



UNIVERSITY
of York

Testing a New Theory for Small Plasma Eruptions in Tokamaks Using Gyrofluid Simulations

Edward Simmons

A Dissertation Submitted to the
School of Physics, Engineering and Technology
at the University of York
in Partial Fulfilment of the Requirements
for the Degree of Master of Science

August 2023

Supervisors:

Dr. Arka Bokshi
Dr. David Dickinson

Abstract

Two electrostatic models of the toroidal ion temperature gradient (ITG) mode, one a fluid model and the other a gyrofluid model, are compared in order to determine whether observations made in the former are also observed in the latter. The behaviour in question is a rapid but transient increase in the global growth rate of the ITG mode, brought on by changes in the equilibrium flow shear, that could be responsible for some classes of small edge-localised modes (ELMs), which are commonly found in magnetically confined tokamak plasmas. Qualitatively strong agreement is found between the two models. As a tangential but relevant exercise, Gaussian process regression (GPR) is presented as a potential method of efficiently finding optimal values for the input parameters of such computational models. It is shown that GPR is a powerful, capable, and accessible tool that could be effectively utilised in many areas of computational plasma physics research.

Contents

1	Introduction	3
1.1	Previous Research	3
1.1.1	Dickinson et al. (2014)	3
1.1.2	Bokshi et al. (2016)	4
1.2	My Research	4
2	Methodology	5
2.1	Physics Models	5
2.1.1	Fluid Model: toroidal-fluid-itg	5
2.1.2	Gyrofluid Model: electrogem	7
2.2	Gaussian Process Regression	8
3	Results	11
3.1	Fluid Model	11
3.1.1	Parameter Optimisation Using GPR	11
3.1.2	Demonstrating the IM - GM Transition at Low Magnetic Shear	16
3.1.3	Growth Rate as a Function of Flow Shear at Low Magnetic Shear	19
3.2	Gyrofluid Model	20
3.2.1	Using the Singh Parameters	20
3.2.2	Demonstrating the IM - GM Transition	21
3.2.3	Growth Rate as a Function of Flow Shear	22
4	Conclusions	24
References		26
Appendix A	Gaussian Process Regression Loop	28

1 Introduction

One of the key issues facing the development of magnetic confinement fusion (MCF) devices, such as tokamaks, is the degradation of confinement due to the turbulent transport of heat and particles. This turbulence is thought to be driven by micro-instabilities [1], which are a class of instability that have a characteristic scale length, perpendicular to the magnetic field line along which they form, on the order of the ion Larmor radius [1, 2]. These micro-instabilities can be electrostatic or electromagnetic in nature [1], which is exploited as a method of simplifying the computational models used to simulate them. Understanding the effect that these micro-instabilities have on the overall stability of the confined plasma, and hence the level of confinement that can be achieved in MCF, is the main motivation for the research undertaken here.

1.1 Previous Research

Both Dickinson et al. and Bokshi et al. focused their research on the ion-temperature gradient (ITG) mode, which is an electrostatic toroidal micro-instability, and is associated primarily with the sound wave [1, 2]. These ion sound waves occur when the frequency of transverse electromagnetic waves, as well as oscillations in the plasma, are sufficiently low enough to couple energy into the ions - in other words, the ions are unaffected by these oscillations at higher frequencies due to their relatively large mass (when compared to the electrons) [3]. The sound waves that form are similar to those found in a gas, but instead of collisional transfers of momentum, the ions have energy coupled into them via the electric field [3].

1.1.1 Dickinson et al. (2014)

Dickinson et al. used a model that solved a 2D eigenvalue equation [2], the results of which were then compared to a 1D model that focused on a more traditional picture of a ballooning mode, most commonly associated with ideal magnetohydrodynamics (MHD) [4]. As Dickinson et al. discuss in their paper, this MHD ballooning mode is shown to be the maximally unstable mode, which is dubbed the isolated mode (IM), and a more stable general mode (GM) exists throughout the plasma [2], typically located at either the top or the bottom of the poloidal plane [2].

Critically, Dickinson et al. found that when adding flow shear to their model, they observed a transition from the IM to the GM at a critical value of flow shear [2]. This addition of flow shear is likened to simulating the effect of toroidal momentum input via neutral beam injection (NBI) [1], which is a typical method of depositing power into tokamak plasmas [5].

Dickinson et al. posited that this rapid transition from the IM to the GM could be the cause of a small edge-localised mode (ELM) [2]. An ELM is a rapid ejection of heat and particles from the confined plasma into the scrape-off layer, commonly found in tokamak plasmas that are operating in the high-confinement mode (H-mode) [6]. They arise due to the rapid collapse of gradients at the plasma boundary, as the steep H-mode gradients grow beyond the MHD stability limit [6]. The reason why Dickinson et al. were specifically interested in small ELMs, is that they could potentially be utilised as

a method of mitigating (or perhaps completely avoiding) the occurrence of much larger Type 1 ELMs, the potential damage from which is a major threat to the success of large scale tokamak devices such as ITER [2, 7].

1.1.2 Bokshi et al. (2016)

Bokshi et al. expanded on the work done by Dickinson et al. by using a time-dependant, linearised, global fluid model of the ITG mode [1], with their focus being on studying the effects of evolving the rate of flow shear with time. The motivation behind varying flow shear with time was the nature of ELMs themselves; when an ELM occurs, the H-mode pedestal plasma profiles collapse, and then build up again before the next ELM [1]. Therefore, they set about studying how the ITG mode develops as it passes through the critical value of flow shear identified by Dickinson et al., and they showed that the mode can transition from the GM-IM-GM under various conditions [1].

1.2 My Research

I was tasked with investigating whether the IM-GM transition observed by both Dickinson et al. and Bokshi et al. could be reproduced when using a more advanced physics model. The fluid model used by Bokshi et al. [1] (discussed in section 2.1.1) is compared against a gyrofluid model, published by Ma et al. [8] (discussed in section 2.1.2). If the same behaviour is observed in the gyrofluid model as in that of the fluid model - i.e. that the IM is the most unstable mode, and that adding flow shear to the model causes the mode to rotate poloidally to form a GM - then the hypotheses formed by Dickinson et al. and Bokshi et al. would be reinforced, and we could be one step closer to identifying a cause of small ELMs in tokamaks.

A statistical method, known as Gaussian process regression (GPR), was used with the aim of efficiently mapping out each computational model's parameter space (and consequently, discovering optimal values for those parameters), was implemented and is discussed in section 2.2.

This GPR optimisation technique was tested on the fluid model to prove its validity, the results of which are discussed in section 3.1.1. These results enabled further research into interesting and newly discovered areas of the fluid model's parameter space (discussed in sections 3.1.2 and 3.1.3). Time became a major constraint of the project, and as a consequence I was not able to carry out the parameter optimisation technique on the gyrofluid model. However, a literature survey conducted at the time provided a suitable parameter space for me to explore, the results of which (and the process of applying said parameters) are discussed in section 3.2.1. Once a suitable set of parameters had been established for the gyrofluid model, it was possible to explore the effects of flow shear on the ITG mode structure, and observe its behaviour under conditions similar to those explored by Bokshi et al., the results of which are discussed in sections 3.2.2 and 3.2.3.

2 Methodology

2.1 Physics Models

2.1.1 Fluid Model: toroidal-fluid-itg

The fluid model, known to me by its repository name "[toroidal-fluid-itg](#)", is the same as that used by Bokshi et al. in their 2016 paper [1]. It is a simple, linearised model of the toroidal ITG mode, that assumes that the electrons in the system are adiabatic [1]. This assumption of an adiabatic electron response, which is also known as the Boltzmann relation [9], has been proven to be a valid assumption [9], and is also assumed in the derivation of the ion sound speed [3]. The model is designed to simulate and measure the perturbed electrostatic potential, with the geometry of the simulation domain set to represent a large aspect-ratio, spherical tokamak [1].

Bokshi et al. are the first to admit that their fluid model is a simplification of the ITG mode, and they mention that a gyrokinetic or gyrofluid model would be required for a full physical description, specifically concerning drift-resonances [1]. They also mention that electromagnetic effects would ideally be taken into account, especially at high values of plasma β [1].

In my research, I focused on four of the fluid model's input parameters. Firstly, the scale factor for the ITG drive's radial profile [1], η_g . Secondly, the density scale length (normalised to the major radius) [1], ϵ_n . Thirdly, the magnetic shear, \hat{s} , which is defined as:

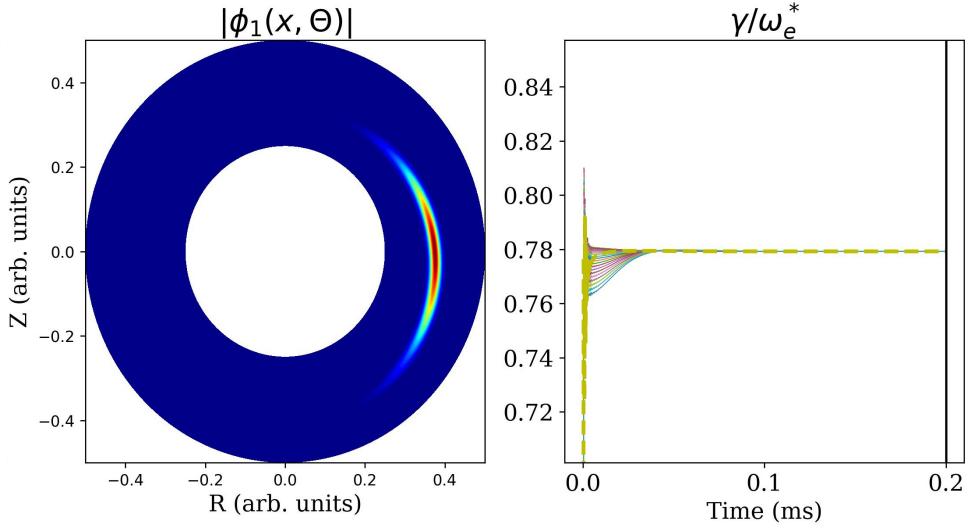
$$\hat{s} = \frac{rq'}{q} \quad (1)$$

Where r is the radial coordinate, q is the safety factor, and q' is the radial derivative of the safety factor profile [1]. Finally, the flow shearing rate, γ_E , which is defined as:

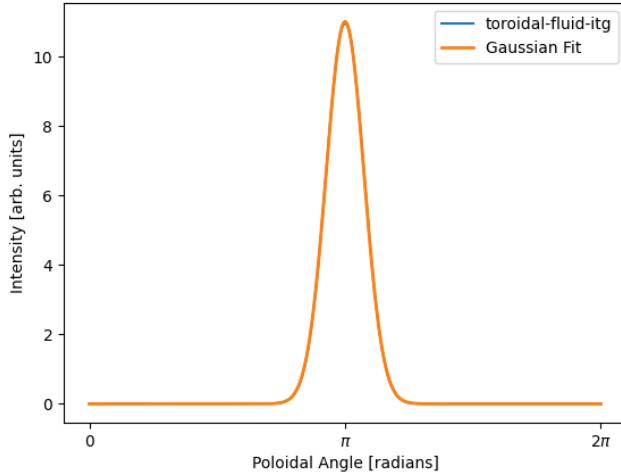
$$\gamma_E = \frac{d\Omega_\phi}{dq} \quad (2)$$

Where Ω_ϕ is the toroidal component of the Doppler shift in the global mode frequency, Ω [1]. Only the toroidal component of the Doppler shift of Ω is considered, as the shear in the toroidal flows is assumed to be dominant - as previously mentioned, the strong drive from toroidal momentum input sources such as NBI, coupled with the fact that poloidal flows experience strong neoclassical damping [1], renders the poloidal component of the flow shear negligible in comparison.

When using the fluid model, my workflow consisted of four main stages, with each step automated in sequence when undertaking multiple iterations. The first was simply to update the input file with new values for the parameters of interest, and run the simulation. If the simulation resulted in a successfully converged mode (this is discussed further in section 2.2) then I needed to identify the radial coordinate of the peak of the measured electrostatic potential. Once I had this coordinate, I could extract a poloidal section from the fluid model's output data, i.e. a line section running from the inboard mid-plane, through the outboard mid-plane, and then back around to the inboard mid-plane ($0 \rightarrow \pi \rightarrow 2\pi$).



(a) Mode structure in the poloidal plane (left), and convergence (right).



(b) A Gaussian curve provides a very close fit to the poloidal line section taken at the peak potential.

Figure 1: Plots from an early, successful run of the fluid model. The parameters used in this run were similar to those published by Bokshi et al. [1]. Notably in this example, the flow shear was set to zero, hence the mode forms, as the IM, at the outboard mid-plane. From the right-hand plot in (a), you can clearly see the incentive to exit the code early; the mode forms in the first quarter of the simulation's total run time, meaning much of the computation undertaken was unnecessary.

When considering the ideal MHD picture of a ballooning mode, the peak of the ITG mode should appear at π radians in this line section - right on the outboard mid-plane - and indeed it does (in the absence of flow shear). The final stage of the workflow was to fit a Gaussian through the line section data - this was used as a means to compare the poloidal width of one result versus another. By measuring the full width at half maximum (FWHM) of the Gaussian fit, I could make an approximate record of the poloidal width of the mode. This workflow is illustrated in figure 1.

The fluid model is a serial FORTRAN code, which takes around 20 minutes to complete 2000 time steps. Each time step, δt , represented 0.0001 s, therefore culminating in a

total simulation time of 0.2 ms, which is consistent with the characteristic timescale of small ELMs observed on experimental devices such as JT-60U [1]. This meant that much of my time was spent carefully orchestrating long, automated runs of the code, with some loops taking several days to complete. My supervisors, after some trial and error, managed to successfully implement a method that would enable the code to terminate early if a mode had clearly converged, which meant that some simulation runs took under five minutes to complete, but unfortunately this improvement was only fully developed towards the end of my project.

2.1.2 Gyrofluid Model: `electrogem`

A gyrofluid model (also sometimes referred to as a gyro-Landau-fluid model [8, 10]) extends upon the physics of a fluid model by adding important kinetic effects, such as Landau damping [10] and gyro-orbit averaging [10] - i.e. the act of averaging out the orbits of charged particles, around magnetic field lines, by treating the orbits as planar discs, reducing the number of degrees of freedom from six to five.

The gyrofluid model that I used was written by my supervisors, and is based on a model published by Ma et al. [8] in 2016. As with the model published by Ma et al., ours is written in the BOUT++ framework; an object-oriented C++ code developed to simulate 3D fluid equations in curvilinear coordinates [11]. BOUT++ has been used in the past to successfully simulate several stages of the ELM life cycle [8], and has become somewhat of a staple in the realm of computational plasma physics research.

The key advantage, from my perspective, of the model being written in the BOUT++ framework, is that many of the confusing, error prone, and frankly laborious tasks involving parallel communication, differential geometry, file input and output, and memory management are all handled by the framework itself. In addition, there is a wealth of experience and expertise within the York Plasma Institute (YPI) when it comes to using BOUT++ (a fact that became invaluable when it came to navigating the package's complicated installation process). As BOUT++ (and hence, the gyrofluid model) is parallelised, it was necessary to execute it on a multi-core server for optimal performance, such as those based at the YPI.

As the toroidal ITG is electrostatic in nature, we chose to drop the electromagnetic terms from the model published by Ma et al. [8], reducing the model from seven evolving equations down to five - a slight performance increase, at the cost of some physics. As mentioned in section 2.1.1, it would be ideal to include electromagnetic effects, but it seemed more appropriate that we conduct a like-for-like comparison between the fluid and gyrofluid models.

My supervisors and I are keen to, eventually, send a pull request that would merge our model into the list of examples available in the BOUT++ repository, for other researchers to use. My contributions to the code base were therefore always approached with this in mind - prioritising clean, easily readable code, that was well documented and tested. The code repository that my supervisors and I worked on collaboratively is called "electrogem".

2.2 Gaussian Process Regression

Regression, in computer science, is the act of mapping a value (or a set of values) produced by a function back to the function itself [12]. Similarly, one can think of it as deriving the independent variable(s) by analysing the dependant variables. Gaussian process regression (GPR) is a non-parametric regression technique [13] (meaning that there is no prior or assumed knowledge of the function), based on Baysean inference statistics [14], which can estimate and fit arbitrarily spaced data in any number of dimensions [13].

The *Gaussian* aspect of GPR enables and sustains its viability as an n-dimensional analysis tool - there's only so much computing time available in the universe, so there needs to be some level of approximation. Each data point is treated as a Gaussian probability distribution function, referred to as a *process* [12], which greatly simplifies the computation required for inference and machine learning [12]. Inference in a Gaussian process enables you to estimate the properties of the function you want to study, and can achieve the same results with a finite number of points as with an infinite number, only with reduced precision [12].

A key advantage of GPR is its ability to simultaneously both predict values and provide uncertainty estimates for it predictions [14]. Upon making a prediction, GPR works by assessing its data set as a whole, and can therefore be described as a global method [14]. One drawback of using GPR is that, depending on the implementation, it can require expensive algebraic computations that scale as $\mathcal{O}(n^3)$ [14], where n is the number of inputs. This effectively limits the use of GPR to data sets that are on the order of a few thousand data points [14], depending of course on the computing power available to you. GPR is therefore most applicable in situations that:

1. You have no prior knowledge of the function that you are trying to map.
2. You are limited in the size of data set that you will feasibly be able to collect, e.g. that your program takes a long time to produce a single data point.

In my case, both of the above points were true. I had very little knowledge of the shape of the function (aside from the parameters published by Bokshi et al. [1]), and as previously mentioned, the fluid model took around 20 minutes to run on average. The gyrofluid model's run time can vary drastically based on both the computational complexity of any one time-step, and the speed of convergence, and hence could vary from being on the order of minutes to hours.

When starting this project, my supervisors and I were uncertain whether GPR would be a suitable method for finding an optimal parameter space for the gyrofluid model, so we decided to first test its use on the fluid model (of which we had a slightly better understanding of the parameter space), and then later apply it to the gyrofluid model if it proved successful, the results of which are discussed in section 3.1.1. Our hope was that it would ultimately save time, as a trial-and-error approach (given such a wide parameter space) could ultimately prove too costly in the time allowed for the project.

Dr. Chris Bowman, a University of York alumnus and researcher at the UK Atomic Energy Authority (UKAEA), has published a python package named "inference-tools" [13],

which contains all of the methods necessary required to carry out GPR, and was in no small part responsible for the success of this project. Chris' documentation is excellent [13], but I have also included a pseudo-code example of GPR in appendix A.

The way that the GPR routine differentiates the merit of one result when compared to the data set as a whole, is through a user-defined *objective function*, the result of which the GPR routine will aim to maximise. In my case, I was trying to find the minimum poloidal width of the formed ITG mode (which would allow me to observe the IM-GM transition more clearly), so my objective function simply returned the negative of the measured poloidal width.

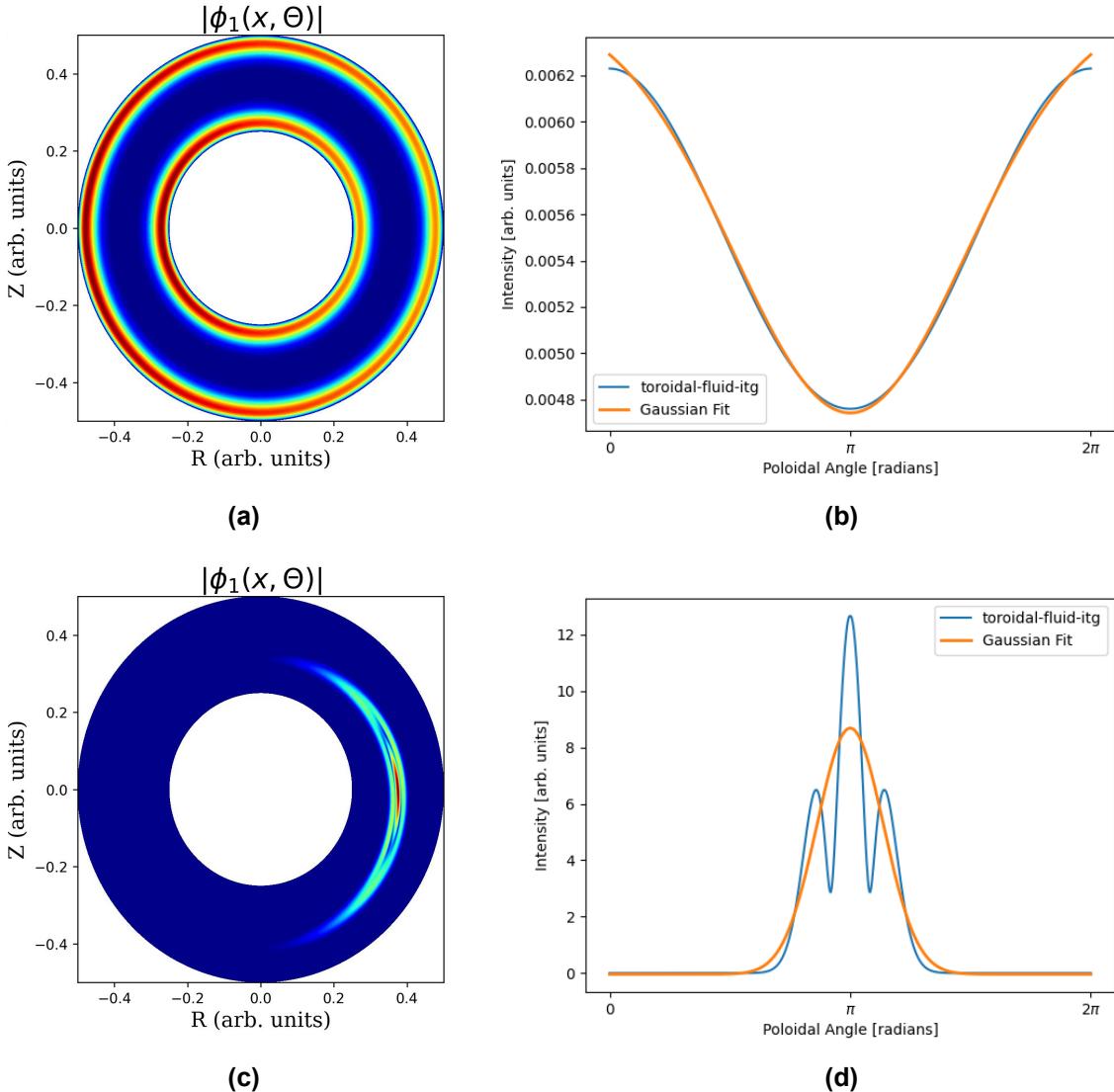


Figure 2: Examples of the fluid model's output that would be assigned the penalty value of -2π , instead of the measured FWHM. (a) shows a mode that has converged at the domain boundary, which is unphysical, and (b) shows the danger in ignoring such a result - the poloidal section is taken at the peak potential as usual, and a Gaussian curve fit can still be achieved, but returning the FWHM in this example would provide an anomalous result. (c) shows a mode that has not fully converged within the simulation time, containing multiple peaks along a single poloidal section. Fitting a Gaussian to such a result, shown in (d), can return a misleadingly positive result, so the penalty value is assigned instead.

In an effort to simultaneously both distinguish anomalous results (i.e. simulation runs that failed to form a physical result) from positive ones, and to discourage the GPR routine from wastefully exploring unpromising areas of the parameter space, I chose to assign simulation runs that failed to meet certain criteria with a "penalty value" that was distinguishably lower than the average result. My penalty value of choice was -2π , i.e. representing a mode that spanned the entire poloidal width of the simulation domain - not a mathematically significant choice, just one that made semantic sense to me. Figure 2 shows the two scenarios, identified through testing the fluid model, that would result in the penalty value being assigned.

An *acquisition function*, a choice of which is required in the initialisation of the GPR routine (see appendix A), is what then defines the behaviour of the GPR algorithm, i.e. are you trying to map out the function's entire parameter space in an attempt to determine the shape of the function, or do you want to prioritise finding a maxima? In the case of the latter, it is possible to unintentionally narrow in on local maxima, as the GPR routine is less focused on exploring the parameter space as a whole. It's for this reason, that I decided to use the *UpperConfidenceBound* [13] as my acquisition function, as I was equally interested in seeing if there were other areas of the parameter space that formed poloidally narrow modes, as well as potentially finding more precise minima in the previously identified parameter regimes found by Bokshi et al. [1].

My experience using GPR was extremely positive (mostly thanks to the work of Dr. Bowman), and I believe it shows great promise, both in this application and in other areas of computational plasma physics. For example, when modelling the peeling-balloonning boundary; one could imagine a GPR algorithm that mapped out the peeling-balloonning boundary by intelligently sampling the parameter space, as opposed to a brute force approach of sampling the entire parameter space in a sparse grid, like the one shown in this paper by Osborne et al. [15]. GPR, in this example, could potentially decrease the required computation time, whilst simultaneously improving the resolution of the acquired data.

3 Results

3.1 Fluid Model

3.1.1 Parameter Optimisation Using GPR

As an initial test of the efficacy of the GPR algorithm I had written, I decided to map out the parameter space of two of the fluid model's parameters of interest, η_g and ϵ_n , to see if GPR would yield similar results as those published by Bokshit et al. [1]. I kept $\hat{s} = 25.0$, as I had been advised by my supervisors that forming a converged ITG mode had a strong dependence on magnetic shear. As this was simply a test of measuring the mode width, the flow shear $\gamma_E = 0.0$, which ensures that the mode will peak at the outboard mid-plane, i.e. form an IM. Figures 3, 4, and 5 show the results from this search.

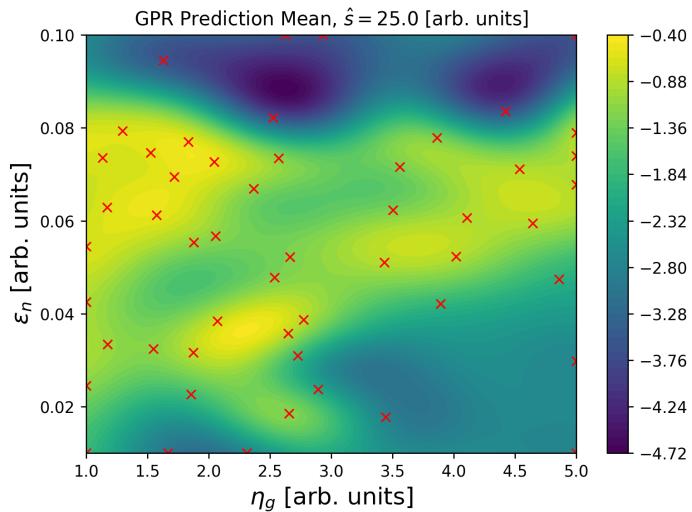


Figure 3: The red crosses indicate the data points, and the colour bar shows the value of the GPR predicted mean.

There are a few notable features in figure 3. Firstly, it is clear that the parameters identified by Bokshi et al. were indeed fairly optimal ($\eta_g = 0.2$, $\epsilon_n = 0.08$ [1]), as there is a distinct cluster of results with a relatively low poloidal width around that region. As denoted by the red crosses on the plots in figures 3 and 4, this run of the GPR algorithm consisted of 50 iterations of the GPR loop (see appendix A for context). It's notable therefore that, in 50 iterations:

- The GPR algorithm found other areas of the parameter space, not published by Bokshi et al. [1], that produced similarly optimal results.
- Running the simulation with the parameters published by Bokshi et al. [1] yields a FWHM of ~ 0.59 radians, and the best result found via GPR was ~ 0.54 radians. It's impressive that GPR was able to improve upon the published result, but it's also remarkable that the published parameters are so close to the optimal value found via GPR. If nothing else, this serves as kudos to the work of Bokshi et al. in discovering this region of parameter space without the aid of machine learning.

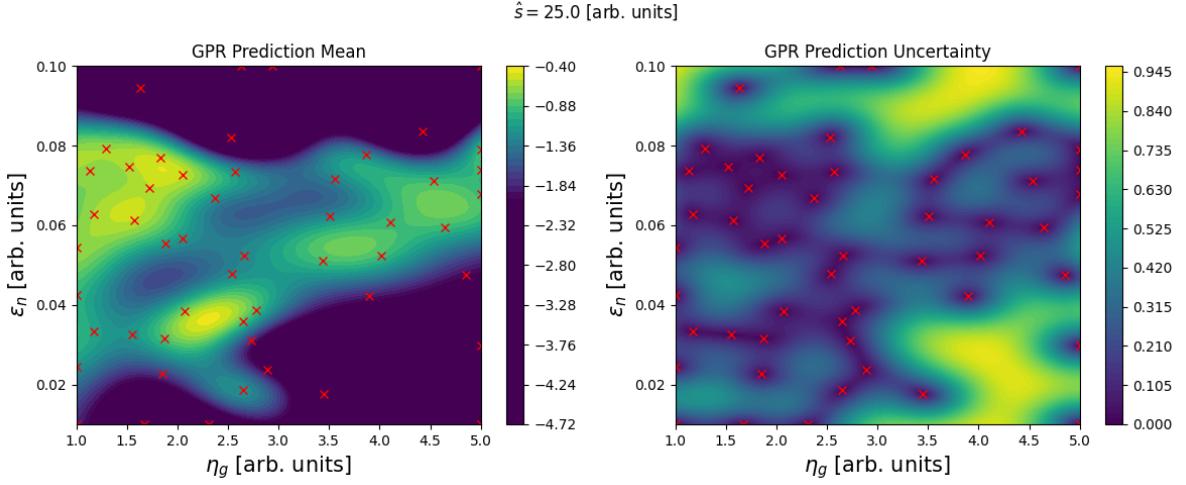


Figure 4: The left-hand plot shows the same results as in figure 3, but with the colour bar's minimum value set to -2.0, in order to better highlight the shape of the function. The right-hand plot shows the 2D uncertainty in the GPR prediction.

In figure 4, the uncertainty in the GPR prediction is shown, and it is notable (and somewhat intuitive) that the uncertainty is minimised locally around each data point. This "uncertainty space" is an illustration of how the `UpperConfidenceBand` acquisition function selects the input parameters for the next iteration of the algorithm; it will attempt to identify the shape of the function being modelled by systematically reducing the uncertainty across the parameter space, making decisions inferred by statistics [13].

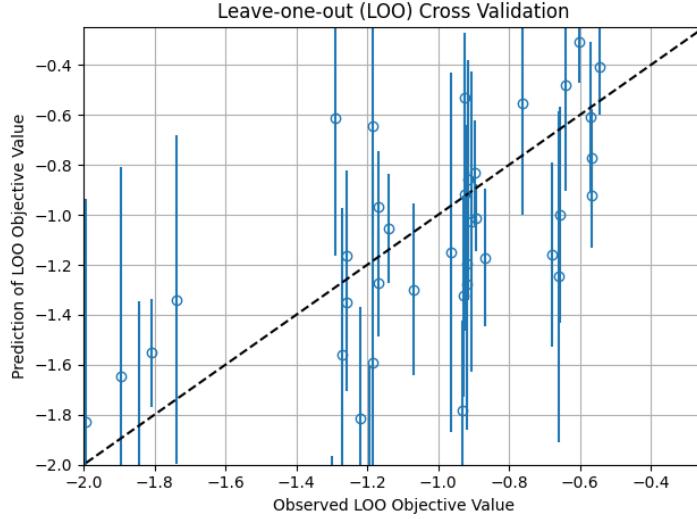


Figure 5: A leave-one-out cross validation plot of the results presented in figures 3 and 4.

In order to determine the effectiveness of GPR in regards to how successful it has been in determining the shape of the function, leave-one-out cross validation (LOO CV), like that shown in figure 5, can be used. LOO CV is a popular diagnostic tool when assessing the efficacy of machine learning algorithms [16, 17], and is a method of quantifying [17] of how effectively your model will be able to predict a future data point [16]. Figure 5 shows that the function that I was trying to model, i.e. the mode structure of the fluid model as a function of its parameter space, is complex in shape, as

the distribution of points on the LOO CV plot is broad (they would be centred around the $y = x$ line otherwise). LOO CV has been shown to be a pessimistic [18] prediction of modelled results, and it is advised not to judge a machine learning algorithm's efficacy on LOO CV results alone [18].

Once the efficacy of GPR had been established, I was keen to further explore the parameter space of the fluid model, particularly when it came to magnetic shear. The results published by Bokshi et al., and also those shown in figures 3 and 4, were obtained using a relatively high value of magnetic shear ($\hat{s} = 25.0$ [1]). Experimental data from tokamak facilities in operation today shows that magnetic shear values are in reality much lower [19].

While magnetic shear does vary from the core to the edge of a tokamak plasma [19], it is still shown experimentally to be much lower than the value used when producing the figures presented thus far. Weak or low magnetic shear is in fact being proposed as an ideal operating scenario for the ITER tokamak [20]. I was therefore interested in whether or not GPR could help to identify regions of the fluid model's parameter space that would enable me to still demonstrate the IM-GM transition behaviour, but at a more realistic level of magnetic shear.

From early testing, the results of which are shown in figure 6, it became apparent that increasing the number of dimensions that the GPR was being asked to optimise necessitated an increase in the number of iterations to be performed. While 50 iterations were sufficient to provide a comprehensive picture of the 2D parameter space, I found that the GPR algorithm required a much greater number of iterations to map out the 3D parameter space. Figures 6 and 7 show the results of 200 and 1000 iteration simulation runs respectively. These simulation runs took several days to complete, as each iteration (as mentioned in section 2.1.1) can take up to 20 minutes to complete.

The plots presented in figures 6 and 7 show six frames of an animation (the same frame is shown in each data set for comparison), where each frame represents a range of values for the magnetic shear that had been selected by the GPR algorithm. The axes of the plots are identical to those in figures 3 and 4, with η_g on the x-axis, and ϵ_n on the y-axis. The end result is something akin to a magnetic resonance imaging (MRI) scan, with each frame showing a "slice" of the magnetic shear's parameter space.

To the 200-iteration run's credit, it did discover the set of parameters that produced the best result of any that I recorded, i.e. the narrowest poloidal mode width at the lowest value of magnetic shear, shown in figure 6b. However, it failed to explore very much of the magnetic shear's parameter bounds - this is demonstrated by the absence of data in figures 6e and 6f. I found this lack of exploration particularly egregious when I realised that this void included the magnetic shear value used in figures 3 and 4. Hence, I decided to embark on the much larger run, but I at least had the sense to seed the GPR routine with the 200 values shown in figure 6, which helped to reduce the total computation time required.

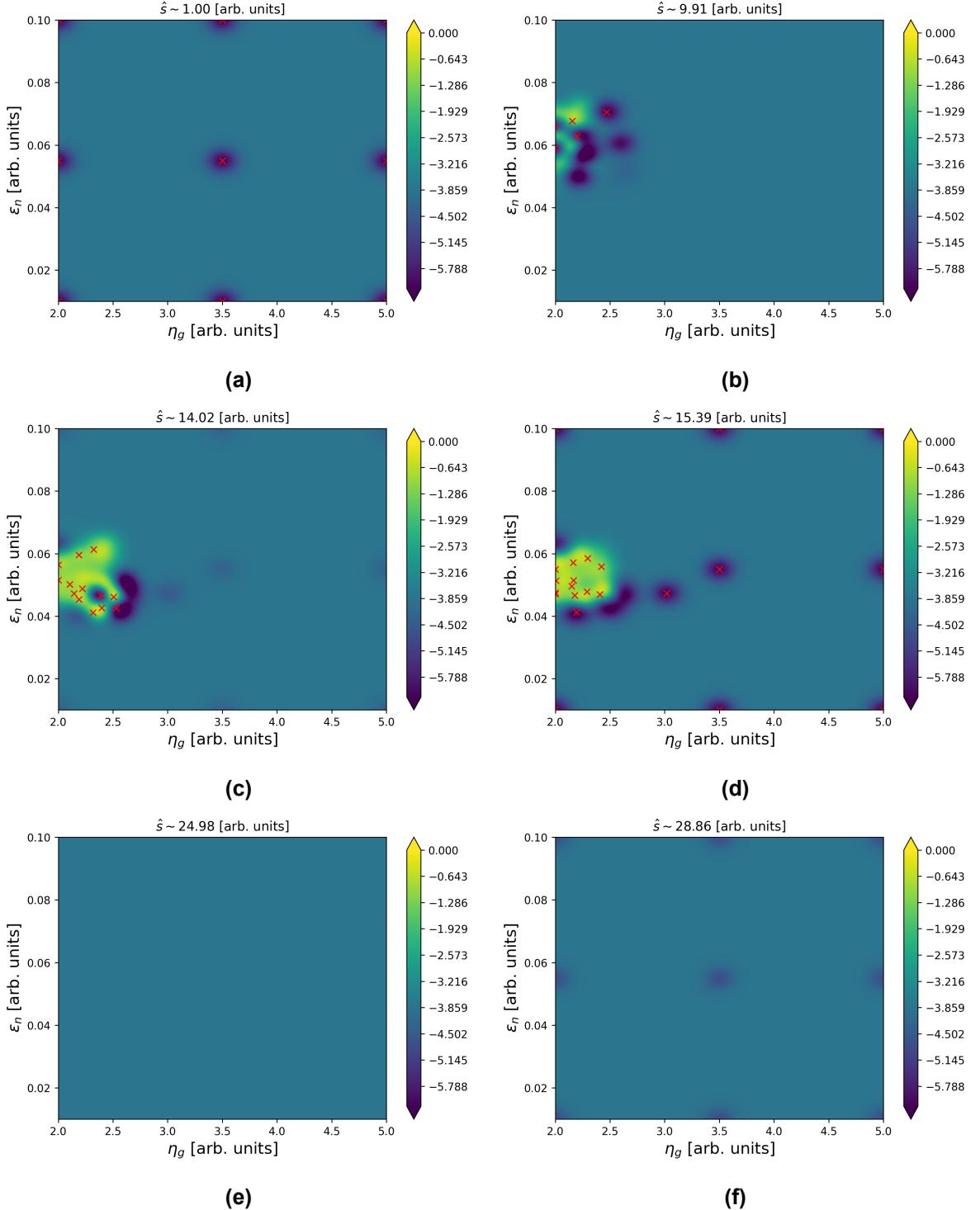


Figure 6: GPR results are shown at varying values of magnetic shear. Each plot denotes the following approximate magnetic shear values: (a) $\hat{s} \approx 1.00$; (b) $\hat{s} \approx 9.91$; (c) $\hat{s} \approx 14.02$; (d) $\hat{s} \approx 15.39$; (e) $\hat{s} \approx 24.98$; (f) $\hat{s} \approx 28.86$. The red crosses in each plot show the data points that were obtained at that approximate value of magnetic shear. The nine points that can be seen in a face centred cubic arrangement in both (a) and (d) are some of the 28 values that were used to seed the GPR routine.

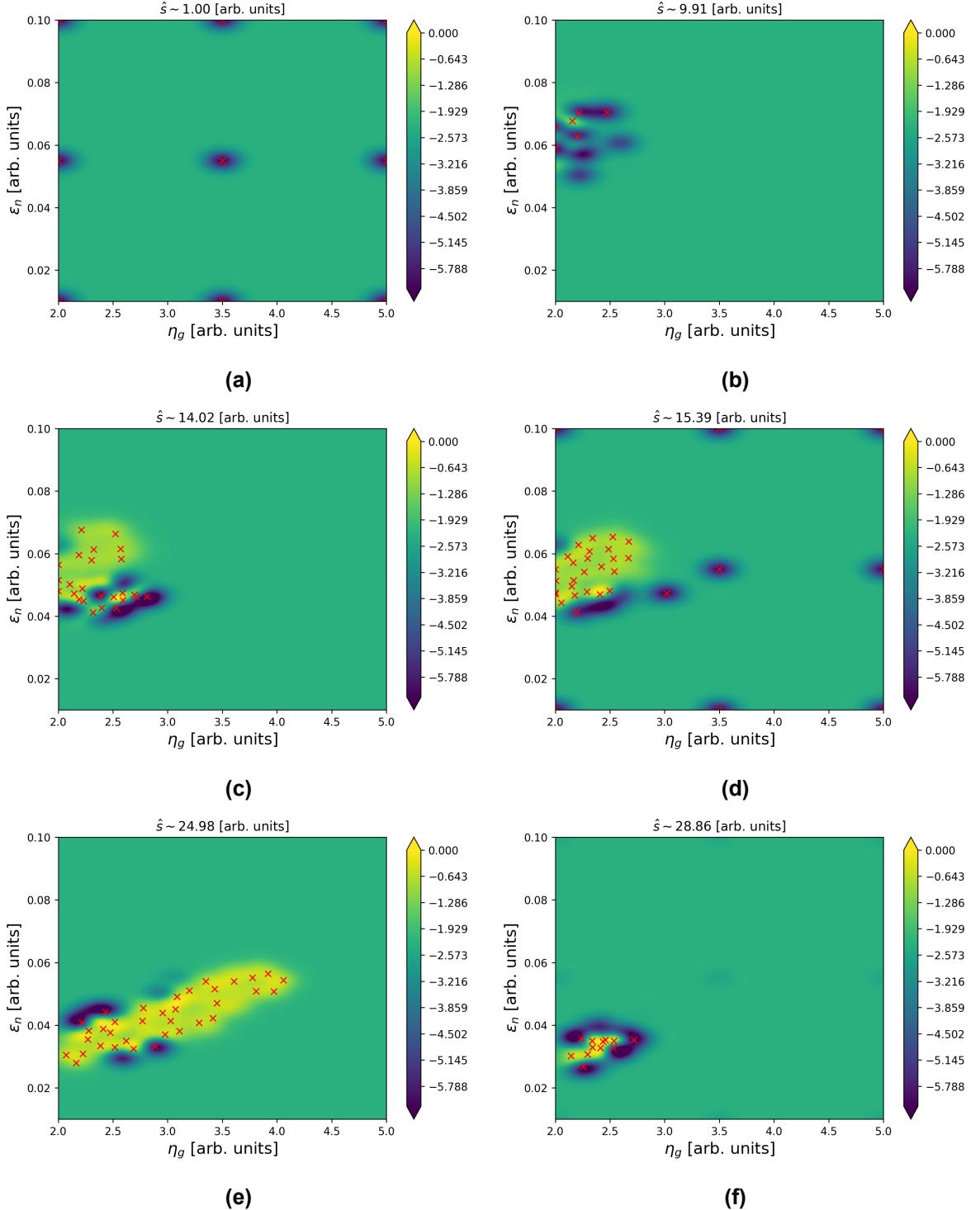


Figure 7: The results of the 1000-iteration GPR run are shown, with the magnetic shear values in each plot identical to those described in figure 6. Increasing the number of iterations five-fold enabled the GPR to explore a greater breadth of the parameter space, but there are still large, unexplored regions, devoid of data points.

The 1000-iteration run (shown in figure 7) did manage to explore a wider variety of magnetic shear values, finding many converged modes at values above and below that used in figures 3 and 4. However, it is clear to me that even 1000 data points is an insufficient number to construct a sufficiently detailed map of the 3D parameter space, especially at the edges each parameter's bounds - the bulk of the data points are found in the region where $10.0 \leq \hat{s} \leq 27.0$, and I suspect this is because the mathematics of the acquisition function slightly incentivises the selection of parameter values in a region of the parameter space with lower uncertainty. It is arguable however, that this strategy of collecting large numbers of data points is juxtaposed with the main incentive for using GPR, i.e. the model you want to run is computationally expensive, therefore you want to converge on a solution with the minimum number of iterations [13].

3.1.2 Demonstrating the IM - GM Transition at Low Magnetic Shear

Finding a converged mode, at low magnetic shear, using the fluid model was a novel discovery, so the natural next step was to test whether adding flow shear would result in the IM-GM transition behaviour that I am interested in.

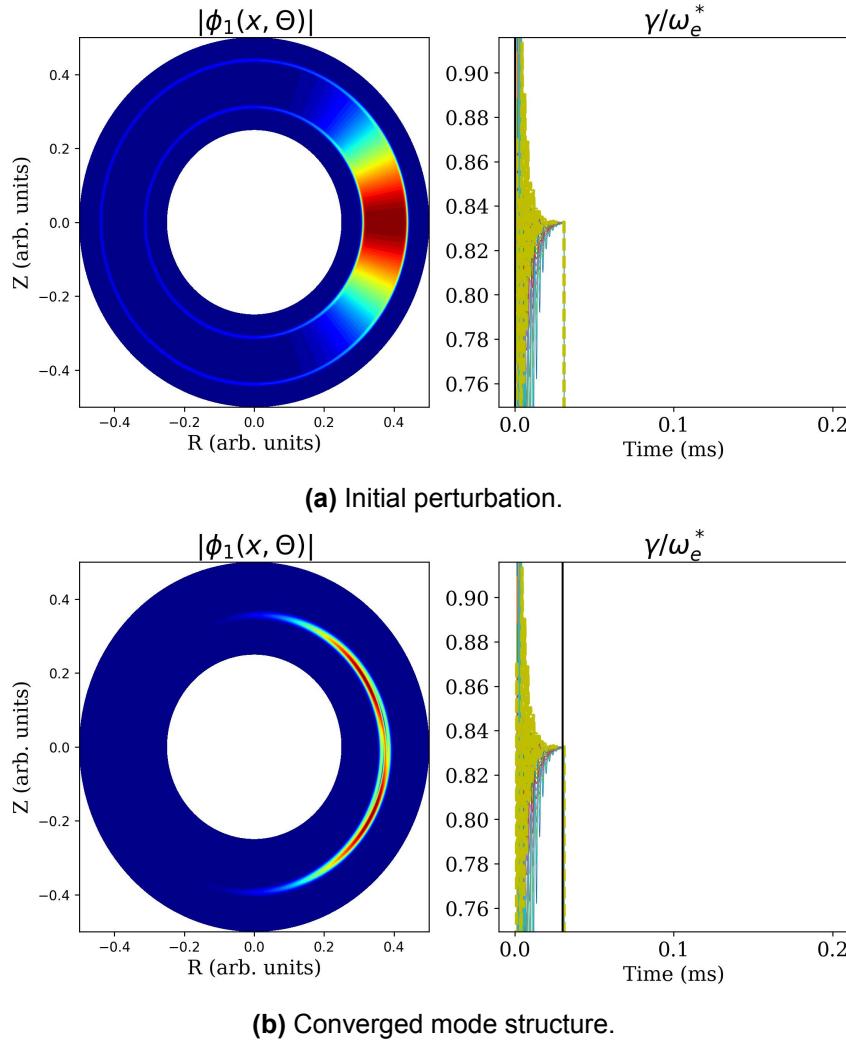


Figure 8: The formation of the IM is shown for $\eta_g \approx 2.16$, $\epsilon_n \approx 0.07$, $\hat{s} \approx 9.90$. The IM only forms at $\gamma_E = 0.0$. The time step of each plot is shown as a vertical black line.

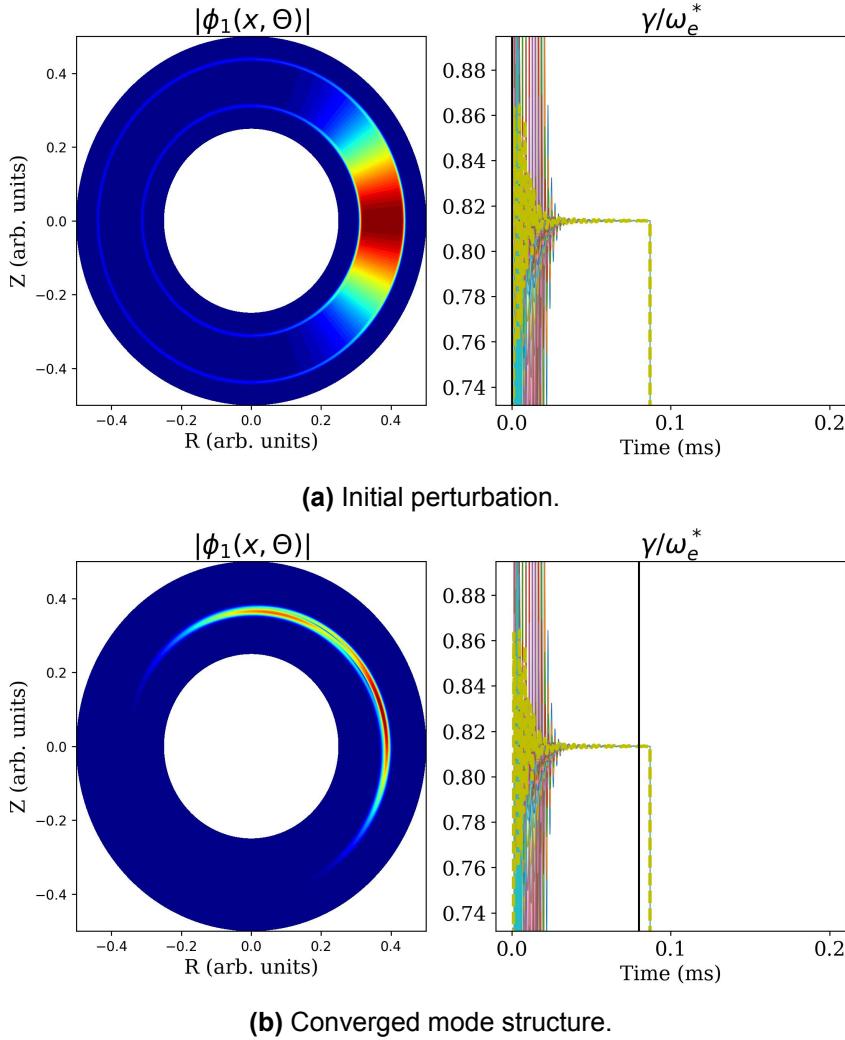


Figure 9: A transition towards the GM can be seen clearly when $\gamma_E = 7.5$.

Figure 8 shows the IM forming rapidly after the initial perturbation. The reason as to why the right-hand plot only covers a small amount of the simulation time, is that the mode had converged to within a predefined tolerance - this allows the code to end its execution early. This is the same data as that highlighted in figure 6b, i.e. the "best" result achieved by the GPR routine.

Figure 9 shows the formation of the GM. Due to the poloidal elongation of the mode, it is difficult to distinguish whether the mode has reached its zenith (in terms of poloidal rotation), but there are still clear signs of it rotating around the poloidal plane.

Figure 10 demonstrates that the sign of the flow shear dictates the direction in which the mode rotates. My observation was that positive flow shear resulted in anti-clockwise rotation of the mode, and negative flow shear the opposite, but I suspect that this is not a significant distinction - the key observation is that the directions of rotation are opposed, not the specific direction.

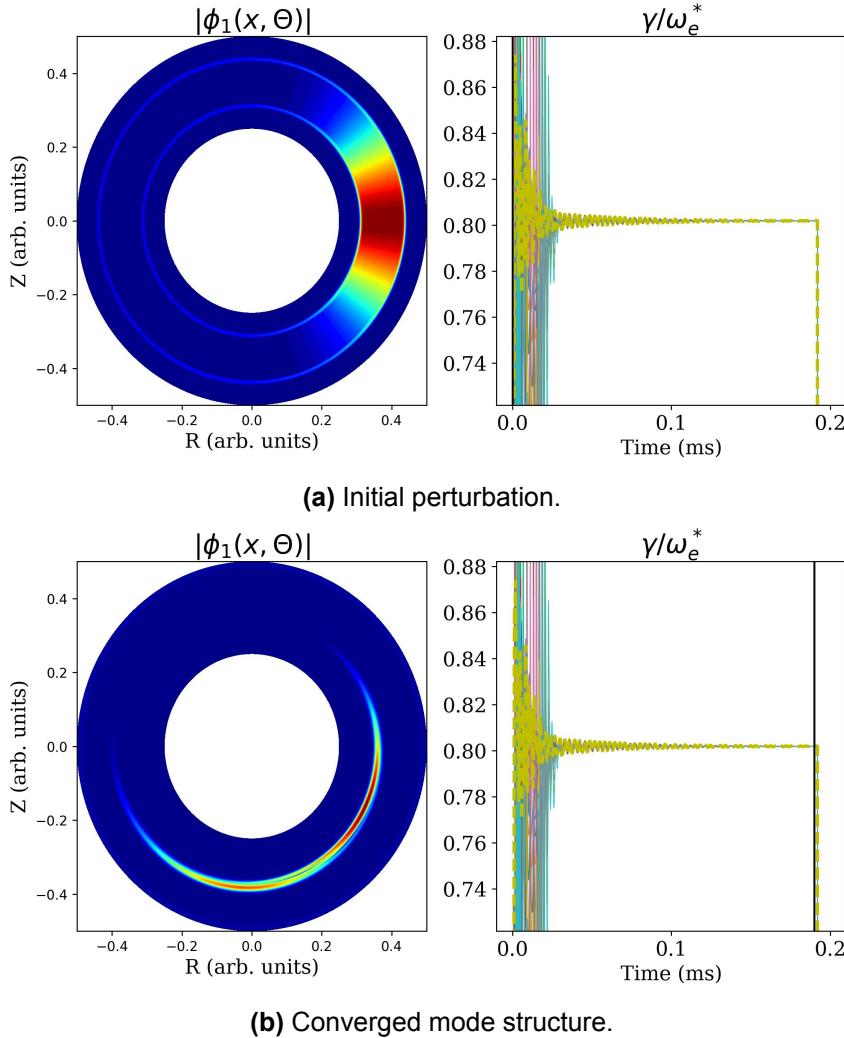


Figure 10: The GM forms at the top of the poloidal section with positive flow shear or, in the case shown here, at the bottom with negative flow shear. Shown here is the GM formed when $\gamma_E = -9.5$.

The final observation that I made was that the time taken for the growth rate to stabilise (i.e. converging to a value low enough for the code to finish executing) increased as the magnitude of the flow shear increased. This is shown (very approximately) in figure 11.

The shape of figure 11 appears to suggest that the number of time steps required for the mode to converge has an exponential relationship with the magnitude of the flow shear. This behaviour was also observed in the gyrofluid model, but as will be discussed in section 3.2.3, this became a limiting factor of the project and was quite problematic. It is interesting to me however, that both models exhibit the same behaviour.

I was unable to successfully measure any values beyond the range shown in figure 11. This is mostly due to the code execution being limited to 2000 time steps (by choice), and measuring a converged mode beyond the values of flow shear shown in figure 11 would require a drastic increase in computation time.

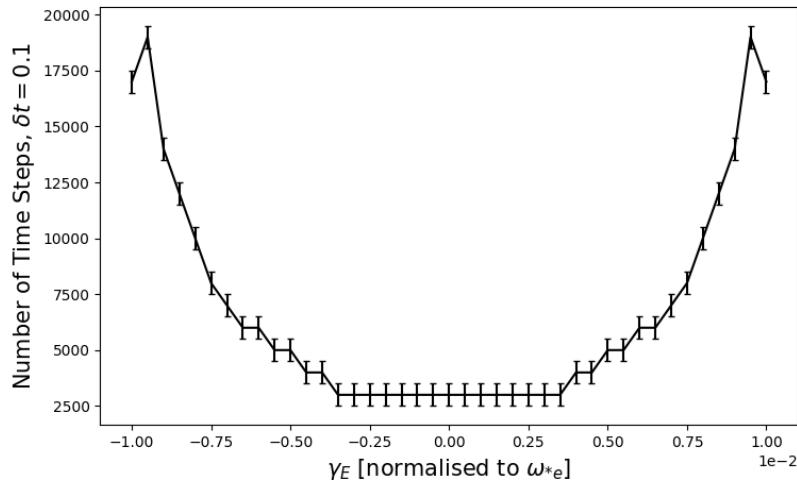


Figure 11: The number of time steps taken for the mode to converge, as a function of the flow shear. The uncertainty shown comes from the fact that the fluid model only exports data at discrete intervals.

3.1.3 Growth Rate as a Function of Flow Shear at Low Magnetic Shear

Taking measurements of the global growth rate and frequency of the converged modes, at varying values of flow shear, demonstrates the smooth evolution from the IM to the GM, and is shown in figure 12.

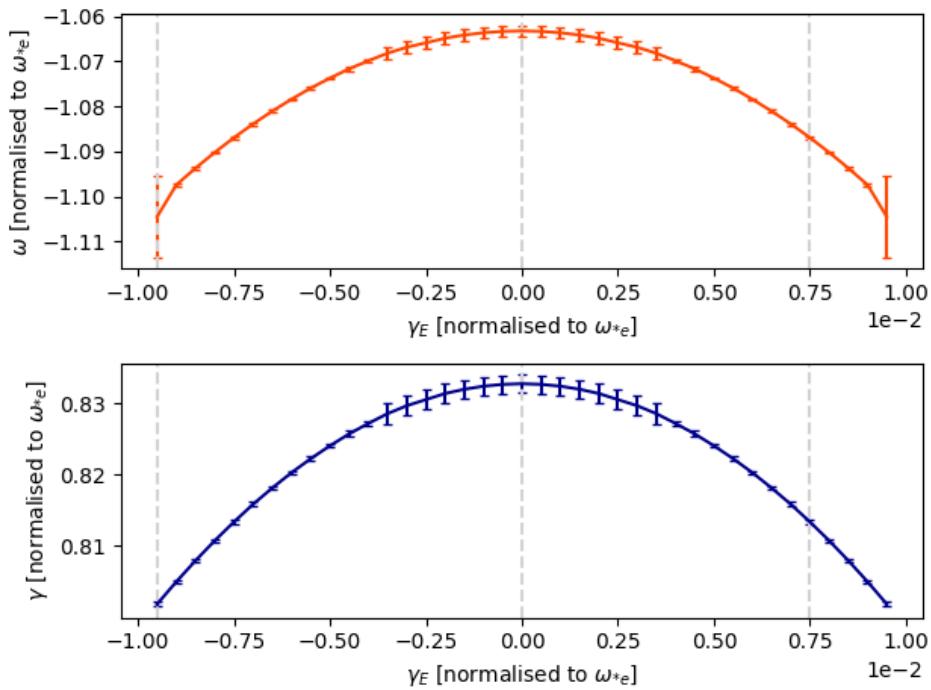


Figure 12: Global growth rate and frequency as a function of flow shear for the fluid model. The dashed lines correspond to values of flow shear used in figures 8, 9, and 10. The uncertainties shown in each plot is simply the standard deviation of the data used to calculate each mean.

The standard deviation in the mean varies quite substantially across the flow shear

spectrum. For the data points centred around the peak, there is a fairly straightforward explanation; the mode converges rapidly at low values of flow shear (see figure 8), so the variance of the data used to calculate the mean is increased. I decided to use the latter half of each data set to calculate the mean, in an attempt to omit the initial perturbation from the values being averaged.

My hypothesis for the sharp increase in uncertainty at the final values recorded for the global frequency (recorded at $\gamma_E = \pm 9.5$), is that the solution being calculated by the fluid model is jumping to a higher harmonic at or above these values of flow shear. For example, I ran the fluid model with a flow shear of $\gamma_E = 40.0$, which took five times longer for the modes to converge (around 10,000 time steps), and the global growth rate increased by around 25%. I believe therefore that the dominant mode being modelled in the range $-9.5 \leq \gamma_E \leq 9.5$ begins to become suppressed by another harmonic when the flow shear is set outside of that range.

It is notable that the difference between the growth rate at the IM and GM is very small. Bokshi et al. explain that this is due to the fluid model's assumption that the tokamak has a large aspect-ratio [1], and they suggest that the IM would be even more unstable if simulated in more realistic geometries [1].

3.2 Gyrofluid Model

3.2.1 Using the Singh Parameters

As mentioned in section 1.2, time became a major constraint of this project. I was unable to feasibly apply the GPR methods of parameter optimisation that I had developed for the fluid model to the gyrofluid model in the time that remained. However, as luck would have it, after conducting a literature review I discovered a paper published very recently (June 2023) by Singh et al. [21], in which they report on a set of plasma parameters, used in a gyrokinetic simulation, that produced a poloidially confined mode structure that would be ideal for my aim; to demonstrate the IM-GM transition using a gyrofluid code.

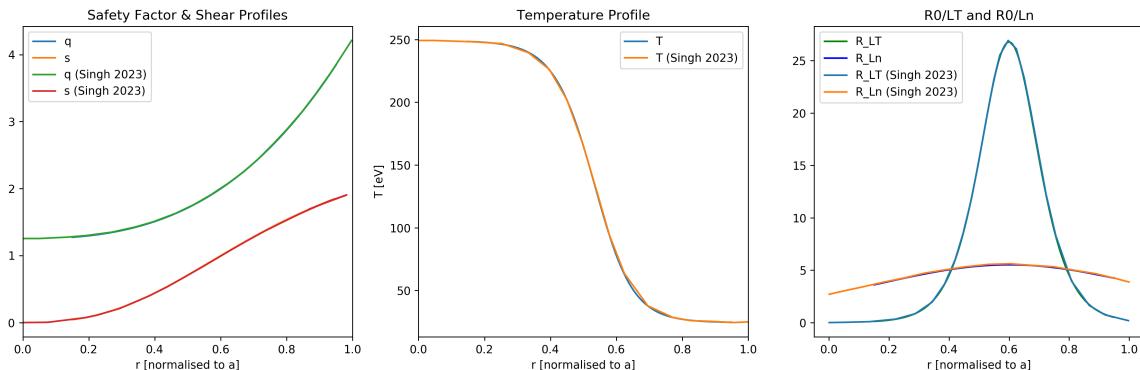


Figure 13: Plasma profiles for the gyrofluid model were matched to those published by Singh et al. [21]. For clarity, the right hand plot shows the profiles of R_0/L_n and R_0/L_T , where R_0 is the tokamak major radius, L_n is the density length scale, and L_T is the temperature length scale.

Matching the plasma profiles was relatively straightforward thanks to the clarity with which they were published [21]. Data from the plots in the paper by Singh et al. [21] was reverse engineered by using an online resource [22] in order for them to then be compared to the plasma profiles being supplied to the gyrofluid model. This comparison is shown in figure 13; I was able to achieve an almost exact match between all five plasma profiles. The grid file produced for the gyrofluid model, derived from these plasma profiles, was used in the procurement of the results discussed in sections 3.2.2 and 3.2.2.

3.2.2 Demonstrating the IM - GM Transition

In my analysis and use of the gyrofluid model, I found that the IM formed at a non-zero value of flow shear, shown in figure 14. This is expected, as the plasma profiles for safety factor, shear, temperature, and density now have a radial dependence, which has been shown to lead to a shift in the critical value of flow shear required to form the IM [1].

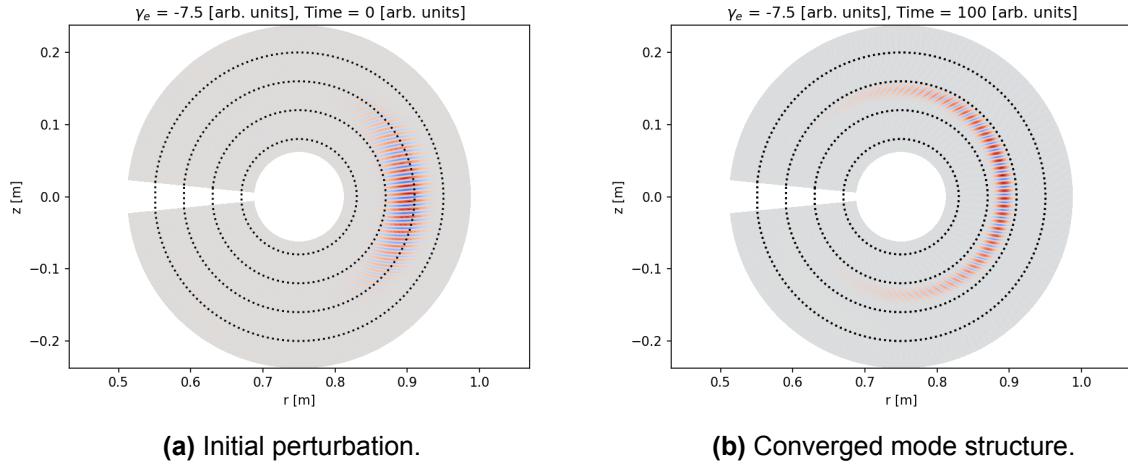


Figure 14: The IM forms at $\gamma_E \approx -7.5$. The resolution with which I studied the IM was severely limited by problems running the gyrofluid model.

The execution of the gyrofluid model proved to be a difficult aspect of collecting data for the IM's critical value of flow shear. A run of the gyrofluid model (bearing in mind that it is fully parallelised, running on 32 cores) would take several hours at low levels of flow shear. My supervisors and I suspected that this excessive computation time was being caused by the magnitude of the calculated growth rate exceeding the memory limit of a stored variable in C++. It was proposed that this could be remedied by regularly normalising the growth rate and including a scale factor when plotting the results. Alas, these problems were becoming apparent in the closing stages of my project, and we ran out of time to implement required fixes to the code. As a consequence, the resolution with which I was able to determine the IM's critical value of flow shear was relatively low, and I can only give an approximation of its value.

The mode structure evolved towards a GM as flow shear was increased, just as I had hoped, and as was shown with the reduced physics of the fluid model. The value of flow shear that I found showed this transition the most clearly was at $\gamma_E = -15.0$, the

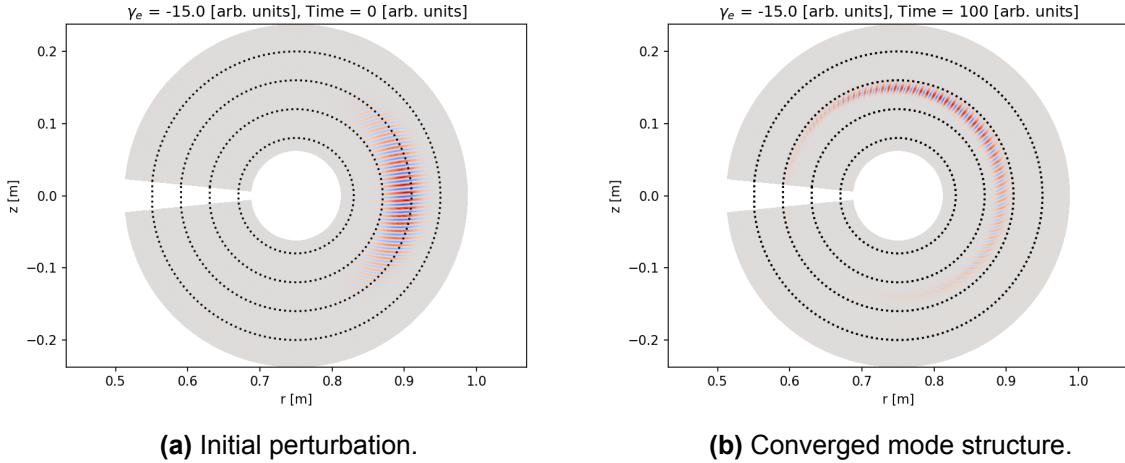


Figure 15: The GM, shown at $\gamma_E = -15.0$.

results from which are shown in figure 15. The moment where I saw these results for the first time was my personal highlight of the project.

Just as with the fluid model, I found that using relatively high values flow shear, e.g. $|\gamma_E| \geq 30.0$, would “over-drive” the mode, causing it to rotate wildly around the poloidal plane, and would have required much longer simulation times to settle into a GM.

3.2.3 Growth Rate as a Function of Flow Shear

Figure 16 shows the data that I was able to collect when performing a scan of the flow shear, using the gyrofluid model. I will be the first to admit that this data set is lacking somewhat in both its breadth and resolution, but I’ll reiterate that collecting this data was troublesome, and the writing of this dissertation demanded more of my attention at the time. Nevertheless, the data that I was able to collect shows a clear trend, very similar to that of the fluid model shown in figure 12.

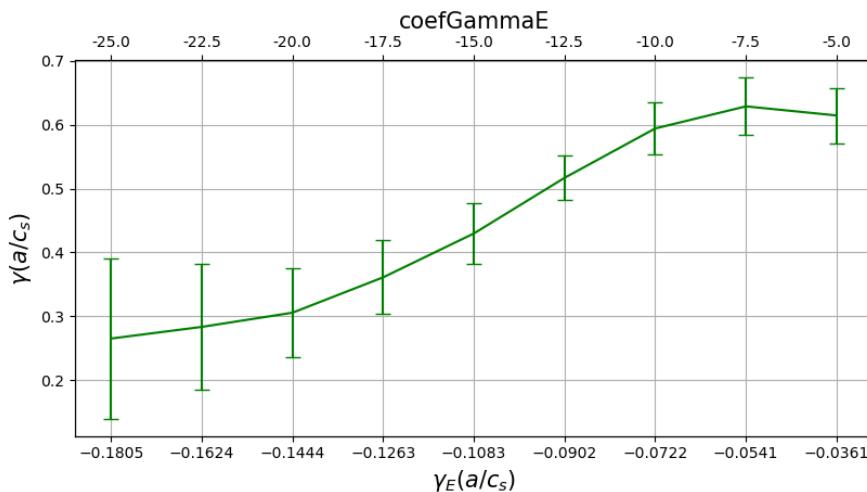


Figure 16: The global growth rate as a function of flow shear is shown, with uncertainties once again calculated from the standard deviation in the mean for each point.

It would have been ideal to have had time to collect data for the positive values of flow shear to mirror those that I did manage to record, with the hope of demonstrating that the value shown at $\gamma_E = -7.5$ is indeed the peak, i.e. the IM. Earlier on in the project, I did produce such a plot, but after consulting with my supervisors, we agreed that I had not run the simulation for long enough in this case (for context, I had used 20 time steps at the time, whereas the results shown in figures 14, 15, and 16 were all gathered from simulations that took 100 time steps). I would confidently predict that the plot, if extended to $\gamma_E = 25.0$, would be mirrored about the peak shown at $\gamma_E = -7.5$.

The most striking thing for me about figure 16, is that when compared to figure 12, the change in the global growth rate is much larger in the gyrofluid model than in the fluid model. This could be due to the differences in the way in which each y-axis was normalised, and it could also be due to the differences in how the mean of the growth rate was calculate in each instance. Tantalisingly however, it could also be evidence of the previously mentioned effect of the fluid model's large aspect-ratio assumption [1]. If that is indeed the case, then these results would certainly justify the speculation made by Bokshi et al.; that the IM would be more unstable if simulated using a more rigorous physics model [1].

4 Conclusions

In summary, I believe that I have primarily demonstrated two things:

1. That GPR is a highly effective machine learning tool when one is attempting to efficiently explore the parameter space of a complex function, and that it could potentially be applied to many areas of computational plasma physics.
2. The IM-GM transition phenomenon presented by Dickinson et al. [2] and Bokshi et al. [1] is still clearly present when extending into a more advanced physics model. The gyrofluid model seems to show the same transient burst in global growth rate (as the mode structure transitions, driven by flow shear, from the GM, through the IM, and back to the GM) as the fluid model, and it is plausible therefore that this rapid increase in growth rate may be associated with a small ELM.

Be that as it may, there are many things that I would still like to have achieved in this project. Given that I spent so much time on exploring the parameter space of the fluid model (using the `UpperConfidenceBound` [13] acquisition function), I would have liked to have focused in on some narrower sections of the parameter space using the `ExpectedImprovement` [13] acquisition function, in order to further investigate some of the more promising results, such as those found at low values of magnetic shear.

It is a shame that despite having spent a great deal of time and effort proving its effectiveness, I did not have time to implement the use of GPR to optimise the input parameters for the gyrofluid model. While time may have been the largest constraint, there were other challenges associated with this potential phase of the project as well. Firstly, the gyrofluid model's data output (e.g. like that shown in figure 15), consistently contained low levels of noise on the inboard side of the simulation domain. This meant that the approach I had developed to assess the fluid model's output, and thus return an objective function value for any prospective GPR routine, would had to have been adapted. Secondly, the parameter space that needed to be explored was unknown (or at least, not immediately apparent) to me, which meant that to explore it fully I would require an amount of computational time that would have exceed the length of my project. This computational restriction was further compounded due to the University's gradual phase-out of the Viking computing cluster, which became an increasingly unreliable source of computing power as the project progressed.

It was becoming clear in the final few days of the project that it would be feasible to run the gyrofluid model, for an adequate number of time steps, with the fixes made to the code by my supervisors. In an ideal world, I would have been able to split my concentration between writing this dissertation and carrying out more investigative work, but I did not find that to be feasible.

The natural extension to this project is to use a gyrokinetic model to capture the physics that is inherently lost in the gyrofluid model. While I agree that would be a valuable investigation to carry out, and publications like that of Singh et al. [21] show that there are gyrokinetic models available, I think it would be equally valuable to first add the non-linear terms that were omitted from the gyrofluid model, and see what effect they would have. The paper from Singh et al. discusses the effects of adding non-linear

terms to their gyrokinetic model, and the change in behaviour, particularly in the mode structure [21], is quite dramatic. The key questions I would like to address are: does the mode structure, despite the difference in shape and appearance, still transition from being maximally peaked at the outboard mid-plane (i.e. the IM) and move elsewhere in the simulation domain (i.e. form a GM); and does the change in growth rate (as a function of flow shear) still show the same transient increase as the flow shear passes through a critical value?

Overall I found this project to be challenging, yet highly rewarding. Another personal highlight was being given the opportunity to visit the UKAEA site in Culham, Oxfordshire, and having had the time to talk to and collaborate with physicists that are working at the cutting edge of fusion energy research. I hope that this dissertation was as enjoyable and instructive for you to read as it was for me to write.

References

- [1] A. Bokshi, D. Dickinson, C. M. Roach, and H. R. Wilson, “The response of toroidal drift modes to profile evolution: a model for small-ELMs in tokamak plasmas?,” *Plasma Physics and Controlled Fusion*, vol. 58, no. 7, p. 075011, 2016.
- [2] D. Dickinson, C. M. Roach, J. M. Skipp, and H. R. Wilson, “Structure of micro-instabilities in tokamak plasmas: Stiff transport or plasma eruptions?,” *Physics of Plasmas*, vol. 21, no. 1, p. 010702, 2014.
- [3] J. Wesson, *Tokamaks; 3rd ed.* Oxford University Press, 2004.
- [4] J. B. Taylor, H. R. Wilson, and J. W. Connor, “Structure of short-wavelength drift modes and transport in a toroidal plasma,” *Plasma Physics and Controlled Fusion*, vol. 38, no. 2, p. 243, 1996.
- [5] G. Duesing, H. Altmann, H. Falter, A. Goede, R. Haange, R. S. Hemsworth, P. Kupschus, D. Stork, and E. Thompson, “Neutral beam injection system,” *Fusion Technology*, vol. 11, no. 1, pp. 163–202, 1987.
- [6] A. W. Leonard, “Edge-localized-modes in tokamaks,” *Physics of Plasmas*, vol. 21, no. 9, p. 090501, 2014.
- [7] A. Loarte, G. Saibene, R. Sartori, D. Campbell, M. Becoulet, L. Horton, T. Eich, A. Herrmann, G. Matthews, N. Asakura, A. Chankin, A. Leonard, G. Porter, G. Federici, G. Janeschitz, M. Shimada, and M. Sugihara, “Characteristics of type I ELM energy and particle losses in existing devices and their extrapolation to ITER,” *Plasma Physics and Controlled Fusion*, vol. 45, no. 9, p. 1549, 2003.
- [8] C. Ma and X. Xu, “Global kinetic ballooning mode simulations in BOUT++,” *Nuclear Fusion*, vol. 57, no. 1, p. 016002, 2016.
- [9] M. A. Meier, R. D. Bengtson, G. A. Hallock, and A. J. Wootton, “Adiabatic electron thermal pressure fluctuations in tokamak plasmas.,” *Physical review letters*, vol. 87, no. 8, p. 085003, 2001.
- [10] G. W. Hammett, M. A. Beer, W. Dorland, S. C. Cowley, and S. A. Smith, “Developments in the gyrofluid approach to tokamak turbulence simulations,” *Plasma Physics and Controlled Fusion*, vol. 35, no. 8, p. 973, 1993.
- [11] B. Dudson, M. Umansky, X. Xu, P. Snyder, and H. Wilson, “BOUT++: A framework for parallel plasma fluid simulations,” *Computer Physics Communications*, vol. 180, no. 9, pp. 1467–1480, 2009.
- [12] C. E. Rasmussen, C. K. Williams, *et al.*, *Gaussian processes for machine learning*, vol. 1. Springer, 2006.
- [13] C. Bowman, “Gaussian process regression, optimisation and inversion.” <https://inference-tools.readthedocs.io/en/stable/gp.html>, 2019. [Accessed: 25/08/2023].
- [14] D. Gogolashvili, B. Kozyrskiy, and M. Filippone, “Locally smoothed gaussian process regression,” *Procedia Computer Science*, vol. 207, pp. 2717–2726, 2022.

- [15] T. Osborne, P. Snyder, K. Burrell, T. Evans, M. Fenstermacher, A. Leonard, R. Moyer, M. Schaffer, and W. West, “Edge stability of stationary ELM-suppressed regimes on DIII-D,” *Journal of Physics: Conference Series*, vol. 123, p. 012014, 2008.
- [16] M. Magnusson, A. Vehtari, J. Jonasson, and M. Andersen, “Leave-one-out cross-validation for bayesian model comparison in large data,” in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (S. Chiappa and R. Calandra, eds.), vol. 108 of *Proceedings of Machine Learning Research*, pp. 341–351, 2020.
- [17] H. Cheng, D. J. Garrick, and R. L. Fernando, “Efficient strategies for leave-one-out cross validation for genomic best linear unbiased prediction,” *Journal of Animal Science and Biotechnology*, vol. 8, no. 1, p. 38, 2017.
- [18] A. Geroldinger, L. Lusa, M. Nold, and G. Heinze, “Leave-one-out cross-validation, penalization, and differential bias of some prediction model performance measures—a simulation study,” *Diagnostic and Prognostic Research*, vol. 7, no. 1, p. 9, 2023.
- [19] M. Yoshida, G. McKee, C. Petty, B. Grierson, M. Nakata, C. Rost, T. Rhodes, D. Ernst, and A. Garofalo, “Magnetic shear effect on plasma transport at $\text{Te}/\text{Ti} \sim 1$ through electron cyclotron heating in DIII-D plasmas,” *Nuclear Fusion*, vol. 61, no. 1, p. 016013, 2020.
- [20] A. C. C. Sips, for the Steady State Operation, and the Transport Physics topical groups of the International Tokamak Physics Activity, “Advanced scenarios for ITER operation,” *Plasma Physics and Controlled Fusion*, vol. 47, no. 5A, p. A19, 2005.
- [21] A. Singh, J. Mahapatra, J. Chowdhury, D. Aggarwal, T. Hayward-Schneider, R. Ganesh, E. Lanti, and L. Villard, “Gyrokinetic simulation of short wavelength ion temperature gradient instabilities in the ADITYA-U tokamak,” *Nuclear Fusion*, vol. 63, p. 086029, 2023.
- [22] A. Rohatgi, “Webplotdigitizer: Version 4.6.” <https://automeris.io/WebPlotDigitizer>, 2022. [Accessed: 11/08/2023].

A Gaussian Process Regression Loop

The following is written in python, and is a pseudo-code representation of the GPR loop that I used in my project:

```
1 # Import the required tools from Dr. Chris Bowman's inference-tools package
2 from inference.gp import GpOptimiser, UpperConfidenceBound
3
4 # Define an "objective function" that the GpOptimiser class will
5 # try to maximise:
6 def objective_function(number: float) -> float:
7     return number * -1.0
8
9 # Create an instance of the GpOptimiser class:
10 # - The x-data needs to have shape (number of points, number of dimensions)
11 # - The y-data needs to be a 1-D array
12 # - The bounds define the parameter space of the x-variables
13 # - The type of acquisition function can be selected
14
15 # The minimum number of data points that you can initialise the GpOptimiser
16 # class with is 2
17 x = [...]
18 y = [...]
19 bounds = (lower_bound, upper_bound)
20
21 GPO = GpOptimiser(x, y, bounds=bounds, acquisition=UpperConfidenceBound)
22
23 # Create an NDArray that spans the length of your parameter space
24 x_gp = [...]
25
26 # Store the current state of the system for plotting later
27 mu, sig = GPO(x_gp)
28 means = [mu]
29 sigmas = [sig]
30 acquis = [array([GPO.acquisition(k) for k in x_gp])]
31
32 # Create an instance of the simulation that you want to run:
33 simulation = ToroidalFluidITG()
34
35 # This is where the magic happens...
36 for i in range(iterations):
37     # Request the proposed evaluation
38     new_x = GPO.propose_evaluation()
39
40     # Evaluate the new point
41     new_y = objective_function(simulation.run())
42
43     # Update the gaussian process with the new information
44     GPO.add_evaluation(new_x, new_y)
45
46     # Store the current state of the system for plotting later
47     mu, sig = GPO(x_gp)
48     means.append(mu)
49     sigmas.append(sig)
50     acquis.append(array([GPO.acquisition(k) for k in x_gp]))
```