Developing Sparse Grid Surrogate Models to Aid In Characterization of Uncertainty in High Fidelity Plasma Simulations

Author: Evan Shapiro

Advisor: Dr. Varis Carey

M.S. Integrated Sciences - CU Denver

May 11, 2020

1 Introduction

The history of nuclear fusion research dates to the 1920's, when Arthur Eddington suggested in his book "The Internal Constitution of Stars" that the transmutation of hydrogen atoms through nuclear fusion into helium is responsible for powering the energy released by the sun.[1]. Eddington's hypothesis motivated experimental research with the goal of ceating the conditions necessary for nuclear fusion. In 1934, Ernest Rutherford and his assistant Mark Oliphant successfully fused 2 deuterium atoms in the following reaction:

$${}^{2}_{1}\mathrm{H} + {}^{2}_{1}\mathrm{H} \longrightarrow {}^{3}_{2}\mathrm{He} + {}^{1}_{0}\mathrm{n} + 3.74\,\mathrm{MeV},$$

where the 3.74 MeV has been converted from nuclear binding energy to kinetic energy,making it an energetically favorable nuclear reaction.[2]. Nuclear fusion of deuterium ${}_1^2H$ and tritium ${}_1^3H$ is more energetically favorable than D-D fusion, and releases more kinetic energy per nucleon than that of nuclear fission. The D-T fusion chain is a safer energy source than fission, since the fuel is radioactively stable and cannot be weaponized. It is also a environmentally safe or clean energy source relative to fossil fuels. The energetic favorability and current climate, both political and literal, explain why nuclear fusion is, and has been for almost a century, an active area of energy research, with the goal of developing a reactor that can sustain and contain nuclear fusion in order to provide clean and safe energy.

While it nice to be idealistic about the amount of energy that nuclear fusion can provide, designing and engineering a device that can create the conditions necessary to sustain nuclear fusion in a net energy positive manner has been a major technical challenge. To make fusion energetically viable the pressure and temperature inside of the plasma must provide enough power to sustain nuclear fusion to some degree, offsetting external heating requirements. Thus it is important to understand the physical mechanisms that can degrade the plasma power inside reactors at operational scale. It is know that the physical mechanism driving the majority of heat transport is the large ion temperature gradient that can exist between the core of a fusion reactor and the reactor walls [3].
Through the use ofcurrent experimental fusion reactors dimensionless scaling relationships have been developed that predict almost all of the conditions that will exist in future operational reactors. However, in current experimental reactors the dimensionless plasma size, $\rho*$, cannot be matched to future operational reactors. This lack of empirical reference at operational scales has led to the development of physical models that produce the underlying physics causing ion temperature gradient driven transport. 5-D Gyrokinetic models have been developed that characterize ion and electron kinetic behavior and collisions in a tokamak environment well enough to predict the transport phenomenon at the scale we are interested in. With the aid of modern computer gyrokinetic simulations, XGC1, and modern parallel computing power at NERSC, we can accurately solve the 5-D equations.

Ion temperature gradient driven heat transport scaling has been shown, under certain conditions, both experimentally and via simulations, to occur in a linear fashion (Bohm-like) with plasma size and power. If true, this will require linear scaling of auxiliary heating with plasma size and power, and is thus important to fully understand transport levels. While experimental and numerical work has been done to predict transport scaling [4], little to no work has been done the developing bounds, or developing enough data to perform statistical analysis on the outputs from the simulations. Experimentally derived ion and electron density and temperature profiles are used as inputs for the gyrokinetic simulations. These profile fits and associated uncertainty bounds are derived using mathematically rigorous fitting methods, i.e. Gaussian process regression [5], from data produced by the Alcator C-Mod III fusion reactor, which means that simulation outputs are random variables. Thus we need to run enough simulations in order to characterize the uncertainty in the simulation results.

To make accurate predictions with these profiles it is necessary to consider the uncertainty bounds, as transport has been shown to be sensitive to variations in the density and temperature profiles. In this project we seek to understand how the uncertainty bounds effect ion heat transport in a tokamak fusion reactor. The amount of time it takes to perform a simulation take brute force sampling methods like Monte Carlo sampling out of consideration for characterizing the transport bounds. To overcome this, we seek to construct a model surrogate to the gyrokinetic model. The intended model surrogate is a sparse grid polynomial interpolant, which will allow us to use far fewer points sampled from the ion temeperature profile parameter space than Monte Carlo sampling would. This model surrogate will allow us to calculate moments of the ion phase space distribution function to calculate the quantity of interest, the expected heat flux at a particular radial distance in the reactor, and other moments of the distribution function.

2 Background

2.1 Nuclear Fusion

Nuclear fusion occurs when the nucleii of two elements combine, and, in the process, produce a heavier element, convert nuclear binding energy into kinetic energy, and release subatomic particles like neutrons. Fusion reactions that release energy occur between lighter elements on the periodic table, like the fusion between two hydrogen particles or two helium particles. For two elements to fuse together, enough kinetic energy must be imparted to the particles to overcome the repelling Coulomb potential barrier, allowing the particles to get close enough for the strong nuclear force to take over and fuse the two particle's nucleii. The radial distance at which fusion will occur is referred to as the impact parameter, β. [6]β is a complex parameter, as it is not only dependent on the physical distance of the fusion particles, but on their relative velocities, since the smaller the relative velocity of the two particles the longer the fusion particles can stay in the vicinity of each other, thus increasing the likelihood of a fusion event. The impact parameter is also affected by quantum tunneling, which, due to the wave properties of the particles, increases the probability of fusion when the wave functions of two particles overlap.

The nuclear fusion reaction that is of primary interest in fusion energy research is the deuterium($^2_1$H)-tritium$^2_1$H, as this reaction has a higher β at lower kinetic energies than other light element nuclear fusion reactions, and releases the most energy of all nuclear fusion reactions. The increase of kinetic energy during a fusion event is what motivates nuclear fusion research, as, at with enough nuclear fusion events, this process can yield enough energy to power a steam turbine to produce electricity, potentially providing a clean, safe source of energy.

The deuterium-tritium fusion reaction is described by the following reaction equation:

$$^2_1\mathrm{H} + {}^3_1\mathrm{H} \longrightarrow {}^4_2\mathrm{He}(3.5\,\mathrm{MeV}) + {}^1_0\mathrm{n}(14.1\,\mathrm{MeV}),$$

where $^2_1$H is deuterium (D), 3 $_1$H is tritium (T), $^4_2$He is an α particle, and $^1_0$n is a neutron. Taking into account the binding energy of each element, the mass of the system before and after this nuclear fusion event is given by:

$$(2-0.000994)\,\mathrm{m}_p + (3-0.006284)\,\mathrm{m}_p \longrightarrow (4-0.0427404)\,\mathrm{m}_p + (1+0.001378)\,\mathrm{m}_p$$

where $m_p = 1.6726 * 10^{-27} kg$ is the proton mass. The difference in mass before and after the reaction is thus $\delta_m = -0.01875 m_p$.

Using Einstein's famous equation $\mathcal{E} = mc^2$, where c is the speed of light, we find that energy-mass conversion to be

$$\mathcal{E} = \delta_m * c^2 = 0.01875 * 1.6726 * 10^{-27} kgc^2 = 17.9 MeV,$$

where the mass has been converted to kinetic energy [7]. A further calculation shows that $3.5 MeV$ of this energy is distributed to the α particle, and the other $14.1 MeV$ is distributed to the neutron. The high energy neutron that is released during D-T fusion is the particle of interest in fusion energy research, as if enough fusion events occur the neutrons can be leveraged to produce electricity via a steam turbine and a transformer.

To produce enough energy nuclear fusion is typically employed using D-T gas that a density of $10^{20} m^{-3}$, which must be raised to temperature on the order of 150 million degrees Celcius. At these temperatures the deuterium and tritium In a fusion reactor these two particles serve two important, and different, functions. While the deuterium, tritium, and α particles are confined to stay in the plasma When the number of fusion events is scaled to a large enough level, the resulting number of high energy neutrons are allowed be leveraged to power a steam turbine, which through the inductive process, produces electricity. The high kinetic energy of the α particles are used to increase the the kinetic energy of supplied D-T fuel, reducing the energy required for continuous external heating. creating a self-sustaining fusion process.

Since both deuterium and tritium have a net positive charge there is a repelling Coulomb force between them that prevents fusion from occuring in at low kinetic energieslow temperature environments. Additionally in low temperature environments the cross sectional area of the D-T mixture is too low to make fusion an energy positive process. To overcome the Coulomb barrier, and increase the effective cross-section, the deuterium and tritium must be supplied enough kinetic energy to surpass the potential energy barrier to get within a particular radial distance from each other.

In equillibrium, the kinetic energy of the D-T fuel mixture in a fusion reaction follows a Maxwellian distribution, and the high energy particles in the upper tail of the distribution are the source of the fusion reactions. The necessary average temperature to achieve desired fusion rates in this tail is 10keV, or around 100 million degrees centigrade. [1] The high temperature environment leads to complete ionization of the D-T mixture, yielding an equal number of electrons n_e and ions n_i. The net equality of charge leads to an electrically neutral system in equillibrium, which is highly responsive to external electric and magnetic fields; a gas in this state is referred to as a plasma.

We would like to keep all of the fusion particles confined inside of the reactor, however the high energy and temperature of the plasma particles preclude using the physical wall of the reactor for confinement.[1] There are two important factors that must be considered when designinng a device to confine a fusion plasma: the plasma state of the plasma the fact that we want to prevent the high energy particles of the plasma from colliding with the material walls of the device. Through the Lorentz force, the ions and electrons of the plasma can be confined with strong magnetic fields interior to the fusion device. A tokamak reactor is a toroidal nuclear fusion device that generates helical magnetic fields that provide a confining path for the ions and electrons of the plasma. The path traversed by the ions and electrons prior to reaching a material wall is a million times the length of the reactor.

3 Power Balance

"A positive energy balance is possible if the fuel particles can be made to interact to lose their energy. To achieve this the particles must retain their energy and remain in the reacting region for a sufficient time." This leads to a necessary confinement time of a sufficient number of particles within a particular region, or number density n, to create the conditions necessary for sustained fusion. The confinement time, τ_E, is the primary parameter of interest in this project, as we are studying diffusive processes that degrade the experimental confinement time below the confinement time necessary for sustained fusion.

For a fusion reactor to be viable from both an energy and economic persepective it must meet power balance condition: $P_{out} >> P_{in}$, where P_{out} is the power retrieved from the high energy neutrons, and P_{in} is the power necessary to sustain these fusion process while the reactor is in operation. [2] If the power necessary to sustain the fusion process is of the same order as the power recovered from the high energy neutrons, then our process will not be a worthy investment. Thus, understanding the input power P_{in} required to sustain nuclear fusion in a working scale tokamak reactor is of critical importance. Predicting P_{in} requires that we account for all of internal fusion plasma power sources, P_{source}, and powers sinks, P_{sink}, which we will do next. Once a full accounting of the sources and sinks is completed, the conditions necessary for a self-sustained fusion reaction can be determined.

Basically, consider a heat bath The primary internal fusion power source are the α-particles that are created during the fusion reaction. The magnetic fields generated by the fusion reactor confines the positively charged α-particles. Under confinement, these α-particles transmit the 3.5MeV they obtained during the fusion reaction to incoming D-T fuel, imparting enough kinetic energy for D-T fusion to occur. Taking into account α-particle velocity, cross-sectional area as a function of velocity $\sigma(v)$, total ion density n, and energy per α-particle, E_α, the α-particles heating or power per unit volume is

$$p_\alpha = \frac{1}{4} n^2 < \sigma v > E_\alpha,$$

with $< \sigma v >$ being the fusion reaction rate assuming a Maxwellian distribution function. Integrating over the entire volume yields the total α-particle power in the plasma.

$$\begin{aligned} P_\alpha &= \int_V \frac{1}{4} n^2 < \sigma v > E_\alpha d^3x \\ &= \frac{1}{4} \overline{n^2 < \sigma v >} E_\alpha V. \end{aligned}$$

with V being the volume of the reactor.

Now we account for power losses in the plasma. Due to temperature gradients and Coulomb collisions, in a tokamak reactor power is continuously leaving the plasma as ions and electrons move from the plasma to the reactor material walls. This lost power should ideally be completely replaced by α-particle heating, and if it is not then external heating must be used to replace the lost power. "The average energy of plasma particles at a temperature T is $\frac{3}{2}T$..." [1]. The electrons and ions have the same temperaturee and quantitity, so if we let the ion number density per unit volume be n, the kinetic energy density of the system is $3nT$. Integrating over the entire volume yields the following total plasma energy

$$\begin{aligned} W &= \int_V 3nTd^3x \\ &= 3\overline{nT}V. \end{aligned}$$

We define the characteristic time τ_E to be the amount of time that heating power, once generated, stays in the plasma. This parameter is very important in nuclear fusion, as it has a minimum threshold that guarantees ignition, or self-sustaining fusion in a fusion reactor.

The amount of heating power that leaves is given by

$$P_H = \frac{W}{\tau_E}.$$

The yields the following power balance equation

$$P_H + P_\alpha = P_L$$

$$P_H + \frac{1}{4}\overline{n^2 < \sigma v >}E_\alpha V,$$

where P_H is the applied heating power.

We now want to consider the power in a fusion reactor after fusion has been initiated, and determine the conditions necessary for self sustaining fusion. For self sustained fusion, there must be enough α-particle heating power to sustain the fusion process, and negate the need for external heating. Thus, in the power balance equation, this means letting $P_H = 0$ and rearranging the power balance equation to arrive at the following inequality

$$n\tau_E > \frac{12}{< \sigma v >}\frac{T}{E_\alpha}.$$

Considering an operating temperature range of 10-20 keV, $< \sigma v >= 1.1 \times 10^{-24}T^2m^3s^{-1}$, we arrive at the following condition for a self-sustaining plasma

$$nT\tau_E > 5 \times 10^{21}m^{-3}keVs$$

in which the density and temperature profile inside the plasma are parabolic, and n, T are the peak values of the profile. Given an operating temperature of 10 keV, and a number density of 10^20m^{-3} yields a time

constant of $\tau_E = 5$ s, or the required energy confinement time in a tokamak reactor is about 5 s. This implies that, if all the power lost from the system is to be balanced by α particle heating, that the characteristic time scale of heat transport should be on the order of 5 seconds. Thus, we need to fully understand all of the physical transport mechanisms inside of the fusion reactor that degrade these conditions, and
Confining energy in the plasma for a time τ_E means that conditions must exist in the reactor that constrain heat from flowing at a rate faster than that dictated by the ignition time. This is one of the major problems being adressed in current nuclear fusion physics: predicting the rate at which heat leaves a fusion plasma via ions and electrons under operational conditions in an ITER scale reactors.

12/5/2019 2:24 PM Ion temperature gradient driven turbulent transport has been shown to scale in Bohm-lik manner is full-f simulations We describe how ion temperature gradients cause turbulent transport via low frequency drift modes, and develop the mathematical model that accurately describes plasma transport at the particle level.

4 Important Tokamak Fusion Plasma Definitions & Behavior

To understand the mechanisms driving heat transport out of a tokamak reactor we must establish the general characteristics of a plasma, and how plasma particles react as they experience the various forces and pressures that exist in the plasma. This section is intended to establish the basics of plasma behavior in a tokamak setting.

4.1 Tokamak Definition & Coordinate System

While there are a few types of experimental nuclear fusions reactors, i.e. stellarators and tokamaks, working fusion reactors are all predicted to be tokamaks. Tokamak is the Russian abbreviation "toroidal'naya kamera s magnitnymi katushkami", which translates to toroidal chamber with magnetic coils. Tokamaks were developed in Russia in the 1950s [?], and adopted as the standard model for nuclear fusion reactors. This is because, with the appropriate confining magnetic field configuration, tokamaks allows for continuous motion of the plasma in the toroidal direction, increasing the plasma confinement time. The appropriate magnetic field configuration, as seen below, is a combination of a toroidal magnetic field, B_ϕ, and a poloidal magnetic field, B_p, resulting in a helical magnetic field. The toroidal magnetic field is generated by coils that surround the tokamak in the poloidal direction, while the the poloidal magnetic fields are induced by a plasma current that flows in the toroidal direction as a result of a transformer at the center of the toroid. Ideally, without instabilities in the plasma, the strength of the helical magnetic fields provides enough force to balance out the plasma pressure, p, which is a product of the particle density and temperature. The magnetic field strength at ITER scale reactors will be on the order of 13 Teslas [**?**].

Nuclear fusion and heating occurs at the core of the fusion plasma resulting in the high temperatures described in the introduction. At the edge of the plasma, close to the reactor walls, the density falls as particles either recombine or are diverted and falls at the edge of the reactor as the ion and electron density Steep ion temperature gradient between the core of the reactor and the edge are necessary. Transport is driven by Coulomb collisions, temperature gradients, and electromagnetic field interactions and fluctuations that perturb the plasma particle velocities away from a Maxwellian distribution function. Thus we need a kinetic theory that predicts the the evolution of the phase space coordinates of the perturbed plasma particles in time. Since coliision frequency is depedent on relative velocities, this theory involves considering all of the local forces acting on all particle velocities in the phase space, which leads to the evolution of the particle distribution function using the Vlasov-Boltzmann equation. The Fokker-Planck collision operator is also considered to account for different species collisions, as well as the transformation of the Vlasoz equation to time averaged gyrokinetic coordinates, since this time averaging reduces the Vlasov system from 6 dimensions to 5 dimensions.

4.2 Energy-Temperature Relatedness

In equillibrium, a plasma particle species j with mass m_j, has a Maxwellian distribution function

$$f(\mathbf{v}) = Aexp(-\frac{1}{2}m_j\mathbf{v}^2/K_BT_j).$$

Taking the second moment, we get that the average kinetic energy has the following relation to temperature:

$$E_{av} = \frac{3}{2}K_BT$$

where $K_B = 1.38 * 10^{-23} J/^{\circ}K$ is Boltzmann's constant. The fact that temperature and average kinetic energy are so closely related means that in plasma literature temperature is given in terms of energy, so in the case of 10 eV plasma, $KT = 10eV$.

4.3 Nuclear

To produce a large enough tritium and deuterium cross sectional area to yield the D-T fusion reaction rate that generates an appropriate amount of fusion power, the average kinetic energy of the individual D-T particles must be around 10 *kev* [**?**]. This will yield enough particles with the necessary energy, 70 keV, [**?**] at the tail end of the Maxwellian distribution. A kinetic energy energy of 10 keV corresponds to D-T temperatures on the order of 100° Celcius, which means that the D-T fuel and α particles are completely ionized, and meet the conditions necessary to be considered a plasma. Plasmas have a few important characteristics that define their behavior, that we discuss here. Plasma exhibit collective behavior due to the fact that they are charged and produce long range electric fields that can interact with distant parts of the plasma. Additionally, motion of the plasma particles produces an electrical current, which in turn generates magnetic fields, which can influence the behavior of distant plasma particles. [**?**].

4.4 Debye Shielding & Length

In a tokamak plasma, the density of the ions and electrons is on the order of $10^{20}\mathrm{m}^{-3}$, which means that only small separations of charge occur due to the strong electric forces generated from the separation. Consider now a sheet of electrons of width d, separated from a sheet of electrons of width d, and a number density of electrons and ions of n. Separation of ions from electrons by a distance d exerts a force per unit area$F$ such that

$$F \sim \frac{(dne)^2}{\epsilon_0},$$

which means in a tokamak that

$$F \sim 10^{13} d^2 N m^{-2}.$$

For a sheet of width $1mm$ this yields a force per unit area of $10^7 Nm^{-2}$. This strong electric force density leads the electron and ion densities being almost equivalent everywhere in the plasma, or they are held in a quasi-neutral state.

The Debye length is the fundamental length of a plasma, and it is characterized as the length or thickness "for which the internal energy of the plasma can provide the energy for a complete separation of the ions and electrons." [?] It is also defined in terms of an positively charged spherical surface, with surface area A. When introduced into the plasma, electrons will move to cover the surface in order to block out the electric field, the width of the sheet of electrons shielding the large ion will be on the order of the Debye length. The Debye length is given by

$$\begin{aligned}\lambda_D &= (\frac{\epsilon_0 T}{ne^2})^{1/2} \\ &= 2.35 \times 10^5 (\frac{T}{n})^{\frac{1}{2}})m\end{aligned}$$

with the units of T in keV. We can see from this formula that the Debye length increases with temperature and kinetic energy, and decreases with density. The Debye length is important when considering short range collisions between plasma particles.

Solving Poisson's equation for the electrical potential in the radial distance away from an ion shielded in a plasma yields

$$\phi(r) = \frac{e}{4\pi\epsilon r} exp(-\sqrt{2}r/\lambda_d).$$

Using this information, we can imagine that, in a quasi-neutral plasma in an equillibrium state, to shield out the electric fields, the plasma electrons surround the plasma ions in a sphere, with a shielding width that is the length of the Debye radius. We refer to this sphere as a Debye sphere. When an electric field with a wavelength λ much greater than the Debye length, or $\lambda_D << \lambda$, is introduced into a region of the plasmas, the charged particles in the Debye sphere will oscillate collectively at the frequency, ω, of the electric field.

4.5 Plasma Frequency

Due to the completely ionized state, plasmas are good conductors, so electrons respond immediately to cancel out any local or externally produced electric field. This allows us to define the characteristic frequency of a plasma. Consider an electrically neutral region of a plasma - if electrons in a plasma in the region are displaced, due to an electromagnetic field or thermal effects, this will cause a charge separation from the ions. This charge separation generates an electric field, that exerts a restoring force on the electrons, and works to move the electrons back to cancel out the electric field. The momentum of the electrons will move them past their original position, "allowing them to recreate their charge separation" [**?**]. This cycle repeats itself, yielding a plasma oscillation that oscillates at the characteristic electron plasma frequency $\omega_p = (\frac{ne^2}{\epsilon_0 m_e})^{1/2}$. The ion oscillation frequency is similarly defined to be $\omega_{pi} = (\frac{ne_i^2}{\epsilon_0 m_i})^{1/2}$.

4.6 Larmor Orbits

In a magnetic field $\mathbf{B}$, particles with charge e_j, where j denotes the species of charged particle, a mass m_j, moving with a velocity $\mathbf{v}$in a region of the field obey the following equation of motion:

$$m_j \frac{d\mathbf{f}}{dt} = e_j \mathbf{v} \times \mathbf{B}.$$

Since we are only concerned with the direction of motion of the charged particle with respect to the magnetic field, we decompose the particles velocity into components that are perpendicular and parallel to the magnetic field:

$$\mathbf{v} = \mathbf{v}_\perp + \mathbf{v}_{||}$$

If the velocity has a component that is perpendicular to the magnetic field, $\mathbf{v}_\perp$, then the charged particle will undergo rotational motion around the field line with frequency $\omega_{cj} = \frac{e_j B}{m_j}$, at a distance $\rho_j = \frac{m_j v \perp}{e_j B}$, which is the Larmor radius of orbit. A thermal charged particle, with a temperature T_j, has a Larmor radius of $\rho_j = \sqrt{2}\frac{m_j v_{T_j}}{|e_j|B}$, where the $\sqrt{2}$ is due to the particle having two degrees of freedom while it undergoes rotational motion. Thus, the general trajectory of a charged particle in a magnetic field is helical, as it travel with a velocity $v||$ along the magnetic field, and with a rotational velocity $v_\perp$ around the magnetic field.

In the relevant case of an ITER scale tokamak reactor with magnetic field strength of 13 Tesla, and a temperature of 10 keV, the ion and electron cyclotron frequencies and Larmor radii are given by

$$\rho_e \approx 7.7 * 10^{-4} mm$$

$$\rho_p \approx 3.5 * 10^{-2} mm.$$

In a tokamak reactor, the helical magnetic field lines are used to confine the plasma ions and electrons in the reactor are not 100% effective in containement. .

4.7 Particle Drifts & Accerations

The above magnetic field was assume to be uniform, and there were no electric fields considered. We thus need to consider particle motion in the following cases:

Gradients in the electric field perpendicular to the magnetic field.

Gradients in the magnetic field, parallel to the direction of the magnetic field.

An electric field perpendiculat to the magnetic field.

A gradient in the magnetic field perpendicular to the magnetic field.

Cu

4.8 Collisions

We now consider collisions between plasma particles. Collisions between charged plasma particles are responsible for diffusion and transport in the plasma, and fusion and α particle heating, in which the α particles impart the energy gained through fusion to the D-T fuel.

5 Magnetic Confinement

6 Traveling Waves & Fluctuations

The majority of plasma transport, including so-called anomolous transport, is a result of long range electric fields that govern the collective behavior of plasma particles grouped in a Debye sph. These low frequency fluctuations create long wavelength electric fields with wave vector k, such that $1 << \frac{\lambda_D}{k}$. These long wavelength modes determine the collective behavior of the collection of charged particles within a Debye sphere. [?] We are interested in studying and characterizing transport levels and scaling driven by ion temperature gradient (ITG) drift-wave fluctuations. The mentioned long wavelength eletric field modes in the plasma allow the free energy in the ion temperature gradient to be accessed, converting thermal energy into kinetic energy [?], resulting in a majority of observed anomolous transport.

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f}{\partial \mathbf{v}} = 0,$$

6.1 H-Mode Operation vs L-Mode Operation

In the 80s it was found that above certain power levels the plasma transitions from a low energy mode (L-Mode) to a high energy mode (H-Mode), and in this high energy mode the confinement time of the plasma is reached. However, instabilities in the plasma edge, or edge localized modes (ELM) can cause the the plasma to crash from the H-mode back down to the L-mode. These edge localized modes are caused by turbulence driven transport, induced by ion temperature gradients in the plasma. flow must fully characterize all of the transport processes driving heat loss. A full characterization of the heat transport processes is beyond the scope of this master's report, but each of the transport mechanisms will be summarized here.

7 Heat transport - Classical, Neoclassical, and Turbulent Transport.

7.1 Parameter Scaling Experiments to Predict Transport

Predicting plasma behavior in fusion reactors is made possible through the use of dimensionless groups of parameters

8 Plasma Models

Description of Models

8.1 Fluid Models

8.2 Kinetic Models - Vlasov-?

To accurately predict ion temperature gradient driven turbulent transport in a fusion interparticle collisions must be taken into consideration

9 Literature Review

$rho*$ parameter can be scaled to reflect larger plasma size than current reactors allow

Predicting the scaling of turbulence driven heat transport in a fusion plasma at future tokamak sized reactors is critical to predicting performance, and to ensure that reactors the nececessary confinement properties for sustained fusion. Thus, scaling studies are of critical importance in fusion reactor design studies. As mentioned previously, the net heat transport must be balanced by auxiliary heating power, so understanding heat transport scaling is vital from an economic perspective. Additionally, stability of the plasma and controlling turbulence is critical to consistent tokamak performance, and edge model simulations may be employed in the future when simulating tokamak performance to ensure the reactor is within the desired operating regimes.

Experimental transport research has focused on ion or electron transport scaling with the dimensionless parameter $\rho*$, as ions and electrons are the primary channels for heat transport out of the plasma. Experiments performed in current fusion reactors have indicated that in H-Mode operation that the scaling of ion driven transport is much larger than electron driven turbulent transport, making the ion channel the the primary channel studied in simulation studies. In H-Mode operation the ion transport scaled linearly (Bohm-like) with plasma size, while electron transport remained constant (gyroBohm-like) as plasma size was scaled. Thus, the focus of most simulation studies is on ion temperature gradient driven turbulent transport.

The scaling of heat transport derived in current experimental reactors cannot necessarily be extrapolated to future operational reactors, as, in current reactors, the dimensionless plasma radius $\rho*$ cannot be scaled to future operational values. However, with the development of parallel computing, modern gyrokinetic theory, and advanced numerical simulations and algorithms, it has become possible to accurately simulate turbulence driven heat transport from first physical principles.

Previous δ_f simulation studies involved fixing the ion temperature gradient, R/Lti and scaling the plasma size via the dimensionless parameter $\rho*^{-1} = a/\rho_{ti}$, where a is the radial length of the tokamak reactor, and ρ_{ti} is the thermal ion Larmor radius, R is major radius of the tokamak, and L_{ti} is the ion temperature length scale, and then observing the normalized ion heat flux, $q_i/(\chi_{GB} n_i T_i/a)$, at a particular radial distance, $r = 0.5a$ with q_i being the ion heat flux, χ_{GB} is the Bohm diffusion constant, $n_i = 10^{19} m^{-3}$, and T_i is the ion temperature.c

In a working fusion reactor, auxiliary heating methods will be used to balance the power lost from the plasma via heat transport. Depending on how transport scales with reactor size, different heating power scaling and strategies will need to be employed: in the case of Bohm like scaling of transport with reactor size, the auxiliary heating will need to increase linearly with reactor size, while gyro-Bohm scaling means that auxiliary be held constant. This difference is critical, as any additional input power affects the global confinement properties of the plasma, and increases heat transport. Additionally, the input power that will be generated by α particle heating in future tokamak reactors is significantly larger than current fusion experiments, and so predicting how α particle heating affects transport scaling is impossible with current experimental reactors. Thus, to accurately model heat transport, it is necessary to incorporate a heating model that captures the input power P_in from auxiliary heating and α particle heating.

In Nakata and Idomura it was shown that, due to ion avalanche events in the plasma, the net ion heat flux "increased rapidly above the nonlinear critical gradient," yielding reduced confinement properties as heating power is scaled. This increase in ion heat flux indicates that maintaining the necessary temperature gradient for H-Mode operation becomes more difficult as auxiliary power increases, which is problematic since that is the operating mode that future reactors will work in. or that the input power degrades the confinement property τ_i. In the same paper,the $\rho*$ scan demonstrated that the PDF for q_i with $\rho* = 600$, and without a heating pows work done in transport scaling research by work performed in [NAKATA, IDOMURA, and other heating power scaling experiments] by incorporating the uncertainty that exists in the heat deposition methods into the heating modeer model, matched the q_i at $(\rho* = 300, P_in = 8MW)$, which suggest that ion transport scaling in the simulated environment is predominantly driven by input power and not plasma size. These results demonstrate important of fully characterizing the role that input power, and input power scaling, play in ion driven heat transport and scaling in a fusion reactor. **Figures from Nakata Idomura 2014 here**

It is critical that the electron and ion density and temperature profiles that are used in the plasma simulation are accurate, as the gradients of these profiles drive the turbulent transport that we are studying. The simulation outputs are highly sensitive to these profiles, so accurate temperature and density profiles derived from fusion experiments are required for accurate predictions of heat transport from the plasma simulations. Any set of measurements and fitting yield profiles along with uncertainty profiles. Methods of measuring particle density and temperature include using Thompson scattering to take measurements of temperature and density at 33 radial positions in 16 ms time increments[8].The density and temperature profiles and profile uncertainties of an Alcator C-Mod L-Mode discharge have been extracted using STRAHL simulation code and Monte Carlo sampling [8]. Recently, [5] developed software that could fit the different profiles, along with their uncertainties in a statistically rigorous manner using Gaussian process regression. Although these profiles have been used for model validation, outside of [5], and [**?**], no significant work has been done on uncertainty propagation of the temperature and density profiles in a high fidelity plasma setting. Thus, we seek demonstrate the viability of such an experiment for uncertainty analysis reasons, and to determine the sensitivity in heat transport to the different profile parameters, such as shape and amplitude.

To address these issues, and build on previous work, we plan to construct a sparse grid interpolant that will approximate the response of gyrokinetic transport model to the heating model parameter space. If succesful this experiment will accomplish the following: it will allow us to to determine the sensitivity of heat transport to the parameters of the heating model, and determine an active set of parameters for future work; it will yield a 4-Dimensional response surface to be used for predictions and uncertainty propagation in a manner that is cheaper than direct numerical simulations; the response surface can be used to make predictions of heat transport scaling under a range of tokamak operating parameters.

10 Uncertainty Quantification

10.1 Introduction

10.2 Heating Sources

Per [Y. Idomura1, H. Urano2, N. Aiba2 and S. Tokuda2], "A source term requires and empirical modeling." This on-axis radial heating model with a fixed power input provides the energy for momentum and heat transport driven by ITG turbulence. [[Y. Idomura1, H. Urano2, N. Aiba2 and S. Tokuda2] The sink model fixes the temperature T_i and parallel velocity $U_{||}$ at the pedestal boundary to reflect that of H-Mode boundary conditions.

The equations of the model are given by

$$S_{src} = \nu_h A_{src}(r)(f_{m1} - f_{m2})$$

$$S_{snk} = \nu_s A_{snk}(r)(f - f_0)$$

where $A_{src}(r)$ is the source deposition profile of radial on-axis heating, and A_{snk} is the heat absorption deposition profile. The heating and cooling rates are given by ν_h and ν_s, f_{M1}, f_{M2} are two shifted Maxwellian distribution functions with different temperatures, and f is the ion and electron dsitribution, and f_0 is the initial distribution. profilef_{M1}, f_{M2}, are chosen to impose fixed power input P_in with no momentum or particle input. The sink equation maintains the initial plasma boundary velocity and temperature using a Krook operator. Possible equations enforcing this condition?

10.3 Defining the Quantity of Interest

We now consider forward propogation of the uncertainty associated with the physical parameters through the 5-D gyrokinetic ion model, and adiabatic electron model. As mentioned, this involves solving a system of partial differential equations that describe the ion and electron distribution functions in a 5-D phase space, which have conservation constrains enforced via various equations, and which are couple to Maxwell's systems of equations to propogate the charged particles through phase space. The solution of these system in the regime we are interested in, with ion and electron density on the order of $10^{19} m^{-3}$, takes NERSC parallel computers on the order of hours to days of computational time.

For the sake of clarity and compactness in writing, we simplify the full 5-D gyrokinetic simulation and heating model to be a black-box mapping, f, which maps the parameter space, $\mathbb{P}$, where $\mathbb{P} \subset \mathbb{R}^4$ are the inputs of the heating model, to a space containing our quantity of interest (QOI), $Q \subset \mathbb{R}$, or (I need to refine the domain and co-domain definition for the paper.)

$$f : \mathbb{P} \to Q.$$

We assume that our parameters are either normally distributed $\mathcal{N}(\mu, \sigma)$, or uniformly distributed $\mathcal{U}(0, 1)$. and we are interested in predicting the expected value of the normalized net heat flux, $\hat{q}_i = q_i/(\chi_{GB} n_i T_i/a)$, as we vary over the entire parameter space, $\mathbb{P}^4$. Thus, we define the normalized heat flux to be a function of the parameter space, or $\hat{q}_i = \hat{q}_i(\mathbf{p})$, where $\mathbf{p} \in \mathbb{P}^4$, as well as a normalized heat flux distribution function $f(\hat{q}_i(\mathbf{p}))$, defined over the entire parameter space. The expectation value is then

$$\mathbb{E}(\hat{q}_i(\mathbf{p})) = \int_{\mathbb{P}^4} \hat{q}_i(\mathbf{p}) f(\hat{q}_i(\mathbf{p})) d\mathbf{p}.$$

10.4 Methods for Determining $\mathbb{E}(\hat{q}_i(\mathbf{p}))$

We consider two methods for determining the normalized heat flux expectation value. We can choose the operating power regime to work in, and use Monte Carlo integration to evaluate $\mathbb{E}(\hat{q}_i(\mathbf{p}))$ directly. The Law of Large Numbers dictates that Monte Carlo integration error decreases at a rate of $\mathcal{O}(\frac{1}{\sqrt{N}})$. Monte Carlo integration is an attractive method in a high degree parameter space, as the convergence rate is independent of the dimension of the parameter space, and is a reliable workhorse for evaluating analytically intractable integrals. However in a parameter space of moderate dimensions, there are other methods that converge at a faster rate, or the error decreases at a rate $\mathcal{O}(N^{-\beta})$, where $\frac{1}{2} < \beta$.

Considering this, we choose to instead approximate the PDF of the heat flux in response to the variation parameter space of the heating model using a polynomial interpolant surrogate model constructed on a sparse grid with Clenshaw-Curtis points.

10.4.1 Interpolation Basics

Polynomial interpolation employs various sets of polynomial basis functions, Ψ_k, like Lagrange, Hermite, or Legendre polynomials, to constructs a model $\hat{f}$ that approximates the full mapping f according to the interpolation error, and is much cheaper to evaluate than f, [?] so

$$\hat{f} = f + \epsilon$$

The basic idea of interpolation is to use a basis of polynomials from a space $\mathbb{P}_k$, of degree $\leq k$ polynomials, and construct a polynomial that matches (interpolates) the map being approximated at select points on the unit domain. It is useful to mention that the theorem of interpolation states eactly 1 polynomial of degree $n-1$ interpolates n points, thus the minimum degree of the polynomial necessary to interpolate the map informs the number of necessary interpolation points. Thus, if the function being interpolated is a polynomial of degree (n-1), then, with n-points, the interpolating polynomial exactly fits the original polynomial.

To construct a polynomial interpolant in 1-D, we first convert the interval, $[a_k, b_k]$, that parameter, x_k, is defined on to the unit interval $[0, 1]$. We then choose points on the interval to interpolate the basis through. The selection of interpolation points can greatly effect the goodness of fit under certain situation. Consider the function $\frac{1}{(1+12x^2)}$ on evenly space points on $[-1, 1]$. This function is 1 at the origin and smoothly converges to $\frac{1}{13}$ at the interval endpoints. However, the slope of the function around he origin causes the interpolating to overestimate the behavior at the endpoints of the interval, causing the interpolant to oscillate in an artificial manner. This problem is fixed by placing more collocation points towards the end of the interval, as this better informs the interpolant about the functions end behavior.

Another factor consider is nesting of interpolation points; consider a scenario in which we have certain set of data: $\{(x_{k,1}, f(x_{k,1}), ..., (x_{k,n}, f(x_{k,n})\}$, and we construct an interpolating polynomial with this data. Then, consider that computational resources become available, so that we can evaluate our expensive model at another point $(x_{k,n+1}), f(x_{k,n+1})$. We want incorporate this new data point into the existing interpolant without changing any existing interpolating points, as this would exceed out computational budget. Thus, we want to use interpolating rules that allow us to add new points to an existing interepolant without reevaluating the existing interpolant, or create a nesting scheme or interpolant points.

The points that are both nested, and smooth out Runge behavior [RUNGE PHENOMENON PICTURE HERE] at the endpoints of the interpolant are the extrema of the Chebyshev polynomials, also known as

Clenshaw-Curtis points:

$$x_{k_j} = \frac{1}{2}[1 - \frac{-cos((j-1)\pi)}{M_l - 1}], \quad j = 1, ..., M_l$$

where M_l is the number of interpolating points at the lth nested level. Thus, the 1-D interpolation rule with the set of interpolating points $x\{(x_{k,1}, f(x_{k,1}), ..., (x_{k,n}, f(x_{k,M_l})\}$ is given by

$$\hat{f}^{M_l}(x) = \sum_{i=1}^{M_l} f(x_{k_i})\phi_{M_l,i}(x).$$

where $\phi_{M_l,i}, i = 1, ..., M_l$ is the set of basis functions being used to interpolate the polynomial.
Now, when considering interpolation for multidimensional parameter space, with parameters $x = \{x_1, x_2, ..., x_n\}$ we can construct an interpolation rule by first constructing a 1-D interpolation formulas,$\mathcal{I}_k^1$ for each parameter. The intervals are converted to the unit interval, so we are interpolating on the unit hypercube, $[0,1]^d$.The interpolant for each parameter may use a different nested level of points thus, we index the Clenshaw-Curtis interpolation points for each nested level:

$$x_l^m = \frac{1}{2}[1 - cos\frac{\pi(m-1)}{M_l - 1}], \quad m = 1, ..., M_l$$

with $M_1 = 1$ and $M_l = 2^{l-1} + 1$, $l > 1$. [?] For each interpolation rule, let $\Theta_l^1 = \{x_l^1, ..., x_l^{M_l}\}$. The new interpolation grid in the n-dimensional parameter space is determined by taking the tensor product of the 1-D rules, which yields the following set of n-tuple interpolation points:

$$\Theta_l^{(p)} = \cup_{\max l' \leq l} \Theta_{l_1}^{(1)} \times \cdots \times \Theta_{l_p}^{(1)},$$

where $l' = (l_1, ..., l_n) \in \mathbb{N}^n$ is a multi-index with $|l'| = \sum_{i=1}^n l_i$, and max $l' \equiv$ max $\{l_1, ..., l_p\}$. This results in the number of interpolation points scaling exponentially with the dimension of parameter space, or

$$(\min M_l)^n \leq M \leq (maxM_l)^n.$$

Additionally, the interpolant

$$\mathcal{I}^{(p)}u(q_1, ..., q_p) = \sum_{i_1=1}^{M_{l_1}} \cdots \sum_{i_n=1}^{M_{l_n}} u(x_1^{i_1}, ..., x_n^{i_n}) L_{i_1}(x_1) \cdots L_{i_n}(x_n),$$

must be evaluated at the $M_1 * M_2 * \cdots M_n$ grid points, which is computationally infeasible. This exponential growth of sampling points with respect to parameter space is referred to as the curse of dimensionality, and developing methods that retain fidelity of approximations to the full model, with fewer computations, is a current field of research in applied mathematics and physics. In our case we are dealing with a 4-D parameter space, and if we choose a level 3 rule with 5 grid points for each 1-D interpolant, then the 4-D grid has a total of $5^4 = 625$ 4-tuple grid points to evaluate, which corresponds to over a year of computation time to develop a model surrogate.

As a basic example of the full tensor product sparse grid, consider a 2 dimensional parameter space, with the first dimension having a level 2 interpolating formula, and the second dimension having a level 3 interpolating formula. This yields the following sets of 1-D nodal points

$$\Theta_{l=2} = \{0, \frac{1}{2}, 1\}$$
$$\Theta_{l=3} = \{0, \frac{1}{2}[1 - \frac{\sqrt{2}}{2}], \frac{1}{2}, \frac{1}{2}[1 + \frac{\sqrt{2}}{2}], 1\},$$

which demonstrates the nested property of the Clenshaw-Curtis points. Taking the tensor product of these points yields 15 points in the 2-D parameter space.

If we use the Lagrange polynomials as the interpolating basis we have the following interpolating polynomials:

$$L_{l=2} = 2(x - \frac{1}{2})(x - 1)f(0) - 4x(x - 1)f(\frac{1}{2}) + 2x(x - \frac{1}{2})f(1)$$
$$L_{l=3} = 16(x - \frac{1}{2}[1 - \frac{\sqrt{2}}{2}])(x - \frac{1}{2})(x - \frac{1}{2}[1 + \frac{\sqrt{2}}{2}])(x - 1) + \cdots + 16x(x - \frac{1}{2}[1 - \frac{\sqrt{2}}{2}])(x - \frac{1}{2})(x - \frac{1}{2}[1 + \frac{\sqrt{2}}{2}]).$$

Without demonstration, we can see from the definition of the tensor product interpolating formula that the tensor product of these 1-D formulas yields a 6th degree polynomial.

This begs the question, why have we gone down the polynomial interpolation if it is computationally infeasible? The answer is routed in the idea that, by taking the tensor product of the interpolant rules, polynomials of degree $M_1 * M_2 * \cdots * M_n$ have been generated, which may be an unnecessarily high degree for accurately interoplating the "true" map. For instance, suppose the "true" map f that we are trying to approximate is well approximated by a third degree polynomial in every direction in our parameter space $\mathbb{P}^4$. Constructing 1-D interpolation rules with 3rd degree polynomials, and taking the tensor products yields a highest order monomial term $(y_1^3 y_2^3 y_3^3 y_4^3)$ of total degree 12. Since this term is unnecessary for accurate interpolation we can discard it, and the associated interpolation point.

In general if we can determine the interpolation level necessary for accurate model prediction we can set an upper bound on the monomial degree necessary for accurate interpolation, and discard unnecessary interpolation points from the full tensor product grid, producing a sparse grid. Thus, we seek a rigorous method that will allow us to construct a a set of weights and grid points, with significantly less grid points than a full tensor product grid construction, without sacrificing accuracy of the interpolant. Although other methods have been developed, Smolyak was the first to describe this method in [?], and is the method we use in this paper. For brevity, we will present the general sparse grid nodal points and interpolation rules, with associated definitions, as well as the nodal point set and interpolation rule for our 4-D space.

[INSERT IMAGES OF GRID COMPARISONS] Let n be the dimension of the parameter space $\mathbb{P}^4$, and let $x_k \in \mathbb{P}_k \subset \mathbb{P}^4$. Let $\{\phi_1, \cdots, \phi_m\}$ denote the set of polynomial basis functions being used to interpolate the function.We define the 1-D interpolant of the map f as follows:

$$\mathcal{I}_l^{(1)} f = \sum_{i=1}^{M_n} f^i \phi_i(x_k)$$

where the operator interpolates the M_n points $\Theta_l^{(1)} = \{x_l^1, ..., x_l^{M_n}\}$ in the l^{th} nested level, and the points x_l^i are the Clenshaw-Curtis points [?]. Per Smith, [?] we use the following difference relations

$$\Delta_l^{(1)} u = (\mathcal{I}_l^{(1)} - \mathcal{I}_{l-1}^{(1)})u \quad , \mathcal{I}_0^{(1)} u = 0.$$

which produces an interpolating formula with shifted weights. Using the difference formulas for each 1-D rule, we get the following sparse grid interpolation formula

$$\mathcal{I}_l^{(n)} u = \sum_{|l'| \leq l+n-1} (\Delta_{l_1}^{(1)} \otimes \Delta_{l_2}^{(1)} \otimes \cdots \otimes \Delta_{l_n}^{(1)}) u.$$

The sparse grid nodal points are given by

$$\Theta_l^{(n)} = \cup_{|l'| \leq l+n-1} \Theta_{l_1}^{(1)} \otimes \cdots \otimes \Delta_{l_n}^{(1)},$$

where the $|l'| \leq l+n-1$ means $\sum_{i=1}^{n} l_i \leq l+n-1$.

In our scenario, with a 4th degree parameter space, and letting the level $l = 3$,

$$l_1 + l_2 + l_3 + l_4 \leq 6,$$

the sparse grid interpolation formula and nodal set becomes

$$\mathcal{I}_l^{(4)} u = \sum_{l_1+l_2+l_3+l_4 \leq 6} (\Delta_{l_1}^{(1)} \otimes \Delta_{l_2}^{(1)} \otimes \Delta_{l_3}^{(1)} \otimes \Delta_{l_4}^{(1)}) u.$$

$$\Theta_l^{(4)} = \cup_{l_1+l_2+l_3+l_4 \leq 6} \Theta_{l_1}^{(1)} \otimes \Theta_{l_2}^{(1)} \otimes \Theta_{l_3}^{(1)} \otimes \Delta_{l_4}^{(1)},$$

which indeed constrains the degree k ofthe monomial terms of the interpolating polynomial to be less than 6.

10.4.2 Sparse Grid Construction

The approximate map is constructed by interpolating $\hat{f}$ through points produced by $f : \mathbb{P}^4 \to \mathbb{R}$, or for select points $\mathbf{x} \in \mathbb{P}^4$:

$$\hat{f}(\mathbf{x}) = f(\mathbf{x}).$$

So, we would like to use the least amount of interpolation points possible that still allows us to capture the interesting behavior in the full model. To do this we need to construct a 4 dimensional grid of points, $\Omega \subset \mathbb{P}^4$, to interpolate

11 UQ Model Problem - Landau Damping

11.1 Physical Background - Landau Damping

As a surrogate to the uncertainty quantification problem we are solving in a high fidelity setting, and to demonstrate the methods we intend to use to create sparse grid interpolants, we work with nonlinear Landau damping. Lev Landau is responsible for properly deriving the dispersion relation for electron plasma oscillations by using contour integration to solve the Vlasov equation in the complex field [**?**]. What was discovered is that, given a traveling electric field fluctuation in a plasma, electrons traveling with a velocity $\mathbf{v}$ close to the phase velocity, $\mathbf{v}_\phi$, of the electric field can exchange energy with the field. An electron with $\mathbf{v} \approx \ \mathbf{v}_\phi$, moves opposite the field, in the field reference frame, and can interact with the field. An electron moving slower than the field will be accelerated when it reaches the peak of the wave in the direction that the wave is traveling, increasing kinetic energy of the electron and reducing the energy of the field. Electrons traveling faster than the phase velocity impart energy to the wave, decreasing their kinetic energy. Assuming the electron velocities have a Maxwell distribution, there are more slow electrons than there are fast electrons, thus there is an overall damping of the field. Interestingly, Landau damping is an example of collective behavior (fields) affecting individual behavior (electrons) in a plasma. Landau damping must be incorporated into plasma models to accurately predict kinetic effects in the plasma. [**?**]

There are two types of Landau damping to consider: linear and nonlinear Landau damping. If the field potential, ϕ, is small then the electric field amplitude does not effect the kinetic energy of the electron, and the field and velocity equations can be linearized. If $\phi$is larger than the kinetic energy of the electron, then the electrons can become trapped in the potential of the electric field. When trapped, electrons completely change directions relative to the wave when reflected off of the potential, thus the velocities of the electron are greatly affected by the oscillating field, and the distribution function $f(v)$ is greatly disturbed, no longer allowing for linearization of our system of equations.

While, Landua damping was discovered mathematically in the 40s, proofs of convergence of the solutions linear and nonlinear Vlasov-Poisson equation in as $t \to \infty$ are drecent developments. Solutions to the linearized Vlasov-Boltzmann equation have been derived, and it was shown in [?] that the solution to the nonlinear Vlasov-Poisson equation does converge under certain stability conditions and initial conditions. In the same paper it was also shown that the expectation values of the distribution function, and the particles density, both converge to a solution as $t \to \infty$. In [?] it was proven analytically that uncertainty in the inputs from the initial data or equillibrium data does not affect the convergence of the various solutions, and the solutions vary smoothly with the input space if the distribution function varies smoothly as time goes to infinity.

As Landau damping occurs in fusion reactors it is not only interesting, but also important to consider how Landau damping affects plasma transport, and overall plasma behavior. Additionally, since Landau damping is incorporated into modern simulations, and the inputs are experimentally derived, there is uncertainty asoocciated with model inputs. To investigate the long term convergence of the distribution function and its moments, we seek to propogate uncertainty in a 2-D input parameter space through a symplectic Runge-Kutta solver for Vlasov-Poisson equation using Monte Carlo sampling. With the results we wish to calculate the expectation value of the kinetic energy of the particles as time approaches ∞, and determine whether convergence occurs. We then wish to construct a sparse grid surrogate to the distribution function in our 2-D parameter space, and use that to calculate the average kinetic energy of the particles to compare the direct solution to the approximate solution.

In our simulation and UQ example we consider both the linear and nonlinear Landau damping examples

11.2 Vlasov-Boltzmann Description

To setup the problem, we consider the most general domain of a 6-D phase space $\Omega = \Omega_x \times \Omega_v = [0, L] \times [0, L] \times [0, L] \times \mathbb{R}^3$. To properly treat Landau damping, kinetic theory is employed to model how the electrons in the distribution close to the electric field phase velocity interact with electric field, as the Maxwellian distribution of particles is greatly disturbed in this region of velocity space. The Vlasov-Boltzmann equation is the standard equation used to model the electric field fluctuations:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{q}{m}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f}{\partial \mathbf{v}} = 0,$$

along with divergence free magnetic field $\nabla\cdot\mathbf{B} = 0$.

Since the ions are massive in size relative to the electrons it is assumed that they do not react to fluctuations in the electric field, so we treat the ion density and related quantities as fixed, which yields $\rho_{ion} = 1$.

Coupling the Vlasov-Boltzmann system to Poisson's equation allows us to solve for quantities of interest such as the potential, charge density, current, and electric field by evaluating the first moment of the electron distribution function:

$$\rho_e = -\int f d^3 v$$

$$-\nabla^2 \Phi = \rho_{ion} + \rho_e$$

$$E := -\nabla \Phi.$$

The kinetic energy of the electrons is calculated by taking the second moment of distribution function, and the electric field energy is calculated with the classical energy equation $\mathcal{H}_E(t) = \frac{1}{2}\int_{\Omega_x} |E|^2 d^3x$, yielding a total time dependent Hamiltonian

$$\mathcal{H}(t) = \frac{1}{2}\int_{\Omega} f v^2 dx^3 dv^3 + \frac{1}{2}\int_{\Omega_x} |E|^2 d^3 x.$$

The Vlasoz-Boltzmann equation is a homogeneous Cauchy equation, and therefore a conservative system, derived by taking the total time derivative of the distribution function $f(\mathbf{X}(t), \mathbf{V}(t), t)$

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \frac{d\mathbf{X}}{dt}\frac{\partial f}{\partial \mathbf{X}} + \frac{d\mathbf{V}}{dt}\frac{\partial f}{\partial \mathbf{V}} = 0,$$

which means that volumetric elements of the distribution stay contant along trajectories in phase space, or

$$f(X(t), V(t), t) = f(X(0), V(0), 0) \qquad \forall t \geq 0.$$

The Poisson equation along with the method of characteristics is used to solve this system, with the characteristics evolving according to the following equations of motion [**?**]

$$\frac{dX(t)}{dt} = V(t)$$
$$\frac{dV(t)}{dt} = -(E(t, X(t)) + V(t) \times B(X(t), t)).$$

We now consider a 1-D electric field oscillation example that demonstrates Landau damping, with an additional small constant background electric field. We consider a uniform plasma with background Maxwellian distribution $f_0(\mathbf{v})$ assumed, and $\mathbf{B}_0 = B_0(\mathbf{r})$ $\mathbf{E}_0 = E_0(\mathbf{r}$,t), with $\mathbf{B}_0$ and $\mathbf{E}_0$ being the background fields. is introduced in the plasma, causing electron oscillations. This yields the perturbed distribution function $f(\mathbf{x{,}v}$,t), with a distribution perturbation $f_1(\mathbf{x{,}v}$,t),

$$f(\mathbf{r{,}v}, t) = f_0(\mathbf{v}) + f_1(\mathbf{r{,}v}, t).$$

If we assume a traveling electric field perturbation, $\mathbf{E}_1$, then the total electric field is given by

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{E}_0(\mathbf{r}, t) + \mathbf{E}_1(\mathbf{r}, t).$$

Now we reduce this system of equations by assuming a 1-D perturbation in the x-coordinate. This simplification still allows us to capture the essential physics under consideration.

The distribution function reduces to

$$f(x, v_x, t) = f_0(x, v_x, t) + f_1(x, v_x, t)$$

s Assume an electric field only in the x-direction, so $\mathbf{E}_0 = E_{0,x}(x, t)$, and assume the following form of the perturbed electric field

$$\mathbf{E}_1 = E_1 e^{i(kx - \omega t)},$$

then total electric field is given by

$$E_x(t) = E_{0,x}(x, t) + E_1 e^{i(kx - \omega t)}.$$

Since the electric field is 1-D,

$$f_1 \propto e^{(i(kx-\omega t)}.$$

We also assume that, since the ions are massive, and essentially not moving on the timescales of interest, that we can ignore their behavior. This means we can solve a single Vlasov-Boltzmann equation for the plasma electrons. This model can be reduced even further by removing the background electric field, and will still retain the Landau damping behavior.

Substituting $E_x(x,t)$, , and $f(x, v_x, t)$ into the Vlasov-Boltzmann equation we arrive at the:

$$\frac{\partial(f_0(\mathbf{v}) + f_1(x,,v_x,t))}{\partial t} + v_x \frac{\partial(f_0(\mathbf{v}) + f_(x,v_x,t))}{\partial x} + (E_{0,x} + E_1 e^{(i(k_x x-\omega t)}) \frac{\partial(f_0(\mathbf{v}) + f_1(x,v_x,t))}{\partial v_x} = 0,$$

which reduces to

$$-i\omega f_1 + v_x i k_x f_1 + (E_{0,x} + E_{1,x} e^{(i(k_x x-\omega t)}) \frac{\partial(f_0 + f_1)}{\partial v_x} = 0,$$

when considering the time and spatial independence of f_0.

In our model we consider $E_{0,x}$ to be a constant background electric field that, over time, alters the distribution function $f(v)$.

Finally, Poisson's equation yields

$$\begin{aligned} \epsilon_0 \nabla \cdot \mathbf{E}_1 &= ik\epsilon_0 E_x \\ &= -en_1 \\ &= -e \int\int\int f_1 d^3v, \end{aligned}$$

which closes our system of equations. In the linear regime of Landau damping the term $E_{1,x} e^{(i(k_x x-\omega t)} \frac{\partial(f_1)}{\partial v_x} = 0$, as this is a second order perturbation. However, when considering electron trapping in the electric field, the distribution $f(v)$ is greatly disturbed around the velocity $v = \omega/k_x$, so the second order perturbation quantity cannot be ignored [9].

11.3 Simulation Methodology and Outline

For work on his PhD [10] developed a spectral 2-D Vlasov-Poisson solver that simulates a symplectic Runge-Kutta method was used to solve the above system in Matlab. Symplectic methods are used as they are a conservative numerical integration method, which is necessary since we are dealing with a conservative Hamiltonian system. The simulation offers the option to turn on a background electric field as a function of space and time, and to vary the perturbation level of the electron distribution functions. **Methods - Uncertainty Quantification Through PDF Construction**

The goal of this project is to make a comparison between a Monte Carlo generated probability density function and a probability the probability density function of the kinetic energy at a particular time point in a 1-D Landau damping simulation generated using two methods. The first method is the Monte Carlo sampling method. The Landau damping model has a few parameters that can be formulated as random variables, for this "experiment" the amplitude of the pertubing wave , as well as the mean kinetic energy, were established as random variables, and a Monte Carlo simulation was performed on these random variables. Uniform $[0, 1]$ distributions were generated in Matlab for both of the model parameters, and 10000 pairs of samples were generated from these distributions, and a Landau damping simulation was run with each pair of parameters. The simulation is run for 500 time steps, to allow for the long term behavior of the linear and nonlinear Landau damping to develop (figure). The quantity of interest chosen to perform uncertainty quantification on was the time centered average of the kinetic energy of the entire simulated system around the 400th time point (KE_{400}). The time centered average was chosen to smooth out fast varying data, and not overfit to spurious data points. Once all of the sample parameter values were run through the simulation, the time centered averages we collected, and the kernel density estimation function from the probability and statistics toolbox in Matlab was used used to estimate the time-centered average kinetic energy pdf. Kernel density estimation is a parameter free probability density estimation method that estimates the probability densit function of a continuous random variables by constructing centering kernel functions, $K(x_i)$ at each of the data points, x_i, and taking the average of all kernels:

$$f(x|\mathbf{x}) \approx \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x_i - x}{h}),$$

where $\mathbf{x}$ is the data vector, K is the kernel function and h is the bandwidth of the kernel, which works as a kernel smoothing parameter. To construct the kinetic energy pdf a normal kernel density estimator function, with a bandwidth of was used.

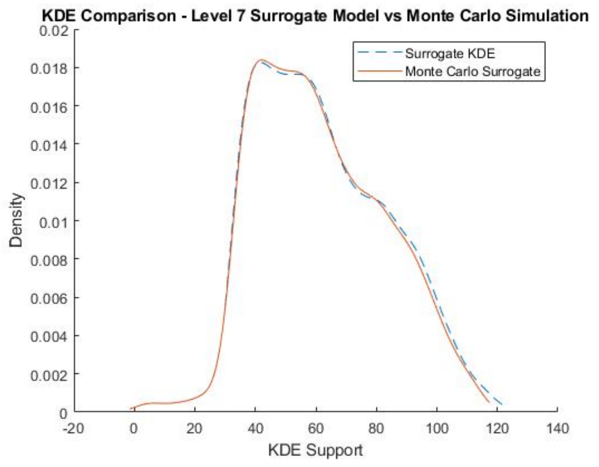
The second method used for constructing the kinetic energy pdf was via sparse grid interpolation of the data, as described in the previous section, using the software package Sparse Grids ++ [11]. A set of 5th degree polynomial Clenshaw Curtis sparse grids of levels 1-7, for a two dimensional parameter space, were generated. Then the sparse grid points were propogated throught the 1-D Landau damping simulation with the resulting KE_{400} values used as the data points for constructing the Clenshaw-Curtis polynomial interpolant. The sparse grids levels were varied to test the limits of the number of grid points (basis functions) necessary to produce an accurate surrogate model, and to compare the theoretical interpolation error to the actual interpolation error.

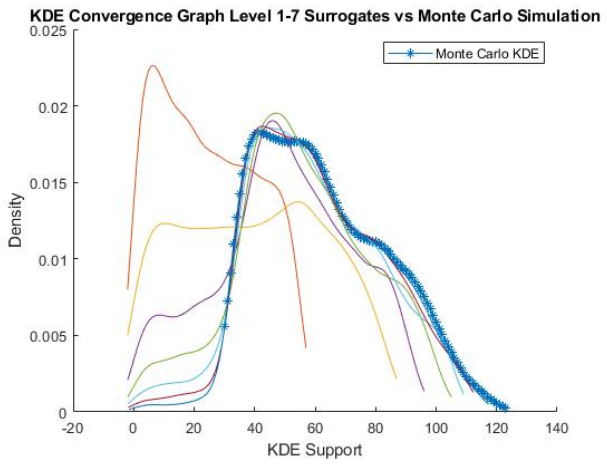
Unit-Test

To ensure that the interpolant was constructed properly (unit-test), the sparse grid points were evaluated as inputs to the interpolant, and their outputs were compared to the exact values obtained from the sparse grid simulation. If these values do not match exactly, the interpolant is rejected.

11.4 Analysis

Two methods of analysis were used to evaluate the quality of the surrogate model. We consider the pointwise, time-centered average kinetic energy results from the direct numerical simulation to be the "true" values of the experiment. We then compare these results to the results from evaluating the surrogate at the same 10000 randomized points that were propogated through the direct numerical simulation. To be clear, these are not the same points used to setup the surrogate model, but the 10000 2-D points from the randomized parameter space used in the DNS. The comparison is done by calculating the normalized, mean-squared sum of differences between the two sets of results. This comparison was done for all 7 levels of the of sparse grid





(a) Level-7 Surrogate Model Generated PDF Vs. Monte Carlo Generated PDF .

(b) Level 1-7 Surrogate Model Generated PDF Vs. Monte Carlo Generated PDF.

Figure 1: Comparison between the pdf generated via kernel density estimation with surrogate model results and the pdf generated with the direct numerical integration results.

surrogate, and plotted, as seen below. Then, the approximate normalized theoretical L^2 interpolation error

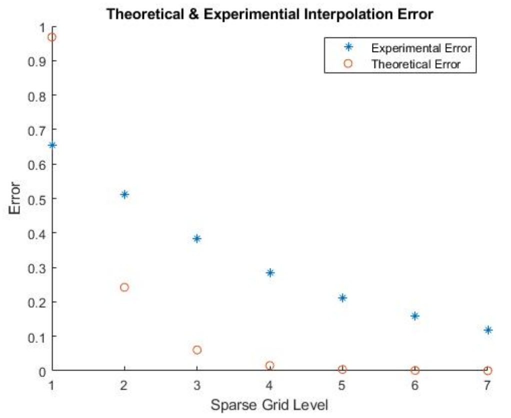
$$\mathcal{O}(2^{-2n}n^{d-1}),$$

where n is the sparse grid level, and d is the dimension of the parameter space, was calculated for sparse grid levels 1-7, and then plotted for comparison. The log of both errors were evaluated and plotted as well, and a linear fit was applied to empiricallog-error set to determine whether the empirical log-error relation to sparse grid level is linear. The linear fit returns a slope value of $m = -0.28782$, with corresponding p-value of $2.681e - 09$, and an intercept value of $b = -0.16111$so it is highly unlikely that we have chosen the incorrect model.

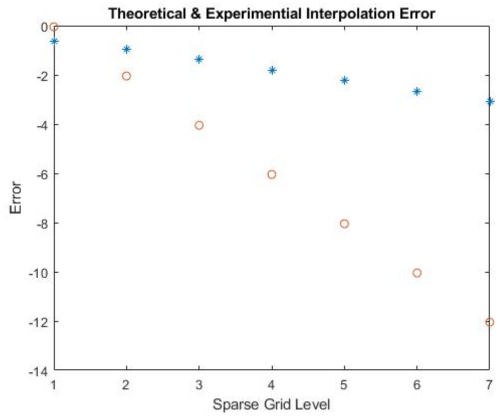
To measure the information lost by using the pdf constructed via KDE on the results from the surrogate, the Kullback-Liebler divergence between the surrogate model pdf and the DNS constructed pdf was calculated. For a continuous probability distribution the Kulbach-Liebler divergence is given by

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x)log(p(x)/q(x))dx$$

where P, Q are both probability density functions, and the KL divergence measure how much P divergence from Q or how much information is lost using P versus Q.



(a) Comparison of Mean Squared Difference Between DNS and Surrogate Model and Theoretical Error



(b) Comparison of Log-Mean Squared Difference Between DNS and Surrogate Model and Log-Theoretical Error

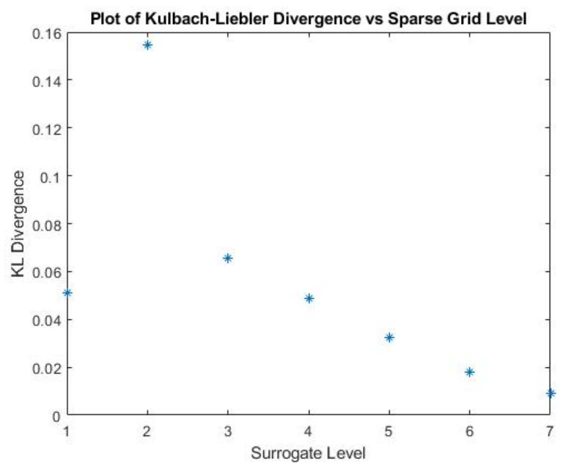
Figure 2: Error Comparison and Log-Error Comparison

To approximate this integral, the support region $[0, 125]$ was divided into increments of length $\Delta x = 0.001$, and trapezoidal quadrature was used, yielding the summation:

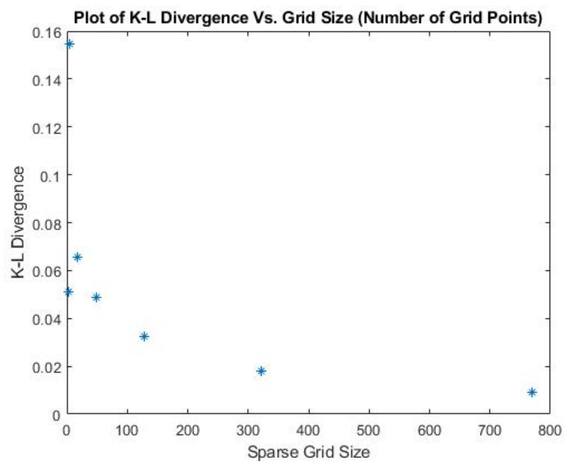
$$\int_{-\infty}^{\infty} p(x)log(p(x)/q(x))dx \approx \sum_{i=1}^{n} \frac{(p(x_{i+1})log(p(x_{i+1})/q(x_{i+1})) + p(x_i)log(p(x_i)/q(x_i)))}{2}\Delta x.$$

In this case we define P_i to be the pdf of the time-centered average kinetic energy generated from the ith level surrogate model, and define Q to be the reference time-centered average kinetic energy pdf generated using the results from the Monte Carlo DNS, so the K-L divergence provides a measure of how much information is lost when using a pdf constructed from the surrogate model, with lower K-L divergence values indicating lower information loss.

The K-L divergence was calculated for all surrogate levels, to compare information loss/gain at each level. It should be noted that the difference in information loss between the 6th and 7th level is 0.0089, which is still a 98 percent difference, but the level 7 sparse grid requires 769 simulation runs, while the level 6 sparse grid requires 321 simulation runs. Based on available computational resources and necessary information loss/gain this problem could be posed as a convex optimization problem and solved numerically or analytically.



(a) Kulblack-Liebler Divergence Vs Sparse Grid Level



(b) Kulblach-Liebler Divernge Vs Number of Grid Points

Figure 3: Kulbach-Liebler Divergence

12 Conclusions and Next Steps

We were interested in studying whether the uncertainty in a quantity of interest from the output of a high fidelity simulation can be characterized just as well using a polynomial interpolant constructed on a sparse grid, which was referred to as the model surrogate. If it can, then the surrogate model can be used to construct the probability density function of the QOI for various operating scenarios, and a computationally expensive portion of a simulation can be replaced with a cheap surrogate. The construction of the model surrogate still requires running the high fidelity simulation a number of times depending on the level of the sparse grid, which determines number of grid points evaluated in the high fiedlity simulation. Thus, it is also necessary to understand the level of information gain/loss from each level of the sparse grid in order to balance the amount of information gained versus the number of simulation runs. In order to familiarize myself with the workflow of surrogate model construction and uncertainty quantification, a a proxy for this scenario was chosen, a 1-D Landau damping simulation. Two parameters in thhe simulation were reformulated as random variables, and one of the simulation outputs, the time-centered average kinetic energy at the 400th time point, was chosen as the QOI. For the Monte Carlo direct numerical simulation, 10000 pairs from the 2-D parameter space were propogated through the model, and the QOI was saved for each paramater pair. Sparse grid surrogate models that predict this QOI based on a 2-D parameter space were then constructed using fixed grid points. Then, the 10000 parameter pairs were evaluated on the surrogate model, and outputs between the high fidelity simulation and surrogate model were compared. The experimental mean square error does obey a linear relationship, although the slope does not match the slope of the theoretical error. Graphically, and by calculating the K-L divergence, it was shown that as sparse grid level is increased the kernel density estimation from a random set of points evaluated on the sparse grid converges to the KDE constructed using the Monte Carlo simulation results. This means that once a sparse grid surrogate is constructed it can be used for future kernel density estimation and replace the high fidelity simulation, thus this research has been a success so far.

It would be useful to determine whether the KDE convergence can be improved by varying the type of basis functions used to construct the sparse grid interpolant, the degree of said basis functions, or by using different kernel density estimator functions. Varying these individually, and then calculating the K-L divergence try to improve the KDE convergence rate, sparse grids with higher accuracy at lower level, and smaller grid size can be constructed. Since each grid point corresponds to a single day of computation time of evluating the high fidelity simulation, reducing necessary grid level means a potential decrease on the orders of hundreds of days of computation time.

Of course, I intend to try to apply the methods demonstrated in this project to a real world high fidelity plasma simulation, and further explore whether surrogate models can used to suplant high fidelity models to predict plasma size power scaling and turbulent transport under uncertainty. My advisor has been generating simulated data on the NERSC computing system with the transport simulation software, and the intention is to try to construct a sparse grid surrogate using the data generated.

Additionally, I would like to turn the scripts that I developed into an easy to use function or sotware package that can be used for sparse grid and KDE construction, and evaluation for use in future experiments.

13 Matlab Code

Landau Damping Monte Carlo Simulation Code

```
%Random variable creation for brute force evaluation of solutions to
%Vlasov-Poisson - Nonlinear Landau Damping
samples1 = 10000;
lower1=0; upper1 = 1;
dist = 'Uniform';
eps1 = makedist(dist,0,1);
%t = truncate(pd, lower, upper);
eps1d = random(eps1, samples1, 1);
%%
samples2 =10000;
dist2 = 'Uniform';
lower2 = 0.0; upper2 = 1;
eps2 = makedist(dist2, 0.0, 1);
%eps2 = truncate(eps2, lower2, upper2);
eps2d = random(eps2, samples2, 1);
%%
epsm = [eps1d,eps2d];
%%
samples3 =10;
```

```
dist3 = 'Normal'
lower3=0; upper3 = 1;
eps3 = makedist(dist,'mu', 0, 'sigma', 0.1);
eps3 = truncate(eps3, lower3, upper3);
%t = truncate(pd, lower, upper);
eps3d = random(eps3, samples3, 1);


%%
%Modified script to incorporate uncertainty in background field and
%instability
%eps=0.5; % Amplitude of perturbation, 0.05 for linear, 0.5 for nonlinear
kx=0.5;     % Wave vector
L=2*pi/kx; % length of domain
qm=-1;     % negative charge to mass = electrons, $\frac{q}{m}=-1$

% Initial condition
f0 =@(x,v,eps1d, eps2d)    (1+eps1d*cos((kx)*x))./sqrt(2*pi).*exp(-0.5.*(v+eps2d).^2);

% Background to be subtracted from the phase space plot
background=@(x,v, mu) 1./sqrt(2*pi).*exp(-0.5.*(v-mu).^2);
% External electric field
E_ext=@(x,t) 0.*x+0.*t;

dt=0.1; % Time step
tmax=50;
rungekutta_order=3; %Order of the runge Kutta time integrator
```

```
Nx=32; % Number of cells in spatial direction

Nv=32; % Number of cells in velocity direciton

vmax=4.5;

vmin=-4.5;

%%

v=linspace(vmin,vmax,Nv+1).';v=v(1:end-1);

x=linspace(0,L,Nx+1).';x=x(1:end-1);


[XX,VV]=ndgrid(x,v);

%%

switch rungekutta_order

    case 1

        rksd=[1, 0]; %Symplectic Euler

        rksc=[0, 1];

    case 2

        rksd=[0.5, 0.5 ];

        rksc=[0, 1];

    case 3

        rksd=[2/3, -2/3, 1 ];

        rksc=[ 7/24, 3/4, -1/24];

    case 4

        rk4sx=real((2^(1/3) +2^(-1/3)-1)/6);

        rksd=[ 2*rk4sx +1 , -4*rk4sx-1, 2*rk4sx+1, 0];

        rksc=[ rk4sx + 0.5 , -rk4sx, -rk4sx, rk4sx +0.5];

end
```

```
%Build Matrix for spatial advection and all stages
XSHIFT=zeros(Nx,Nv,length(rksd));
m=fftshift(((1:Nx)-Nx/2-1));
for jdx=1:length(rksd)
    XSHIFT(:,:,jdx)=exp((-1j*dt*rksd(jdx)*2*pi/L)*(v*m).');
end
%Preallocate matrix for velocity advection
VSHIFT=zeros(Nx,Nv);


%diagnostic
numt=ceil(tmax/dt);
time = linspace(0,tmax,numt);
fieldenergy=zeros(numt,1);
kineticenergyr=zeros(numt,samples1);
meankineticenergy=zeros(samples1);
sh_entropy=zeros(numt,1); %Shannon entropy
kl_entropy=zeros(numt,1); %Kullback-Leibler entropy

EE=zeros(Nx,numt);



% Set initial condition
%%
%Defining a matrix for all of the distributions based off of the perturbed
%fields

for i = 1:samples1
```

```
        fr(:,:,i,j)=f0(XX,VV,epsm(i,1), epsm(i,2));
        fr(:,:,i,j)=fft(fr(:,:,i,j),[],1);
    end

%%

%%
%Unit Test for UQ
%figure('Name', 'Phase Space Density', 'Numbertitle','off');
%Adding loop to consider perturbations in the electric field
for i =1:samples1
    for tdx=1:numt
    for sdx=1:length(rksd)
        rho=sum(fr(:,:,i),2).*(vmax-vmin)/Nv; %rho(x) -> integrate over v
        rho(1)=0; %remove (ion) background
        E=rho./(-1j*(2*pi/L)*fftshift(((1:Nx)-Nx/2-1).'));
        % remove constant fourier mode
        E(1)=0;

        if (sdx==length(rksd))
            %fieldenergy(tdx)=sum(permute(E,[2 1])*E)/2; %L2 norm of Electric field
            %EE(:,tdx)=E;
        end

        E=ifft(E,'symmetric');
        E=E+E_ext(x,tdx*dt);
        %plot(x,ifft(E,'symmetric'));
             fr(:,:,i)=ifft(fr(:,:,i),[],1,'symmetric'); %back transform
        % f is fully backtransformed, calculate diagnostics
        if (sdx==length(rksd))

                  kineticenergyr(tdx,i)=sum(fr(:,:,i),1)*v.^2*(vmax-vmin)/Nv;
                 %sh_entropy(tdx)=sum(sum(f1.*log(abs(f1))))*L/Nx*(vmax-vmin)/Nv;
            %kl_entropy(tdx)=sum(sum(f1.*log(abs(f1./f0(XX,VV)))))...
                 %*L/Nx*(vmax-vmin)/Nv;

        end

        % Fourier transform in v
        fr(:,:,i)=fft(fr(:,:,i),[],2);
        % Build Matrix for spatial advection
         m=fftshift((1:Nv)  -Nv/2-1);
         VSHIFT=exp((1j*dt*rksc(sdx)*2*pi/(vmax-vmin))*E.*m);

        fr(:,:,i)=fr(:,:,i).*VSHIFT;
        fr(:,:,i)=ifft(fr(:,:,i),[],2,'symmetric');
        %Advection in X
        fr(:,:,i)=fft(fr(:,:,i),[],1);
        fr(:,:,i)=fr(:,:,i).*XSHIFT(:,:,sdx);
    end


    end
    %Plot 1 kinetic energy sample from a given pertubration
```

```
end


%%
%Setting up arrays to save simulation outputs to perform statistical
%analysis on, evaluate on the sparse grid interpolant for comparison, and
%whatever else is needed.
save ('RandLD1.mat', 'rho', 'E', 'kineticenergyr', 'fr', 'epsm', 'eps1d', 'eps2d', '-v7.3')

%%
keqoir = kineticenergyr(390:410,:,:);
mkeqoir = zeros(100,1);
for i = 1:100
    mkeqoir(i) = mean(keqoir(:,i,1));
end
%%
%Creating a vector of the mean electron kinetic energy values, the
%quantitiy of interest in this UQ experiment.
meankineticenergyr = zeros(samples1,1);
  for i = 1:samples1
        meankineticenergyr(i) = mean(keqoir(1,i,:));
  end




%%
```

```
%%
figure('Name','Kinetic Energy','Numbertitle','off');
for i = 1:10
    for j = 1:3
    semilogy(time, kineticenergyr(:,i,j));
    xlabel('time'); %grid on;
    ylabel('kinetic energy');
    hold all
    end
end
%%
veci = zeros(samples2,1);

for i = 1:samples2
    veci(i) = keqoir(1,1,i);
end


%%
figure('Name','Kinetic Energy @ t = 400','Numbertitle','off');
    plot(eps2d, v, '*');
    xlabel('Perturbation in Amplitude'); %grid on;
    ylabel('kinetic energy');
    hold all
```

Sparse Grid Construction Code, KDE Construction Code, and Graphical Code

```
a%function(  grid  , kldiv) =  surconstruct(dim, level , type grid? Implement options to cho
dim = 2;
```

```
level = 5;

grid = sgpp.Grid.createPolyGrid(dim,6);

grid.getGenerator().regular(level);

gridStorage = grid.getStorage();


%%

%Retrieving Linear Clenshaw−Curtis Grid Points

y = zeros(gridStorage.getSize(),dim);

    for i = 0:gridStorage.getSize()−1

            gp = gridStorage.getPoint(i);

            y(i+1,:) = [gp.getStandardCoordinate(0), gp.getStandardCoordinate(1)];

    end

    gridsize(level) =length(y);

%%

% Sparse grid, grid point conversion for sparse grid surrogate construction

% of Vlasov−Poisson Nonlinear Landau Damping Problem

% %Convert [0,1] to [a,b] for each random variable (f^(−1) = x*(b−a) +a)

% lower1 = −1; upper1 = 1;

% lower2= 0.0 ; upper2 = pi/2;

% xconv = zeros(length(y),2);

% xconv(:,1) = y(:,1)*(upper1 − lower1) + lower1; %Scaling sparse grid points to be from [

% xconv(:,2) = y(:,2)*(upper2 − lower2) + lower2;

% eps1d = xconv(:,1);

% eps2d = xconv(:,2);

% samples = length(xconv(:,1));

% ynew = cat(2, eps1d, eps2d);

%%
```

```
%Modified script to incorporate uncertainty in background field and
%instability
%eps=0.5; % Amplitude of perturbation, 0.05 for linear, 0.5 for nonlinear
kx=0.5;    % Wave vector
L=2*pi/kx; % length of domain
qm=-1;    % negative charge to mass = electrons, $\frac{q}{m}=-1$

% Initial condition
f0 =@(x,v,eps1d, eps2d)   (1+eps1d.*cos((kx)*x))./sqrt(2*pi).*exp(-0.5.*(v + eps2d).^2);

% Background to be subtracted from the phase space plot
background=@(x,v, mu) 1./sqrt(2*pi).*exp(-0.5.*(v-mu).^2);
% External electric field
E_ext=@(x,t) 0.*x+0.*t;

dt=0.1; % Time step
tmax=50;
rungekutta_order=3; %Order of the runge Kutta time integrator



Nx=32; % Number of cells in spatial direction
Nv=32; % Number of cells in velocity direciton
vmax=4.5;
vmin=-4.5;
%%
v=linspace(vmin,vmax,Nv+1).';v=v(1:end-1);
x=linspace(0,L,Nx+1).';x=x(1:end-1);
```

```
[XX,VV]=ndgrid(x,v);
%%
switch rungekutta_order
    case 1
        rksd=[1, 0]; %Symplectic Euler
        rksc=[0, 1];
    case 2
        rksd=[0.5, 0.5 ];
        rksc=[0, 1];
    case 3
        rksd=[2/3, -2/3, 1  ];
        rksc=[ 7/24, 3/4, -1/24];
    case 4
        rk4sx=real((2^(1/3) +2^(-1/3)-1)/6);
        rksd=[ 2*rk4sx +1 , -4*rk4sx-1, 2*rk4sx+1, 0];
        rksc=[ rk4sx + 0.5 , -rk4sx, -rk4sx, rk4sx +0.5];
end


%Build Matrix for spatial advection and all stages
XSHIFT=zeros(Nx,Nv,length(rksd));
m=fftshift(((1:Nx)-Nx/2-1));
for jdx=1:length(rksd)
   XSHIFT(:,:,jdx)=exp((-1j*dt*rksd(jdx)*2*pi/L)*(v*m).');
end
%Preallocate matrix for velocity advection
VSHIFT=zeros(Nx,Nv);
```

```
%diagnostic
numt=ceil(tmax/dt);
time = linspace(0,tmax,numt);
fieldenergy=zeros(numt,1);
kineticenergyi=zeros(numt,gridStorage.getSize());

EE=zeros(Nx,numt);



% Set initial condition
%%
%Defining a 3-D Array the initial particle distributions based off of the
%perturbed data.



for i = 1:gridStorage.getSize()


fi(:,:,i)=f0(XX,VV,y(i,1), y(i,2));
fi(:,:,i)=fft(fi(:,:,i),[],1);


end


% for i = 1:samples
%       b(:,:,i) = background(
% end
```

```
%%
%Unit Test for UQ with Sparse Grid Points
%figure('Name', 'Phase Space Density', 'Numbertitle','off');
%Adding loop to consider perturbations in the electric field
%Unit Test for UQ
%figure('Name', 'Phase Space Density', 'Numbertitle','off');
%Adding loop to consider perturbations in the electric field
 for i =1:gridStorage.getSize()



    for tdx=1:numt
    for sdx=1:length(rksd)
        rho=sum(fi(:,:,i),2).*(vmax-vmin)/Nv; %rho(x) -> integrate over v
        rho(1)=0; %remove (ion) background
        E=rho./(-1j*(2*pi/L)*fftshift(((1:Nx)-Nx/2-1).'));
        % remove constant fourier mode
        E(1)=0;

        if (sdx==length(rksd))
            %fieldenergy(tdx)=sum(permute(E,[2 1])*E)/2; %L2 norm of Electric field
            %EE(:,tdx)=E;
        end

        E=ifft(E,'symmetric');
        E=E+E_ext(x,tdx*dt);
        %plot(x,ifft(E,'symmetric'));
            fi(:,:,i)=ifft(fi(:,:,i),[],1,'symmetric'); %back transform
        % f is fully backtransformed, calculate diagnostics
        if (sdx==length(rksd))

                kineticenergyi(tdx,i)=sum(fi(:,:,i),1)*v.^2*(vmax-vmin)/Nv;
               %sh_entropy(tdx)=sum(sum(f1.*log(abs(f1))))*L/Nx*(vmax-vmin)/Nv;
           %kl_entropy(tdx)=sum(sum(f1.*log(abs(f1./f0(XX,VV)))))...
               %*L/Nx*(vmax-vmin)/Nv;

        end

        % Fourier transform in v
        fi(:,:,i)=fft(fi(:,:,i),[],2);
        % Build Matrix for spatial advection
         m=fftshift((1:Nv)   -Nv/2-1);
         VSHIFT=exp((1j*dt*rksc(sdx)*2*pi/(vmax-vmin))*E.*m);

        fi(:,:,i)=fi(:,:,i).*VSHIFT;
        fi(:,:,i)=ifft(fi(:,:,i),[],2,'symmetric');
        %Advection in X
        fi(:,:,i)=fft(fi(:,:,i),[],1);
        fi(:,:,i)=fi(:,:,i).*XSHIFT(:,:,sdx);
    end

    end
    %Plot 1 kinetic energy sample from a given pertubration

end
```

```
%% Finding the time centered average around the 400th time point of the simulation.

keqoi = kineticenergyi(390:410,:); %Setting up an array for the quantity of interest at a t
% when the system is in a steady state, across all sparse grid points.
mkeqoi = zeros(gridStorage.getSize(),1);
for i = 1: gridStorage.getSize()
    mkeqoi(i) = mean(keqoi(:,i));
end
%%
% Creating a coefficients vector for the interpolant - Running more than
% once with different data causes the script to crash
alpha = sgpp.DataVector(gridStorage.getSize());
alpha.setAll(0);
for i = 0:gridStorage.getSize()-1
    gp = gridStorage.getPoint(i);
    alpha.set(i, mkeqoi(i+1));
end
%Hierarchisation of coefficients of the polynomial interpolant - I am not
%sure what is going on here.
sgpp.createOperationHierarchisation(grid).doHierarchisation(alpha);


%%
% Mapping random sample points to grid values, and then creating a 2-D grid
% of input values for the interpolant.
```

```
load('RandLD1.mat', 'epsm', 'kineticenergyr') %Loading the necessary data from brute force
%% Turn this into a function or program - Find the KDE for one
%Finding the mean kinetic energy for every perturbation in first dim, fixed 1
%point for second dim.
mkeqoi1 =zeros(length(epsm(:,1)),1);
for i = 1:length(epsm(:,1))
    mkeqoi1(i) = mean(kineticenergyr(390:410,i));
end






%% Test to ensure interpolant is setup properly by evaluating the interpolant at the initia
for i = 1:gridStorage.getSize()
    p = sgpp.DataVector(dim);
    p.set(0, y(i,1));
    p.set(1, y(i,2));
    opEval = sgpp.createOperationEvalNaive(grid);
    a = opEval.eval(alpha,p)
if a ~= mkeqoi
    fprintf('Function not interpolated properly')
    break
end
end
%% MSE Calculation and Vector Setup for KDE
u1 = zeros(length(epsm(:,1)),1);
```

```
%Evaluating interpolant at a set of perturbation points from the brute
%force simulation.
for i = 1:length(epsm(:,1))
p = sgpp.DataVector(dim);
p.set(0, epsm(i,1));
p.set(1, epsm(i,2));
opEval = sgpp.createOperationEvalNaive(grid);
u1(i) = opEval.eval(alpha,p);
clear p opeval
end
%%
save('dist.mat', 'u1', 'mkeqoi1')
%%
%Calculating interpolant error by evaluating MSE between sparse grid and
% brute force results.
for i = 1:length(epsm(:,1))
    e(i) = (u1(i) - mkeqoi1(i))^2;%
end
    ef1 = sqrt(sum(e))/length(epsm(:,1))
%% Creating KDEs
pd = fitdist(mkeqoi1, 'Kernel')
x = min(mkeqoi1)-2:1:max(mkeqoi1) + 2;
p = pdf(pd,x);
pd1 = fitdist(u1, 'Kernel')
x1 = min(u1)-2:1:max(u1) + 2;
p1 = pdf(pd1,x1);
```

```
%% Kullback-Liebler Divergence Calculation
%Choosing the min/max values of the data to setup a region of integration
if min(u1) <= min(mkeqoi1)
    minv = min(u1);
else
    minv = min(mkeqoi1);
end


if max(u1) <= max(mkeqoi1)
    maxv = max(mkeqoi);
else
    maxv = max(u1);
end
minv = fix(minv);
maxv = ceil(maxv);
dx = 0.001
range = minv:dx:maxv;
%Trapezoidal rule for integration
P = pdf(pd, range);
Q = pdf(pd1, range);
klint = P.*log(P./Q);
kldiv = (klint(1) + klint(length(klint))+ 2*sum(klint(2:length(klint)-1)))*dx/2;
%%
save('pdf.mat', 'klv', '-append')
%%
load('pdf.mat', 'x7', 'p7','x','p')
figure(1)
```

```
hold all

plot(x,p, '--')
plot(x7,p7)

title('KDE Comparison - Level 7 Surrogate Model vs Monte Carlo Simulation') % of Time Cent
xlabel('KDE Support')
ylabel('Density')
legend('Surrogate KDE', 'Monte Carlo Surrogate')
%%
load('pdf.mat', 'x','p','p1','x1', 'p2','x2', 'p3', 'x3', 'p4', 'x4', 'p5', 'x5', 'p6', 'x6
%%
figure(2)
hold all

%x = min(mkeqoi1)-2:1:max(mkeqoi1) + 2;

plot(x,p, '-*', x1,p1, x2,p2, x3,p3, x4, p4, x5, p5, x6, p6, x7, p7)

%pd = fitdist(u1, 'Kernel')
% x = min(u1)-2:1:max(u1) + 2;
% p = pdf(pd,x);
% plot(x,p)
title('KDE Convergence Graph Level 1-7 Surrogates vs Monte Carlo Simulation') % of Time Ce
legend('Monte Carlo KDE')
ylabel('Density')
xlabel('KDE Support')
```

```
%% Error analysis graph plotting theoretical polynomial interpolation error vs sparse grid
%%
hold all
pd = fitdist(mkeqoi2', 'Kernel')
x = 73:1:105;
p = pdf(pd,x);
plot(x,p)

pd = fitdist(u2, 'Kernel')
x = 73:1:100;
p = pdf(pd,x);
plot(x,p)

%%
figure(3)
x = linspace(1,7,7)
plot(x, klv, '*')
title('Plot of Kulbach-Liebler Divergence vs Sparse Grid Level')
xlabel('Surrogate Level')
ylabel('KL Divergence')
figure(4)
plot(gridsize, klv, '*')
title('Plot of K-L Divergence Vs. Grid Size (Number of Grid Points)')
xlabel('Sparse Grid Size')
ylabel('K-L Divergence')
%% Plotting theoretical error vs sparse grid level and actual error vs sparse grid level
%Find the normalized error
```

```
hold all
x = linspace(1,7, 7);
norme = error./rssq(error);
etheory = 2.^(-2*x);
normet = etheory./rssq(etheory);
let = log2(normet);
le = log2(norme);
figure(1)
title( ' Interpolation Normalized MSE Comparison')
plot(x, norme, '*', x, normet, 'o')
legend('Experimental Error', 'Theoretical Error')
title('Theoretical & Experimential Interpolation Error')
xlabel('Sparse Grid Level')
ylabel('Error')


figure(2)
plot(x, le, '*', x, let, 'o')
title( 'Normalized Semi-Log Error Graph')
title('Theoretical & Experimential Interpolation Error')
xlabel('Sparse Grid Level')
ylabel('Error')
%%
figure('Name','Kinetic Energy','Numbertitle','off');
for i = 1:10
    semilogy(time, kineticenergyi(:,i));
    xlabel('time'); %grid on;
    ylabel('kinetic energy');
    hold all
end
%%
x = 1:1:7
plot(x,error,'*')
```

References

[1] A.S. Eddington. *The Internal Constitution of Stars.* Cambridge University Press.

[2] M. L. E. Oliphant, P. Harteck, and Lord Rutherford. Transmutation effect observed with heavy hydrogen. 144:692 – 703.

[3] W. Horton. Drift waves and transport. 71(3):735–778.

[4] C.C. Petty et al. Gyroradius scaling of electron and ion transport. 74(10):4.

[5] M.A. Chilenski et al. Improved profile fitting and quantification of uncertainty in experimental measurements of impurity transport coefficients using gaussian process regression. 55(23012):21.

[6] Jeffrey P. Friedberg. *Plasma Physics and Fusion Energy.* Cambridge University Press, 1st edition.

[7] John Wesson. *Tokamaks.* Oxford University, 3rd edition.

[8] N.T. Howard, M. Greenwald, D.R. Mikkelsen, M.L. Reinke, A.E. White, D. Ernst, Y. Podpaly, and J. Candy. Quantitative comparison of experimental impurity transport with nonlinear gyrokinetic simulation in an alcator c-mod l-mode plasma. 52(6):063002.

[9] Francis F. Chen. *Introduction to Plasma Physics and Controlled Fusion.* Springer, 2nd edition.

[10] Jacob Ameres. Stochastic and spectral particle methods for plasma physics.

[11] Jochen Garcke. Sparse grids in a nutshell. In Jochen Garcke and Michael Griebel, editors, *Sparse Grids and Applications*, pages 57–80. Springer Berlin Heidelberg.