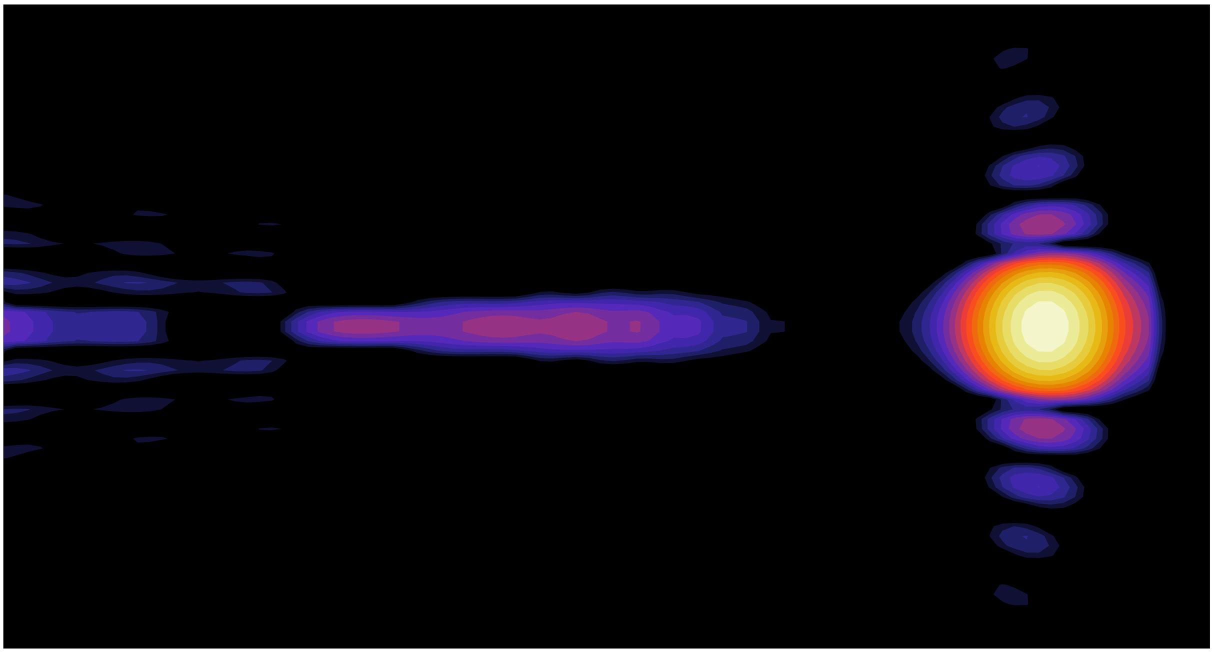
CHALMERS
UNIVERSITY OF TECHNOLOGY



Numerical Modelling of Runaway Electrons in Suddenly Cooling Fusion Plasmas

Bachelor's thesis

OSKAR VALLHAGEN
LUCAS UNNERFELT
JOHANNES HANSSON
ALBERT JOHANSSON
ERIK HERMANSSON
OSKAR ERIKSSON

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2019

Bachelor's thesis 2019

Numerical Modelling of Runaway Electrons in Suddenly Cooling Fusion Plasmas

OSKAR VALLHAGEN

ALBERT JOHANSSON

LUCAS UNNERFELT

ERIK HERMANSSON

JOHANNES HANSSON

OSKAR ERIKSSON

Department of Physics
Division of Subatomic and Plasma Physics
Plasma Theory
Chalmers University of Technology
Gothenburg, Sweden 2019

Numerical Modelling of Runaway Electrons
in Suddenly Cooling Fusion Plasmas
OSKAR VALLHAGEN
ALBERT JOHANSSON
LUCAS UNNERFELT
ERIK HERMANSSON
JOHANNES HANSSON
OSKAR ERIKSSON

Supervisors:
Tünde Fülöp, Department of Physics
Linnea Hesslow, Department of Physics
Co-supervisor:
Klara Insulander Björk, Department of Physics
Examiner:
Jan Swenson, Department of Physics

Bachelor's Thesis TIFX04-19-05
Department of Physics
Division of Subatomic and Plasma Physics
Plasma Theory
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Momentum-space distribution of runaway electrons calculated self-consistently with the evolution of the electric field in a tokamak disruption.

Typeset in LaTeX
Gothenburg, Sweden 2019

Numerical Modelling of Runaway Electrons
in Suddenly Cooling Fusion Plasmas

OSKAR VALLHAGEN
ALBERT JOHANSSON
LUCAS UNNERFELT
ERIK HERMANSSON
JOHANNES HANSSON
OSKAR ERIKSSON
Department of Physics
Chalmers University of Technology

Abstract

A serious problem facing fusion devices with large currents are plasma-terminating disruptions. These events sometimes result in the acceleration of electrons to very high energies, so-called runaway electrons. The harmful effects of disruptions and associated runaway electrons can be mitigated by material injections of argon or neon, often referred to as impurities. This work is dedicated to the modelling of runaway electrons in disruptions.
Firstly, we investigate how runaway dynamics is affected by the presence of impurities. To correctly capture the interaction of fast electrons and impurities, we improve runaway generation formulae in a numerical model that calculates the runaway electron current and the corresponding electric field evolution. The results of the simulations with the improved model show that material injections drastically affect the calculated runaway generation in a way that was not captured with the previous model. With pure argon gas injections, the net effect is generally an increase in runaway generation.

Secondly, we couple the model with a solver for the electron momentum distribution, which replaces the runaway generation formulae. This combined tool allows studying the spatiotemporal evolution of the runaway electron energy spectrum. It can also be used to validate the original model. We find that the original model gives too low runaway currents in the case of fast temperature decay, but the discrepancy is lower for slower temperature decays. We also address the numerical efficiency in the combined tool and show that the simulation time can be reduced by an order of magnitude without compromising accuracy.

Keywords: Runaway electron, tokamak, fusion, plasma physics, kinetic modelling.

Sammandrag

Ett allvarligt problem som kan uppstå i fusionsreaktorer med höga strömmar är disruptioner. Dessa kan bilda högenergetiska elektroner, så kallade skenande elektroner. Genom att injicera orenheter i plasmat, i form av exempelvis argon eller neon, kan skadliga effekter av de skenande elektronerna samt disruptionerna förhindras. I det här arbetet modelleras skenande elektroner i disruptioner.
Först undersöks dynamiken hos skenande elektroner då orenheter injiceras i plasmat. För att studera interaktioner mellan snabba elektroner och orenheter på ett korrekt sätt uppdateras modellerna för genereringstakt av skenande elektroner i en numerisk modell som beräknar strömmen av skenande elektroner och den motsvarande utvecklingen av det elektriska fältet. Simuleringar med den uppdaterade modellen visar att materialinjektioner drastiskt påverkar genereringen av skenande elektroner på ett sätt som inte fångas upp av den tidigare modellen. Med rena argoninjektioner fås generellt en ökning i antalet skenande elektroner.
Därefter kopplades modellen till en lösare för fördelningen av elektronernas rörelsemängd i syfte att studera utvecklingen i tid och rum av de skenande elektronernas energispektrum. Det kopplade verktyget användes även till att verifiera den ursprungliga modellens giltighet. Den ursprungliga modellen observeras generera för låga strömmar av skenande elektroner i scenarion där plasmatemperaturen sjunker mycket snabbt men denna skillnad minskar för långsammare temperaturfall. Till sist undersöker vi den numeriska effektiviteten i simuleringsverktygen och visar hur simulationstiden kraftigt kan reduceras utan att resultatets precision påverkas.

Sökord: Skenande elektroner, tokamak, fusion, plasmafysik, kinetisk modellering.

Contents

Chapter 1

Introduction

One of the greatest challenges facing human civilisation today is the ever-increasing use of fossil fuels [1]. The burning of fossil fuels releases large quantities of greenhouse gasses into the atmosphere, which, in turn, is the main cause of global warming. It is therefore necessary to find alternative sources of energy. Harnessing nuclear energy can potentially be a solution to this problem, providing cheap and environmentally sound energy [1], [2].

Nuclear energy is stored inside the nuclei of atoms, and today the only way to convert this energy to electricity is through the use of fission [1]. Fission has many positive aspects, such as near zero emission of greenhouse gasses and small fuel requirements, but there are also several dangers associated with it. The radioactive isotopes that are produced as waste must be safely stored for millions of years and there is a risk of the power plant malfunctioning and releasing these dangerous isotopes into the environment.

To prevent these problems, the other approach to nuclear energy, fusion, can be used [1]. Fusion works by combining two light nuclei into a heavier one, which releases energy. Fusion has no long-lived waste products, and there is no way for the process to lead to a large scale disaster. If a fusion power plant was to be constructed, it would be the ideal source of energy, since it will have ample fuel reserves as well as very little environmental impact [3].

However, controlling fusion is difficult, due to the extremely high temperatures that are needed to overcome the repulsive forces between nuclei. Therefore, confinement with a wall is not possible since it would melt. Instead, the particles are confined by a magnetic field, which works since the fuel becomes a plasma at these temperatures. The *tokamak* is one of the most promising fusion reactor designs [4]. It confines the plasma in a torus configuration, using a strong magnetic field which is partially created by coils outside the confinement and partially by a large plasma current.

Having large plasma currents can cause problems. In the case of an event that rapidly cools the plasma, referred to as a *disruption*, the current would suddenly drop. In such an event, a large electric field is created, that can accelerate a significant portion of the electrons to high speeds. Unlike in normal gasses, the friction force acting upon fast electrons decreases with increasing velocity, allowing them to continue to accelerate until reaching relativistic speeds. These are called *runaway electrons* or just *runaways*, and the study of them is of great importance to the future of fusion energy due to the risk of them hitting important components of a reactor wall and causing significant damage [5].

During the past decades, there has been an international collaboration with the aim of building a reactor-scale tokamak, called ITER (International Thermonuclear Experimental Reactor) [2]. ITER is currently under construction in France and is expected to be completed in 2025. Tokamaks of this size have to run at significantly higher plasma currents than previous designs. This might be problematic due to an increase in the creation of runaways. Once an initial runaway population is established, referred to as *primary runaways*, these can collide with other electrons in the plasma so that both colliding electrons become runaways. This phenomenon is known as the *avalanche* effect. Under certain approximations, the avalanche is exponentially sensitive to the plasma current, increasing the runaway generation in ITER significantly compared to present day tokamaks [6].

If an unmitigated disruption was to happen in ITER, the runaway beam produced is predicted to be able to melt several kilograms of the material from the inner wall [5]. Designing a system that can safely dissipate the thermal energy and the plasma current without generating unacceptably high runaway currents is, therefore, one of the main concerns for the construction of ITER [7]. The currently most researched method to achieve this is to inject neutral gases in the plasma. In spite of research being made, the effects of the massive gas injections are not completely understood. To devise countermeasures to the runaway events, it is necessary to create numerical tools capable of predicting the behaviour of runaways in scenarios with massive material injection. This is the focus of the thesis.

1.1 Method

This work is based on simulations with numerical tools. Within recent years, several such tools have been developed by the plasma theory research group at Chalmers University of Technology. Two tools that are used extensively in this work are GO and CODE. GO is used to model the behaviour of current, temperature, density and electric field as a function of radius and time during a disruption [8], [9]. The generation of runaway electrons is calculated using approximate analytic formulas and all the runaway electrons are assumed to travel at the speed of light. In contrast, CODE (COllisional Distribution of Electrons) determines the runaway generation rates by calculating the distribution of electron momentum as a function of time, but does not have a radial coordinate [10], [11].

To accurately describe the dynamics during a disruption, the position and momentum dynamics of electrons have to be simulated simultaneously. Although CODE is well-suited for calculating the momentum-space distribution of the fast electrons, the lack of spatial information is a severe drawback, due to the fact that the electron dynamics is sensitive to the plasma parameters. Coupling of GO and CODE would allow a description of the radial transport and evolution of background plasma parameters self-consistently. However, this is a challenging task for currently available computational resources. An initial study to perform such a coupling is presented in [12], where it was concluded that it is currently not feasible to use the coupled tool to model scenarios in large tokamaks or make extensive parameter scans with all important effects taken into account. This raises a demand to improve more approximate and numerically efficient models such as updating the accuracy of GO, in addition to creating hybrid tools that simulate the dynamics simultaneously.

1.2 Purpose

The thesis is focused on the numerical modelling of runaway electron dynamics during disruptions in tokamaks. This will be done through simulations made with improved versions of GO and GO+CODE. The avalanche and primary runaway generation rates in GO will be updated to take into account effects in a partially ionised plasma. This is necessary to accurately model disruption mitigation by massive gas injections. Finally, a coupled GO+CODE will be developed, and the numerical efficiency of it improved. Simulations performed with GO+CODE will be used to investigate cases with very fast temperature drops, where the validity of the approximate runaway generation models used in GO are questionable.

1.3 Limitations

The focus of this thesis is on the implementation of models already published in the literature. Limitations will be set by the approximations made in the numerical tools GO and CODE. These approximations are either motivated by computational cost or the properties of a tokamak. The tokamak is approximated as a cylinder and all quantities are assumed to only depend on the distance from the central axis. This works as an approximation of a torus if the ratio between the major and minor radii is large enough. Further limitations of the tools will be explained in detail in Chapter 3.

1.4 Social and ethical aspects

As energy consumption continues to increase, a full-scale fusion reactor is one of the few, albeit theoretical, promising large-scale, carbon-free and environmentally friendly energy sources. Fusion reactors are also safe as the energy production has to be actively controlled and can be shut down immediately at any malfunction. The disadvantages of fusion are connected to the difficult engineering and scientific challenges. Understanding runaway dynamics is one of these challenges and it is essential for the future success of fusion. No known weapon research or any other sensitive ethical dilemmas are connected to the runaway problem.

1.5 Thesis Outline

The structure of the thesis is as follows: in Chapter 2, the relevant background theory of plasma properties and electrodynamics will be explained, leading to a more elaborate description of runaway electrons and how they are created in tokamak disruptions. In Chapter 3, GO and CODE are described in more detail. Improvements and simulations done with GO are presented in Chapter 4 and improvements and simulations with coupled GO and CODE are presented in Chapter 5. Finally, conclusions are summarised and suggestions for future development discussed in Chapter 6.

Chapter 2

Plasma Theory and Disruptions

Plasma physics models can be derived from first principles using only classical mechanics and electrodynamics. The complications first appear when the collective behaviour of the large number of particles is considered. This chapter will begin with an overview of the main features of a tokamak, and how it functions. Later the motion of particles in a plasma, due to external fields and collisions, will be described. These relationships will then be used to explain the runaway phenomenon and how runaway electrons are generated during disruptions in tokamaks. Finally, some of the requirements for a safe disruption mitigation system are discussed.

2.1 Tokamaks

The high temperatures required to initiate and maintain the fusion process demands a fusion reactor design that can confine the plasma without its high temperature damaging the reactor. Though several such designs exist, the tokamak is the most studied design due to its relative simplicity and stability [13, p. 289]. The tokamak design, illustrated in Fig. 2.1, consists of a toroidal vacuum chamber in which the fusion fuel, most commonly consisting of deuterium and tritium,[1] is confined. The major and minor radii of the tokamak chamber are denoted as R and a respectively, and the minor radius coordinate is denoted by r. The magnetic axis is the centre of the cross-section of the tokamak, where $r = 0$.

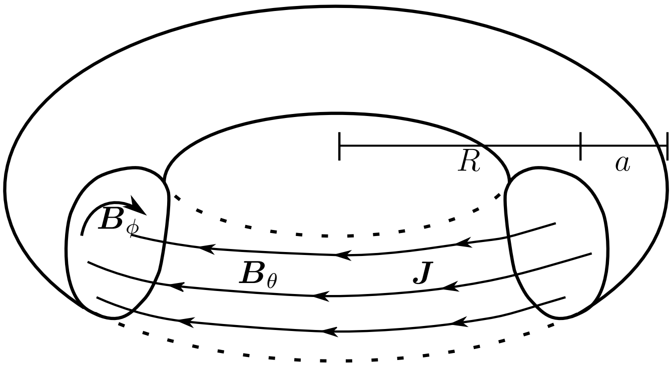


Figure 2.1: Schematic illustration of a tokamak with major radius R and minor radius a. The toroidal $\boldsymbol{B}_\theta$ and poloidal $\boldsymbol{B}_\phi$ of the magnetic field, as well as the plasma current $\boldsymbol{J}$, are also depicted.

[1] When generating fusion energy, the fusion reaction taking place is between deuterium and tritium. However, the models used in this thesis do not make any differences between isotopes with the same atomic number, and we can, therefore, consider plasmas consisting of only deuterium.

Since the fuel is ionised, magnetic fields induced by coils placed around the chamber allow for the plasma to be controlled and confined. In the presence of a magnetic field, the charged particles will follow helical paths around the magnetic field lines, called gyro-orbits. The toroidal magnetic field $\boldsymbol{B}_\theta$ induced by the external coils will, however, be stronger closer to the centre of the tokamak. The radius of the particle orbits around the magnetic field lines will, therefore, be smaller closer to the centre of the tokamak than on the rest of the orbit. This causes a vertical drift out of the confinement. To avoid this, a strong current $\boldsymbol{J}$ in the order of a few MA is induced in the plasma, as shown in figure Fig. 2.1. This current gives rise to a poloidal component $\boldsymbol{B}_\phi$ of the magnetic field lines, which reduces the vertical drift. Plasma parameters for selected tokamaks can be found in Appendix A.

2.2 Plasma Definition

The defining characteristic of a plasma is that the electrons and nuclei are free to move independently of each other [13, p. 3]. This stands in contrast to the other states of matter in which the electrons are bound to the nuclei, forming atoms and molecules. In a fully ionised plasma, the material can be considered to consist of only ions and electrons. A partially ionised plasma is a plasma in which not all atoms have been completely ionised. In either case, particle interactions are dominated by repelling and attractive forces between charges.

A plasma differs in several ways from the other states of matter. One way is that charges are shielded from each other in a process known as *Debye shielding* [1, p. 121]. Negatively charged electrons will gather around any positively charged particle, thus making the positive charge practically invisible beyond a certain length scale. This length is known as the *Debye length*, λ_D. Debye shielding implies that the plasma is quasi-neutral, that is, the plasma at large has no observable net charge.

For the statistical treatment of a plasma to be applicable, the plasma dimension, L, has to be much larger than the Debye length [13, p. 11]. Another requirement is that the number of particles inside a sphere with the radius of a Debye length, N_D, must be very large. This is to make sure that there are particles present able to shield the charge. If there are neutral particles present, the frequency of collisions between them and electrons, ν, have to be smaller than the plasma oscillation frequency,[2] ω, for the material to behave like a plasma. These requirements can be summarised as

$$L \gg \lambda_\mathrm{D}, \quad N_\mathrm{D} \ggg 1, \quad \omega > \nu.$$

2.3 Statistical Description

Due to the large number of particles in a plasma, it is infeasible to describe the motion of every single one. Instead, distribution functions that give statistical information about the distribution of particles in phase space (the combined momentum and position space) are used. The distribution function for particles of species k is denoted f_k and is normalised such that $f_k(\boldsymbol{x}, \boldsymbol{p})\,\mathrm{d}^3\boldsymbol{x}\,\mathrm{d}^3\boldsymbol{p}$ is the number of particles of species k inside the phase space

[2] The plasma oscillation frequency is rather involved and will therefore not be described in this thesis.

volume element $\mathrm{d}^3\boldsymbol{x}\,\mathrm{d}^3\boldsymbol{p}$, where $\boldsymbol{x}$ and $\boldsymbol{p}$ are the coordinates in position and momentum space respectively. Several species at the same time can be considered, for example ions and electrons, $(f_\mathrm{i},\ f_\mathrm{e})$.[3] The position dependence of the distribution function can be dropped, in which case

$$n_k = \int f_k(\boldsymbol{p})\,\mathrm{d}^3\boldsymbol{p}$$

where n_k is the particle density of particle species k. The time dependence of f_k is also implicitly dropped.

To describe the evolution of the distribution function, the conservation equation [14]

$$\frac{\partial f_k}{\partial t} + \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}\cdot\frac{\partial f_k}{\partial \boldsymbol{x}} + \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t}\cdot\frac{\partial f_k}{\partial \boldsymbol{p}} = 0,$$

is used. The change in momentum of a particle is governed by the electromagnetic fields, $\boldsymbol{E}$ and $\boldsymbol{B}$, through the Lorentz-force

$$\frac{\mathrm{d}\boldsymbol{p}_k}{\mathrm{d}t} = q_k(\boldsymbol{E} + \boldsymbol{v}_k \times \boldsymbol{B}),$$

where q_k, m_k, $\boldsymbol{v}_k$, $\boldsymbol{p}_k$ are the charge, mass, velocity and momentum of the particle k.

The charges themselves also have an impact on the electromagnetic fields, but these will be inconvenient to consider, since they are rapidly changing in both time and space. Instead, the electromagnetic fields are assumed to be macroscopic and averaged such that

$$\frac{\partial f_k}{\partial t} + \boldsymbol{v}\cdot\frac{\partial f_k}{\partial \boldsymbol{x}} + q_k(\boldsymbol{E} + \boldsymbol{v}_k \times \boldsymbol{B})\cdot\frac{\partial f_k}{\partial \boldsymbol{p}} = C_k\{f_k\}, \tag{2.1}$$

where the *collision operator* $C_k\{f_k\}$ is included to account for the otherwise excluded interactions between particles. This equation is called the *kinetic equation.*

In the absence of external forces, collisions will make the distribution approach a *Maxwellian* [14, p. 35],

$$f_{\mathrm{M}k}(v) = \frac{n_k}{\pi^{3/2}v_{\mathrm{T}k}^3}e^{-(v/v_{\mathrm{T}k})^2}, \tag{2.2}$$

where $v_{\mathrm{T}k} = \sqrt{2T_k/m_k}$ is the most probable speed, and T_k is the temperature of species k. As is customary in plasma physics the temperature is defined as the average kinetic energy of particles, which differs from the thermodynamic definition by a factor of the Boltzmann constant.

2.4 Coulomb Collisions

When two charged particles come close to each other they exchange momentum through the Coulomb force. These events are referred to as *Coulomb collisions*, and they behave differently compared to collisions between particles in an ordinary gas. In Coulomb collisions, the particles do not actually hit each other, but they are instead deflected smoothly

[3] Throughout this text, the indices i and e denote ions and electrons respectively.

by the Coulomb force. In such collisions, a higher impact speed can be shown to result in a smaller change in momentum of the incoming particle [1, p. 190]. This is the basic phenomenon behind the behaviour of the electrical conductivity and drag force felt by electrons in the plasma, in turn giving rise to the mechanisms by which runaway electrons are generated during tokamak disruptions.

2.4.1 Small Angle Collisions Between Two Charged Particles

Most interactions will not significantly change the velocity of a particle in a plasma, but the accumulation of many small angle collisions still have a larger impact than the few large angle collisions [1, p. 184]. To account for the small angle collisions between particles, the change in velocity of a particle k due to an interaction with a particle of type l is of interest. A non-relativistic calculation of this change of velocity is presented here.

Let b be the distance between the two particles at closest approach if they were not to interact. This distance is called the *impact parameter*. If the masses of the particles are comparable, both k and l will change velocities during the collision. This is a classical two-body problem. To turn it into a one-body problem, the relative coordinates, $\boldsymbol{x} = \boldsymbol{x}_k - \boldsymbol{x}_l$, $\boldsymbol{u} = \mathrm{d}\boldsymbol{x}/\mathrm{d}t$, are used. An illustration of these coordinates is shown in Fig. 2.2.

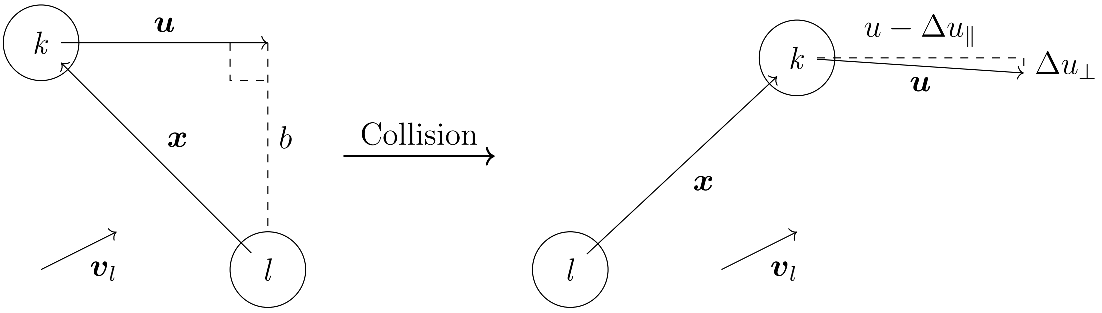


Figure 2.2: Coordinates used when calculating the collision frequency. Prior and posterior coordinates are illustrated. The coordinate system travels with $\boldsymbol{v}_l$, the speed of the l species.

Let $m^* = (m_k^{-1} + m_l^{-1})^{-1}$, Newton's laws of motion as well as Coulomb's law then gives

$$\frac{\mathrm{d}^2\boldsymbol{x}}{\mathrm{d}t^2} = \frac{\mathrm{d}^2\boldsymbol{x}_k}{\mathrm{d}t^2} - \frac{\mathrm{d}^2\boldsymbol{x}_l}{\mathrm{d}t^2} = \frac{1}{m^*}\frac{q_k q_l}{4\pi\varepsilon_0 x^3}\boldsymbol{x}. \tag{2.3}$$

Let $\Delta u_\perp$ be the perpendicular change in $\boldsymbol{u}$ during the collision. Since only small changes in velocity are considered, the approximation $x^2(t) \approx b^2 + u^2t^2$ can be made, where t is a time parameter such that $t = 0$ at closest approach. Equation (2.3) can then be used to calculate

$$\Delta u_\perp \approx \int_{-\infty}^{\infty} \frac{1}{m^*}\frac{q_k q_l}{4\pi\varepsilon_0 x^2}\frac{b}{x}\,\mathrm{d}t = \frac{1}{m^*}\frac{q_k q_l}{4\pi\varepsilon_0}\int_{-\infty}^{\infty}\frac{b}{(b^2+u^2t^2)^{3/2}}\,\mathrm{d}t = \frac{1}{m^*}\frac{q_k q_l}{2\pi\varepsilon_0 u b}.$$

The smallest b considered, $b_{\min}$, is when $\Delta u_\perp \sim u$, giving $b_{\min} := (q_k q_l)/(2\pi\varepsilon_0 u^2 m^*)$. Collisions with impact parameter smaller than $b_{\min}$ are considered large angle collisions, and can thus be justified to be ignored.

From geometry and the conservation of energy $(u - \Delta u_\parallel)^2 + (\Delta u_\perp)^2 = u^2$. This reduces to $\Delta u_\parallel = -(\Delta u_\perp)^2/(2u)$ by assuming that $\Delta u_\parallel / u$ is small. From the relationship $\boldsymbol{x}_k(m_k + m_l) = m_k \boldsymbol{x}_k + m_l \boldsymbol{x}_l + m_l \boldsymbol{x}$, conservation of momentum gives

$$\boldsymbol{v}_k = \frac{m_l}{m_k + m_l} \boldsymbol{u}$$

The changes in $\boldsymbol{v}_k$ during the collisions can then be summarised with

$$\begin{aligned} \Delta v_{k\perp} &= \frac{m_l}{m_k + m_l} \frac{1}{m^*} \frac{q_k q_l}{2\pi\varepsilon_0 u b} = \frac{q_k q_l}{2\pi\varepsilon_0 u b m_k} \\ \Delta v_{k\parallel} &= -\frac{m_l}{m_k + m_l} \left(\frac{1}{m^*} \frac{q_k q_l}{2\pi\varepsilon_0 u b} \right)^2 \frac{1}{2u} = -\left(1 + \frac{m_k}{m_l}\right) \left(\frac{q_k q_l}{2\pi\varepsilon_0 m_k} \right)^2 \frac{1}{2b^2 u^3} . \end{aligned}$$

Note that the parallel and perpendicular parts are in relation to $\boldsymbol{u}$, or in other words $\Delta \boldsymbol{v}_k = \Delta v_{k\parallel} \hat{\boldsymbol{u}} + \Delta v_{k\perp} \hat{\boldsymbol{b}}$, where $\hat{\boldsymbol{b}}$ is in the direction of the impact parameter. Due to symmetry, the exact direction of $\hat{\boldsymbol{b}}$ will not be important.

2.4.2 Drag Force

During a set of events under the time period $\mathrm{d}t$ the average change in velocity of a single particle of type k due to collisions with particles of type l will be[4]

$$\mathrm{d}\langle \Delta \boldsymbol{v} \rangle^{kl} = \int_{b=b_{\min}}^{b=\lambda_\mathrm{D}} 2\pi b \, \mathrm{d}b \int_{\boldsymbol{v}'} f_l(\boldsymbol{v}') u \, \mathrm{d}t \, \Delta \boldsymbol{v}^{kl} \, \mathrm{d}^3 \boldsymbol{v}' \, .$$

The upper limit of the b integral is chosen to exclude the effect of particles with distance larger than λ_D due to the particles being sufficiently screened at those length scales. The lower limit is chosen due to the approximations being invalid below that point.

By observing that changes in $\Delta \boldsymbol{v}$ perpendicular to the velocity will average out due to symmetry, the change in velocity along the z-axis can be written as

$$\begin{aligned} \frac{\mathrm{d}\langle \Delta v_z \rangle^{kl}}{\mathrm{d}t} &= \int_{b=b_{\min}}^{b=\lambda_\mathrm{D}} 2\pi b \, \mathrm{d}b \int_{\boldsymbol{v}'} f_l(\boldsymbol{v}') u \Delta v_\parallel^{kl} \hat{u} \, \mathrm{d}^3 \boldsymbol{v}' \\ &= -\left(1 + \frac{m_k}{m_l}\right) \frac{q_k^2 q_l^2 \ln \Lambda}{4\pi\varepsilon_0^2 m_k^2} \int_{\boldsymbol{v}'} \frac{f_l(\boldsymbol{v}')}{u^2} \frac{u_z}{u} \, \mathrm{d}^3 \boldsymbol{v}' \, , \end{aligned} \tag{2.4}$$

where $\ln \Lambda = \ln\big(\lambda_\mathrm{D}/b_{\min}\big)$ is approximated to not be dependent on the relative velocity. The last equality comes from expanding the parallel velocity and integrating with respect to b.

The factor $\ln \Lambda$ is called the *Coulomb logarithm.* In an ordinary plasma it is roughly 10 to 20, and is proportional to the logarithm of the number of particles in the Debye sphere, N_D [14]. Therefore, by the definition of a plasma, the Coulomb logarithm can always be considered to be large.

[4] All collisions during time $\mathrm{d}t$ and impact parameter b take place in the cylindrical shell with the length $u \, \mathrm{d}t$, circumference $2\pi b$ and thickness $\mathrm{d}b$. Then the integral is just a cause of the definition of expectation value.

Since the equilibrium distribution f_l of the particles of species l will approach a Maxwellian in the absence of external forces, f_l can often be approximated as such. In that case, by noting the isotropy of f_l and similarities with inverse square forces, Newton's shell theorem can be used to carry out the integral [15, Proposition 70-71]

$$\int_{\boldsymbol{v}'} f_l(\boldsymbol{v}')\frac{1}{u^2}\frac{u_z}{u}\,\mathrm{d}^3\boldsymbol{v}' = \frac{1}{v_k^2}\int_{|\boldsymbol{v}'|\leq v_k} f_l(\boldsymbol{v}')\,\mathrm{d}^3\boldsymbol{v}' = \frac{n_l}{v_{\mathrm{T}l}^2}\frac{\mathrm{erf}(s) - s\,\mathrm{erf}'(s)}{s^2} = \frac{2n_l}{v_{\mathrm{T}l}^2}G(s),$$

where $s = v_k/v_{\mathrm{T}l}$, $\mathrm{erf}(s)$ is the error function, and $G(s) = (\mathrm{erf}(s) - s\,\mathrm{erf}'(s))/(2s^2)$. The latter integral is solved using spherical coordinates. As can be seen, the dynamical friction is proportional to $G(s)$, the *Chandrasekhar function*, which has the shape plotted in Fig. 2.3. The asymptotic behaviour of G is $2s/(3\sqrt{\pi})$ when $s \to 0$, and $1/(2s^2)$ when $s \to \infty$. This result agrees with a relativistic derivation for particle energies much smaller than the particle rest energy, despite the derivation being non-relativistic [14, p. 47].

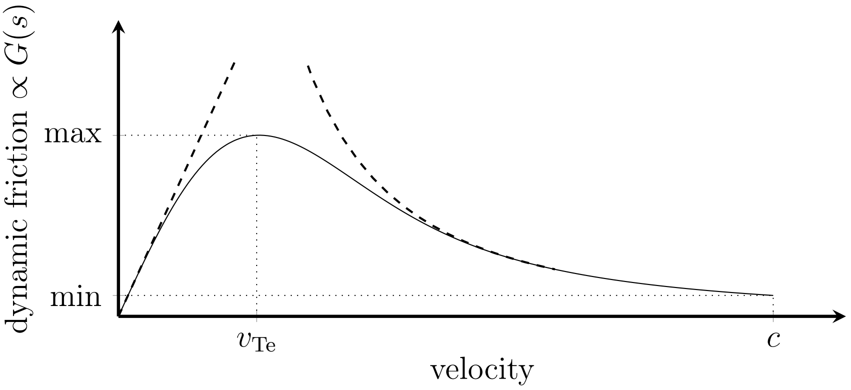


Figure 2.3: Frictional force plotted against velocity. Asymptotic behaviours of the Chandrasekhar function are in dashed lines. Velocities are not necessarily to scale.

2.4.3 Collision Frequency

It is useful to have a measure of how often collisions are happening. Therefore, a reasonable definition of the *collision frequency*, ν_{kl}, is how often the velocity is significantly changed due to collisions [14, p. 39],

$$\frac{\mathrm{d}\langle\Delta v\rangle^{kl}}{\mathrm{d}t} = -v_z\nu_{kl}.$$

For simplicity, only electron-ion collisions are considered, in which case $m_{\mathrm{e}}/m_{\mathrm{i}} \approx 0$ and $f_{\mathrm{i}}(\boldsymbol{v}) \approx \delta(\boldsymbol{v})$, where δ is the Dirac delta function. Eq. (2.4) is then simplified significantly, which gives

$$\nu_{\mathrm{ei}} = \frac{n_{\mathrm{i}}q_{\mathrm{e}}^2q_{\mathrm{i}}^2\ln\Lambda}{4\pi\varepsilon_0^2m_{\mathrm{e}}^2v_{\mathrm{e}}^3}.$$

The inverse of the collision frequency is the *collision time*, $\tau_{kl} := 1/\nu_{kl}$.

2.4.4 Conductivity

External electric fields cause charged particles in a plasma to move, thus creating currents. For currents carried by electrons with velocities similar to the thermal velocity, the relation between the current and the electric field can be described by Ohm's law, $\boldsymbol{J}_{\text{ohm}} = \sigma \boldsymbol{E}$, where σ is the conductivity [13, p. 160].

Let $\langle \boldsymbol{v}_{\text{d}} \rangle$ be the average drift velocity of electrons. The current is then given by $\boldsymbol{J}_{\text{ohm}} = n_{\text{e}} q_{\text{e}} \langle \boldsymbol{v}_{\text{d}} \rangle$. The contribution to the current from the motion of ions is negligible due to the ions having much higher inertia than electrons. Only electron-ion collisions need be considered when estimating the conductivity since the conservation of momentum causes electron-electron collisions to have no impact on the current. The conductivity can then be calculated by assuming that the electron distribution is a shifted Maxwellian, and then solving for the drift speed that equates the frictional force due to electron-ion collisions and electric force. The relationship

$$\sigma = \frac{(4\pi\varepsilon_0)^2 T^{3/2}}{\pi Z e^2 m_{\text{e}}^{1/2} \ln \Lambda}, \tag{2.5}$$

where Z is the atomic number of the ion and e is the elementary charge, is then found [16]. Note, however, that Ohm's law is only valid in a plasma for currents carried by thermal electrons and cannot describe currents carried by runaway electrons.

2.5 Runaway Electrons

As can be seen by the shape of the Chandrasekhar function in Fig. 2.3, the drag force becomes smaller when the velocity is increased. If a strong enough electric field is present, electrons with a velocity above a certain *critical velocity*, v_{c}, could therefore be accelerated indefinitely. These are called runaway electrons. The dominating drag force at these speeds will be from electron-electron collisions [14]. By inserting the asymptotic behaviour of the Chandrasekhar function in Eq. (2.4), this force can be approximated to be

$$F^{\text{ee}} = \frac{q_{\text{e}}^4 n_{\text{e}} \ln \Lambda}{2\pi\varepsilon_0^2 m_{\text{e}} v_{\text{e}}^2}. \tag{2.6}$$

The Chandrasekhar function has a maximum at $s = 1$ ($v_{\text{e}} = v_{\text{Te}}$), and then decreases monotonically as s approaches infinity. Therefore, in the presence of an electric field that causes an accelerating force stronger than the friction force at $s = 1$, all electrons will become runaways. A measure of this electric field is obtained by inserting $s = 1$ in the asymptotic behaviour of the Chandrasekhar function for large s. It is called the *Dreicer field*, and is given by [14, p. 39]

$$E_{\text{D}} := \frac{q_{\text{e}}^3 n_{\text{e}} \ln \Lambda}{4\pi\varepsilon_0^2 T_{\text{e}}}.$$

The actual value for the electric field required to make all electrons become runaways is $0.21 E_{\text{D}}$. Since the friction force decreases monotonically with velocity for $s \geq 1$, a minimum friction force is obtained for electrons travelling at the speed of light. The electric field needed to overcome this friction force defines a lower limit for the electric field below which no runaways can be created. From a relativistic derivation, this electric field, called the *critical electric field*, is found to be [17]

$$E_{\mathrm{c}} := \frac{q_{\mathrm{e}}^3 n_{\mathrm{e}} \ln \Lambda}{4\pi\varepsilon_0^2 m_{\mathrm{e}} c^2}, \tag{2.7}$$

A similar expression can be found by inserting $v_{\mathrm{e}} = c$ in Eq. (2.4).

2.5.1 Disruptions

In a fusion tokamak, events called disruptions can occur where the plasma confinement is suddenly lost and the temperature drops rapidly [5]. The time phase in which the temperature drops is called the *thermal quench*. In this phase, the conductivity also drops, since $\sigma \propto T^{3/2}$ as seen in Eq. (2.5). Due to induction, the current cannot drop arbitrarily fast, and there must be a rise in the electric field to compensate for the reduced conductivity. The ohmic current then drops over a longer time scale, in a phase called the *current quench*.

In a simplified model, where the temperature decreases from T_1 to T_2 instantly, the change in the electric field can be estimated using conservation of ohmic current, which gives $E_2 \sim E_1(T_1/T_2)^{3/2}$. This mechanism can generate electric fields much larger than the critical field, E_{c}, potentially causing the creation of runaways.

More generally, one can use the electric field diffusion equation to get the electric field evolution. This reads

$$\nabla^2 \boldsymbol{E} = \frac{\partial}{\partial t}(\mu_0 \boldsymbol{J}), \tag{2.8}$$

and is obtained from Maxwell's equations by neglecting the displacement current and assuming quasi-neutrality. Intuitively, when the current decreases an electric field is induced to try and maintain the current. The total current can be written as the sum of the ohmic current and the runaway current, so that Eq. (2.8) can be written as

$$\nabla^2 \boldsymbol{E} = \mu_0 \frac{\partial}{\partial t}(\sigma \boldsymbol{E} + \boldsymbol{J}_{\mathrm{run}}). \tag{2.9}$$

2.5.2 Runaway Generation Mechanisms

For an electron to become a runaway its velocity has to get above the critical velocity. In tokamak disruptions, the most important generation mechanisms are *Dreicer*, *avalanche*, and *hot-tail*. In addition to these, runaways can also be produced by tritium decay or by gamma-rays generated from the radioactive decay of the wall material, but these will be neglected in this thesis.

The Dreicer generation of runaways is caused by particles randomly getting a velocity larger than the critical velocity, v_{c}, due to the statistical nature of Coulomb collisions [18]. The hot-tail generation comes into play when the temperature drops quickly during a disruption [19]. In this case, v_{c} will decrease faster than the electrons have time to slow down due to the decreasing temperature. Electrons with velocities above v_{c} will then become runaways.

The avalanche generation is due to large-angle collisions between electrons [6]. It can become significant if an electron with a velocity above v_c transfers a significant amount of its momentum to a slow electron, pushing it above the critical velocity. This new electron can then also potentially collide with yet another, causing an exponential growth of the runaway electron population, an “avalanche”. Since this mechanism amplifies a pre-existing runaway population, referred to as a *runaway seed* or just *seed*, runaway electrons created by the avalanche mechanism are called *secondary runaway electrons.* Runaways generated by other mechanisms are called primary runaway electrons.

There are also several other mechanisms affecting the runaway generation which are not included in this thesis. For example, the runaway electrons do not necessarily stay in the plasma. A significant loss mechanism of runaways is diffusion partially caused by perturbations in the magnetic field. Another generation mechanism excluded is γ-generation where high energy γ-rays transfer energy to electrons pushing them over the critical velocity.

2.6 Disruption Mitigation

Disruptions can cause severe damage to an ITER-sized tokamak, both by runaway currents striking the wall and other mechanisms [7]. The amount of energy to be dissipated is approximately 350 MJ of thermal energy and 1 GJ of magnetic energy. The details of the damages caused by disruptions are beyond the scope of this thesis, but some constraints that must be satisfied by the mitigation system, according to [7], are listed below.

When designing a disruption mitigation system, there are mainly three parameters one wants to control: the thermal quench time, the current quench time and how much runaway current is formed. The currently most promising way to mitigate disruptions is to inject neutral gas into the plasma. If a sufficient amount of gas is injected, the thermal energy can be radiated away faster than it is lost by convection, resulting in a more uniform distribution of the heat load on the vessel. Suitable gases for this purpose are noble gases such as argon or neon.

Disruptions in ITER are expected to have a warning time of about 20 ms and an unmitigated thermal quench time in the order of 1 ms to 2 ms. To limit the heat load on the vessel the disruption mitigation system should radiate away the thermal energy faster than that. To avoid unacceptable electromagnetic forces on the vessel, the current quench time should be longer than 50 ms in order to limit the induced currents in the tokamak structure surrounding the plasma. In addition, the current quench time should not be longer than 150 ms so that the current quench is over before the plasma drifts into the wall. The current quench time is mostly determined by the post-disruption temperature, which in turn depends on the amount of injected gas. Finally, the runaway current striking the wall should not be higher than 2 MA.

To dampen the runaway generation, argon or neon injections can be combined with extra deuterium. The extra deuterium raises the electron density in order to increase the critical electric field, while only having a small effect on the radiation loss time scale and current quench time scale. The increased critical electric field would, in turn, dampen the runaway generation, especially the secondary generation.

Chapter 3

Numerical Tools

When modelling a plasma, simulating every single particle present is not remotely feasible due to the density of particles being on the order of $10^{20}\,\mathrm{m}^{-3}$. Instead, the kinetic equation, Eq. (2.1), is used to determine the electron distribution function. This is referred to as *kinetic modelling*. Solving the kinetic equation over the whole phase space is however still not computationally possible, so further simplifications have to be made. There are several ways this can be done. One way is to assume that the processes studied are happening on time scales longer than the collision time and that the velocity distribution at every point, therefore, is close to a steady state. This reduces the equation to only three dimensions, and is called *fluid modelling*. For shorter timescales, fluid modelling can be insufficient due to the lack of information about the velocity distribution. In that case, the spatial dependence could instead be neglected, therefore only simulating the velocity distribution. This is a type of kinetic modelling. There are also several ways of making use of symmetries to reduce dimensionality. In the following sections, it is described how this is done in the simulation tools GO and CODE.

3.1 GO

GO is a fluid model which calculates the self-consistent evolution of the electric field and current during a disruption scenario [8], [20]. The current density is divided into an ohmic part, $J_{\mathrm{ohm}} = \sigma E$, and a runaway part J_{run}. The runaway part is calculated by approximate analytical expressions for the growth rates of the previously described runaway generation mechanisms. All runaways are assumed to travel with the speed of light, so that $J_{\mathrm{run}} = q_{\mathrm{e}} c n_{\mathrm{run}}$ where n_{run} is the runaway electron density. The main part of the code solves the coupled system of differential equations consisting of the electric field evolution, which is given by Eq. (2.9) and analytical formulae for the runaway growth rate. In GO, the plasma is approximated as an axially symmetric cylinder, but toroidal corrections are taken into account in the expression of the plasma conductivity. With cylindrical symmetry, Eq. (2.9) reduces to

$$\frac{1}{r}\frac{\partial}{\partial r} r \frac{\partial E_{\|}}{\partial r} = \mu_0 \frac{\partial J}{\partial t} = \mu_0 \frac{\partial}{\partial t}(\sigma E_{\|} + J_{\mathrm{run}}),$$
$$J_{\mathrm{run}} = q_{\mathrm{e}} c n_{\mathrm{run}}$$

where $E_{\|}$ is the component of the electric field parallel to the magnetic axis.

The temperature is assumed to decay exponentially in time between a given initial profile $T_{\mathrm{i}}(r)$ and final profile $T_{\mathrm{f}}(r)$ according to $T(r,t) = T_{\mathrm{f}}(r) + \big(T_{\mathrm{i}}(r) - T_{\mathrm{f}}(r)\big)\exp\big(-t/t_0\big)$, where $t_0 \sim 1\,\mathrm{ms}$, referred to as the thermal quench time, is given by the user. Using predetermined input is useful if the temperature and density development is known from experiments, and in numerical model developments, which is focused on in this thesis.

To calculate the runaway current, analytical models for different runaway mechanisms are used [20]. More specifically, the Dreicer mechanism is modelled by

$$\left(\frac{\partial n_{\mathrm{run}}}{\partial t}\right)^{\mathrm{Dreicer}} = \frac{n_{\mathrm{e}}}{\tau}\left(\frac{m_{\mathrm{e}}c^2}{2T_{\mathrm{e}}}\right)^{3/2}\left(\frac{E_{\mathrm{D}}}{E_\parallel}\right)^{3(1+Z_{\mathrm{eff}})/16} \exp\left(-\frac{E_{\mathrm{D}}}{4E_\parallel} - \sqrt{\frac{(1+Z_{\mathrm{eff}})E_{\mathrm{D}}}{E_\parallel}}\right), \tag{3.1}$$

where $Z_{\mathrm{eff}} = (\sum_i Z_i^2 n_i)/n_{\mathrm{e}}$ is the effective ion charge [17] and $\tau = 4\pi\varepsilon_0^2 m_{\mathrm{e}}^2 c^3/(n_{\mathrm{e}}e^4 \ln\Lambda)$ is the collision time for relativistic electrons. The avalanche is modelled by the analytical formula of Rosenbluth and Putvinski

$$\begin{aligned}\left(\frac{\partial n_{\mathrm{run}}}{\partial t}\right)^{\mathrm{avalanche}} =& n_{\mathrm{run}}\frac{E_\parallel/E_{\mathrm{c}}-1}{\tau\ln\Lambda}\sqrt{\frac{\pi\varphi}{3(Z_{\mathrm{eff}}+5)}}\cdot \\ & \left(1-\frac{E_{\mathrm{c}}}{E_\parallel}+\frac{4\pi(Z_{\mathrm{eff}}+1)^2}{3\varphi(Z_{\mathrm{eff}}+5)(E_\parallel^2/E_{\mathrm{c}}^2+\frac{4}{\varphi^2}-1)}\right)^{-1/2},\end{aligned} \tag{3.2}$$

where $\varphi = (1+1.46\epsilon^{1/2}+1.72\epsilon)^{-1}$ and $\epsilon = a/R$ is the inverse aspect ratio of the tokamak [6]. These two formulae are derived assuming a fully ionised plasma.

The hot-tail generation is calculated from a solution to a simplified version of the kinetic equation. Here, the effect of the electric field is not taken into account other than that it determines the momentum boundary of the runaway region. The kinetic equation being solved for f is

$$\frac{\partial f}{\partial t} = C\{f\} = \frac{q_{\mathrm{e}}^4 n\ln\Lambda}{8\pi\varepsilon_0^2 m_{\mathrm{e}}}\frac{1}{v^2}\frac{\partial}{\partial v}\left[v^2 G(v/v_{\mathrm{T}})\left(\frac{1}{T}f(v)+\frac{1}{m_{\mathrm{e}}v}\frac{\partial f(v)}{\partial v}\right)\right]. \tag{3.3}$$

In the case of an exponential-like temperature development, this equation can be treated analytically according to [21]. Otherwise, Eq. (3.3) is solved numerically [20]. The solution is then used to calculate the hot tail generation using

$$\left(\frac{\partial n_{\mathrm{run}}}{\partial t}\right)^{\mathrm{hot-tail}} \approx 4\pi\frac{\mathrm{d}}{\mathrm{d}t}\int_{v_{\mathrm{c}}}^{\infty}(v^2-v_{\mathrm{c}}^2)f\,\mathrm{d}v\,.$$

The total runaway density evolution is thus given by

$$\frac{\partial n_{\mathrm{run}}}{\partial t} = \left(\frac{\partial n_{\mathrm{run}}}{\partial t}\right)^{\text{Dreicer}} + \left(\frac{\partial n_{\mathrm{run}}}{\partial t}\right)^{\text{hot-tail}} + \left(\frac{\partial n_{\mathrm{run}}}{\partial t}\right)^{\text{avalanche}}.$$

In this work, runaway generation due to tritium decay or gamma-rays originating from the activated wall, as well as losses of runaways, are neglected.

All derivatives are discretised by finite difference approximations. The electric field diffusion equation and runaway diffusion term is discretised by a Crank-Nicolson scheme. As a standard for the kind of simulations performed in this thesis, the time resolution is 1 µs and the radial discretisation is homogeneous with a spacing of $a/101$.

3.2 CODE

CODE solves for the time-evolution of the electron momentum distribution using kinetic modelling [10], [11]. To be able to do this efficiently, some simplifying assumptions are made. The first is to assume that the plasma is homogeneous in space. The second assumption is that the distribution in momentum space has cylindrical symmetry around the electric and magnetic field lines. This assumption can be made since the gyro-orbits of the electrons are very fast compared to the other relevant time scales, and thus averages out. Another assumption that is made is that the electromagnetic fields, $\boldsymbol{E}$ and $\boldsymbol{B}$, are parallel, which is approximately the case either close to the magnetic axis, or in a tokamak with a major radius much larger than the minor radius.

As CODE uses kinetic models, it solves Eq. (2.1), which, with the spatial gradient removed due to the assumed homogeneity, becomes

$$\frac{\partial f_\mathrm{e}}{\partial t} + q_\mathrm{e}(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f_\mathrm{e}}{\partial \boldsymbol{p}} = C\{f_\mathrm{e}\} + S\{f_\mathrm{e}\},$$

where a source term, S, has been added in order to model avalanche generation and to numerically control the density and temperature evolution. The exact form of the collision operator and the source terms are rather involved and are therefore not included here. A description of the original implementation is found in [10].

CODE parameterises $\boldsymbol{p}$ using the spherical coordinates p, θ, and ϕ, where p is the magnitude, θ is the angle with respect to $\boldsymbol{E}$, and ϕ is the gyro-angle around $\boldsymbol{E}$. The expansion of the gradient into these new coordinates gives

$$\frac{\partial f_\mathrm{e}}{\partial \boldsymbol{p}} = \frac{\partial f_\mathrm{e}}{\partial p}\hat{\boldsymbol{p}} + \frac{1}{p}\frac{\partial f_\mathrm{e}}{\partial \theta}\hat{\boldsymbol{\theta}} + \frac{1}{p \sin\theta}\frac{\partial f_\mathrm{e}}{\partial \phi}\hat{\boldsymbol{\phi}}.$$

The last term vanishes due to the assumed cylindrical symmetry, and thus the coordinate ϕ is not used in CODE. The dot product can be distributed using the relationships $(\boldsymbol{v} \times \boldsymbol{B}) \cdot \hat{\boldsymbol{p}} = (\boldsymbol{v} \times \boldsymbol{B}) \cdot \hat{\boldsymbol{\theta}} = 0$, $\boldsymbol{E} \cdot \hat{\boldsymbol{p}} = E\cos\theta$, and $\boldsymbol{E} \cdot \hat{\boldsymbol{\theta}} = -E\sin\theta$. These are simply an effect of how the parameterisation was chosen, and because of the assumption that $\boldsymbol{B}$ is in the direction of $\boldsymbol{E}$. Combining these, the effects of the magnetic field vanishes and only scalar quantities remain,

$$\frac{\partial f_\mathrm{e}}{\partial t} + q_\mathrm{e}E\left(\cos\theta\frac{\partial f_\mathrm{e}}{\partial p} - \frac{\sin\theta}{p}\frac{\partial f_\mathrm{e}}{\partial \theta}\right) = \frac{\partial f_\mathrm{e}}{\partial t} + q_\mathrm{e}E\left(\xi\frac{\partial f_\mathrm{e}}{\partial p} + \frac{1-\xi^2}{p}\frac{\partial f_\mathrm{e}}{\partial \xi}\right) = C\{f_\mathrm{e}\} + S\{f_\mathrm{e}\},$$

where $\xi = \cos\theta$.

The collision operator can be described as the sum of the contributions from electron-electron and electron-ion collisions, $C\{f_\mathrm{e}\} = C_\mathrm{ee}\{f_\mathrm{e}, f_\mathrm{e}\} + C_\mathrm{ei}\{f_\mathrm{e}, f_\mathrm{i}\}$. The electron-ion contribution is linear in f since the Coulomb collision operator is bilinear, that is to say

$$\begin{aligned} C_{kl}\{f_k + g_k, f_l\} &= C_{kl}\{f_k, f_l\} + C_{kl}\{g_k, f_l\} \\ C_{kl}\{f_k, f_l + g_l\} &= C_{kl}\{f_k, f_l\} + C_{kl}\{f_k, g_l\} \\ C_{kl}\{c_k f_l, c_k f_l\} &= c_k c_l C_{kl}\{f_k, f_l\}, \end{aligned}$$

for any distributions f_k, g_k, f_l, g_l, and constants c_k, c_l [14]. This, however, makes the collision operator for electron-electron collisions nonlinear, since using the relations above give $C_{\mathrm{ee}}\{2f_{\mathrm{e}}\} = C_{\mathrm{ee}}\{2f_{\mathrm{e}}, 2f_{\mathrm{e}}\} = 4C_{\mathrm{ee}}\{f_{\mathrm{e}}\}$.

Since the distribution function is close to a Maxwellian, a simpler linear collision operator can be derived. In this case, $f_{\mathrm{e}} = f_{\mathrm{Me}} + f_{\mathrm{e0}}$ where f_{Me} is the Maxwellian part, and $f_{\mathrm{e0}} \ll f_{\mathrm{Me}}$. The operator can now be approximated as

$$\begin{aligned}C_{\mathrm{ee}}\{f_{\mathrm{e}}\} &= C_{\mathrm{ee}}\{f_{\mathrm{Me}} + f_{\mathrm{e0}}\}\\ &= C_{\mathrm{ee}}\{f_{\mathrm{Me}}, f_{\mathrm{Me}}\} + C_{\mathrm{ee}}\{f_{\mathrm{Me}}, f_{\mathrm{e0}}\} + C_{\mathrm{ee}}\{f_{\mathrm{e0}}, f_{\mathrm{Me}}\} + C_{\mathrm{ee}}\{f_{\mathrm{e0}}, f_{\mathrm{e0}}\}\\ &\approx C_{\mathrm{ee}}\{f_{\mathrm{Me}}, f_{\mathrm{e0}}\} + C_{\mathrm{ee}}\{f_{\mathrm{e0}}, f_{\mathrm{Me}}\} =: C^l_{\mathrm{ee}}\{f_{\mathrm{e}}\},\end{aligned}$$

where the quadratic term in f_{e0} has been neglected. Collisions between two Maxwellian distributions with the same temperature vanish since they are at the same thermodynamical equilibrium. The linear collision operator $C^l_{\mathrm{ee}}\{f_{\mathrm{e}}\}$ is used in CODE. For performance reasons, the field particle term, $C_{\mathrm{ee}}\{f_{\mathrm{Me}}, f_{\mathrm{e0}}\}$, is sometimes also neglected because of its weak effect on runaway dynamics.

When the collision operator is linearised, the kinetic equation can be written as

$$\frac{\partial f_{\mathrm{e}}}{\partial t} + M f_{\mathrm{e}} = S, \tag{3.4}$$

where M is a linear operator such that

$$M f_{\mathrm{e}} = q_{\mathrm{e}} E\left(\xi \frac{\partial f_{\mathrm{e}}}{\partial p} + \frac{1-\xi^2}{p}\frac{\partial f_{\mathrm{e}}}{\partial \xi}\right) - C\{f_{\mathrm{e}}\}.$$

The ξ parameter is discretised in N_ξ Legendre polynomials, P_L such that

$$f_{\mathrm{e}}(p,\xi) = \sum_{L=0}^{\infty} F_L(p)P_L(\xi) \approx \sum_{L=0}^{N_\xi - 1} F_L(p)P_L(\xi).$$

Internally in CODE, the normalised momentum $y = p/(m_e v_{\mathrm{Te}})$ is used instead of p. Using finite differences, the new parameter y is discretised such that the distribution function can be stored as

$$F = \begin{bmatrix}\begin{bmatrix}F_0(y_1) & \dots & F_0(y_{\max})\end{bmatrix} & \dots & \begin{bmatrix}F_{N_{\xi-1}}(y_1) & \dots & F_{N_{\xi-1}}(y_{\max})\end{bmatrix}\end{bmatrix}^{\mathrm{T}}.$$

The reason for choosing this discretisation is that the operator M becomes a sparse matrix, which makes numerical calculations efficient. The number of discrete y coordinates is denoted by N_y.

Equation (3.4) is then solved using an implicit method, most commonly backward Euler, taking time steps of length Δt. In that case Eq. (3.4) is written as

$$\frac{F_{t+\Delta t} - F_t}{\Delta t} + M F_{t+\Delta t} = S,$$

which can be rewritten as

$$F_{t+\Delta t} = (I + \Delta t M)^{-1}(F_t + \Delta t S),$$

where I is the identity matrix. For numerical speed, CODE uses a sparse LU-factorisation of $I + \Delta t M$. After being created, the M matrix does not change unless the temperature, the electric field, or similar plasma parameters are updated. Therefore, the LU-factorisation can be reused over all the time steps with similar plasma quantities.

Chapter 4

Improved Fluid Model Simulations

In this chapter, improvements and simulation results obtained with GO are described. As noted in Section 3.1, the equations used to model the Dreicer and avalanche mechanisms in GO, Eq. (3.1) and Eq. (3.2), are only valid in a fully ionised plasma. When mitigating disruptions using neutral gas injection, not all injected particles will be fully ionised. Interactions with bound electrons are therefore important to model accurately in such a scenario. In Section 4.1, a neural network trained on kinetic models to calculate Dreicer runaway generation in the presence of partially ionised impurities is described and some initial simulation results obtained with the neural network are shown. In Section 4.2, the calculation of the avalanche growth rate is updated to take effects of partially ionised impurities into account, and the updated growth rate is investigated in ITER-like scenarios.

4.1 Dreicer Calculation with Impurities

The calculation of the Dreicer growth rate using CODE is computationally expensive. Instead of directly running CODE, a neural network that has been trained on a large number of CODE simulations was used. The neural network functions as an interpolation between the input containing information about the plasma, and the output representing the corresponding growth rate. It is able to predict a simulation output from a collection of similar simulations. The network has been developed in another project and is used here due to its applicability to GO simulations.

Several parameters relating to the densities of the impurities, as well as the electric field, and temperature are used as input to the neural network, through the vector, $\boldsymbol{x}$. The atomic number and the charge of species a will be denoted by Z_a and Z_{0a} respectively. For instance, the ion Ar^{5+} gives $Z = 18$, $Z_0 = 5$. The input vector has the information

$$\boldsymbol{x} = \left(\frac{E}{E_\mathrm{D}}, \log\left(\frac{T_\mathrm{e}}{\mathrm{eV}}\right), \log\left(\sum_a (Z_a - Z_{0a}) n_a\right), \log\left(\sum_a Z_{0a} n_a\right), \right. \\ \left. \log\left(\sum_a (Z_a^2 - Z_{0a}^2) n_a\right), \log\left(\sum_a Z_{0a}^2 n_a\right)\right),$$

where the third and fourth elements can be seen as the logarithm of the density of free electrons, and the logarithm of the density of bound electrons. For numerical stability, the input vector is normalised within the range $[0, 1]$ and is denoted by $\tilde{\boldsymbol{x}}$.

The output of the network, $\tilde{G}$ is the logarithm of the growth rate, which is also normalised to $[0, 1]$. The network itself can then be described as the matrix equation

$$\tilde{G} = \mathbf{W}_3 g(\mathbf{W}_2 g(\mathbf{W}_1 \tilde{\boldsymbol{x}} + \boldsymbol{b}_1) + \boldsymbol{b}_2) + \boldsymbol{b}_3,$$

where $\mathbf{W}_1 \in \mathbb{R}^{20\times 6}$, $\mathbf{W}_2 \in \mathbb{R}^{20\times 20}$, $\mathbf{W}_3 \in \mathbb{R}^{1\times 20}$ are *weight* matrices, $\boldsymbol{b}_1 \in \mathbb{R}^{20\times 1}$, $\boldsymbol{b}_2 \in \mathbb{R}^{20\times 1}$, $\boldsymbol{b}_3 \in \mathbb{R}^{1\times 1}$ are *bias* vectors, and the *activation function* $g(x) := \big(1+\exp(-x)\big)^{-1}$ is applied element-wise to each matrix separately. This type of network is widely used, and is called a multilayer perceptron [22, p. 178]. To find appropriate values for weights and biases, the network was trained on results from approximately a million CODE simulations, using backpropagation, a standard machine learning technique [22, p. 183].

The final neural network is several orders of magnitude faster than running CODE directly,[1] since it only consists of a few matrix multiplications, vector additions, and applications of the g function. By training the network for different temperatures, densities, and ionisation degrees, it could then be used in other simulation tools, such as GO.

In the rest of this section, some results obtained from implementing the neural network in GO will be described. Since the neural network treats the Dreicer generation mechanism, a case similar to the TEXTOR tokamak where the Dreicer generation is the dominant source of runaways is studied. The parameters of the TEXTOR-tokamak are given in Appendix A. In its present state, the neural network can only handle temperatures between 100 eV and 10 eV and therefore a flat final temperature profile with a temperature of 10 eV is assumed. For temperatures above 100 eV, the expression in Eq. (3.1) is used. The Dreicer generation mechanism does not get started until the thermal quench is close to finished, so this should have a very small effect on the result.[2] Once the Dreicer generation gets started, however, it takes place over a timescale comparable to the rise in the electric field. If the Dreicer generated current is large enough to have a significant effect on the electric field, the interaction between the Dreicer generation and electric field during the end of a thermal quench must, therefore, be resolved. This requires a non-zero thermal quench time. Therefore, an exponential drop time scale of $t_0 = 0.5\,\text{ms}$ is used.

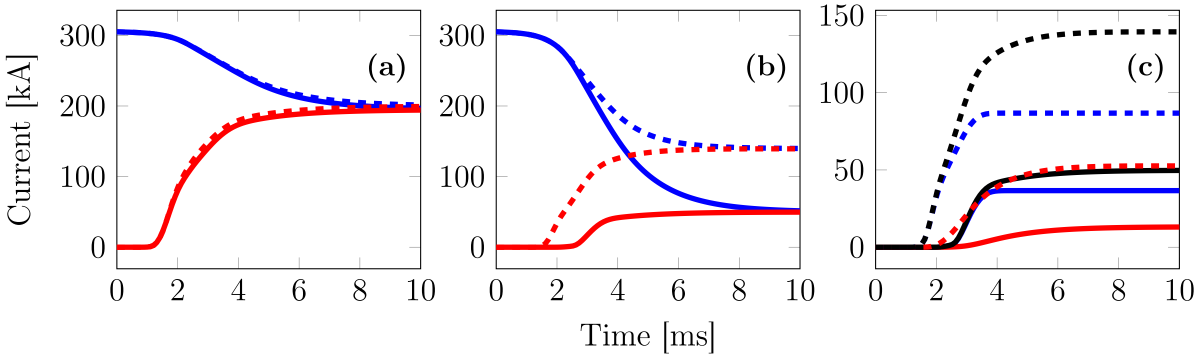


Figure 4.1: a) Runaway current (red) and total current (blue) for a fully ionised plasma. b) Runaway current (red) and total current (blue) with an argon impurity density of half the deuterium density. c) Dreicer current (blue) and avalanche current (red) with an argon impurity density of half the deuterium density. Solid lines indicate results using neural network, dashed lines using the original Dreicer formula.

[1] These kinds of CODE simulations can take on the order of 10^{-1} s to run, while the neural network takes 10^{-6} s.

[2] Until the thermal quench is close to finished the current is approximately constant. Since the initial current is ohmic, it is given by $J_\Omega = \sigma E \propto T^{3/2}E$, which, for a constant J_Ω gives $E \propto T^{-3/2}$. Dreicer generation is very sensitive to E/E_D, where $E_\text{D} \propto T^{-1}$. This gives that $E/E_\text{D} \propto T^{-1/2}$ so that the Dreicer generation will be small compared to its maximum before the end of thermal quench.

As a benchmark, a simulation without any partially ionised impurities was performed, where the neural network and the formula in Eq. (3.1) should agree well.[3] The temporal evolution of the current for such a simulation is shown in Fig. 4.1a. The results obtained with the neural network agree well with the results obtained with Eq. (3.1). In Fig. 4.1c, the current evolution from a simulation with an argon impurity density of half the deuterium background density is shown. Figure 4.1b shows the runaway current generated by the Dreicer and avalanche mechanisms separately (hot-tail generation is negligible). The maximum initial temperature is 1.3 keV, meaning that the temperature will be lower than 100 eV at all radii after a time $\ln(1290/90)t_0 \approx 2.7t_0$. In all simulations, no significant runaway current is generated within the first 1.5 ms (corresponding to $3t_0$), confirming the validity of the approximation.

In the case with an argon density present, the neural network gives less than half the Dreicer current obtained with Eq. (3.1). The lower Dreicer current also results in a lower avalanche current so that the total runaway current is about a third of the runaway current without the neural network. In this case, the Dreicer current is large enough to have a significant influence on the electric field. This means that a change in the Dreicer current will be counteracted by a change in the electric field which dampens the change in the Dreicer current. Therefore, it is possible that the effect can be even more significant in situations where the Dreicer current has a smaller effect on the electric field.

4.2 Avalanche Mechanism Model with Impurities

As described in Section 2.5.2, once an initial runaway electron population is established, this population is amplified by the avalanche effect. How this mechanism is affected by the presence of partially ionised impurities has been addressed in [23]–[26]. In this section, the avalanche model in GO is improved to take effects of partially ionised impurities into account, and the improvements are investigated in ITER-like scenarios.

An increase in the electron density increases both the electron drag force and the number of target particles for the avalanche mechanism, but the balance between these two effects is different for free and bound electrons [26]. In a fully ionised plasma, these effects cancel each other at high electric fields, but when the electric field becomes lower, the increased drag force causes a net decrease in the avalanche runaway generation.[4] As can be seen in Eq. (3.2), the avalanche growth rate also decreases as a function of the effective ion charge. According to previous calculations, it therefore seemed that a mitigation system based on gas injections would suppress the avalanche growth rate [27]. If bound electrons are included, however, these effects no longer cancel each other at high electric fields. The partially ionised impurities make the avalanche growth rate increase stronger with the electric field than in a fully ionised plasma, as shown in Fig. 4.2. For low electric fields, the avalanche growth rate is lower than in a fully ionised plasma, but for electric fields typical for a tokamak disruption, the avalanche runaway generation is significantly increased. Such an increase in the avalanche growth rate can have a dramatic effect on the final runaway current.

[3] They are not expected to agree exactly since Eq. (3.1) is not an exact steady state solution.

[4] More specifically, the avalanche growth rate is independent of the electron density in a fully ionised plasma as long as the electric field is much larger than the critical electric field E_c given in equation Eq. (2.7)

4.2.1 Implementation and Simulation Result

To investigate the effect of the recently derived avalanche growth rate, the results from [26] were implemented in GO. The updated avalanche growth rate is given by

$$\Gamma := \frac{\dot{n}_{\text{run}}}{n_{\text{run}}} = \frac{e}{m_{\text{e}} c \ln \Lambda_{\text{c}}} \frac{n_{\text{e}}^{\text{tot}}}{n_{\text{e}}^{\text{free}}} \frac{E_{||} - E_{\text{c}}^{\text{eff}}}{\sqrt{4 + \overline{\nu}_{\text{S}}(p_\star)\overline{\nu}_{\text{D}}(p_\star)}}. \tag{4.1}$$

Here, $\ln \Lambda^{\text{c}} \approx 14.6 + 0.5 \ln (T_{\text{eV}}/n_{\text{e}20})$ is the relativistic Coulomb logarithm where T_{eV} is the electron temperature in eV and $n_{\text{e}20}$ is the free electron density in units of $10^{20}\,\text{m}^{-3}$. The effective critical electric field $E_{\text{c}}^{\text{eff}}$ is a more accurate calculation of the lowest electric field for which runaways can be created, calculated according to [24]. In contrast to the expression in Eq. (2.7), the effect of partially ionised impurities and energy losses due to bremsstrahlung and synchrotron radiation are taken into account here. Finally, $\overline{\nu}_{\text{S}}$ and $\overline{\nu}_{\text{D}}$ are the electron slowing down frequency and deflection frequency, respectively. These are normalised to $1/\tau_{\text{c}}$, where $\tau_{\text{c}} = 4\pi\varepsilon_0^2 m_{\text{e}}^2 c^3/(n_{\text{e}} e^4 \ln \Lambda^{\text{c}})$ is the relativistic electron collision time with the energy-dependent Coulomb logarithm. The momentum $p_\star$ is a representative momentum at which $\overline{\nu}_{\text{S}}$ and $\overline{\nu}_{\text{D}}$ are calculated, which is defined as $p_\star = \sqrt[4]{\overline{\nu}_{\text{S}}(p_\star)\overline{\nu}_{\text{D}}(p_\star)}/\sqrt{\max(E_{||}, E_{\text{c}}^{\text{eff}})/E_{\text{c}}}$. Since the frequencies $\overline{\nu}_{\text{S}}$ and $\overline{\nu}_{\text{D}}$ are rather involved functions of momentum and $p_\star$ is only defined implicitly, $p_\star$ is calculated numerically.

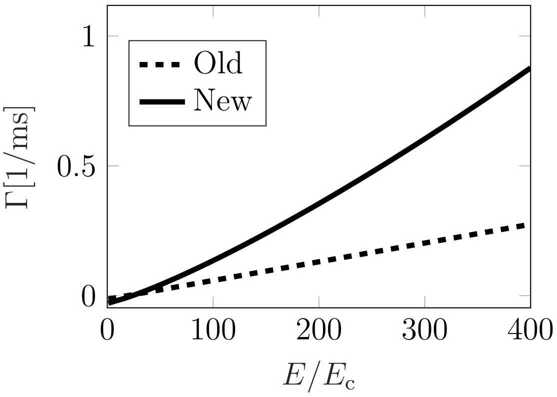


Figure 4.2: Avalanche growth rate, Γ, as a function of E/E_{c}. Calculated both widh the old formula Eq. (3.2), as well as the updated avalanche growth rate given by Eq. (4.1). The range of E/E_{c} is representative for ITER disruption simulations with GO. A density of $10^{20}\,\text{m}^{-3}$ is assumed for both deuterium and argon respectively, as well as a temperature of 5 eV.

The implemented avalanche calculations were tested in ITER-like scenarios. Details of the parameters for ITER are given in Appendix A. The magnitude of the runaway electron seed varies depending on e.g. how short the thermal quench is, which determines the efficiency of the hot tail mechanism. Therefore, the effect of the avalanche mechanism is isolated by making the seed a free parameter to be scanned over. The heat transport during a disruption is also not completely understood [7]. For simplicity, the studied scenarios are therefore chosen to have an instantaneous drop in temperature and a flat, constant post-disruption temperature profile. This is justified in this case, since the avalanche takes place over a much longer time scale than the thermal quench. The seed is given as input, and thus not affected by the thermal development. Losses of runaways are neglected, corresponding to a worst-case scenario.

The impurity density profile is assumed to be similar to the background plasma density profile, and the distribution of density over the different ionisation states is estimated by assuming equilibrium between ionisation and recombination[5] rates for all charge states.

[5] Recombination is the reverse of the ionisation process.

These rates are calculated by interpolating data from the ADAS database [28]. It can be shown that this equilibrium is established over a timescale of less than 10 ms, which is significantly shorter than the current quench times relevant for ITER that should be between 50 ms and 150 ms [29].

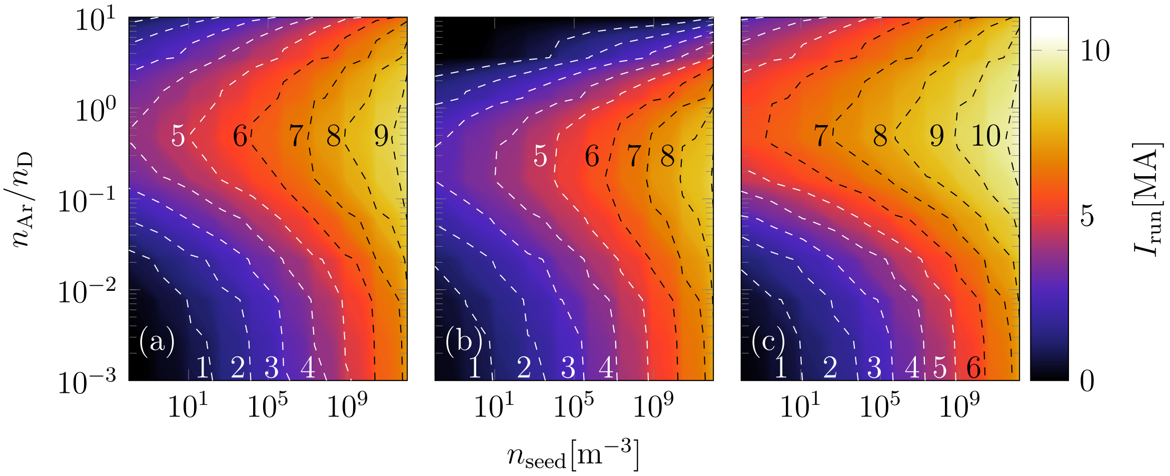


Figure 4.3: Final runaway current as a function of n_{seed} and $n_{\mathrm{Ar}}/n_{\mathrm{D}}$. a) $T = 5\,\mathrm{eV}$ and $n_{\mathrm{D}} = n_{\mathrm{D},0}$. b) $T = 5\,\mathrm{eV}$ and $n_{\mathrm{D}} = 11 n_{\mathrm{D},0}$. c) $T = 10\,\mathrm{eV}$ and $n_{\mathrm{D}} = n_{\mathrm{D},0}$.

Figure 4.3a shows the final runaway current as a function of the seed density n_{seed}, the ratio of argon impurity density n_{Ar} and deuterium density n_{D}. The temperature is chosen to a typical post-disruption value of 5 eV. Figure 4.3b shows a similar scan with an extra deuterium injection $n_{\mathrm{D,inj}} = 10 n_{\mathrm{D},0}$ where $n_{\mathrm{D},0}$ is the initial deuterium density, so that the total deuterium density is $n_{\mathrm{D}} = 11 n_{\mathrm{D},0}$. A similar scenario with the temperature changed to 10 eV is shown in Fig. 4.3c. In the investigated cases, the final runaway current is approximately logarithmically sensitive to the seed when the seed is big enough. The reason for this weak dependence is that when the increase in the runaway current becomes comparable to the decay rate of the ohmic current, the electric field is reduced. This, in turn, reduces the avalanche growth rate. The dependence on $n_{\mathrm{Ar}}/n_{\mathrm{D}}$ is quite weak for $n_{\mathrm{Ar}}/n_{\mathrm{D}} \lesssim 10^{-2}$ and for $n_{\mathrm{Ar}}/n_{\mathrm{D}} \gtrsim 10^{-1}$, but is stronger in between. The region $10^{-2} < n_{\mathrm{Ar}}/n_{\mathrm{D}} < 10^{-1}$ roughly corresponds to where the number of electrons contributed by the argon (including bound electrons) is comparable to the number of electrons contributed by the deuterium. This is the region where $n_{\mathrm{e}}^{\mathrm{tot}}/n_{\mathrm{e}}^{\mathrm{free}}$ changes significantly. For the highest $n_{\mathrm{Ar}}/n_{\mathrm{D}}$ included in the scan, a decrease in the runaway current as a function of $n_{\mathrm{Ar}}/n_{\mathrm{D}}$ was seen. This happens because the effective critical electric field becomes larger and terminates the avalanche at an earlier stage of the current quench.

In the worst-case scenario, a seed density of $10^{-2}\,\mathrm{m}^{-3}$ is enough to result in a final runaway current of about 6 MA. Such a seed density corresponds to about 10 runaway electrons in the whole volume of ITER. If an argon impurity contributes significantly to the electron density, this scan indicates that the density must be many times larger than the initial deuterium density or some substantial loss mechanism of runaway electrons must be present to avoid an unacceptably high runaway current. In reality, it is quite problematic to assimilate such high impurity densities in the plasma sufficiently fast [7].

For lower impurity densities, the final runaway current becomes larger than the desired maximal runaway current of 2 MA when the impurity density becomes larger than about $n_\mathrm{D}/30$ even for a seed density of $10^{-2}\,\mathrm{m}^{-3}$ in all three investigated scenarios. Therefore, the results suggest that impurity densities higher than $n_\mathrm{D}/30$ should be avoided.

4.2.2 Relevance for Disruption Mitigation

Indications from the results concerning the possibility to mitigate disruptions with massive gas injections are now discussed. To be able to do this, the dependence of the timescales for the current quench and thermal energy loss by radiation on the amount of injected argon and deuterium is of interest. Calculating these timescales accurately is a very complicated problem, but they can still be estimated with a rather simple model. These estimates are then discussed in the context of the limitations on the injected impurity densities set by the runaway avalanche.

First, the radiation loss timescale is estimated. The total initial thermal energy contributed by both the electrons and the background plasma is given by $E_\mathrm{th} = 3n_{\mathrm{e},0}T_0$, with $n_{\mathrm{e},0}$ being the initial electron density and T_0 the initial plasma temperature. If line radiation is the dominant energy loss mechanism, as is desirable from a disruption mitigation perspective, the loss effect per unit volume is given by $P_\mathrm{rad} = \sum n_i n_\mathrm{e} L_i(n_\mathrm{e}, T_\mathrm{e})$. The summation is over all ionisation states of all ion types and the coefficient L_i is calculated by interpolating data from the ADAS database [28]. By assuming the same equilibrium as in Section 4.2.1, the distribution of density over different ionisation states is calculated. A representative temperature, at which L_i and the distribution of ionisation states are calculated, is taken to be the temperature obtained if the initial thermal energy is distributed equally over all species present in the plasma after the injection. The radiation loss timescale is then calculated as $t_\mathrm{rad} = E_\mathrm{th}/P_\mathrm{rad}$. A contour plot of t_rad as a function of injected argon and deuterium density is shown in Fig. 4.4a.

The current quench time scale is now estimated, assuming a cylindrical plasma with constant conductivity. For simplicity, the effect of the runaway current on the electric field is also neglected. In this case it can be shown that the current will decay exponentially such that it decreases by a factor e^{-1} over a timescale $t_\mathrm{cq} = \sigma\mu_0 a^2/x_1^2$ [26]. Here, $x_1 \approx 2.4$ is the first zero of zeroth Bessel function of the first kind. Since $\sigma \propto T^{3/2}$ as described in Section 2.4.4, this timescale is mostly determined by the post-disruption temperature. This temperature is estimated by assuming equilibrium between line radiation and ohmic heating. The ohmic heating effect per unit volume is given by $P_\mathrm{ohm} = J^2/\sigma$. The current density used when calculating the ohmic heating effect is determined by dividing the total initial plasma current by the cross-section area of the tokamak. Setting P_ohm equal to P_rad calculated as earlier, solving numerically for T_2 and inserting in the expression for t_cq gives the result shown in Fig. 4.4b.

In Fig. 4.4a and Fig. 4.4b, a solid white line shows the boundary of $n_\mathrm{Ar}/n_\mathrm{D} = 1/30$ where the avalanche mechanism was observed to make an unacceptable runaway current unavoidable. In Fig. 4.4b, a white and a black solid contour line show the desired limits within which t_cq should be to get an acceptable current quench time. To get a total current quench time between 50 ms and 150 ms, t_cq should be between 22 ms and 66 ms [26], [29]. From this model, the densities required to get the current quench time within a desirable range does not seem to be very limited by the restriction caused by the avalanche

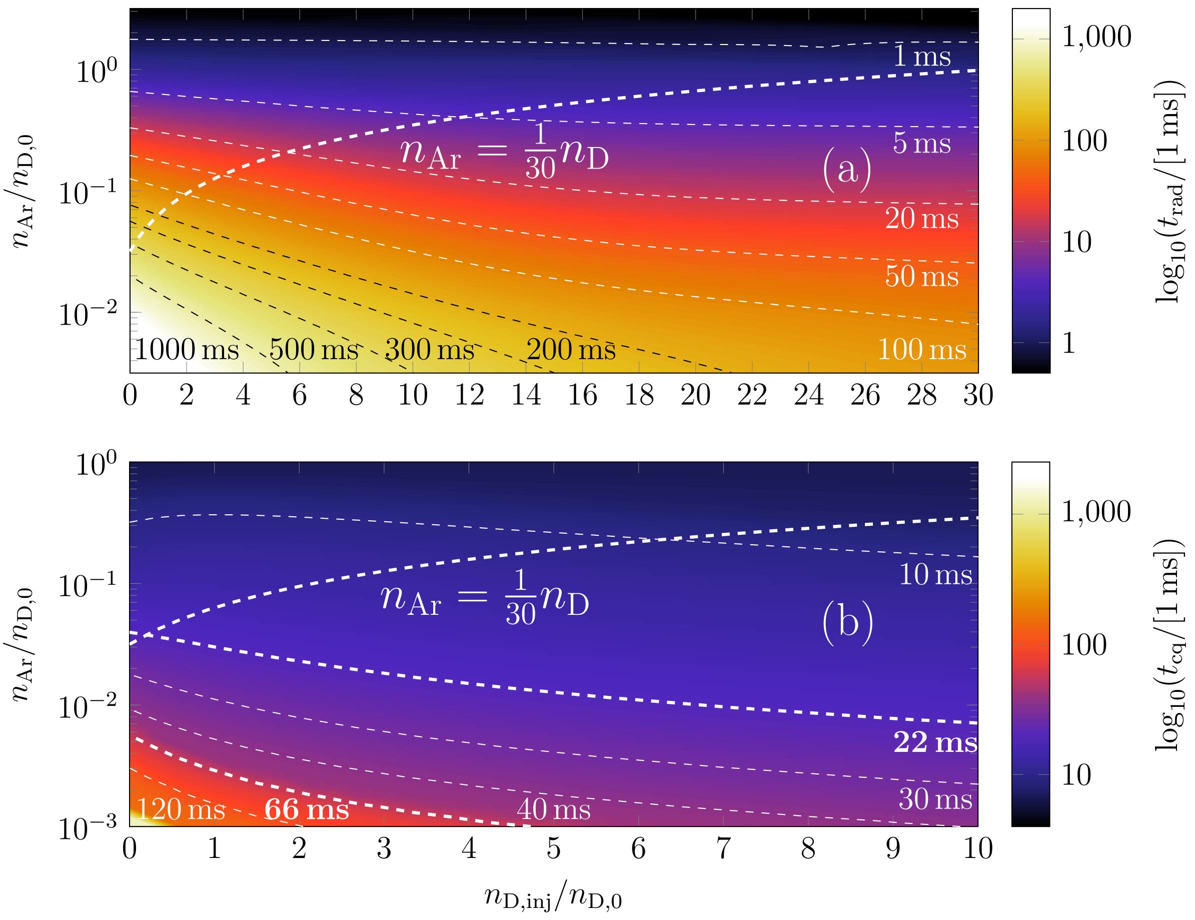


Figure 4.4: Estimate of (a) t_{rad} and (b) t_{cq} as a function of injected argon and deuterium density. A line marks the smallest argon density that was observed to result in an unacceptable runaway current. Thick lines in (b) show the limiting cases for acceptable current quench time.

mechanism. Without any deuterium injection, however, one would have to be very close to the boundary according to this model, and since this model is rather crude it might still be necessary to inject some amount of deuterium.

The time it takes to radiate away the thermal energy is much more limited. As mentioned in Section 2.6, the time from the detection of an upcoming disruption to the end of an unmitigated disruption is expected to be about 20 ms in ITER. Depending on how quickly the mitigation system will be able to start the controlled thermal quench, t_{rad} might have to be as short as a few milliseconds. To achieve this, Fig. 4.4b shows that an argon density comparable to the initial deuterium density is needed. The amount of deuterium that needs to be injected to dampen the runaway avalanche would then have to be of the order of ten times the initial deuterium density. Such an amount of injected deuterium would probably be very problematic to assimilate in the short time required [7]. What is also noteworthy is that this impurity density is about an order of magnitude larger than what is estimated to give an acceptable current quench time.

Chapter 5

Coupled Position and Momentum-Space Modelling

To properly simulate runaway generation, simulation of the full phase-space must be done. As noted before, this is computationally heavy. Therefore, we couple the momentum-space dynamics calculated in CODE, to the position space dynamics calculated in GO. First, the full method is presented in Section 5.1, later results from simulations using parameters from the TEXTOR tokamak are presented in Section 5.2. This is followed by a numerical efficiency analysis of the numerical tools in Section 5.3.

5.1 GO+CODE

GO+CODE is a hybrid fluid-kinetic solver that couples GO described in Section 3.1 with CODE described in Section 3.2. CODE is used to calculate the runaway current instead of the approximate growth rates used in GO. Each radial point in GO therefore has one separate instance of CODE which evolves the distribution function in momentum space and is used to calculate the runaway current. This makes it possible to avoid the simplification $J_{\text{run}} = n_{\text{run}} q_{\text{e}} c$. However, if the electric field falls below the critical electric field, the number of runaway electrons is zero even if there still might be a large current carried by relativistic electrons. Therefore, the current calculated by CODE that is coupled to GO is the current carried by electrons with a speed higher than a certain threshold chosen to $0.25c$. It should also be noted that even though there is a density gradient, the cross term $\boldsymbol{v} \cdot \partial f_k / \partial \boldsymbol{x}$ in Eq. (2.1) is still disregarded for simplicity.

5.2 Simulation Results

To begin with, a demonstration of the ability to calculate the full evolution of currents and electric field as well as the distribution function is shown. The illustrated case is similar to the TEXTOR tokamak, assuming an exponential temperature decay with $t_0 = 1.5\,\text{ms}$. The evolution of the temperature is shown in Fig. 5.1 for reference. It also shows the evolution of the currents and electric field. As explained in Section 2.5.1, the decrease in temperature gives rise to an increase in the electric field. This, in turn, gives rise to a runaway current, which starts to grow significantly after about 4 ms. The evolution of the distribution function in the centre of the tokamak is shown in Fig. 5.2 at 50 %, 60 %, and 70 % of the total simulation time. This corresponds to $t = 5.4\,\text{ms}$, $t = 6.6\,\text{ms}$ and $t = 7.7\,\text{ms}$ respectively, corresponding to when the runaway current has just been generated. Here, the runaway population can be seen in the form of a tail accelerating to

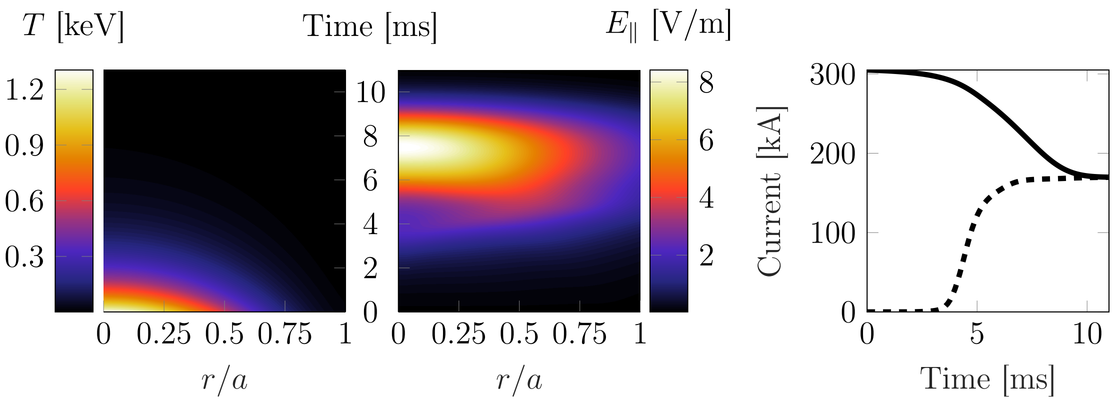


Figure 5.1: Evolution of temperature (left), electric field (middle) and current (right) from a simulation of a TEXTOR disruption with GO+CODE. The solid and dashed lines indicate total and runaway current respectively.

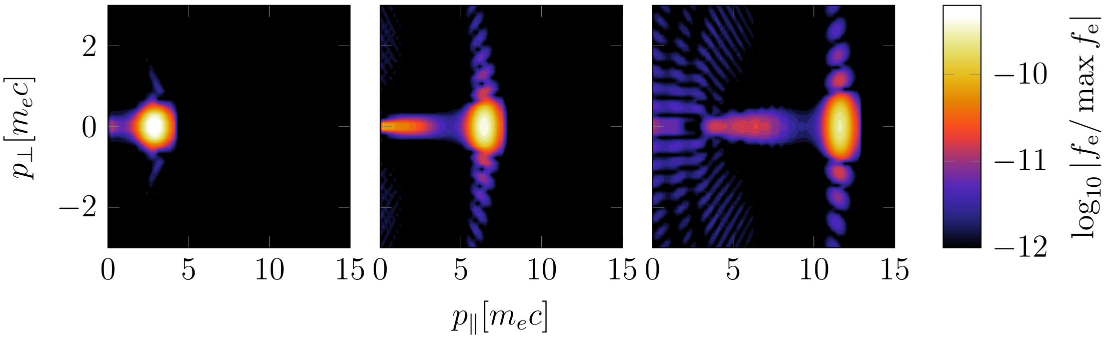


Figure 5.2: Evolution of the distribution function in the centre of the tokamak from a simulation of a TEXTOR disruption with GO+CODE. Snapshots are taken at 50 %, 60 %, and 70 % of the total simulation time, corresponding to $t = 5.4$ ms, $t = 6.6$ ms and $t = 7.7$ ms respectively. The distribution function is normalised to the maximum value of the distribution in the first time step. Note the logarithmic colour scale.

higher and higher momentum. Some oscillating features due to numerical noise can also be seen, but at several orders of magnitude lower amplitude than the main features of the distribution function.

5.2.1 Fluid Model Comparison

In the initial pilot project of creating a GO and CODE coupling [12], the runaway current was found to be somewhat larger compared to using GO alone. Since then, various improvements have been made to CODE, including an update of the collision operator, which are beyond the scope of this thesis. Because of the updates, and because the full reason for the discrepancy was not understood at the time, the difference between GO and GO+CODE was examined again.

As in [12], TEXTOR parameters are used, the details of which can be seen in Appendix A. An exponential temperature decay is assumed with a final temperature of 5 eV in the centre decreasing at a constant rate to 1 eV at the edge, and a time independent density profile. A scan over the thermal quench time[1] is performed.

The final runaway current found using GO and GO+CODE, respectively, for the investigated thermal quench times is shown in Fig. 5.3. While the current found by GO+CODE is about the same as in [12] for the shortest thermal quench times, the difference to GO quickly drops between about $t_0 = 0.5\,\text{ms}$ and $t_0 = 1.5\,\text{ms}$ to a final value of around 15 % higher than GO. For even longer thermal quench times there is no data from the previous study, but the difference seems to stay somewhat constant at 15 %.

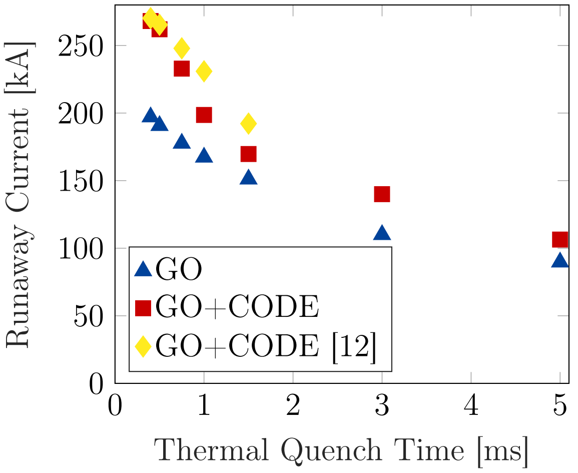


Figure 5.3: Calculated runaway current in TEXTOR assuming an exponential thermal quench using GO and GO+CODE. Results using an older version of GO+CODE, first shown in [12], are included for reference.

The discrepancies would be expected at shorter time scales, as this is where fluid models, which uses steady-state growth rate models, might not be valid. Both hot-tail and steady state approximations for the Dreicer growth rate are explanations of the discrepancy in the total runaway current. In Fig. 5.4 the differences in the runaway current between kinetic and fluid models can be seen. As seen in the kinetic model (solid lines), the runaway generation starts earlier than in the fluid model (dashed lines). As explained in Section 4.1, the Dreicer generation starts at the end of the thermal quench whereas hot-tail generation occurs during the thermal quench. The earlier start time of the generation therefore implies that the effect from hot-tail is too small in the analytical model implemented in GO. This is also observed in [11], where the analytical model was compared with numerical simulations for a constant electric field without spatial dependence, and a discrepancy of approximately an order of magnitude was found. In this case, the electric field is calculated self-consistently with the development of the plasma current with radial spatial dependency. The larger hot-tail leads to more seed electrons and as shown in Fig. 4.3 the seed will be multiplied through the avalanche mechanism and even small differences can have a substantial effect on the final runaway current.

The effect of hot-tail should be lower for slower thermal quench times. This can be seen in Fig. 5.4 where the runaway generation starts somewhat faster in kinetic models than fluid models for the 0.5 ms graph whereas they start at the same time in the 5 ms graph. It also seems that a discrepancy is created even after the thermal quench, which implies a difference in either Dreicer generation, avalanche generation or both. To examine which of hot-tail and other growth rates affects the final results the most, further studies should be done where the different growth rates are isolated in GO+CODE. This identification can determine where improvements in present fluid models should be made. These improvements are important in order to obtain accurate results from the fluid models, which run substantially faster than kinetic models.

The persisting difference at longer thermal quench time is unexpected. One explanation could be that hot-tail is still important at a thermal quench time of 5 ms, or that

[1] The thermal quench time, t_0, is used as defined in Section 3.1

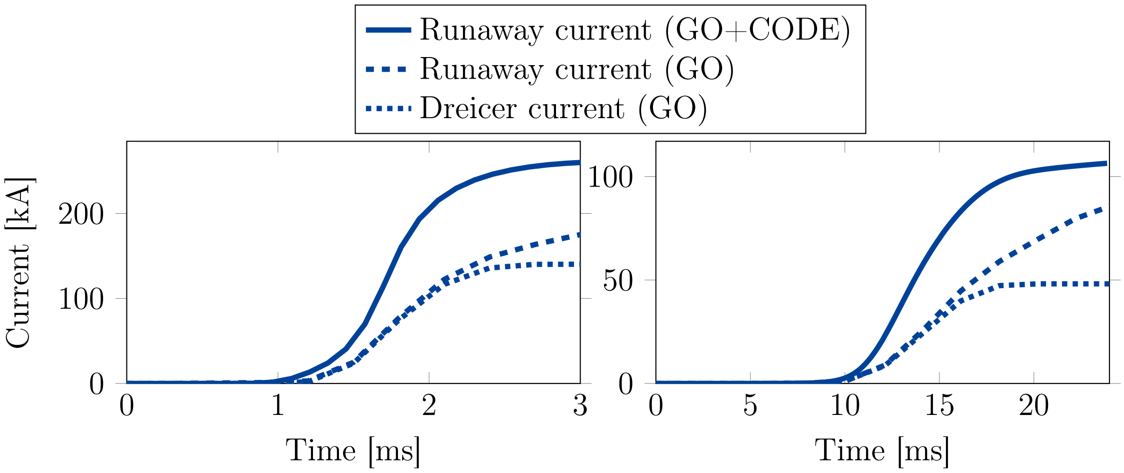


Figure 5.4: Time evolution of runaway current during thermal quenches of 0.5 ms (left) and 5 ms (right) using GO and GO+CODE respectively. The dotted line divides the runaway current calculated by GO into Dreicer and avalanche generation. The hot-tail portion of the current in GO was below 1 % for both simulations and is therefore not illustrated.

other approximations in growth rates in GO is still not valid at 5 ms timescale. A more likely explanation is the chosen boundary for fast electrons in kinetic modelling. If a too high threshold is used, an increased electric field could be created due to the electric field not reacting to the accelerated electrons until the electrons reach the threshold. Then, electrons are accelerated without dampening the electric field, increasing the runaway generation. This can explain the constant difference at 15 %. By lowering the threshold the discrepancy could possibly be lowered. However, too low of a threshold could mean that a large portion of the ohmic current would be double-counted since there is no distinction between ohmic current and other currents in kinetic modelling.

Therefore, to avoid this problem, the electric field should be modelled with the total current calculated from kinetic simulations. In practice, this would require solving Eq. (2.8) for the electric field. With the current's dependence of the electric field written out explicitly, this reads

$$\nabla^2 \boldsymbol{E} = \frac{\partial}{\partial t}(\mu_0 \boldsymbol{J}(\boldsymbol{E})).$$

Observing that the only time derivative present is that of the current, it can be concluded that to solve for the electric field in time, either the inverse of the current operator needs to be estimated, or some time expansion of the Laplacian operator must be made. Since the separation between ohmic and fast current allows for the numerical stable Crank-Nicolson scheme to be used, total current modelling was not performed in this study but should instead be addressed in further studies.

5.3 Resolution and Convergence in GO+CODE

The resolution of numerical simulations affects both the run time and the accuracy of the results. Simulations with high resolution take a long time to run but give the best results. Conversely, simulations with low resolution can run quickly, but might not give accurate results. Some of the simulations presented in Fig. 5.3 took up to several days to run. It therefore becomes necessary to perform benchmarks and convergence tests with respect to resolution. These benchmarks give information about how well the resolution parameters were chosen for a particular physical scenario. Particular emphasis was placed on the importance of resolution parameters in CODE when used together with GO.

5.3.1 Structure of benchmarks

The benchmarks were designed as parameter scans. A physical scenario was simulated with high numerical resolution, where the run time and result of the simulation were recorded. The resolution was then lowered and the process repeated. The same scenario was simulated several times with continuously decreasing resolution. After a specified number of repetitions, the result of each simulation was compared to the simulation with the highest resolution. However, at the highest resolution it is important to manually confirm that the accuracy barely increases with increasing resolution. This indicates that the solution has converged. The relative error in the result of each simulation could therefore be described as a function of numerical resolution. The relative error was computed for the total current density and the runaway current density.

Several resolution parameters were identified that could affect the results and the variables N_ξ, N_y, and Δt were chosen for further study. These three parameters were suspected to have a major impact on both accuracy and run time. Two additional parameters were instead suspected to only affect simulation accuracy, leaving the run time unaffected. These are explained in more detail in Appendix B.3.

The first benchmarks were performed on CODE alone, corresponding to a single GO time step in GO+CODE. This was done in order to get an intuitive understanding of what the different resolution parameters do and how they affect the outcome. The results of a GO+CODE simulation of the TEXTOR tokamak were then used to study the choice of resolution made by GO+CODE. Emphasis in this benchmark was placed on ensuring the CODE results converge before continuing with the next step in the GO simulation. A secondary objective was minimising run time.

Since the GO+CODE simulations depend on a large number of previous time steps, it was necessary to study the effects of error propagation. A benchmark was made to study a single radial grid point throughout the entire time-evolution. This test shows how errors in the beginning can affect the end result. The findings of the benchmarks were then used to write a function that automatically chooses resolution parameters for CODE. This function was then implemented in GO+CODE and the TEXTOR simulation was performed again. Finally, differences compared to previous simulations of the same scenario were analysed.

5.3.2 Benchmark and Convergence Tests for CODE

The parameters N_ξ and N_y were the first parameters to be studied in the CODE part of GO+CODE. It was found that the relative error decreases continuously with increasing N_y. The lowest required value for the resolution parameter N_y in a typical CODE scenario is generally a few hundred to several hundred. Increasing the resolution further leads to diminishing accuracy improvements. It was also found that the parameter N_ξ only seems to affect the resolution up to a certain point, after which accuracy barely increases. Sufficient accuracy was often achieved with a value of N_ξ of roughly 6 to 8. The two resolution parameters do not seem to interact noticeably. Additionally, the simulation run time was found to be approximately proportional to $(N_\xi N_y)^2$. This dependence implies that a small decrease in resolution can lead to a significant reduction in simulation run time. A more detailed description of this benchmark can be found in Appendix B.1.

The third resolution parameter studied was the time step Δt. Instead of varying Δt directly, the number of time steps, $t_{\text{max}}/\Delta t$, was studied, where t_{max} is the maximum time of the scenario. It was found that the same TEXTOR scenario that was studied earlier required more than 100 time steps in the chosen time frame in order to have a relative error of less than 1 %. This is equivalent of having a Δt value of 5 or less thermal collision times. It was also found that the simulation run time scaled linearly with the number of time steps, with a constant offset. This indicates that there is some room for optimisation of the simulation run time, but not as much as the parameters N_ξ and N_y have. For additional details about this benchmark, refer to Appendix B.2.

5.3.3 Benchmark and Convergence Tests for GO+CODE

Benchmarks on the CODE part of GO+CODE can indicate how GO+CODE will perform. However, the GO part of GO+CODE must be included in the benchmarks in order to study a more useful scenario. The benchmarks in the previous section studied the CODE simulation in a single GO time step. The following section studies how errors propagate through the simulation when several GO time steps, with their corresponding CODE simulations, are taken.

It was found that error propagation is not a significant issue in GO+CODE as long as sufficient resolution is employed in CODE. Sufficient resolution in GO is, however, also required. Increasing resolution further in CODE will not affect the results noticeably. The current version of GO+CODE increases the values of N_ξ and N_y continuously as time progresses to compensate for the grid extensions that are performed when the particle momentum increases. However, the resolution turned out to become unnecessarily large in the later parts of the simulation. A few exceptions were nonetheless identified. It appears that N_y should be slightly greater in the beginning, but can be significantly reduced in the end compared to previous versions of GO+CODE. N_ξ can most of the time be chosen to take small values, i.e. less than 10, in regions with a weak electric field. This value must be increased when stronger electric fields are considered.

The parameters N_ξ and N_y controlled a large portion of the run time in GO+CODE. The parameter Δt did, however, still have a significant effect on the accuracy. GO+CODE used only a single time step for CODE in the first GO steps. A significant difference of more than 8 % was observed in the final current density when the GO+CODE value of Δt was compared to similar simulations performed with a greater number of time steps, i.e. smaller Δt. Since smaller time steps increase resolution, it is probable that the new parameters give a more accurate result.

A few additional observations were made regarding the scaling of the resolution requirements. It was found that the required value of N_ξ scaled approximately with E. Furthermore, it was found that the resolution requirement for N_y seemed to scale approximately with the quantity E/E_{D}. The value of Δt did not seem to require scaling and was chosen to give good overall convergence in all parts of the simulation. The value of t_{max} in CODE is equal to the GO time step.

For simplicity, the selection algorithm for N_ξ and N_y was chosen to be a linear function of E and E/E_{D}, respectively. The constant offsets were chosen in order to guarantee that a minimum resolution was employed. Values for the coefficients were found by correlating the required resolution to the variables E and E/E_{D} at certain points in the simulation. The required resolution was found by manually optimising the resolution in different areas of a GO+CODE simulation. The values were then adjusted to achieve good simulation run times without sacrificing accuracy. Finally, the coefficient values were increased to safeguard against poor resolution. The final model can then be described by

$$N_\xi = \mathrm{round}\left(4.5\,\frac{E}{1\,\mathrm{V/m}} + 10\right), \tag{5.1}$$

$$N_y = \mathrm{round}\left(7000\,\frac{E}{E_{\mathrm{D}}} + 300\right), \tag{5.2}$$

$$\Delta t = \frac{t_{\mathrm{max}}}{100}. \tag{5.3}$$

Equations (5.1) and (5.2) feature rounding in order to only provide integer values for the resolution. Note that E/E_{D} comes from the previous CODE result in the same radial grid point. The other values, E and t_{max}, are taken from the parameters that are about to be used in the current simulation.

This choice of parameters was implemented in GO+CODE. The increase in simulation run time due to the increased resolution in the beginning was more than offset by the reduction of resolution in the end. This led to an overall decrease in the simulation time of the investigated TEXTOR scenario from slightly over six and a half hours to just under 30 minutes, a factor 14 decrease. Furthermore, the reduction in resolution led to a reduction in peak memory usage, as reported by GO+CODE, from 5341 MiB to 989 MiB. This corresponds to a reduction in memory usage by a factor 5. The benchmarks were performed with MATLAB R2018b on an Intel i5-8400 CPU running at 3.8 GHz with 16 GiB RAM.

The new algorithm for choosing resolution parameters include increased resolution in parts of the simulation that previously suffered from poor resolution. This includes increased N_y and number of time steps in the early parts of the simulation. It might therefore be beneficial to repeat previously performed GO+CODE simulations to verify if poor resolution might have given erroneous results. Note that the algorithm was developed using the TEXTOR tokamak as a reference. Other tokamaks or scenarios that differ significantly from the studied scenario might need slightly different models.

The increased numerical efficiency that follows from the new selection algorithm means that a given simulation can be completed more quickly and with less computational resources than before. It also means that more complex scenarios can be simulated on current computational hardware. The new selection algorithm takes an important step towards the goal of simulating the ITER tokamak in GO+CODE. Simulating ITER scenarios require significantly more computational power but the new selection algorithm could, when properly adjusted to ITER, decrease those demands considerably.

Chapter 6

Conclusion

In this thesis, important steps have been taken towards accurate modelling of the runaway dynamics in disruptions by updating and testing the numerical tools to include the effect of partially ionised impurities in the runaway generation rate. This includes both an implementation of a neural network to address Dreicer generation and an improved model for the avalanche generation that includes the effect of impurities. This is relevant since disruption mitigation in tokamaks can be done by injecting impurities in the plasma.

The partially ionised impurities were observed to have a great impact on both Dreicer and avalanche runaway generation. Their presence generally decreases the Dreicer runaway generation, but a drastic increase in the avalanche generation was observed. Results from avalanche generation indicate that even an extremely small population of initial runaways could lead to a runaway current of extreme magnitude in the presence of certain densities of partially ionised impurities. Therefore, in most cases, the final result will be an increase in the number of runaways.

Although more accurate models were implemented in the self-consistent fluid model, their validity on shorter timescales, characteristic of a sudden cooling of the plasma in a tokamak disruption, could be questioned. To ensure that present fluid models are valid and to find where discrepancies arise, a kinetic model was used. This model was compared with fluid modelling in a fully ionised plasma. The results of kinetic modelling of the runaway dynamics did not completely agree with the results of fluid modelling, but they did seem to converge towards each other with a constant offset in the case of slow temperature decay. The convergence is to be expected as fluid models handle effects on longer time scales, while kinetic models also cover short time scale effects. A simplified version of the kinetic modelling was used, which could be the cause of the discrepancy to the fluid model. Investigations of what exactly causes the discrepancy are necessary to develop accurate models. These are outside the scope of this thesis and are left for further studies.

The numerical efficiency of the kinetic model has also been investigated. Through optimisation of internal numerical parameters a drastic decrease in run time, without compromising accuracy, has been achieved. The decreased run time means that more elaborate models could be implemented to further increase the accuracy of simulations.

All of the above improvements contribute to the accurate predictions needed when designing a disruption mitigation system in future tokamaks. To be able to mitigate disruptions in a way where the tokamak takes close to zero damage is a great challenge to overcome and would be a crucial step towards a fusion energy power plant. A successful fusion power plant would, in turn, be a giant leap towards solving the ever-growing demand for an environmentally friendly energy source.

Bibliography

[1] J. P. Freidberg, *Plasma physics and fusion energy*, 1st ed. Cambridge, UK: Cambridge university press, 2008.

[2] ITER organization, `http://www.iter.org`.

[3] F. Chen, *An indispensable truth: how fusion power can save the planet.* New York, NY, USA: Springer Science & Business Media, 2011.

[4] J. Wesson, *Tokamaks*, 4th ed. Oxford, UK: Oxford university press, 2011.

[5] T. Hender *et al.*, "Chapter 3: MHD stability, operational limits and disruptions", *Nuclear Fusion*, vol. 47, no. 6, S128–S202, Jun. 2007. DOI: `10.1088/0029-5515/47/6/S03`.

[6] M. Rosenbluth and S. Putvinski, "Theory for avalanche of runaway electrons in tokamaks", *Nuclear Fusion*, vol. 37, no. 10, pp. 1355–1362, Oct. 1997.

[7] E. Hollmann *et al.*, "Status of research toward the iter disruption mitigation system", *Physics of Plasmas*, vol. 22, no. 2, pp. 021802-1–021802-16, Feb. 2015. DOI: `10.1063/1.4901251`.

[8] H. Smith, P. Helander, L.-G. Eriksson, D. Anderson, M. Lisak, and F. Andersson, "Runaway electrons and the evolution of the plasma current in tokamak disruptions", *Physics of Plasmas*, vol. 13, no. 10, pp. 102502-1–102502-13, Oct. 2006. DOI: `10.1063/1.2358110`.

[9] T. Fehér, H. M. Smith, T. Fülöp, and K. Gál, "Simulation of runaway electron generation during plasma shutdown by impurity injection in ITER", *Plasma Physics and Controlled Fusion*, vol. 53, no. 3, p. 035 014, Feb. 2011. DOI: `10.1088/0741-3335/53/3/035014`.

[10] M. Landreman, A. Stahl, and T. Fülöp, "Numerical calculation of the runaway electron distribution function and associated synchrotron emission", *Computer Physics Communications*, vol. 185, no. 3, pp. 847–855, Mar. 2014. DOI: `10.1016/j.cpc.2013.12.004`.

[11] A. Stahl, O. Embréus, G. Papp, M. Landreman, and T. Fülöp, "Kinetic modelling of runaway electrons in dynamic scenarios", *Nuclear Fusion*, vol. 56, no. 11, p. 112 009, Nov. 2016. DOI: `10.1088/0029-5515/56/11/112009`.

[12] G. Papp, A. Stahl, M. Drevlak, T. Fulop, P. W. Lauber, and G. Pokol, "Towards self-consistent runaway electron modeling", in *42nd EPS Conference on Plasma Physics, Lisbon, Portugal, ECA*, vol. 39, 2015, p. 1173.

[13] F. F. Chen, *Introduction to Plasma Physics and Controlled Fusion*, 1st ed. New York, NY, USA: Springer International Publishing, 1974.

[14] P. Helander and D. Sigmar, *Collisional Transport in Magnetized Plasmas.* Cambridge, UK: Cambridge University Press, 2005.

[15] I. Newton, *Philosophiæ Naturalis Principia Mathematica*, 1st ed. London, UK, 1687.

[16] L. Spitzer Jr and R. Härm, "Transport phenomena in a completely ionized gas", *Physical Review*, vol. 89, no. 5, pp. 977–981, Mar. 1953. DOI: `10.1103/PhysRev.89.977`.

[17] J. Connor and R. Hastie, "Relativistic limitations on runaway electrons", *Nuclear Fusion*, vol. 15, no. 3, pp. 415–424, Jun. 1975.

[18] H. Dreicer, "Electron and ion runaway in a fully ionized gas I", *Physical Review*, vol. 115, pp. 238–249, 2 Jul. 1959. DOI: `10.1103/PhysRev.115.238`.

[19] P. Helander, H. M. Smith, T. Fülöp, and L. G. Eriksson, "Electron kinetics in a cooling plasma", *Physics of Plasmas*, vol. 11, no. 12, pp. 5704–5709, 2004. DOI: `10.1063/1.1812759`.

[20] T. Fehér, "Simulation of runaway electron generation during plasma shutdown by impurity injection", Licentiate Thesis, Chalmers University of Technology, 2011.

[21] H. M. Smith and E. Verwichte, "Hot tail runaway electron generation in tokamak disruptions", *Physics of Plasmas*, vol. 15, no. 7, p. 072 502, Jul. 2008. DOI: `10.1063/1.2949692`.

[22] S. Haykin, *Neural networks: a comprehensive foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1994.

[23] L. Hesslow *et al.*, "Effect of partially screened nuclei on fast-electron dynamics", *Phys. Rev. Lett.*, vol. 118, pp. 255001-1–255001-5, 25 Jun. 2017. DOI: `10.1103/PhysRevLett.118.255001`.

[24] L. Hesslow, O. Embréus, G. J. Wilkie, G. Papp, and T. Fülöp, "Effect of partially ionized impurities and radiation on the effective critical electric field for runaway generation", *Plasma Physics and Controlled Fusion*, vol. 60, no. 7, p. 074 010, Jun. 2018. DOI: `10.1088/1361-6587/aac33e`.

[25] L. Hesslow *et al.*, "Generalized collision operator for fast electrons interacting with partially ionized impurities", *Journal of Plasma Physics*, vol. 84, no. 6, 2018. DOI: `10.1017/S0022377818001113`.

[26] L. Hesslow, O. Embréus, O. Vallhagen, and T. Fülöp, "Influence of massive material injection on avalanche runaway generation during tokamak disruptions", Submitted for publication in Nuclear Fusion.

[27] J. R. Martín-Solís, A. Loarte, and M. Lehnen, "Runaway electron dynamics in tokamak plasmas with high impurity content", *Physics of Plasmas*, vol. 22, pp. 092512-1–092512-13, Sep. 2015. DOI: `10.1063/1.4931166`.

[28] H. P. Summers, *The ADAS user manual, version 2.6*, http://www.adas.ac.uk, 2004.

[29] J. R. Martín-Solís, A. Loarte, and M. Lehnen, "Formation and termination of runaway beams in ITER disruptions", *Nuclear Fusion*, vol. 57, no. 6, p. 066 025, Apr. 2017. DOI: `10.1088/1741-4326/aa6939`.

Appendix A

Parameters for selected tokamaks

Dimensions and plasma parameters used for the simulations of different tokamaks are given below. The parameters are the same as in previous publications using GO [12], [20].

Table A.1: Tokamak and plasma parameters for tokamaks used in simulations.

Tokamak or plasma parameter	ITER	TEXTOR
Tokamak major radius, R [m]	6.2	1.8
Plasma minor radius, a [m]	2	0.4
Tokamak conducting wall radius[1], b [m]	2.15	0.6
Initial plasma current I_0 [MA]	15	0.305
Magnetic field, B [T]	5.3	2.1
Coulomb logarithm, $\ln\Lambda$	17	16
Density	See Fig. A.1a	See Fig. A.2a
Initial temperature	See Fig. A.1b	See Fig. A.2b
Initial current profile	See Fig. A.1c	See Fig. A.2c

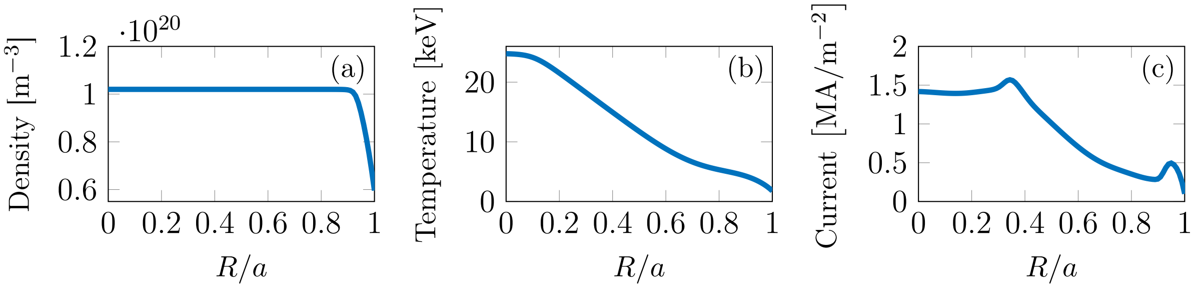


Figure A.1: Initial radial profiles of (a) density (b) temperature and (c) current for ITER

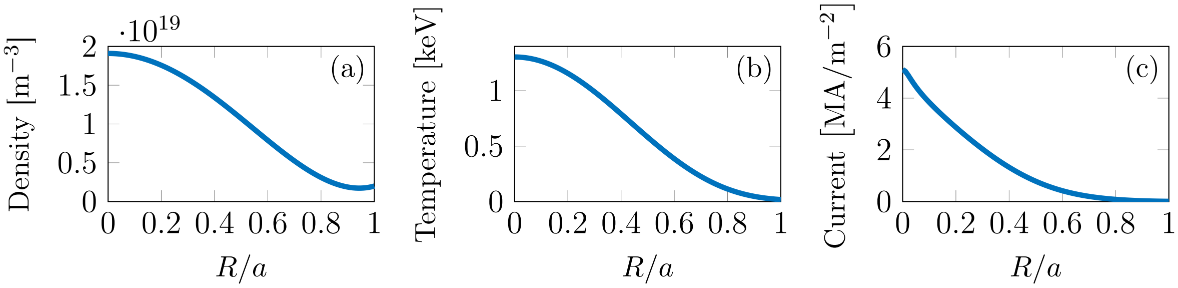


Figure A.2: Initial radial profiles of (a) density and (b) temperature of TEXTOR

[1] Used for setting a boundary condition for the electric field

Appendix B

Tests of GO+CODE Convergence

The following sections provide further details to the resolution and convergence tests mentioned in Section 5.3.

B.1 Benchmark of N_ξ and N_y for CODE

This section presents a benchmark performed on a single CODE scenario. The relative error in the runaway current density as a function of resolution parameters N_ξ and N_y is presented in Fig. B.1. The graph indicates that increasing values of N_ξ increases accuracy, as can be expected. However, accuracy no longer increases appreciably once N_ξ has reached a value of approximately 6 to 8. This suggests that N_ξ can be chosen to be the value at which accuracy barely increases anymore. The parameter N_y seems to affect accuracy more continuously. It can be seen from the graph that accuracy improves gradually with increasing N_y. The relative error was through curve fitting found to be approximately proportional to $1/N_y^3$. Note the direction of the N_y axis. An observation that can be made from the graph is that the parameters do not seem to interact noticeably.

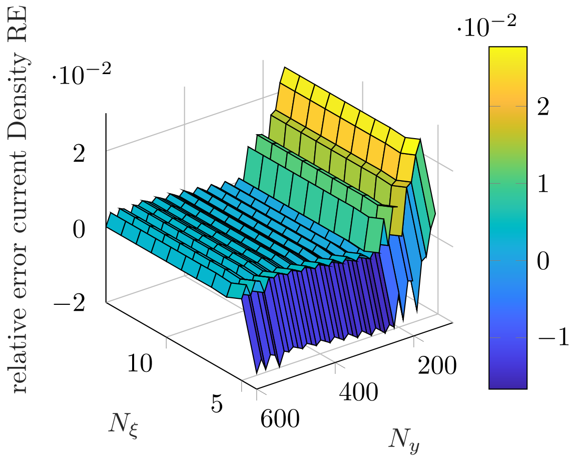


Figure B.1: Relative error for the runaway current density as a function of resolution parameters N_ξ and N_y. Increasing resolution leads to diminishing accuracy improvements. Note the direction of the N_y axis.

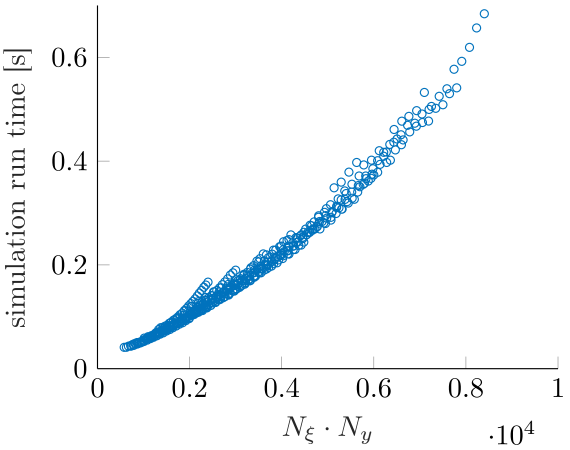


Figure B.2: Simulation run time as a function of the product $N_\xi \cdot N_y$. Simulation run time is approximately proportional to $(N_\xi \cdot N_y)^2$.

During a simulation, CODE performs one or more matrix inversions on a matrix with dimensions $N_\xi \cdot N_y$. It is therefore possible to extract information about the simulation by regarding the product $N_\xi \cdot N_y$. Fig. B.2 shows simulation run time as a function of

$N_\xi \cdot N_y$. It can be shown by curve fitting that the simulation run time is approximately proportional to $(N_\xi \cdot N_y)^2$. There are, however, artefacts in the figure that indicate that other factors affect the outcome. One of these is that there are sections in which run time increases more rapidly. This can be seen as small divergences that break off from the main body, especially between values of 2000 to 4000 for $N_\xi \cdot N_y$.

The scenario simulated during this benchmark had the following parameters: $T = 500\,\mathrm{eV}$, $n = 5 \cdot 10^{19}\,\mathrm{m}^{-3}$, $Z = 1$, $E = 1\,\mathrm{V/m}$, $B = 2\,\mathrm{T}$, $\Delta t = 1$, $t_{\max} = 500$, $y_{\max} = 50$. Avalanche sources were turned off. All other parameters were set to their default values. The simulation used a polynomial grid spacing for the y grid. This increases resolution of the thermal bulk of electrons while simultaneously providing coverage for high-energy electrons.

B.2 Benchmark of Time Step Parameter Δt for CODE

This benchmark evaluates how the size of the time step parameter Δt affects run time and relative error. The same physical scenario was used as in the previous section but with $N_\xi = 25$ and $N_y = 250$. A slightly different metric that can be used when investigating the time step length is the number of time steps that have to be taken. This corresponds more clearly to the computational complexity since every step in the time-advance has a similar computational requirement. The results in this section are therefore presented in terms of number of time steps instead of in terms of Δt. The number of time steps is equal to $t_{\max}/\Delta t$. It is in other words inversely proportional to the size of the time step.

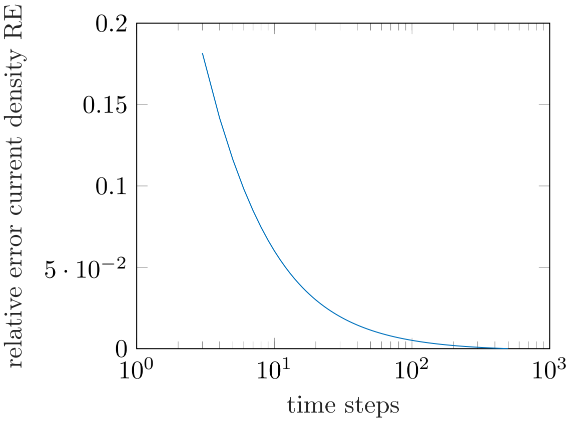


Figure B.3: Relative error for the runaway current density as a function of the number of time steps. A higher number of time steps gives a smaller error.

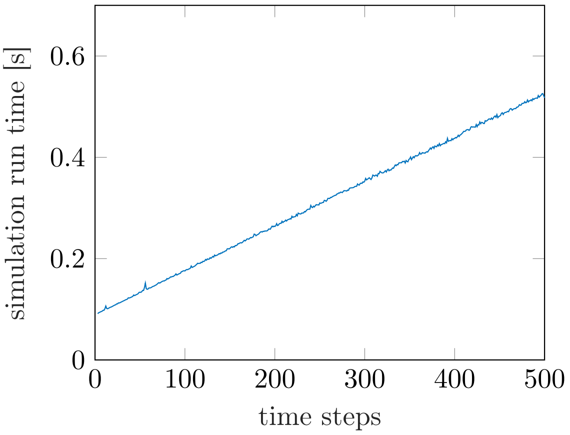


Figure B.4: Simulation run time as a function of the number of time steps. Note that the relationship is linear, with a constant offset.

The results of the benchmark are presented in Fig. B.3 and Fig. B.4. The general characteristics in Fig. B.3 seem to resemble the behaviour of N_y. The relative error was through curve fitting determined to be inversely proportional to the number of time steps. The time dependence is shown to be almost exactly linear, with a constant offset. This offset is due to other calculations, for example matrix inversion, in the CODE simulation.

These other calculations are only performed once, hence a constant offset. The rest of the simulation run time is then made up solely by time-advance calculations.

Fig. B.3 shows that in order for the simulation to converge to the correct value, a large number of time steps must be taken. The simulation of this scenario seems to require 100 or more time steps in order to achieve relative errors of less than one percent. The fact that simulation run time scales linearly with the number of time steps leaves some room for parameter optimisation, although not as much as the parameters N_y and N_ξ did. It is however of great importance to choose a sufficiently large number of time steps in relation to the rate of change in physical parameters. When the physical parameters change quickly, a large number of time steps is required. This is equivalent to choosing a sufficiently small Δt.

B.3 Study of Parameters y_{max} and `gridParameter`

Two additional resolution parameters were studied, as mentioned in Section 5.3.1. These were the parameters y_{max} and `gridParameter`. The parameter y_{max} describes the number of grid points in y and `gridParameter` controls the spacing of the grid according to the formula $y = s^2 + \texttt{gridParameter} \cdot s$, where `gridParameter` is a constant and s is a uniformly spaced variable. Parameter scans similar to those for N_ξ, N_y and Δt were performed for y_{max} and `gridParameter`. As expected, no effect on simulation run time was observed. It appears that `gridParameter` does not significantly affect the simulation results either, as long as values close to the default value are chosen. It appears that the default choice works well so it will not be investigated further. The parameter y_{max} must, however, be chosen with greater care. This parameter must be great enough to accommodate the entire distribution function f. If it is too small the observed results will quickly diverge from the correct results. The simulation is not as sensitive when y_{max} is chosen to be greater than what is required by f, but y_{max} cannot be too great or else the simulation will suffer from the relatively low number of grid points N_y. The parameter should be chosen to be slightly greater than what is strictly required in order to guarantee complete coverage of f. It is nevertheless advantageous to choose y_{max} to be as small as practically possible in order to increase the coverage of the distribution function. When chosen correctly, it might be possible to decrease N_y, thereby reducing simulation run time.