the error in the density is constant and the accuracy of the computed thermal energy scales with the number of grid points squared. The reason for the error in the density being constant is that a central difference scheme is used. Since the exact solution for the density is a straight line, it is reproduced exactly by the discretisation scheme (see 3.1 for an example).

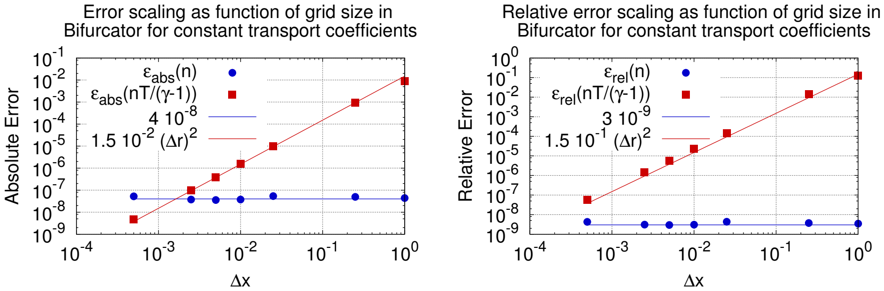
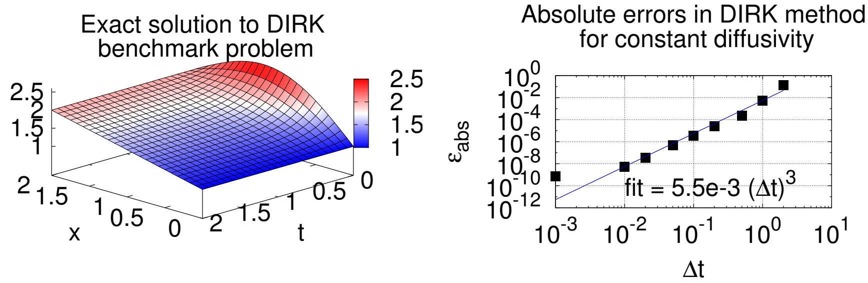


Figure 4.1: Scaling of the absolute error (left) and relative error (right) in the density n and the thermal energy $\frac{nT}{\gamma-1}$ with the grid size. The error in the density is constant, as the central difference scheme reproduces linear functions exactly, and thus the accuracy is capped by the printing precision. This is confirmed by the error in the thermal energy for the largest grid size. Clearly the accuracy with which the thermal energy is computed increases quadratically with the number of grid points.

Table 4.1: Grid sizes and obtained errors for spatial discretisation benchmark. The domain length R was 50. Note that the error in the density is constant. This is because the steady state profile for the density is linear, and central differences exactly reproduce polynomials of degree 2 and less. The accuracy of the obtained results are thus capped by the printing precision, which is in the order of 10^{-8}.

Δx	$\varepsilon_{\text{abs}}(n)$	$\varepsilon_{\text{abs}}\left(\frac{nT}{\gamma-1}\right)$	$\varepsilon_{\text{rel}}(n)$	$\varepsilon_{\text{rel}}\left(\frac{nT}{\gamma-1}\right)$
$5.0\ 10^{-4}$	$5.2381\ 10^{-8}$	$4.7238\ 10^{-9}$	$4.2715\ 10^{-9}$	$5.7020\ 10^{-8}$
$5.0\ 10^{-3}$	$3.5714\ 10^{-8}$	$3.8657\ 10^{-7}$	$2.9582\ 10^{-9}$	$5.7072\ 10^{-6}$
$2.5\ 10^{-3}$	$3.7619\ 10^{-8}$	$9.6592\ 10^{-8}$	$3.0956\ 10^{-9}$	$1.4261\ 10^{-6}$
$1.0\ 10^{-2}$	$3.7619\ 10^{-8}$	$1.5467\ 10^{-6}$	$3.0666\ 10^{-9}$	$2.2835\ 10^{-5}$
$2.5\ 10^{-2}$	$5.3810\ 10^{-8}$	$9.6716\ 10^{-6}$	$4.3522\ 10^{-9}$	$1.4279\ 10^{-4}$
$2.5\ 10^{-1}$	$4.9524\ 10^{-8}$	$9.5801\ 10^{-4}$	$3.7379\ 10^{-9}$	$1.4144\ 10^{-2}$
$1.0\ 10^{-0}$	$4.3810\ 10^{-8}$	$8.6545\ 10^{-3}$	$3.4671\ 10^{-9}$	$1.2777\ 10^{-1}$



(a) The analytic solution to the problem against which the DIRK is benchmarked.

(b) Errors of the DIRK method at various time step sizes. Linear regression shows that initial scaling is with $(\Delta t)^3$.

4.1.2 Time integration scheme

As mentioned in the chapter about the numerical methods used, Bifurcator uses a diagonally implicit Runge-Kutta (DIRK) method of order 3, with an embedded DIRK method of order 2. Their stability properties are already discussed in section 3.2.6. Their accuracy, however, is not. As a second benchmark, the following problem is solved numerically:

$$\begin{aligned}\frac{\partial n}{\partial t} = D\frac{\partial^2 n}{\partial x^2}, \qquad & n(x,0) = 1 + x/R + \sin\left(\frac{\pi}{R}x\right) \\ 0 \leq x \leq R, \qquad & n(0,t) = 1, \qquad n(R,t) = 2\end{aligned}$$

Its analytic solution is given by

$$n(x,t) = 1 + x/R + \sin\left(\frac{\pi}{R}x\right) e^{-\frac{D\pi^2}{R^2}t} \tag{4.1}$$

The reason for choosing this test problem is because it can be implemented into Bifurcator quite easily by fixing the temperature and electric field profiles. The solution is plotted in figure 4.2a for the value of the benchmark, which had $D \approx 1.62$. For this value, $D\pi^2/R^2 = 4$. To confirm that the DIRK is indeed third order consistent, the system of equations is solved using Bifurcator, keeping T and Z fixed. The grid chosen for this benchmark is exceptionally large (10000 grid points). The reason for this is to make absolutely sure that the error observed is due to the DIRK method, rather than the spatial discretisation. The observed errors, as well as a linear regression are shown in figure 4.2b. These confirm that the DIRK method is indeed third order consistent.

4.2 Zohm's model

From theory it is known that Zohm's model should contain at least hysteresis, sharp transitions and dithering [8, 9]. This can be confirmed using Bifurcator. Also, there is reason to believe that hysteresis can vanish for certain boundary conditions. To test this, the hysteresis region was located for a boundary condition $\partial Z/\partial x = Z/\lambda_Z$ for various λ_Z.

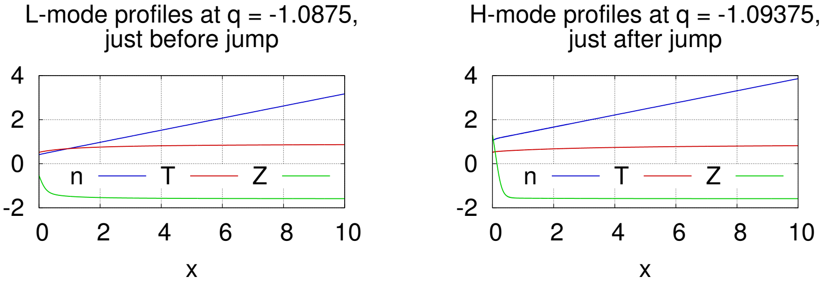


Figure 4.2: Sharp jumps in the electric field and density profiles occur when increasing or decreasing the heat flux very slightly. This is a typical characteristic of the L-H transition observed in experiments. The heat fluxes are shown in the figures. For this result $b = -2.5$. The critical heat fluxes were found using the parameter scan functionality of Bifurcator.

4.2.1 Sharp jumps and hysteresis

To show sharp jumps the heat flux is ramped up to the L-H-transition value. This is shown in figure 4.2. The sharp jump in Z is clearly visible. Note that the increase in heat flux is only 0.00625, which is about 0.5%, making the resulting change of Z quite sharp indeed. From theory it is expected that these sharp transitions occur at the Hopf bifurcation [9]. However, for this result, the transition is much closer to the fold bifurcation instead. Possible causes for this discrepancy will be discussed in section 4.4.

Hysteresis was also observed in Zohm's model in the same parameter region. The results are shown in figure 4.3. In the figure, six graphs are plotted. The left three follow the path of increasing heat flux, starting at $q_\infty = -0.02$ up to $q_\infty = -1.32$. The right three graphs follow the path of decreasing heat flux instead, starting at $q_\infty = -1.32$, as indicated by the arrows. Clearly, in the intermediate region around $q_\infty = -0.32$, either an L-mode profile is obtained when increasing the heat flux or an H-mode profile is obtained when decreasing the heat flux. This is clearly hysteresis, as the obtained result depends not only on the current parameter settings, but also on the history of the solution.

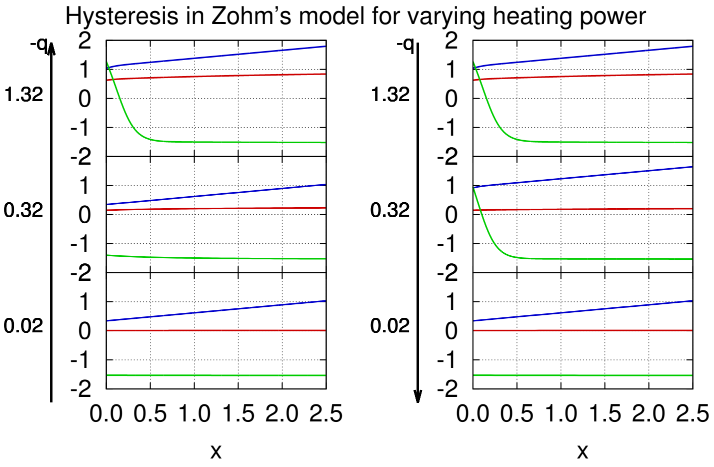


Figure 4.3: Hysteresis in Zohm's model. The left column contains three profiles obtained when increasing the heat flux from $q_\infty = -0.02$ to $q_\infty = -1.32$, in which the profile at $q_\infty = -0.32$ is an L-mode profile. In the right column, profiles at the same three heat fluxes are shown, but this time for decreasing the heat flux from $q_\infty = -1.32$ to $q_\infty = -0.02$. In this case, the profile at $q_\infty = -0.32$ is clearly an H-mode profile. This illustrates hysteresis in Zohm's model: the profile obtained at $q_\infty = -0.32$ depends on which heat flux the profiles comes from. For these results, $b = -1.9$. The colour coding is the same as in figure 4.2.

4.2.2 Dithering

In Zohm's model, dithering is observed. Typical for dithering in Zohm's model is the oscillations in the particle and heat fluxes at the edge. These are shown in figure 4.4. By drawing a phase plot of such solutions, it can be verified that the oscillation is really a limit cycle (i.e. the oscillatory solution is really the "final" solution). This is done in figure 4.5. The limit cycle shows the time evolution of the electric field in $x = 0$ against the density in $x = 0$. Note that the crossing of lines in this plot is purely because there is a third variable in T that is invisible, because the lines are projected on the Zn-plane. Two interesting things happen in figure 4.5. The first interesting thing is that all the profiles starting with a high Z "snap" together before converging to the limit cycle. Furthermore, they all join together with the actual limit cycle at approximately the same point. Taking a closer look at the profiles in this point reveals that there the profiles are pretty alike. They are shown in figure 4.6a. In figure 4.6b two profiles for different Z_{init} are shown after having travelled half a limit cycle. After this time, the profiles are already almost identical. Secondly, there are crossing limit cycles. This is not possible in the 2-dimensional ODE model introduced in section 2.1.2. In this case, however, there are three variables to consider, and it turns out that the crossing of the limit cycles is just an artifact caused by the projection of the 3D limit cycle (of $n(x = 0)$, $T(x = 0)$ and $Z(x = 0)$) onto the n,Z-plane. The profiles on the crossing part do not match the ones found later on the limit cycle in that point. This is shown in figures 4.7 and 4.8, where the limit cycles in a three-dimensional n,T,Z-space are drawn with a black dot indicating the point on the limit cycle the profiles on the right correspond to. The results are for $Z_{init} = 0.5$, and clearly the two profiles are not the same, even though they appear to belong to the same point on the limit cycle. The extra dimension in

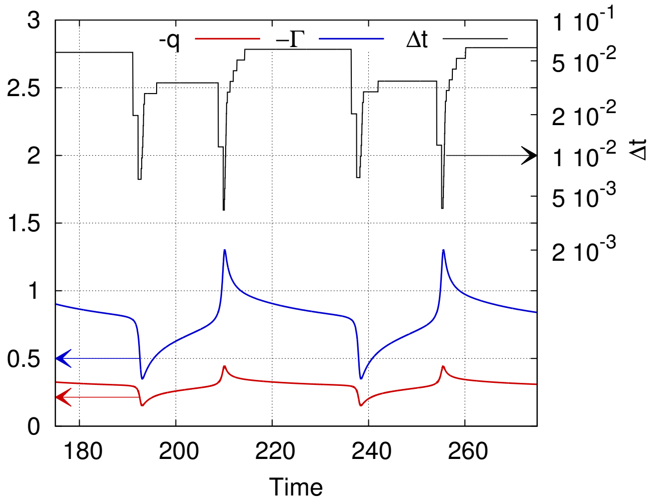


Figure 4.4: Dithering in Zohm's model, as seen in the time trace of the particle and heat flux. Also plotted is the dynamic behaviour of the time step size (black), which clearly adapts to the fast changes in q (red) and Γ (blue) when jumping between L and H-modes by decreasing, and to the transport time scales in L and H-mode by increasing. The fluxes appear to be perfectly reproduced each cycle.

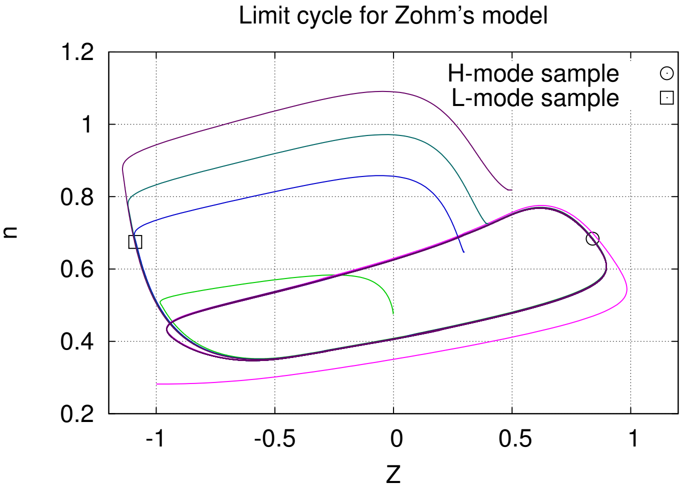
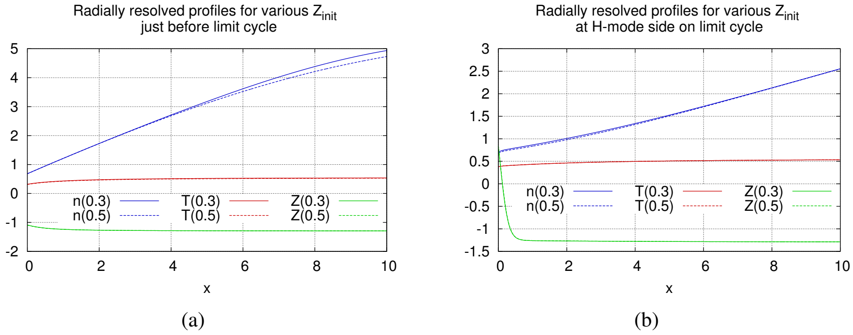


Figure 4.5: Limit cycle of dithering in Zohm's model of the electric field and density edge values. Various seed profiles are used, each with $Z(t=0,x) = const$.

T makes it possible for limit cycles to cross when projected on a plane. The relevant parameters for the results were $q_\infty = -0.3$, $D0 = 0.2$, $D1 = 4.0$, $b = -0.9$, $\varepsilon = 0.3$ and $\gamma = 2.5$. There were simulations for various Z_{init}, namely at 0, -1, 0.3, 0.4 and 0.5.



(a)

(b)

Figure 4.6: Different profiles belonging to different seed profiles (i.e. different Z_{init}, value indicated between parenthesis), sampled in points relatively close on the limit cycle shown in figure 4.5 (black square and circle). The results show that there is only a small quantitative difference in the densities visible. This difference comes from the fact that at the start of the simulation, there is plenty of excess particles in the plasma. These particles are ejected through a slow diffusive process. Only after a few "rounds" in the limit cycle are the profiles expected to be equal.

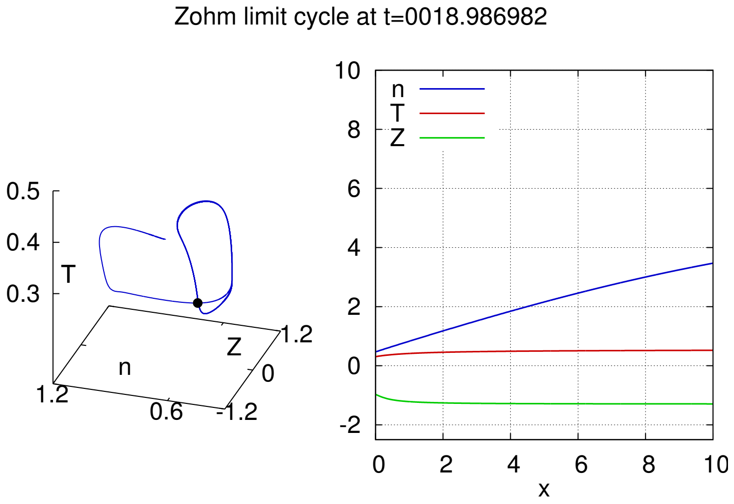


Figure 4.7: The limit cycle for dithering in Zohm's model shown in the n, T, Z-space, with a non-steady state profile in the "crossing" shown on the right.

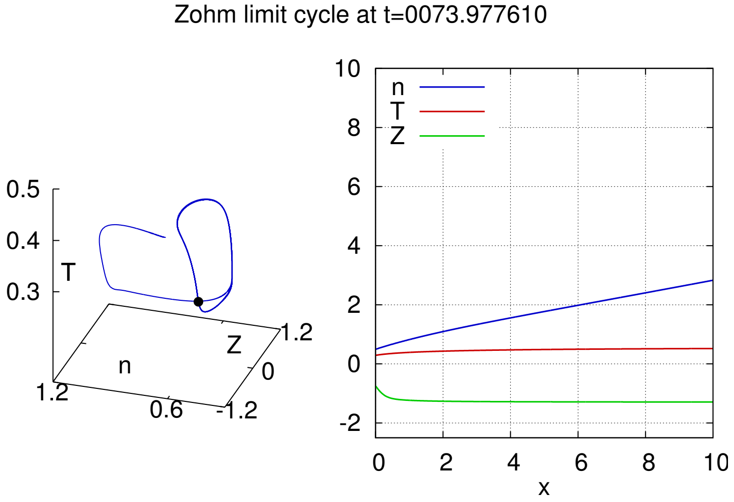


Figure 4.8: The limit cycle for dithering in Zohm's model shown in the n, T, Z-space, with a more relaxed profile in the "crossing" shown on the right.

4.2.3 Effect of different boundary conditions

Some time was taken to study the effect of the boundary conditions on the bifurcations in the system for Zohm's model. Studies with Auto, a program that can numerically find bifurcations in ODE models, showed that for the boundary condition $\partial Z/\partial x = Z/\lambda_Z$ hysteresis should disappear for sufficiently large values of λ_Z [14].
Some preliminary testing was done with Bifurcator, to observe if this is the case and also to try if changing a different parameter can "enable" hysteresis again. Unfortunately, this study is inconclusive, but the results will be listed here nonetheless. This study was especially useful to see how well Bifurcator can narrow down regions where bifurcations are. The results are shown in table 4.2.
The results show that changing the parameter λ_Z from a small number (which is close to the boundary condition $Z(x=0)=0$) to a higher number decreases the width of the region in which there is hysteresis. To determine if the region really disappeared or simply moved in phase space, the parameter b was then decreased in an attempt to relocate the hysteresis region. It was found that the back transition occured at $-q < 0.1$, but an exact value was not recovered. The fact that the forward transition was obtained at or above 1 implies that for $-2.5 < b < -1.5$ there should again be hysteresis regions: they simply moved in phase space.

Table 4.2: Table showing the heat fluxes at which the transition from L-mode to H-mode and back occurs for various b and λ_Z. All other parameters were kept constant and are listed in appendix A. This table shows that changing only λ_Z influences the width of the hysteresis region, and thus that the boundary conditions influence qualitatively the bifurcation landscape of the system.

b	λ_Z	$-q_{L\rightarrow H}$	$-q_{H\rightarrow L}$
-1.5	0.001	1.847 ± 0.003	1.465 ± 0.004
-1.5	1.0	1.848 ± 0.004	1.840 ± 0.004
-2.5	1.0	1.053 ± 0.003	< 0.1

4.3 Flow shear model

In this section, the results of the flow shear model will be discussed. Since it couples reduction of transport to ∇Z (which is equivalent to $\nabla v_{\mathbf{E}\times\mathbf{B}}$) this is a more realistic model than Zohm's model. If it turns out that the model still contains all relevant bifurcations, then this gives hope that more complex models will also preserve the bifurcation structure.

4.3.1 Sharp transitions and hysteresis

The flow shear model also exhibits sharp transitions and hysteresis. Using Bifurcator, the heat fluxes for the transitions were found to be $q_{L\to H} = -1.881 \pm 0.002$ and $q_{H\to L} = -1.377 \pm 0.002$ for $b = -2.8$. The sharp jump is illustrated in figure 4.9. The transition is indeed sharp, as the electric field at the edge jumps from $Z < 1$ to $Z \approx 3$. Hysteresis was also observed, and is illustrated in figure 4.10.

4.3.2 Dithering

Dithering can also be observed in the flow shear model. In figure 4.11 a limit cycle of the flow shear dithering behaviour is shown. Again the limit cycle is projected onto the n, Z-plane. The used parameters that differed from the default configuration were $b = -0.8$, $q_\infty = -2.0$ and $a_1 = 1$. Note that for this limit cycle, the edges are a lot straighter than for Zohm's model, which is explained by the reduced value for ε. After

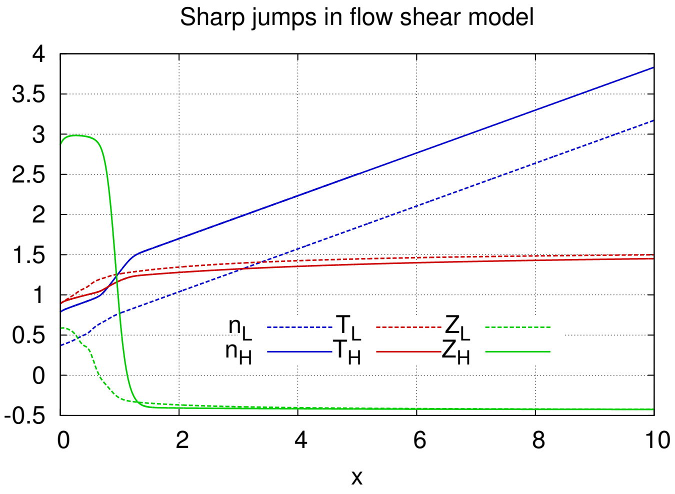


Figure 4.9: Sharp jumps in the flow shear model. The dashed lines represent the L-mode at $q_\infty = -1.875$ and the solid lines represent the H-mode at $q_\infty = -1.882$. For these results, $b = -2.8$. The density at $x = 2.5$ is 20% more in H-mode, and a density pedestal is clearly visible. One artefact of the model is that the temperature in L-mode is actually slightly higher than the temperature in H-mode. This is however not really problematic as the thermal energy $nT/(\gamma - 1)$ is clearly higher in H-mode due to the increased density.

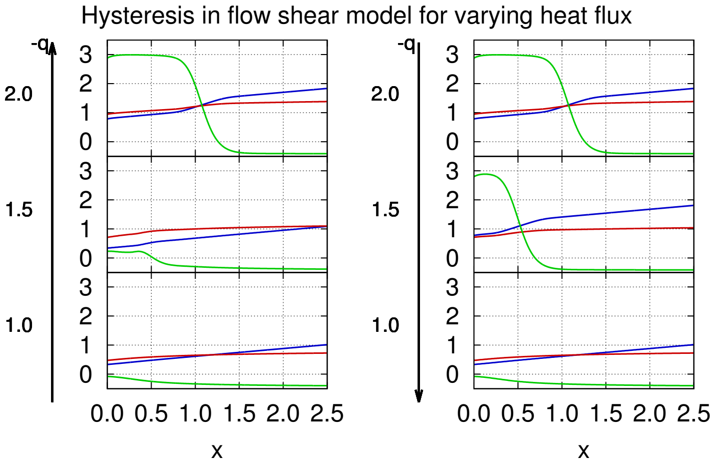


Figure 4.10: Hysteresis in the flow shear model for $b = -2.8$. On the left is the path from L-mode to H-mode, which in the intermediate region $q_\infty = -1.5$ is still in L-mode. At $q_\infty = -2$ the profiles are in H-mode by virtue of the transition. When the path is taken from H-mode to L-mode, the profiles in the intermediate region are now still in H-mode, similarly to the case for Zohm's model (figure 4.3). The colour coding is consistent with figure 4.9.

all, by reducing ε the time scale at which Z evolves is increased. This means in fuzzy terms that Z can change more before the density realises and changes to accommodate for the new profile for Z.

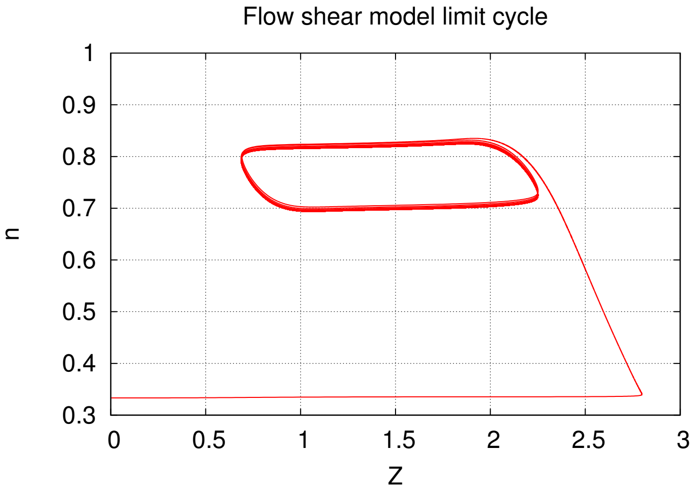


Figure 4.11: A limit cycle of a dithering solution for the flow shear model. The initial seed profile at $Z_{\text{init}} = 0$ jumps to the high Z-side at the time scale $1/\varepsilon$, at which the density increases with a transport time scale. The increase in density causes the value for Z to move down slightly towards the L-mode side until it hits the limit cycle at $Z \approx 2,\ n \approx 0.8$.

4.4 Discrepancies between theoretical predictions and numerical results

Using Bifurcator, the location of bifurcations in parameter space have been located, both for Zohm's model and the flow shear model. The results are shown in figure 4.12. These figures should be read the same way as the ones in figure 2.6 in chapter 2: The regions to the left in between the orange and red branch is the hysteresis region, above the red line and green dots is the H-mode regime, and below the orange line and green dots is the L-mode regime. Finally the green dots indicate a dithering region. There are some discrepancies between the theoretically and numerically obtained bifurcation landscapes. The biggest difference for Zohm's model is that the numerically obtained hysteresis region seems to lie in between the fold lines, rather than inside the Hopf lines. Since the theoretical prediction is based on various approximatinos, this implies that there is some added effect of the radially resolved profiles the theoretical predictions do not take into account. After all, the theoretical predictions are based on $\mu \to 0$, and solely on the edge values (boundary conditions at the plasma edge are substituted into the evolution equations).

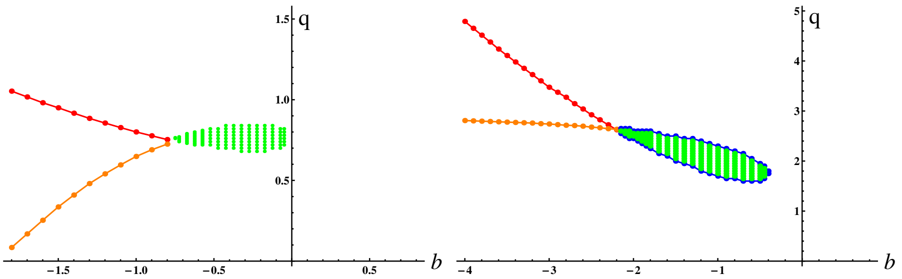


Figure 4.12: Numerically obtained bifurcation landscapes. On the left is for Zohm's model, on the right for the flow shear model. Note the qualitative agreement with each other and with the theoretical prediction in figure 2.6.

For the flow shear model it is even worse: the location of the various bifurcation regions seem a complete mismatch. However, qualitatively the bifurcation characteristics are the same for all models. They all show the correct bifuractions (sharp jumps, hysteresis, dithering), and they do this with the same ordering (hysteresis to the left of dithering, etc.). Therefore, the only difference is in the location of these regions in parameter space, implying that the underlying bifurcation structure is very robust.

Chapter 5

Discussion and conclusion

Since this research basically had two sides, namely the investigation of suitable numerical techniques for dealing with bifurcations as well as determining which bifurcations are contained within the models, the results of these two parts will be discussed separately.

5.1 Numerical methods

Based on the obtained numerical results, it seems that the numerical methods eventually implemented in Bifurcator are adequate for dealing with the encountered bifurcations, which were Hopf and fold bifurcations. Each of the numerical methods chosen will be discussed in more detail in the subsections below.

5.1.1 Time integration methods

The time integration method ultimately implemented was a so-called DIRK, a diagonally implicit Runge-Kutta method. The main reasons for choosing this method over the implicit Euler and BDF methods is because it combines unconditional stability, a fairly high accuracy order and the ability to use embedding for cheap error estimations and time step size control. The downside is that the computation of new time levels can be fairly expensive when compared with e.g. backward differentiating formula (BDF) methods of the same order. However, when studying bifurcations one can expect long periods of little activity and short periods of high activity, and therefore this downside is far outweighed by the benefit of having adaptive time step control combined with the lack of a stability constraint on the time step size. Even in the case of non-linear equations, where the BDF method only has to perform Newton iteration once while an s-stage DIRK methods has to do so s times each time step, cheap and effective time step size control still outweighs the added disadvantage of more computational work. Additionally, the extra work can be reduced, because adaptive time step control can be used to try to keep the number of Newton iterations required relatively low, so that while the DIRK method has to perform Newton iterations s times each time step, each iteration individually only takes a few iterates. In my opinion, the added stability and time step size control is so useful that DIRK methods should be considered for all applications where stability is important and the different time scales are far apart (e.g.

stiff ODEs). In conclusion:

Diagonally implicit Runge-Kutta methods are especially suitable for solving systems of stiff equations that contain periods of high activity and periods of low activity.

5.1.2 Spatial discretisation methods

For spatial discretisation second order central differences were used. It is especially important that the numerical scheme obeys physical conservation laws whenever possible (as illustrated in section 3.1.2), unless there is a very good reason not to (e.g. a necessary performance boost). Therefore, rewriting the system of equations in conservative form whenever possible via a help variable (in this research $U := \frac{nT}{\gamma-1}$) is highly advantageous.

5.1.3 Adaptive grid refinement

While adaptive grid refinement was implemented for the spatial discretisation, there was not enough time available to do a thorough analysis of the accuracy of this refinement, and about the added benefit. Furthermore, it is not even really clear what good criteria are for the flow of information between the grids (i.e., which boundary conditions for the patches should be used for using the information of the parent grid). In the latest versions of Bifurcator the heat and particle flux are kept constant along the grids, which resulted in fairly good results, but these might not have been the optimal choice. Furthermore, it is not clear at all what a good condition would be for the electric field, or for any other variables that do not correspond to a conservation law for that matter. Running Bifurcator through a profiling benchmark showed that fairly little time was spent on refining the grid and on the communication between the grids, so there should be little overhead by the refining. The slowdown associated with refining is simply because there are more equations to solve, resulting in slower solving of the systems of equations. Since fine grids are required to accurately position the bifurcation points, adaptive grid refinement provides a useful way to accurately compute these positions without doing a lot of unnecessary work in regions that do not need finer grids.

Adaptive grid refinement is a useful tool to accurately resolve bifurcations in PDE models, as there does not appear to be much slow down associated with the extra communication between grids, and fine grids are necessary to accurately locate bifurcation points.

However, proper boundary conditions for transferring the information between patches are not trivial to derive if there is no conservation law associated with the variable(s). Extra study might reveal local boundary conditions that make sense regardless of the variable considered, but these were not found during this study.

5.2 L-H transition models

Using Bifurcator, numerical results exhibiting bifurcations were obtained. These results, their implications for the validity of the studied models and the influences of boundary conditions in these models on the bifurcations will be discussed here.

5.2.1 Zohm's model

Based on the numerical results obtained with Bifurcator, it seems that the bifurcations qualitatively describing the transition from L-mode to H-mode are indeed present in this model, which is consistent with the observations by Zohm [8]. For the right parameters, it is possible to generate L-mode and H-mode solutions, which are separated by regions of hysteresis. Furthermore, for other parameter regions dithering is obtained.
The location of the transition points, i.e. the value of the heating power for which the profiles change from L-mode to H-mode, differ from their theoretical predictions. Numerically, the "jump" points seem to lie nearer to the fold bifurcation curve than to the Hopf curve. The reason for this is that while theoretically the steady state solution (e.g. the L-mode profile) should become unstable, in the numerical case this might not be the case due to the added damping introduced by the parameters ε and μ. Instead, the solution remains in L-mode until this solution ceases to be a steady state solution entirely, which happens when crossing the fold curve. The discrepancy is visible in this case because in numerical studies the values of the parameters ε and μ always have to be finite, while the theoretical predictions are in the limit $\varepsilon, \mu \to 0$. Some analysis of the scaling of this discrepancy with the values of ε and μ was done, and while inconclusive, they did show a very slow convergence to the theoretically predicted value for smaller values of μ and ε. Since the computation time increases heavily when decreasing μ and ε, this scaling becomes hard to check at some point.
Finally, qualitatively the bifurcation curves look similar to the theoretical ones. This is not quite trivial, as the theoretical analysis is actually for a system of ODEs instead of PDEs. It turns out that in this case, the situation from ODEs generalises quite well to PDEs. Reason for this is probably the fact that all equations are damped. When considering hyperbolic equations the results might be vastly different.

> Zohm's model does indeed contain all bifurcations relevant for the L-H transition in fusion experiments. Furthermore, the shape of the parameter space (i.e. the location of the various solution regions) qualitatively matches that of the theoretical analysis. This confirms the theoretical predictions. Furthermore, this shows that numerical analysis is a feasible alternative to theoretical analysis for models that are too complex for analytic study.

5.2.2 Flow shear model

Similar to Zohm's model, for the flow shear model numerical results showing L-modes, H-modes, hysteresis and dithering have also been observed. For this model, it also turned out that the theoretical predictions (as far as they were possible) qualitatively match the numerical findings but not quantitatively. Still, it is very reassuring that even when using a more complex but realistic expression for the diffusion, the bifurcations of the L-H transition are still intact.

> The flow shear model exhibits all bifurcations corresponding to the L-H transition in fusion experiments. Also, the solution regions in phase space are qualitatively similar to the one obtained for Zohm's model.

The fact that the bifurcation characteristics obtained are similar to the ones in Zohm's model implies that the transport coefficients might be generalised even further without qualitative consequences for the bifurcations, which would make bifurcation analysis still suitable for more complex and realistic models. For such models, analytic bifurcation analysis will probably be hard or impossible, but numerical studies with Bifurcator can reveal the bifurcation landscape.

5.3 Bifurcations in PDE models in general

Based on the research done in the study of bifurcation models for the L-H transition, some results were observed that suggest implications for bifurcations in PDE models in general. Most notably, during the analysis of the limit cycle in Zohm's model it turned out that some lines in phase space did not overlap but were very close to each other (see figures 4.5 and 4.6a). This seems to generalise the way bifurcations in ODE models are. There, similar points in phase space imply similar values for the variables considered. In a PDE model however, this generalises from values of variables to profiles of variables, due to the fact that they are radially resolved. However, it still seems to hold true that similar points in phase space imply similar profiles, which is not quite trivial. This might indicate that bifurcation predictions based on a non-radially resolved model might generalise quite well to a PDE model, allowing at least some theoretical predictions.

Based on the results observed in this research, it seems that predictions about bifurcation behaviour based on ODE models generalise quite well to similar but inherently spatially resolved PDE models.

Note that this research was only for a quite specific model, and that much more research is needed to verify the above claim. In particular, it might be that this generalisation holds only for parabolic equations, as their damping properties make their solutions relatively well-behaved. For hyperbolic equations, this could be vastly different.

Chapter 6

Outlook

With Bifurcator, a useful numerical tool for the study of bifurcations in 1D parabolic models has been delivered. Applying it to Zohm's model and the flow shear model has shown that it is well capable of resolving numerically the bifurcations of interest. The next logical step would be to apply Bifurcator to other proposed models in the literature. Parameter scans can then reveal if these models also predict correctly the bifurcations observed in experiments. Also an important path of new research is the implementation of more flow shear candidates. Examples include the coupling of zonal flows to local turbulence levels for example. These can be implemented straightforwardly by adding another evolution equation, and adding proper coupling to the other variables. Parameter scans then could reveal under which circumstances which of the different candidates is most responsible for the reduction in transport. Finally, when the transition to first-principles models is made, it is also worthwile to determine how to convert experimental parameters into the theoretical parameters in the model. This will also enable a more quantitative comparison between models and possibly with experiments as well.

Finally, another interesting, not directly fusion related, point of research is simply bifurcations in 1D transport models themselves. It might be possible to find numerically other bifurcations in either Zohm's model or the flow shear model (or any new models that might be studied), that have natural analogues in ODE models. This is in itself a useful study as it can give more insight in how bifurcations generalise from ODEs to PDEs.

Chapter 7

Acknowledgements

First of all I would like to thank my daily supervisors Wolf and Hugo. Hugo's detailed and enthousiastic explanations on bifurcations were very inspiring, and Wolf's patient explanations on the theoretical study of bifurcations made sure I always knew what I was doing. Furthermore, I'd like to thank Wolf for being such a reliable beta-tester for Bifurcator. A word of thanks is also in order for Jan ten Thije Boonkkamp for suggesting the DIRK methods, as well as for computer science guru Tom Boshoven for suggesting the use of eigen [15], the coolest C++-library for linear algebra, and for helping me out with various random questions on programming in C++.
This work, supported by the European Communities under the contract of Association between EURATOM/FOM, was carried out within the framework of the European Fusion Program. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Bibliography

[1] F. Wagner, G. Becker, K. Behringer, D. Campbell, A. Eberhagen, W. Engelhardt, G. Fussmann, O. Gehre, J. Gernhardt, G. Gierke, *et al.*, “Regime of improved confinement and high beta in neutral-beam-heated divertor discharges of the ASDEX Tokamak,” *Physical Review Letters*, vol. 49, no. 19, pp. 1408–1412, 1982.

[2] F. Wagner, “A quarter-century of H-mode studies,” *Plasma Physics and Controlled Fusion*, vol. 49, no. 12B, p. B1, 2007.

[3] P. Sauter, T. Pütterich, F. Ryter, E. Viezzer, E. Wolfrum, G. Conway, R. Fischer, B. Kurzan, R. McDermott, S. Rathgeber, and the ASDEX Upgrade Team, “L- to H-mode transitions at low density in ASDEX Upgrade,” *Nuclear Fusion*, vol. 52, no. 1, p. 012001, 2012.

[4] R. Groebner, T. Carlstrom, K. Burrell, S. Coda, E. Doyle, P. Gohil, K. Kim, Q. Peng, R. Maingi, R. Moyer, C. Rettig, T. Rhodes, D. Schissel, G. Staebler, R. Stambaugh, D. Thomas, and J. Watkins, “Study of H–Mode threshold conditions in DIII–D ,”

[5] K. Burrell, “Effects of E x B velocity shear and magnetic shear on turbulence and transport in magnetic confinement devices,” *Physics of Plasmas*, vol. 4, no. 5, pp. 1499–1518, 1997.

[6] M. A. Malkov and P. H. Diamond, “Weak hysteresis in a simplified model of the L-H transition,” *Physics of Plasmas*, vol. 16, no. 1, p. 012504, 2009.

[7] S. Paquay, *Bifurcator user manual*.

[8] H. Zohm, ASDEX-Upgrade Team, and NI and ICRH Group, “Dynamic behaviour of the L-H transition,”

[9] W. Weymiens, H. J. de Blank, G. M. D. Hogeweij, and J. C. de Valenca, “Bifurcation theory for the L-H transition in magnetically confined fusion plasmas,” *Physics of Plasmas*, vol. 19, no. 7, p. 072309, 2012.

[10] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic problems*. Springer Series in Computational Mathematics, Springer-Verlag, second ed., 2002.

[11] R. Mattheij and J. Molenaar, *Ordinary Differential Equations in Theory and Practice*.

[12] C. F. Curtiss and J. O. Hirschfelder, “Integration of Stiff Equations.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 38, pp. 235–243, Mar. 1952.

[13] P. van der Houwen and B. Sommeijer, “Diagonally implicit runge-kutta methods for 3d shallow water applications,” *Advances in Computational Mathematics*, vol. 12, pp. 229–250, 2000. 10.1023/A:1018969203026.

[14] D. Zhelyazhov, “Private communications.”

[15] G. Guennebaud, B. Jacob, *et al.*, “Eigen v3.” `http://eigen.tuxfamily.org`, 2010.

Appendix A

Default configuration file

The default configuration file for Zohm's model is listed below. There is also a default flow shear configuration file, which has the following key differences: Z_Shift = 1.4, diffusion_model = 1 and boundary_condition_electric_field = 0.

```
# This is a Bifurcator configuration
# file generated by Bifurcator version 1.1.2a.
# Some keywords might have a comment behind them
# that says "Do not change this!"
# This means that there are some values for which
# this keyword results in incorrect
# behaviour. You can of course change them if you
# really want to,  but that will probably result
# in garbage results. You have been warned.

################################
##### Physical parameters  #####
################################
diffusion_model = 0
core_particle_flux = -0.8
core_heat_flux = -0.2

D0 = 0.8
D1 = 3.0

a1 = 0.0
a2 = 0.0
a3 = 1.0

a = -1.0
b = -1.5

zeta = 1.1

mu = 0.05
epsilon = 0.05
```

```
ZShift = 0

gamma = 1.66666666

labdan = 1.25
labdaT = 1.5
labdaZ = 1.5

boundary_condition_electric_field = 3

cn = 1.1
cT = 0.9


ZInitial = 0.0

###########################################
##### Parameters for time integration #####
###########################################
tmax = 1e4
tstart = 0.00

adaptive_time_step = 1

dltt = 0.05
dltt_maximum_growth_factor = 1.05
time_derivative_stop_threshold = 1e-10

time_integration_type = 4 # <-- Do not change this!
theta = 1.0

maxit = 5
time_step_error_tolerance = 1e-3
newton_tolerance = 2e-4
always_keep_going = 0

initial_profile_types = 0

update_temperature_via_energy = 1 # <-- Do not change this!

no_smart_scan = 0
smart_scan_tolerance = 1e-2
Hmode_determining_threshold = 0.1

#####################################################
##### Parameters for the spatial discretisation #####
#####################################################
solver_type = 1

R = 50
```

```
initial_grid_size = 500
maximum_grid_rank = 40

adaptive_grid_refinement = 1
adaptive_grid_tolerance = 1e-2

#################################
##### Parameters for output #####
#################################
dirk_be_silent = 1
debug_output = 0

write_to_file = 1
file_write_interval = 1
file_write_interval_in_time_units = 50

time_traces_to_print = 0
time_traces_to_plot = 0

output_directory_name = bifurcator_output
output_file_name = final_profiles

plot_profiles_when_done = 1
plot_fluxes_when_done = 1
```