Zusammenfassung der Dissertation
zum Thema: “Finite Difference Code for 3-D Plasma Edge Modeling”

vorgelegt von
Neil McTaggart

Modeling ergodic regions in magnetized plasmas offers special difficulties due to the strong anisotropy of the transport and the complex geometry of the computational domain. In the present work a method is developed for finite difference discretization of a transport equation in order to model electron heat conduction in the plasma edge of fusion experiments. This is to gain insight into the physics of such devices and to optimize their design and performance.

Using field line tracing, meshes are generated to represent plasmas in W7-X and NCSX. The transport equation is discretized along the field lines to allow accurate evaluation of the extreme anisotropic transport with an anisotropy factor between parallel and perpendicular transport of $\sim 10^4 - 10^7$. This large anisotropy poses a real challenge to avoid numerical diffusion. Using an optimization criterion in the mesh generation it is possible to reduce numerical diffusion to $\sim 10^{-5} m^2 s^{-1}$. This is much less than the physical diffusion, $\sim 1.0 m^2 s^{-1}$. The problem of ergodicity is handled by using local magnetic coordinates. This allows a finite difference discretization of the transport equation as long as the distance between the mesh points is well below the typical length where neighboring field lines diverge exponentially (Kolmogorov length). The use of local magnetic coordinates allows a complete description of the system without additional approximation. However, the price to pay for this complete model is a full metric tensor including non-diagonal terms. The computational process is summarized in Fig.0.1.

In the present work, a variety of studies are done to investigate the physics in the plasma edge in the presence of ergodicity. The results show that the W7-X finite beta case allowed cascading of energy into regions which were inaccessible in the vacuum case, resulting in a smoothing of the island structures in the edge region. Thus, the broadening of the solution in the islands is due to an indirect ergodic effect. The island structures are not prominent in the solution, being masked by the general flattening of the temperature profile in the edge region.

Holding the core boundary condition constant and reducing the radial transport coefficient resulted in a decoupling, in temperature space, of the open field lines in the edge region from the closed field lines in the core region, thus inhibiting transport from the core. By comparison, reducing the core boundary condition, while holding the radial transport constant resulted only in a scaling down of the temperature solution while *not* inhibiting transport from the core.

By investigation of flux densities at the ends of the open field lines it was found that the longer open field lines played a greater role in removing heat from the plasma core because they had more contact with the ergodic region surrounding the plasma core.

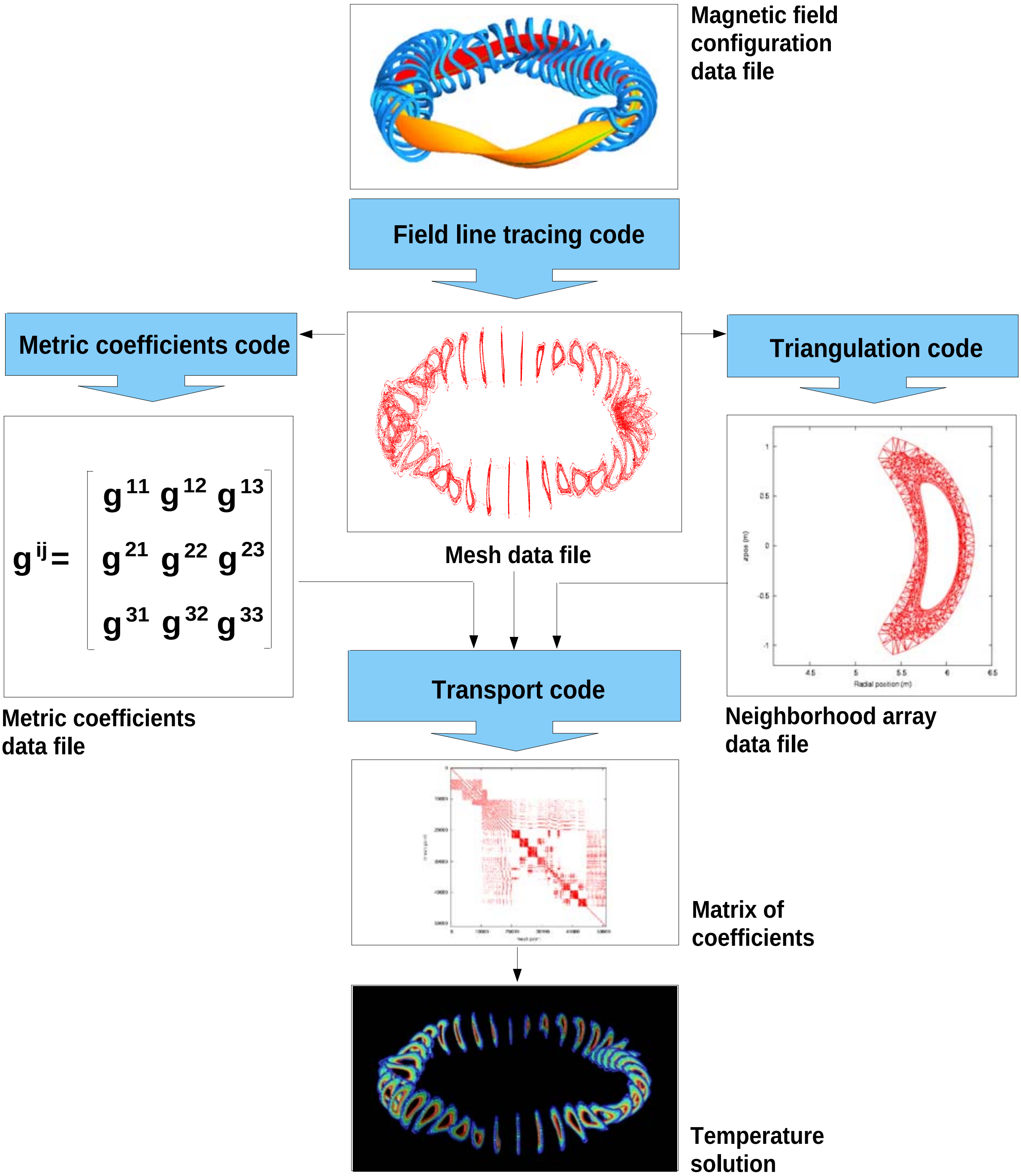
Magnetic field configuration data file
Field line tracing code
Metric coefficients code
Triangulation code
$g^{ij} = \begin{bmatrix} g^{11} & g^{12} & g^{13} \\ g^{21} & g^{22} & g^{23} \\ g^{31} & g^{32} & g^{33} \end{bmatrix}$
Mesh data file
Transport code
Metric coefficients data file
Neighborhood array data file
Matrix of coefficients
Temperature solution

Figure 0.1: *Summary of computational process.*

Contents

1 Introduction **7**

2 Motivation **9**

3 Magnetic confinement concepts and plasma edge physics **11**

3.1 Transport along field lines 14

3.2 Plasma wall interaction 15

3.3 Ergodicity and transport in the plasma edge 19

4 State of the art **21**

5 Mesh generation **23**

5.1 Tracing closed field lines 27

5.2 Tracing open field lines 29

5.3 The stencil 30

6 The Transport Equation 33

6.1 3D Rectangular form of the transport equation. 34

6.2 Vector form of the transport equation 35

7 Local Magnetic Coordinate Systems 37

7.1 The Metric Tensor 38

7.1.1 Coordinate surfaces and coordinate curves 39

7.1.2 Basis Vectors 40

7.1.3 Metric coefficients, g_{ij} and g^{ij} 40

7.1.4 The metric tensor in the transport equation 42

8 Triangulation of Poincaré plots 45

8.1 Delaunay triangulation. 45

8.2 Optimizing the triangulation algorithm. 46

8.3 Secondary points. 49

9 Constructing the Coefficient Matrix 53

9.1 Poloidal Flux Divergences 55

9.2 Toroidal flux divergence 60

9.3 Mixed flux divergences, $ij = 13, 23$ 62

9.4 Mixed flux divergences, $ij = 31, 32$ 63

9.5 Fundamental matrices 66

9.6 Component matrices 67

9.7 Time derivative 68

9.8 Constructing the final coefficient matrix 68

10 General Ansatz **71**

11 Results and Discussion **75**

11.1 Numerics 75

11.1.1 Numerical diffusion 75

11.1.2 Testing the sensitivity to the metrics 83

11.2 Physics 84

11.2.1 W7-X 84

11.2.2 NCSX 105

12 Conclusions and outlook **111**

13 Appendices **113**

13.1 Evaluating the metric coefficient 113

13.2 Poincaré plots in W7-X 114

13.3 Triangulations in W7-X 117

13.4 Least squares method 120

13.4.1 The Least Squares fit function 121

13.4.2 Differentiating the Least Squares fit function 121

13.4.3 The resulting system of equations 122

13.4.4 Inverting the least squares coefficient matrix [α] 123

13.4.5 Matrix [A0], poloidal gradients at the secondary points 124

13.4.6 Matrices [A3] and [A5], poloidal gradients at the primary points . . 125

13.5 Sparse matrix storage and manipulation 126

13.5.1 Coordinate format 126

13.5.2 Compressed Sparse Row format 127

13.5.3 Summary of matrix operations in the transport code 128

14 Acknowledgements **135**

15 Curriculum Vitae **137**

1 Introduction

The present work is concerned with developing a finite difference code for modeling the physics of the plasma edge in fusion devices. We are interested in a quantitative understanding of the energy transport from the plasma core to the vessel wall in order to optimize steady state operation of magnetic fusion devices. One problem to consider is the broadening of the power deposition pattern on the wall structures. This broadening is necessary to avoid power loads that exceed the engineering limits. In particular, it is desirable to keep the interaction between the plasma and the wall structures away from the hot core so as not to adversely affect core performance. The present work is focused on W7-X [1] which is characterized by a complex 3D geometry and an ergodic structure of field lines in the plasma edge. This introduces the need for the development of new concepts and tools for handling such situations.

In any transport code for modeling the physics in a magnetized plasma the following characteristics must be considered. First, the extreme anisotropy of energy transport caused by very strong electron heat conduction along the magnetic field lines. This parallel conduction is typically $10^4 - 10^7$ [2, 3] times higher than the transport across the field lines. Second, conduction along the field lines is mainly by Coulomb collisions and is strongly non-linear, with $\kappa_{\|} \propto T_e^{5/2}$. By contrast, conduction across the field lines is driven by turbulence in the plasma [4] and can be described as a diffusive process with $\chi_{\perp} \approx 1 m^2 s^{-1}$. Third, the geometry of the plasma may further complicate the application of the computational method, especially for 3D devices such as W7-X. Fourth, in the edge region of a fusion plasma, the field lines may have an ergodic geometry, meaning spatially chaotic trajectories.

The special advantage of a finite difference code in modeling plasma physics in fusion experiments is its applicability to any device regardless of anisotropy, ergodicity or geometry. Essentially, the code described in the present work requires only a magnetic field configuration data file and specified boundary conditions in order to generate temperature solutions on cross-sections of the plasma. This work lays the foundation for a complete finite difference fluid code. A finite difference code for modeling transport in magnetized plasmas, is an additional computational tool to be used alongside existing particle and finite volume codes, and for comparison with experimental results.

A key question addressed in the present work is the mechanism of energy transport from the core to the divertor plates. For example, what are the effects of the radial and parallel transport coefficients, and are the shorter or the longer open field lines in the plasma edge more important in energy transfer?

The present work is structured as follows. The first part introduces, after a short motivation, the basics of magnetic confinement and edge physics followed by a summary of the current state of the art in edge modeling. Chapters 5 to 8 describe the process of field line tracing to generate a mesh on which the transport equation, using the concept of local magnetic coordinates, is later discretized. Chapter 9 describes the process of discretizing the transport equation and generating the matrix of coefficients. The whole computational process is summarized in chapter 10. Chapters 11 and 12 give the results and discussion of various studies to investigate the numerics of the transport code and the physics of the transport in the plasma edge, followed by conclusions and outlook.

2 Motivation

The development and maintenance of a technological society is constrained by the amount of energy available to it. When the energy supply is based on finite resources and produces harmful by-products, as in the combustion of fossil fuels or the splitting of heavy isotopes, then it becomes imperative to find alternative sources. Energy generation by nuclear fusion has the potential to play a primary role in ending our dependence on fossil fuels and nuclear fission, and to supply our long term energy needs.

The main approach to realizing a commercial fusion reactor is the magnetic confinement concept [5] in which powerful magnetic fields confine a plasma to create the conditions necessary for fusion reactions to occur. For steady state operation of a fusion reactor the plasma must be kept stable and free of impurities.

In a steady state fusion experiment, fuel is pumped into the reactor so that the plasma may continue burning with fusion reactions which produce heat and helium ash. The ash must be removed from the core to prevent contamination of the plasma [6]. Further contamination may come from the reactor walls and the divertor plates as a result of plasma erosion. If the impurities in the plasma core exceed a critical threshold, only a few percent, then the fusion reactions will stop [7].

The goal of this work is to develop a finite difference code for modeling electron heat transport in order to further understand the particle and power exhaust in a fusion reactor [8, 9, 10]. This is a crucial element because it defines strict limits on the operation of a reactor. Such an operational space must take into account the power load and erosion of the target plates, and dilution of the core plasma by helium ash. The present work considers only the power loading problem. A simple estimate of the power load $q_\perp$, assuming no additional transport channel, can be made for W7-X:

$$q_\perp = \frac{P_{heat}}{n \times 2\pi R \times \Delta_e} \quad \Rightarrow \quad \frac{24MW}{10 \times 2\pi 5.5m \times 0.05m} \quad = \quad 13.9MW/m^2 \tag{2.1}$$

where P_{heat} is the heating power, n is the number of sets of target plates in W7-X, R is the average major radius of the experiment, and Δ_e is the energy decay length at the target plates, found empirically. Equation 2.1 therefore defines the power load problem but the large value obtained can be reduced by angling the target plates [11], broadening the temperature profile in the edge region by ergodic effects, and by neutral losses and impurity radiation.

Fig.2.1 shows a promotional picture of a temperature solution on a W7-X mesh obtained by the finite difference code developed in the present work.

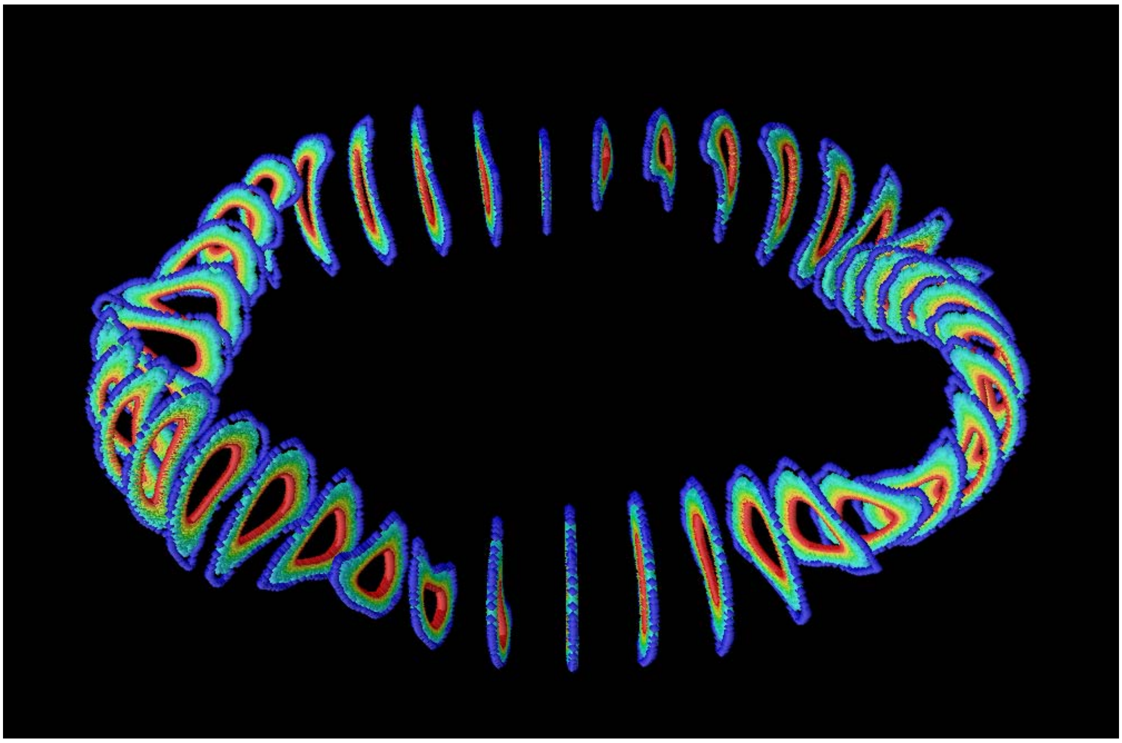

Figure 2.1: *An electron temperature solution on a W7-X mesh containing 40 cuts (total points* ~ 51000*. Each mesh point is colored according to temperature. The red core points are at 200 eV. The purple outer boundary points are at 10 eV.*

3 Magnetic confinement concepts and plasma edge physics

In a magnetic confinement fusion reactor a plasma is confined by powerful magnetic fields in order to create the conditions necessary for fusion reactions to occur in the plasma core and release energy [12]. This energy can be converted to heat and used to drive steam turbines to generate electricity. The most promising designs for the magnetic confinement concept [5] are tokamaks and stellarators which are toroidal devices. Fig.3.1 shows a generalized tokamak configuration.

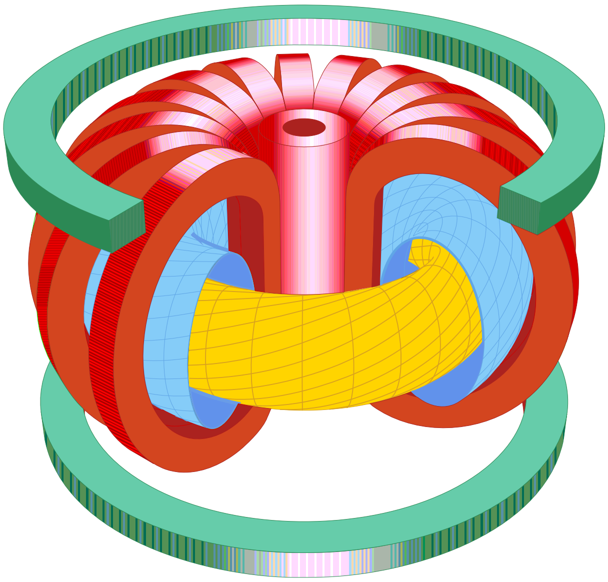

Figure 3.1: *In a tokamak the plasma cross-section remains a constant shape at all toroidal positions, therefore the tokamak may be treated as a 2D device. A central transformer coil, shown in pink, drives a current in the plasma.*

In the tokamak the central transformer coil adds a helical twist to the magnetic field lines. This short-circuits the creation of electric fields and an outward drift which would destabilize the plasma [13, 14]. The current flowing in the plasma boosts the core temperature by Ohmic resistance, though only at relatively low temperatures. Higher temperatures are achieved by additional heating methods such as Neutral Beam Injection (NBI), and Ion and Electron Cyclotron Resonance Heating (ICRH, ECRH). The use of the transformer coil means that the tokamak operates as a pulsed device, [5].

A successful fusion power plant must operate in a steady state mode over long time intervals. Therefore the plasma must be kept clean and stable and we are faced with the problem of particle and power removal from the plasma core. Fig.3.2 shows two cross-sections of a tokamak plasma and two basic concepts of power and particle removal.

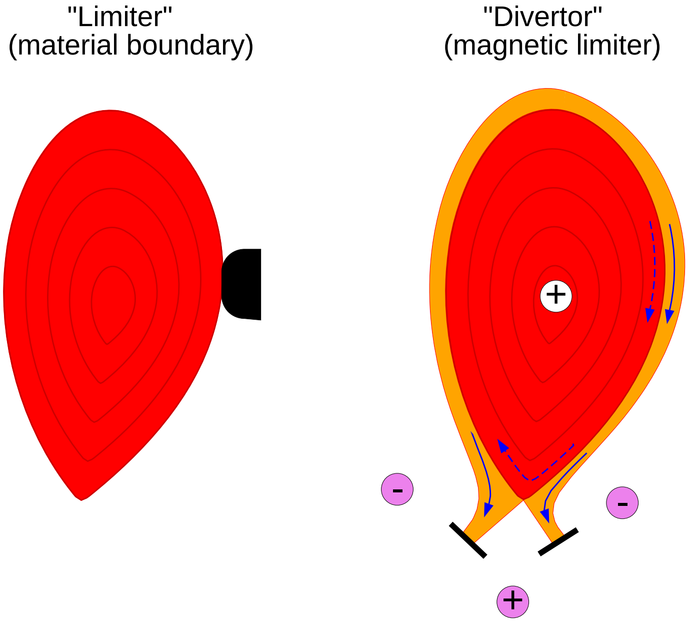

Figure 3.2: *(left) A limiter intersecting the plasma edge. (right) Additional poloidal coils diverting the plasma edge onto 2 divertor plates at the bottom.*

In the first concept a limiter structure directly impinges on the hot plasma core. The advantage of this concept is its simplicity and its direct influence on the core. However, direct contact with the core can produce high power loads on the limiter structure, resulting in severe erosion and impurity problems. In the second concept, additional coils create an x-point in the separatrix (the boundary between closed and open field lines). In this way, the outer part of the plasma is scraped off and diverted onto the divertor plates [15]. This so-called Scrape-Off-Layer (SOL) is the region of open field lines in the plasma edge which intersect the divertor plates. It plays a critical role in maintaining the purity and stability of the plasma. The purpose of such plasma edge diversion is to move the interaction zone between the divertor plates and the plasma, away from the plasma core. This offers the possibility of good impurity control [16] and also the reduction of the heat flux to the divertor plates due to radiation losses. The divertor concept is now the standard solution for all reactor designs [17].

Fig.3.3 shows the W7-X stellarator configuration. In a stellarator there is no central transformer coil and practically no plasma current. The lack of a central transformer coil allows the possibility of steady state (non-pulsed) operation [18]. The short-circuiting of electric fields is achieved by the complex shape of the plasma.

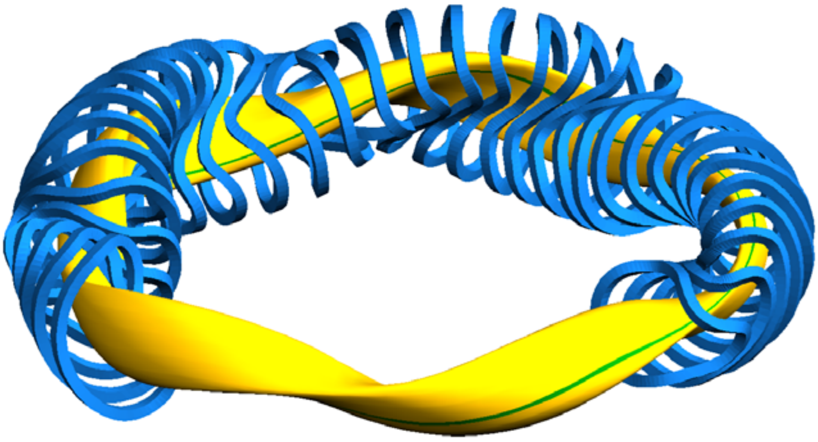

Figure 3.3: *In the W7-X stellarator the cross-section of the plasma, shown in yellow, changes with toroidal position, therefore a stellarator can only be treated as a fully 3D device. The plasma is shaped by a series of non-planar field coils shown in blue.*

In W7-X, which is an optimized stellarator [1], we make use of the intrinsic islands which naturally divert the exhaust power fluxes from the core into the islands. By placing target plates at the interaction zones we can introduce a similar concept of plasma wall interaction control away from the core, like in the tokamak divertors. Therefore, such islands are called island divertors, and because of the 3-dimensionality of the plasma in W7-X these divertors have a more complex shape. See Fig.3.4.

In the stellarator concept, there is the possibility to operate at higher plasma density without the danger of disruptions because the stellarator is internally current-free. Also, one can use the 3-dimensionality for impurity and neutral screening. Potential problems are the danger of producing local instabilities even earlier than in the tokamak divertor, and the more complex baffling and pumping due to the plasma geometry.

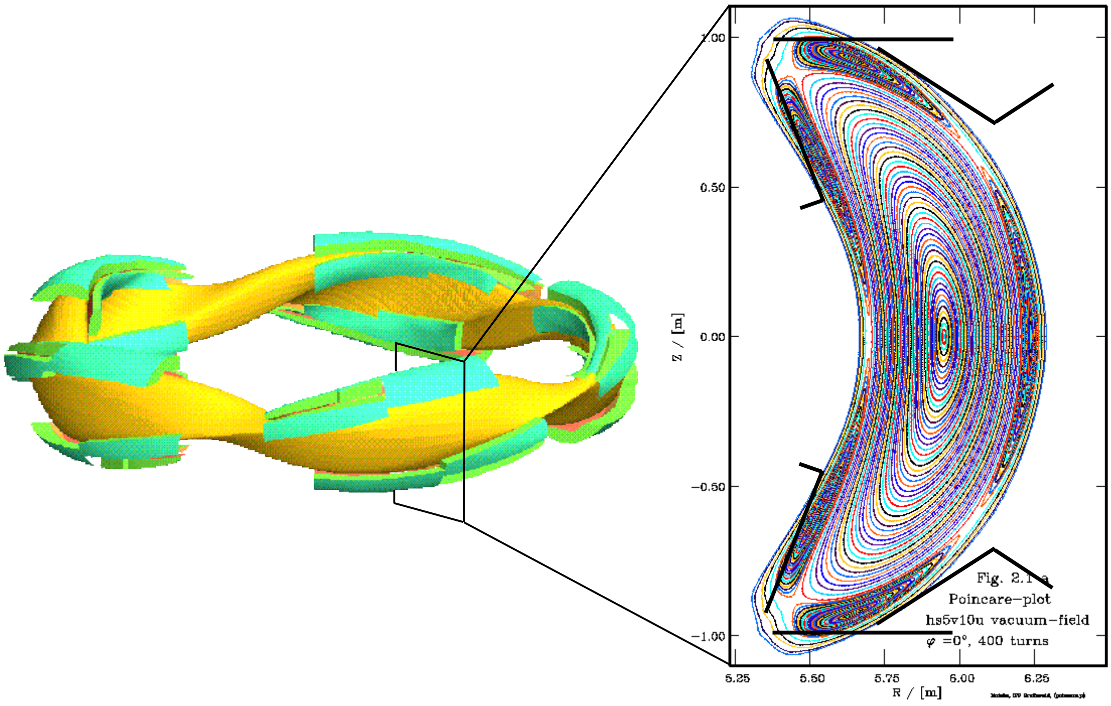

Figure 3.4: *Possible island divertor configuration in W7-X. The divertor structures are shown in green. At right is a plasma cross-section showing how the divertor plates intersect the plasma at toroidal position* $\phi = 0°$.

3.1 Transport along field lines

The dominant mode of heat transfer in the magnetized plasma is electron heat conduction along magnetic field lines which is governed by Coulomb collisions [19]. The electron mean-free path length $\lambda_e = v\tau_e = \sqrt{3}v_{th}\tau_e$ for a test particle with energy $mv^2/2 = \frac{3}{2}kT$ and a thermal velocity $v_{th} = (kT/m)^{1/2}$ is:

$$\lambda_e \quad = \quad 1.5 \cdot 10^{16} \times \frac{T_e^2}{n} \tag{3.1}$$

where T is temperature in electron Volts (eV) and the other quantities are in MKS-units. This is the sum of the small angle scattering producing a 90° deflection. According to the Rutherford scattering formula the mean-free path scales with v^4. The ion-ion and electron-ion mean-free paths are practically identical to the electron-electron mean-free path, within a factor of $1/\sqrt{2}$. The electron heat conduction along the magnetic field is:

$$q_{e,\parallel} \quad = \quad -\kappa_{e,\parallel}\nabla_\parallel T_e \quad = \quad -\chi_{e,\parallel} n_e \nabla_\parallel T_e \tag{3.2}$$

where q_e and T_e are the electron heat flux and temperature respectively. The heat diffusivity is given by:

$$\chi_{e,\|} \quad \approx \quad 1.8 \cdot 10^3 \times T_e^{5/2} \tag{3.3}$$

This defines a very strong non-linearity in the heat conduction.

3.2 Plasma wall interaction

A plasma in contact with a boundary forms a sheath layer at the boundary. This is due to the electron velocity being much higher than the ion velocity (mass difference) [19, 20]. The sheath layer can be divided into different regions (sheath and presheath) which can be characterized qualitatively by the properties of the plasma in this region [21] (see Fig.3.5). The recyling neutrals get ionized at a distance λ_{ion} in the presheath where ions are accelerated to sound speed at the sheath edge x_S, as described by the Bohm criterion [2, 3].

For fluid models the sheath effects are introduced as effective boundary conditions at the sheath edge, that is, the acceleration of the plasma ions to velocities which satisfy the Bohm condition [22]:

$$v_{i,potential} \quad \geq \quad \sqrt{\frac{kT_e}{m_i}} \tag{3.4}$$

and an effective boundary condition for the heat flux:

$$Q_{e,x}(x_s heath) \quad = \quad \delta_e T_{e,sheath} \Gamma_{e,x} \tag{3.5}$$

taking into account the electron reflection properties of the sheath and introducing a kinetic factor of about $\delta_e = 5$.

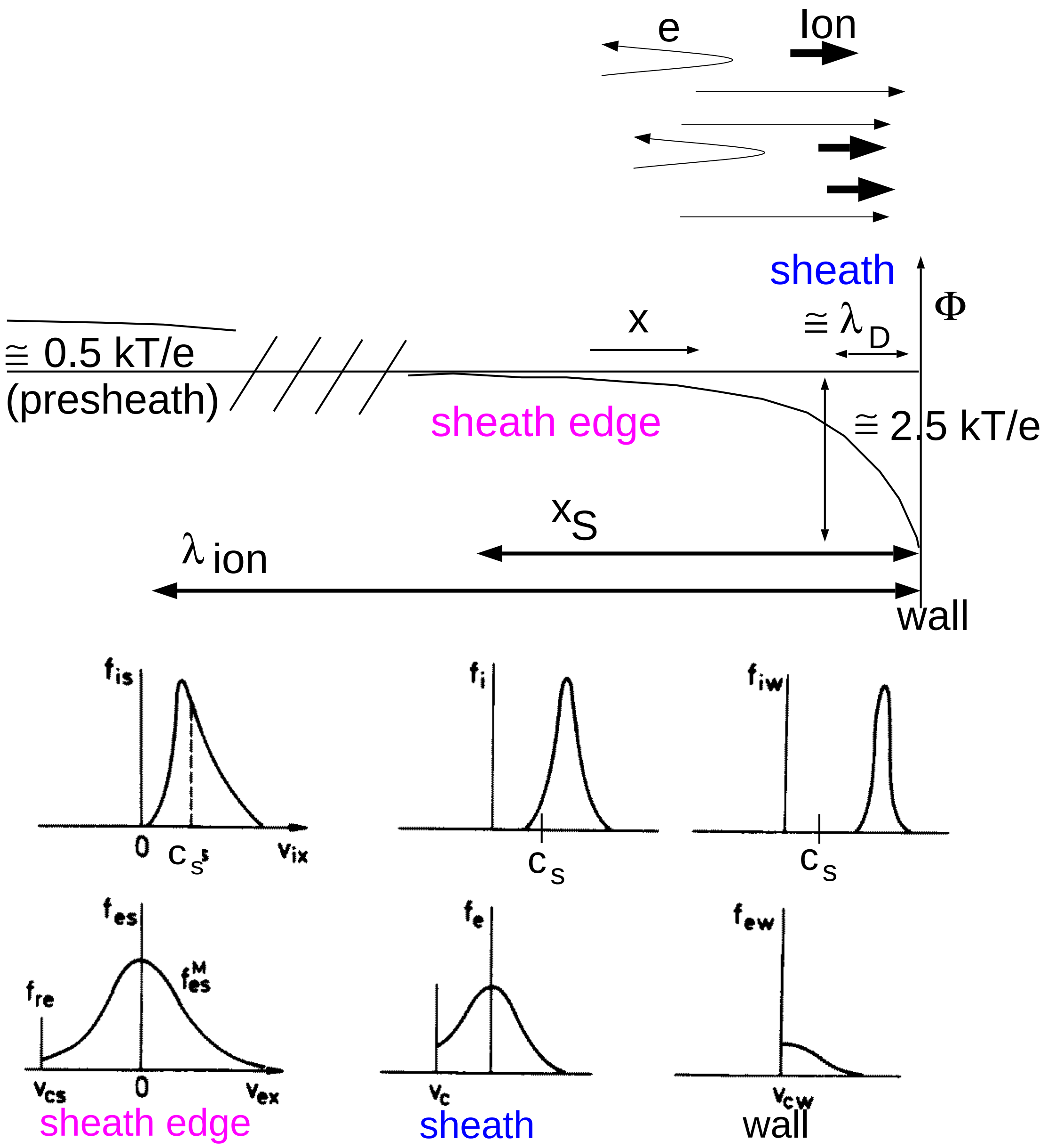

Figure 3.5: *Schematics of the sheath potential in front of the wall without magnetic field (top). The relevant lengths scales are shown: Debye-length* λ_D*, sheath edge position* x_S *and ionization length for neutrals* λ_{ion}*. At the bottom, the velocity distribution functions for ions* f_i *and electrons* f_e *are shown at the sheath edge, in the sheath, and at the wall. The electrons have a cut-off Maxwellian (with a cut-off velocity* v_cs *at the sheath edge). At the wall no electrons are going back (half Maxwellian). At the sheath edge the ions already have a non-Maxwellian distribution function as they are accelerated to* c_S *or larger because of the Bohm criterion. (Figure from R. Chodura)*

The general edge problem is much more complex because of the interaction with neutrals and impurities and other phenomena which have to be included [23]. However these effects are beyond the scope of the present work. Plasma physics [24] combines many disciplines:

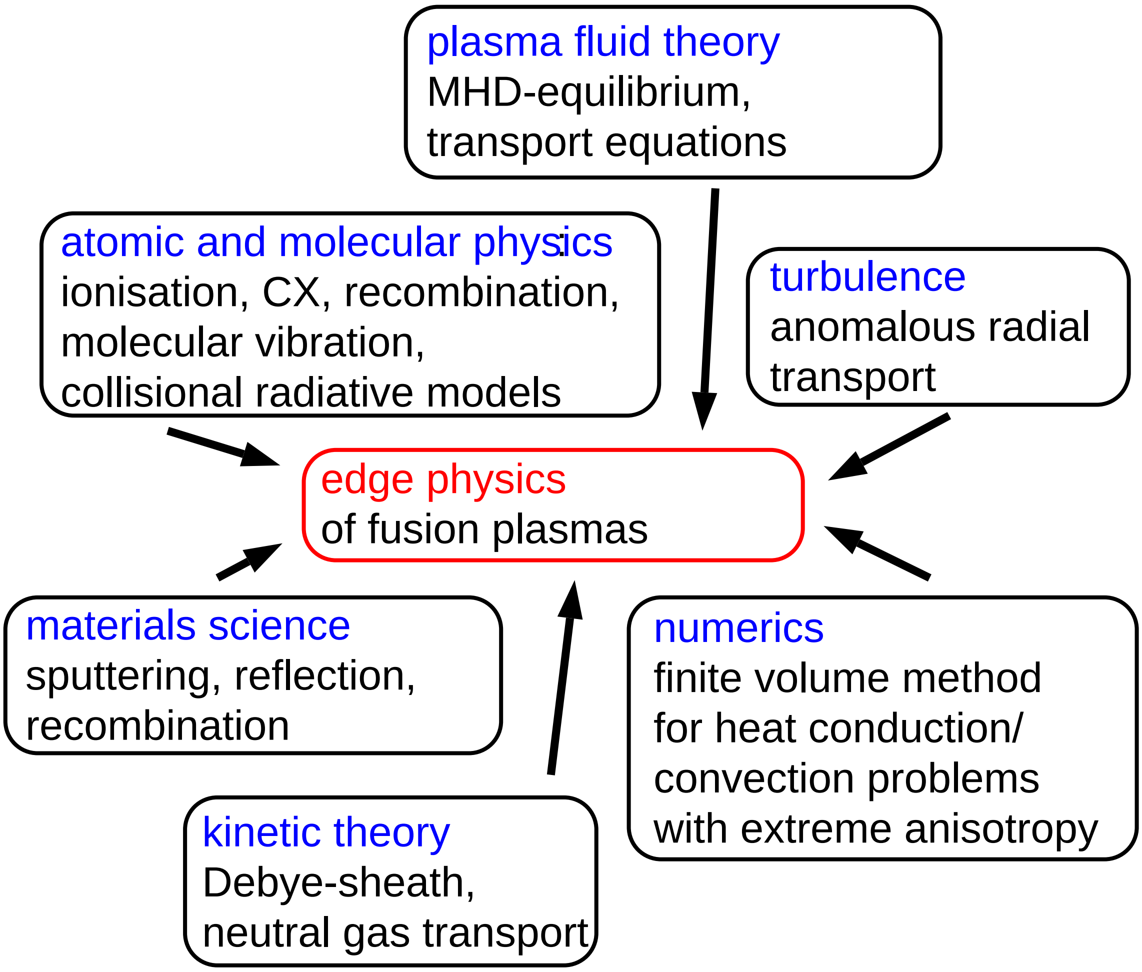
plasma fluid theory
MHD-equilibrium, transport equations
atomic and molecular physics
ionisation, CX, recombination, molecular vibration, collisional radiative models
turbulence
anomalous radial transport
edge physics
of fusion plasmas
materials science
sputtering, reflection, recombination
numerics
finite volume method for heat conduction/ convection problems with extreme anisotropy
kinetic theory
Debye-sheath, neutral gas transport

Figure 3.6: *Summary of edge physics.*

1. ***Magnetohydrodynamic equilibrium studies*** are a necessary prerequisite in edge physics. From plasma fluid theory we obtain the transport equations which govern edge transport.

2. ***Kinetic theory*** describes the sheath forming in front of a plasma-facing structure. This defines the boundary conditions for fluid model transport codes. The classical transport of neutrals strongly determines the operational parameters in divertor plates.

3. ***Turbulence and anomalous radial transport.*** Due to these effects, scaling laws or direct coupling to turbulence codes is necessary for good modeling. Drift-wave turbulence in full geometry has the potential for a full predictive capability

for the edge transport. Anomalous transport is still the largest uncertainty in edge physics.

4. ***Atomic and molecular processes*** strongly affect the plasma through ionization, charge exchange and recombination. Therefore we use collisional-radiative models. These may also account for molecular vibration.
5. ***Materials science*** and plasma wall interaction involve the study of sputtering, reflection and recombination.
6. ***Numerics.*** Transport codes for heat conduction/convection problems may use various numerical methods, but they must allow for physical characteristics such as strong anisotropy, ergodicity and complex geometry.

3.3 Ergodicity and transport in the plasma edge

Further complication of the problem of energy transport in the plasma edge in W7-X [25] is ergodicity which is a spatially chaotic structure of field line trajectories [26]. In W7-X this ergodicity may be created by additional control coils in the divertor plates or by intrinsic plasma effects (finite beta).

The plasma core is a non-ergodic region in which the field lines form a tightly packed series of nested flux surfaces, and there is an ordered temperature gradient across the surfaces. In the plasma edge some field lines are ergodic which means that they fill out volumes, and neighboring field lines diverge from each other exponentially, on average. The Kolmogorov length, defined in Fig.3.7, is a measure of field line ergodicity. Note that in the SOL, the open field lines have varying lengths. Those that are shorter than the Kolmogorov length do not exhibit ergodicity, and their structure is laminar.

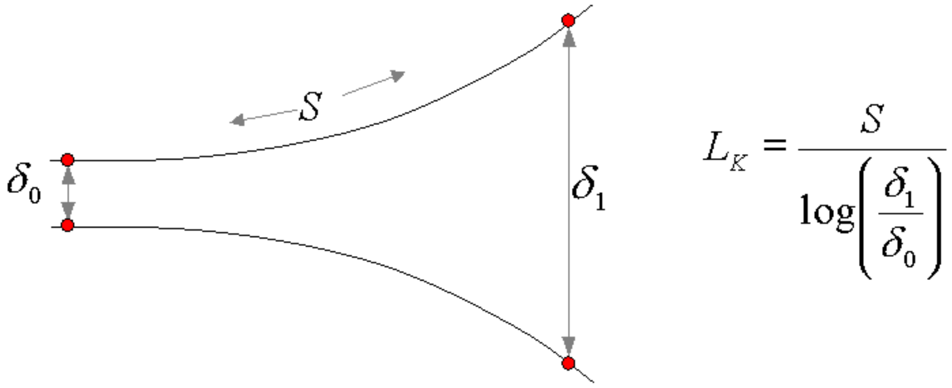
S
δ_0
δ_1
$L_K = \frac{S}{\log\left(\frac{\delta_1}{\delta_0}\right)}$

Figure 3.7: *In the edge region, consider two neighboring field lines initially separated by* δ_0*, after following them for a distance* S*, they are separated by* δ_1*. The Kolmogorov length is proportional to the ratio of distances* δ_1 *and* δ_0*.*

In terms of field line trajectories, ergodicity means that as a field line travels around a torus it undergoes excursions in the radial direction. This results in a mixing of the strong parallel transport with radial transport. Thus, the radial transport is enhanced and there is a flattening of the temperature profile across the edge region. Fig.3.8 summarizes the characteristics of the edge region in a cross-section of TEXTOR-DED [27], which is a tokamak with a circular plasma cross-section.

Rechester-Rosenbluth [28] give an analytical estimate of energy transport in ergodic regions in which the radial diffusion is proportional to field line diffusivity and parallel diffusion. Based on the field line diffusion coefficient, D_{fl}, one can calculate the diffusion of each population of free streaming particles along the field lines [29]. The diffusion is then simply $D_{fl}v_{th}$ where v_{th} is the typical velocity of each species, namely the thermal velocity. The ratio of electron to ion transport is therefore $\sqrt{m_i/m_e}$. Using this order of magnitude, one readily expects a large plasma response for 'electronic' fields, such as the electron temperature, the toroidal component or the radial electric field, and a relatively small direct response for the 'ionic' fields, ion temperature, density and plasma momentum.

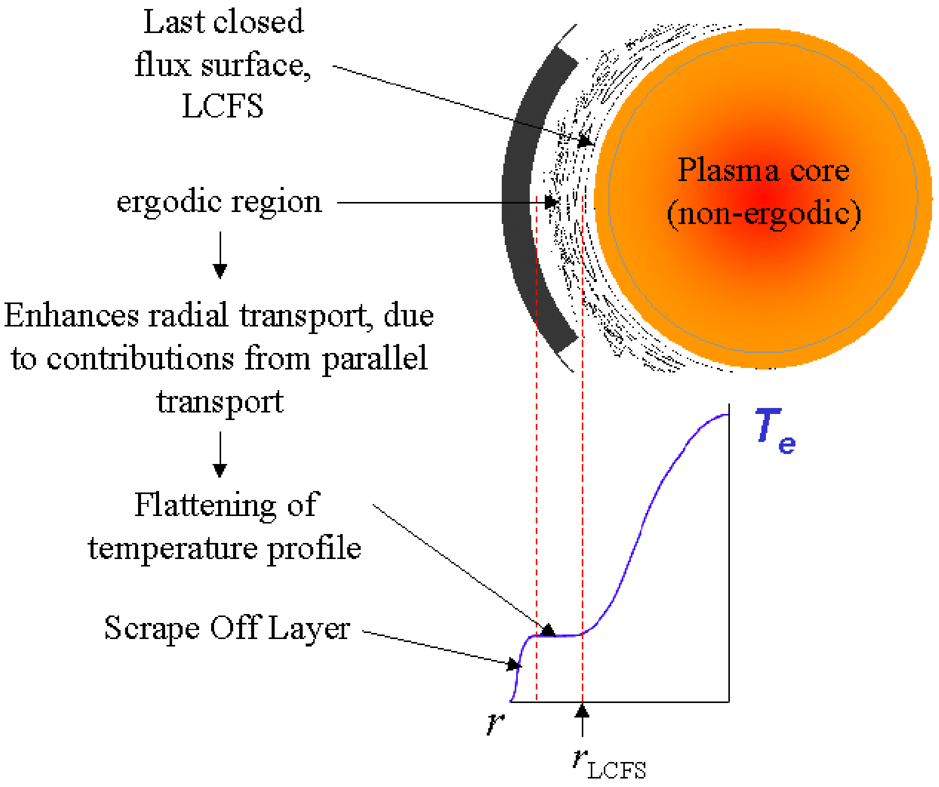
Last closed flux surface, LCFS
ergodic region
Enhances radial transport, due to contributions from parallel transport
Flattening of temperature profile
Scrape Off Layer
Plasma core (non-ergodic)
T_e
r
r_{LCFS}

Figure 3.8: *Effect of ergodicity on transport in the edge region.*

Ergodic layers can also help to overcome the power loading problem at the walls and divertor plates. If we introduce a small amount of Argon into the reaction chamber, say, a few percent of the plasma volume, it forms a radiating mantle around the plasma core. This mantle has a radiating maximum at ∼40 eV which is roughly the temperature of the plasma edge in W7-X under normal operating conditions. By creating an ergodic region close to the maximum radiating efficiency of Argon, we can enhance the radiation from the plasma edge. This occurs because, in an ergodic region, the radial temperature profile is flattened and the radiation is spread out over a larger area, which translates to a large volume in a 3D device. In this way we can reduce the power load at the divertor plates. The ergodic region surrounding the core allows us to maximize the radiation losses with a minimum of impurities.

4 State of the art

In recent years a number of codes have been developed for the numerical solution of the plasma fluid transport equations in tokamak and stellarator 3D configurations. The motivation is the need for tools which can handle strong anisotropy in complex 3D topologies including flux surfaces, islands, ergodic regions and complicated surfaces of plasma-surface interaction. Presently, the most advanced codes in this field of computational physics are the BoRiS [30], EMC3-EIRENE [31, 32], and E3D [33, 34] codes

BoRiS is a 3D Scrape-Off-Layer transport code for solving a system of coupled partial differential equations using a finite volume method. It has been developed for 3D edge modeling in W7-X but is also applicable to other devices without ergodicity. The code is characterized by interpolation for mixed convection-diffusion, generalized magnetic coordinates and the Newton method. The development of BoRiS was influenced by experience with 2D codes like B2-Eirene [35] and UEDGE [36, 37]. Fig.4.1 shows the system of sub-grids and coordinates used in the BoRiS code.

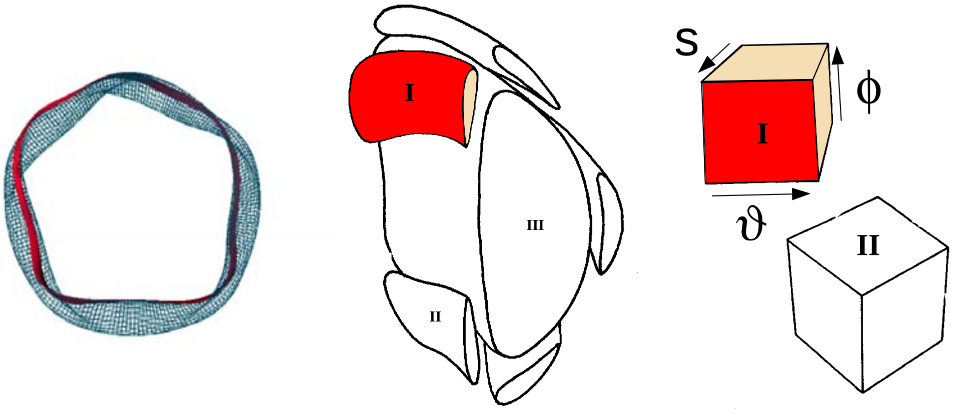

Figure 4.1: *Seven sub-grids with individual magnetic coordinates correspond to five island flux tubes, the plasma core and the outer Scrape-Off-Layer, forming a complete W7-X grid..*

The EMC3-EIRENE code uses a Monte-Carlo technique in real space using a field-aligned local orthogonal vector basis which reduces the diffusion tensor to a diagonal form. The parallel and perpendicular transport are separated by integrating the parallel transport along the field lines. Presently the code is being applied to W7-AS, W7-X and TEXTOR-DED.

The E3D code uses a multiple coordinate system approach (MCSA) with local magnetic coordinates and the appropriate full metric tensor. The local or piecewise treatment of field lines allowed in the Monte Carlo method makes it applicable for modeling in ergodic regions. Also, the grid used in the Monte-Carlo approach can be refined at point resolution. Disadvantages of the Monte Carlo approach are the computational expense and the noise in the solutions which make it difficult to identify a steady state. E3D was developed for modeling in TEXTOR-DED[38] but has also been applied to W7-X. Work has also been done on coupled fluid and Monte-Carlo models, [39]

Rechester-Rosenbluth [28] give an analytical estimate of energy transport in ergodic regions in which the radial diffusion is proportional to field line diffusivity and parallel diffusion. Based on the field line diffusion coefficient, D_{fl}, several estimates improving this original work and discriminating different ergodic regimes are existing (see [29]).

A general finite volume method requires a continuum of the computational grid which is only available for certain magnetic configurations with closed flux surfaces. By comparison, using Monte Carlo methods, transport in ergodic regions can be treated locally which allows greater flexibility in the mesh construction. Regardless of the numerical method used, the strong anisotropy of the transport requires a complete separation of parallel and radial transport terms. This may be achieved by a clear identification of the parallel direction such as the magnetic coordinates used in BoRiS, the local magnetic coordinates used in E3D, and the finite flux tube coordinates used in EMC3.

A finite difference code is an attractive addition to these codes because it has the possibility to overcome the disadvantages of the finite volume and Monte Carlo methods. That is, modeling in ergodic regions, which is problematic in the finite volume method, and clearly identifiable steady state solutions obtained with less computation effort than in the Monte Carlo approach. The price of the finite difference code is the complexity of the grid generation process (also, the grid can be refined only at field line resolution, not at point resolution), and the inability to directly confirm flux conservation.

5 Mesh generation

The first stage in the modeling process is to generate a mesh. This is a 3-dimensional array of points representing the plasma. Due to the extreme anisotropy the field line start points must be pre-selected to ensure a correct solution (minimizing numerical diffusion as discussed later). Fig.5.1 shows the general shape of the W7-X plasma, and the target plates which intersect the plasma edge.

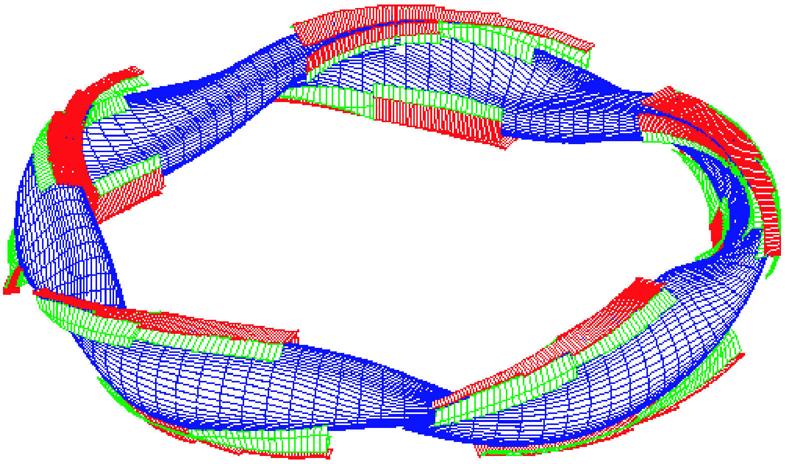

Figure 5.1: *The W7-X plasma core (in blue). The red and green structures are target plates. In W7-X the magnetic field has a 5-fold symmetry which means that the plasma cross-section repeats itself every* $72°$.

Starting with a specified magnetic field configuration (stored in a data file) a field line tracing code is used to trace field lines around the torus of the experiment. Mesh points are generated at specified toroidal angles along these field lines (eg every $9°$) and stored with their x, y, z coordinates. By following many different field lines, many times around the torus, many points are generated to fill the volume of the plasma. The transport equation is solved on this computational domain in order to model the physics in the plasma. See Fig.5.2

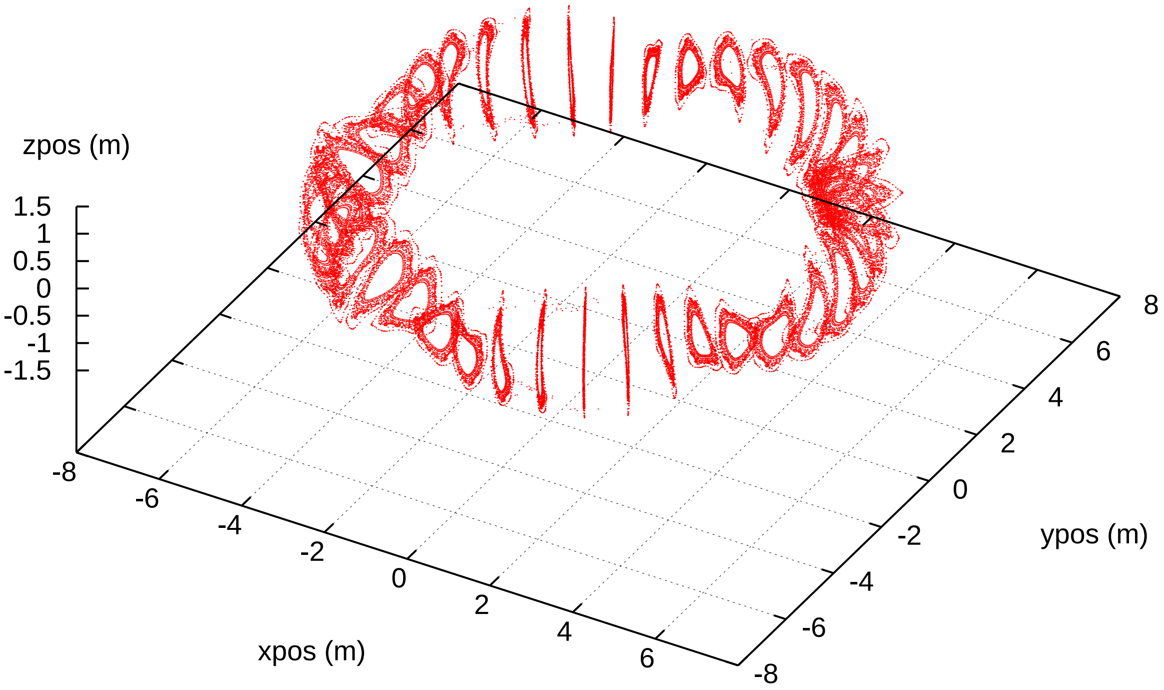

Figure 5.2: *A 40 cut mesh generated in a W7-X geometry. Total points* ~ 51000.

The field line tracing code uses a 4th order Runge-Kutta integration on the equation for a field line:

$$\frac{dx}{dB_x} = \frac{dy}{dB_y} = \frac{dz}{dB_z} \tag{5.1}$$

where x, y, z are global Cartesian coordinates with the origin in the center of the torus, and B_i are magnetic field components. The points are generated at regular toroidal angles, and therefore the mesh exists as a series of cross-sectional planes, called Poincaré plots, each with many points. A Poincaré plot is a 2D scatter plot lying on a plane of constant toroidal angle. The points on the scatter plot are the intersections of field lines with that plane. If the points are generated at 9° intervals, as in Fig.5.2, then the mesh is a set of 40 Poincaré plots or cuts.

In order to have good quality modeling of the physics in the plasma, the mesh must have certain characteristics. First, The distance between Poincaré plots must be less than the Kolmogorov length, L_K so that, in the edge region, the local coordinate system at each point is too small to 'see' the local field line ergodicity. In W7-X, with a 40 cut mesh the toroidal point separation distance is $\sim 1m$, while L_K is typically $10 \rightarrow 30m$. Second, the number of cuts also determines the toroidal resolution of the metric coefficients along the field lines. This must be of sufficient quality. Fig.5.3 shows $\sqrt{g}$ plotted along part of a field line through 360°, at different toroidal resolutions.

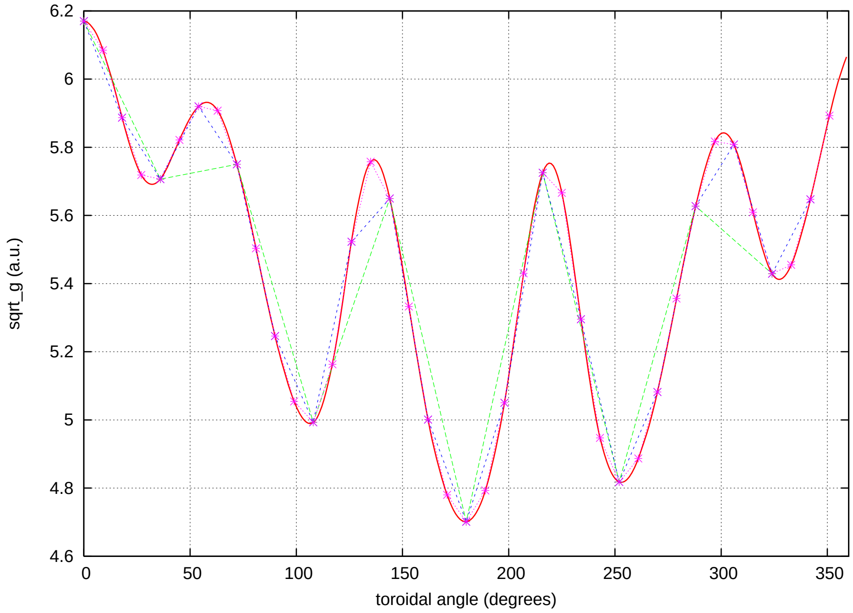

Figure 5.3: *Plot of* $\sqrt{g}$ *along a field line for 1 pass around the torus of W7-X, with different toroidal resolutions. The red curve represents a high resolution mesh with 360 cuts. The 5 peaks indicate the 5-fold symmetry in the W7-X plasma. The green curve represents a low resolution mesh with 10 cuts. Meshes with 20 and 40 cuts, represented by the blue and pink lines, give intermediate resolutions of the metric coefficient along the field line.*

A mesh with 360 cuts, ie a Poincaré plot every 1°, gives excellent resolution of the metric coefficients but is too large to be practical, that is, such a mesh would have so many points that the transport code would take a long time to obtain a solution. At the other extreme, a solution can be obtained very quickly on a mesh having only 10 cuts, ie a Poincaré plot every 36°, but then the metric coefficient is poorly resolved, resulting in a poor quality solution. Meshes with 20 or 40 cuts give a satisfactory resolution of the metric coefficients without having too many mesh points.

Finally, the poloidal resolution (the density and distribution of points on each Poincaré plot) must also be of sufficient quality. This is to ensure the resolution of details in the temperature solution. The mesh quality can be refined by visual inspection of the Poincaré plots and then adding or subtracting field lines from the mesh.

The Poincaré plot shown in Fig.5.4 shows a 'standard' poloidal density used in meshes in the present work. It shows the magnetic field structure in the plasma and the different types of field lines which make up the mesh. These include closed field lines in the core region and open field lines in the edge region.

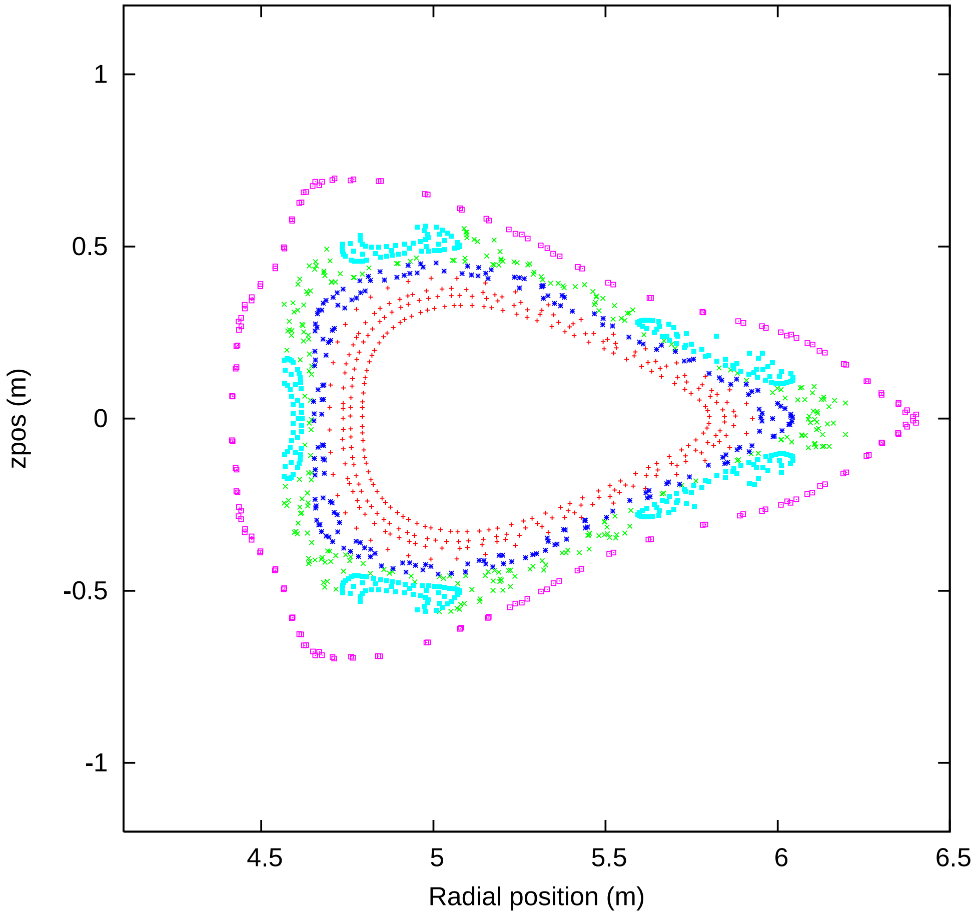

Figure 5.4: *Poincaré plot at* $\phi = 36°$ *from a W7-X mesh. Red points lie on field lines which form closed flux surfaces in the plasma core. The core is not filled to the center because we are mainly interested in modeling the plasma edge. Dark blue points lie on field lines which form closed ergodic field lines. These form a thin ergodic layer around the plasma core. Green points lie on open field lines in the edge region. Light blue points lie on field lines which form nested flux surfaces in the islands. The islands are intersected by the divertor plates, so these flux surfaces are 'open'. Pink points lie on a closed field line which approximates the vacuum vessel wall. They serve as an outer boundary for applying poloidal boundary conditions. This plot has* ~ 1270 *points.*

5.1 Tracing closed field lines

In the 'core region' of the plasma there are closed non-ergodic field lines which form a tightly packed series of nested flux surfaces in the plasma core. There are also closed ergodic field lines which form a thin ergodic layer surrounding the plasma core. For closed field lines the mesh is optimized to minimize numerical diffusion. This is done by selecting field lines which satisfy a 'closed' criterion: A field line must travel a large number of turns, N, around the torus and close on itself to within some small distance, Δ. This criterion can be written as:

$$\Delta \quad \ll \quad L_{\parallel}\sqrt{\frac{\chi_{\perp}}{\chi_{\parallel}}} \tag{5.2}$$

where:
Δ : *distance between field line start and end points on the starting cut.*
$L_{\parallel}$: *length along the field line from start to end points,* $L_{\parallel} \approx 2\pi RN$
R : *plasma major radius,* $R \approx 5.5m$ *in W7-X*
N : *number of toroidal turns made by a field line, eg* $N = 100$
$\chi_{\perp}$: *radial diffusivity,* $\chi_{\perp} \approx 1m^2s^{-1}$
$\chi_{\parallel}$: *parallel diffusivity,* $\chi_{\parallel}/\chi_{\perp} \gtrsim 10^6$

Fig.5.5 shows how this criterion is applied.

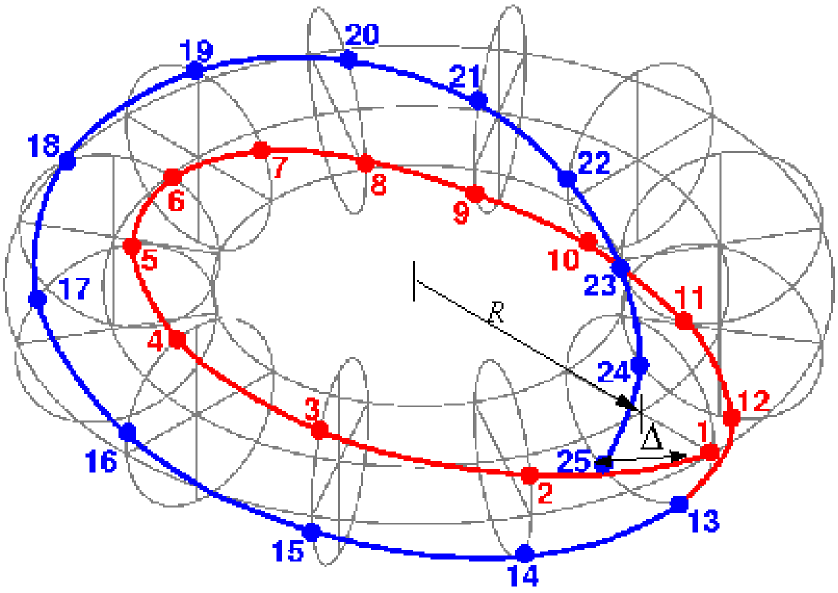

Figure 5.5: *Simulated field line making $N = 2$ passes on a simple toroidal geometry. The closing distance, Δ is measured on the plane of the starting cut.*

For the purposes of the present work, a suitable maximum closing distance for a field line is one electron gyro-radius; $\Delta \approx 1mm$. With $N \approx 100$ the numerical diffusion is of the order of $10^{-3}m^2s^{-1}$, ie much less than the physical diffusion $\chi_\perp$. The physics argument is that for $\Delta < 1mm$ we should not expect physics differences. Fig.5.6 shows a single closed field line traced in the core of W7-X.

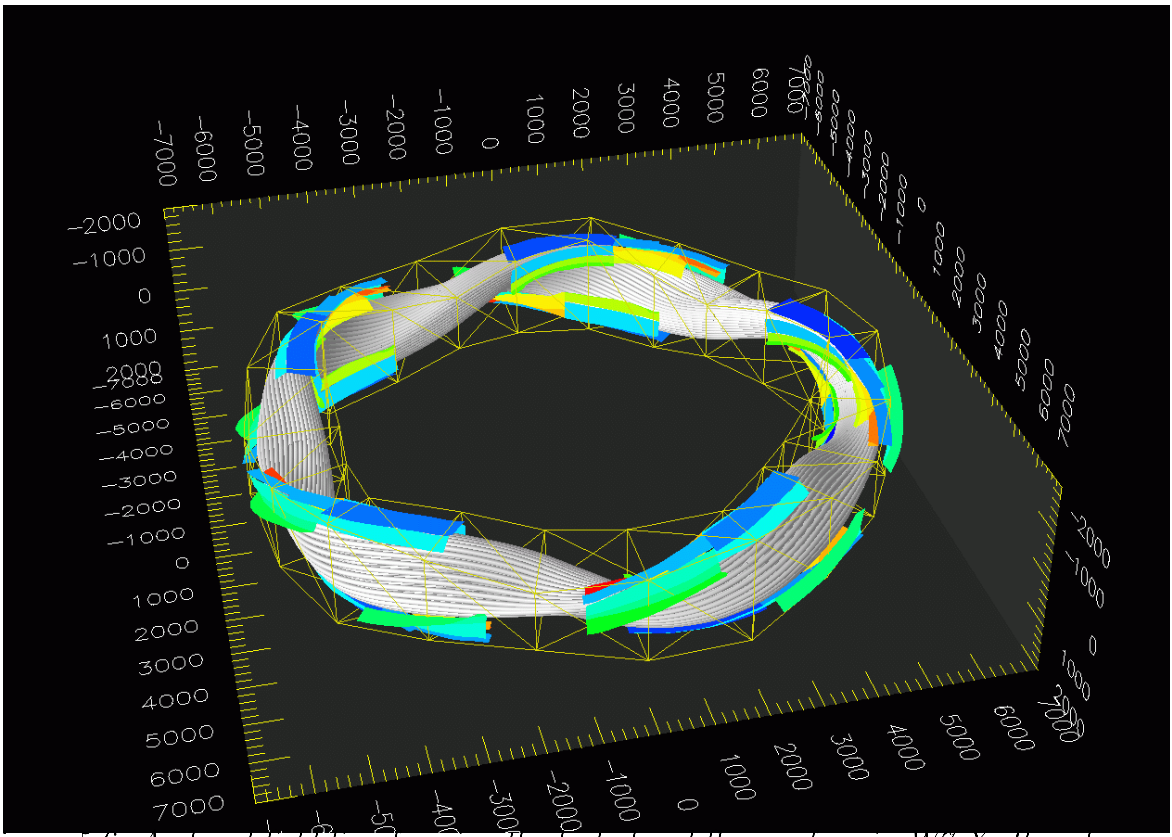

Figure 5.6: *A closed field line forming the last closed flux surface in W7-X. It makes only 43 passes around the torus but the closing distance* Δ *is much less than 1mm, so the resulting numerical diffusion is still sufficiently minimized. The colored surfaces are the target plates.*

We cannot control the exact number of toroidal passes made by a field line before it satisfies the criterion. Instead, we trace many field lines, storing the corresponding values of Δ and N, and then we select the best field lines which are available. These field lines are then used in the construction of the mesh. Each field line has a different number of points but this is not a problem if the variation in the number of points from one flux surface to the next is not large.

Note that we are guaranteed to find suitable field lines in the non-ergodic region because of the large number of irrational values of the rotational transform, ι (the ratio of toroidal to poloidal turns made by field lines). The closed field line whose points form the outer boundary on each Poincaré plot is also selected using the 'closed' criterion.

5.2 Tracing open field lines

In the 'edge region' of the plasma there are many open field lines of different lengths, usually much shorter than the closed field lines in the core region. Essentially, an open field line is traced from a target plate, around the torus until it hits another target plate. No optimization criterion is applied. Some open field lines form nested flux surface structures in the islands. Others lie in the space between the islands and the core region. Fig.5.7 shows an open field line traced in the edge region W7-X.

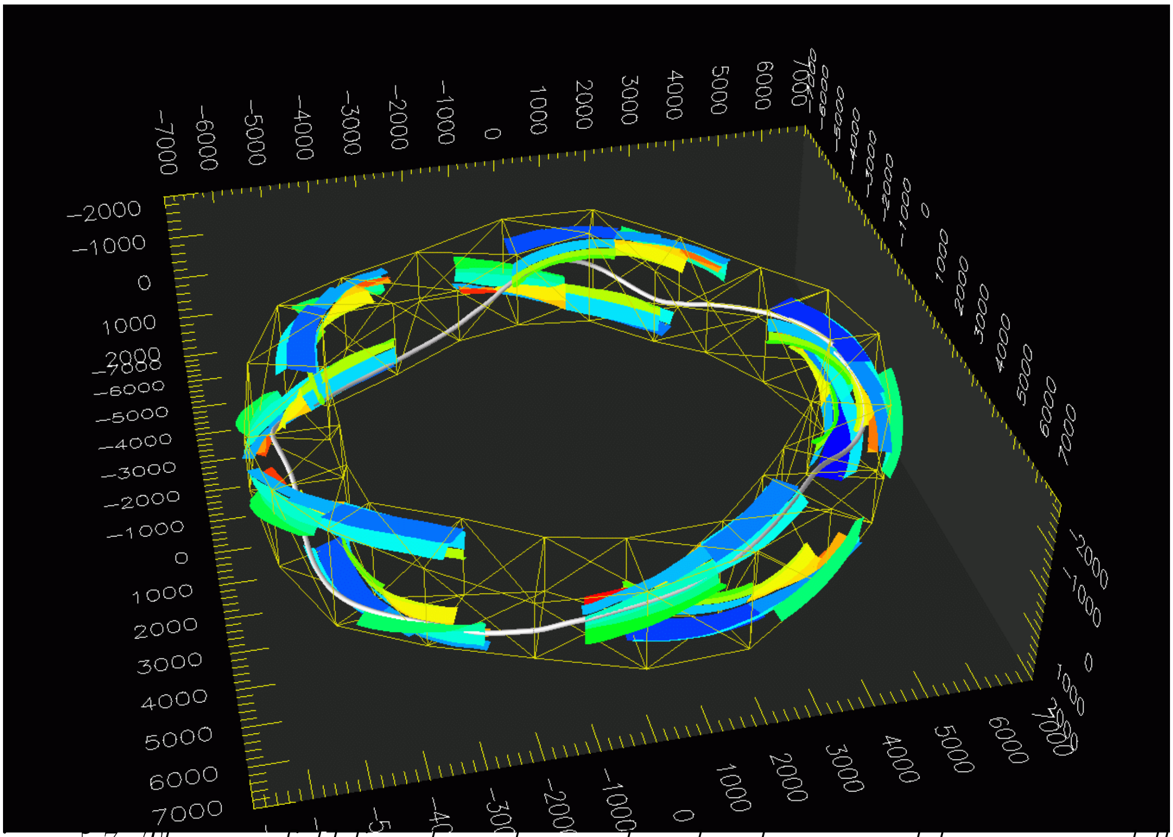

Figure 5.7: *The open field line shown here makes almost one complete pass around the torus. The start and end points lie on the top target plate (dark blue) at the right hand side. In this case, the field line comes close to closing on itself.*

The edge region must be filled with points with a suitable uniformity and density. If the point distribution on a Poincaré plot has poor uniformity, or the density is too low, then there is poor resolution of the radial temperature profile. If the density is too high then the system of transport equations is correspondingly large and the transport code takes longer to reach a solution.

The actual proces of tracing an open field line is as follows. To ensure that the edge region is filled more or less uniformly we specify 'floating' start points in the edge region on a cut, between the core and the islands. We then trace field lines from these start points in the clockwise and counterclockwise directions until they intersect a target plate. Mesh points are generated at regular toroidal intervals. Thus, for each 'floating' start point there are 2 field line parts. These are concatenated and the mesh points are renumbered in order to form a complete open field line which extends around the torus from one target plate to another target plate. More open field lines are traced from similar start points on other cuts around the torus, until the edge region is filled with points.

The start and end points of open field lines are also stored as valid mesh points and they lie on the target plates. However, open field lines never intersect a target plate exactly on a Poincaré plot. In practice, the start and end points of open field lines always lie in between the plots. These points can be seen by close inspection of the mesh in Fig.5.2. In W7-X meshes the number of such points is typically $1 \rightarrow 2$ % of the total mesh points.

When enough closed and open field lines have been traced, they are collected into a mesh data file. This is a list of x, y, z coordinates for each mesh point. Additional information is also stored for each point such as Poincaré plot number, field line number, flux surface number, point type (ie boundary or inner) etc. This is used for optimizing the triangulation process (discussed in chapter 8), and for easier access to the mesh points in the transport code.

5.3 The stencil

A stencil is a diagram showing, in generalized form, the spatial arrangement of neighboring points for the discretization of a transport equation around a typical 'inner' mesh point. An inner mesh point is any point which does not lie on the innermost flux surface of the plasma core, or on the outer boundary, or at the start or end of an open field line. A stencil represents a characteristic part of a mesh, on which the transport equation is discretized. A stencil for the 'standard' W7-X mesh used in the present work is shown in Fig.5.8.

The stencil shows 3 characteristic properties of the mesh. First, the mesh is ***unstructured***. This means that the total number of neighbors for each mesh point is not a constant because the number of poloidal neighbors varies. Second, the mesh is ***non-rectilinear***. This means that the mesh points do not lie on mutually orthogonal axes. The field lines never intersect the Poincaré plots at exactly 90° and the points on each Poincaré plot lie scattered in a non-rectilinear pattern. Third, the mesh is ***Non-uniform*** because the spacing between points is not a constant, both along the field line and on the plane of each Poincaré plot. These 3 characteristics make the discretization of the trans-

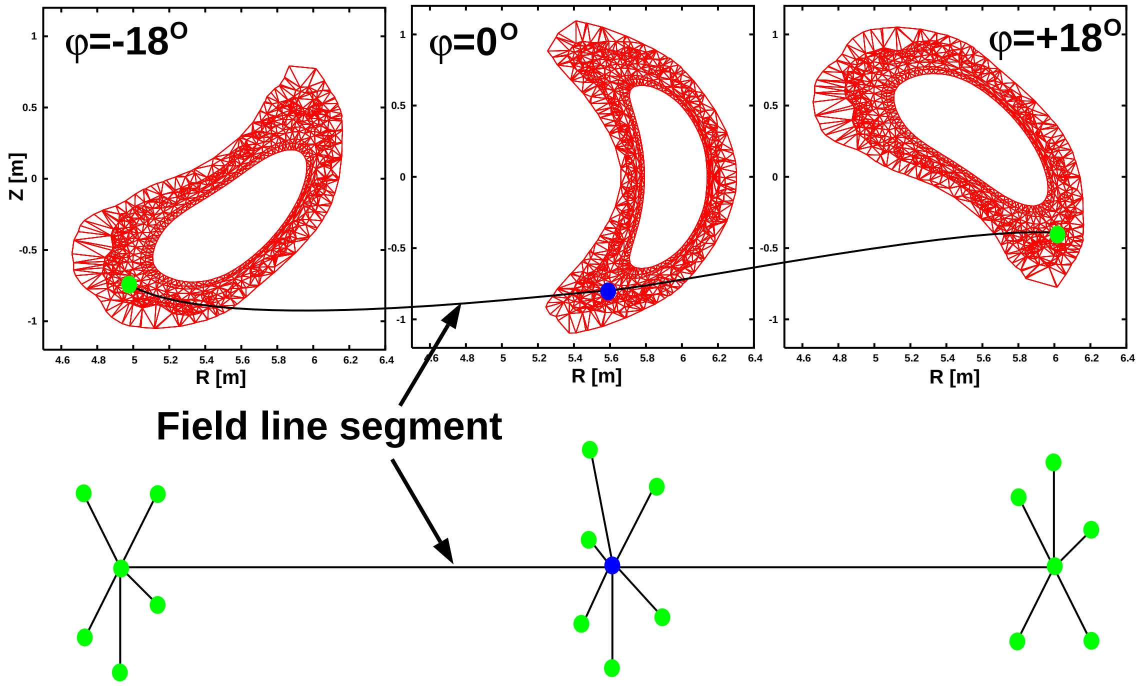

Figure 5.8: *Mesh stencil for a W7-X mesh. Consider a field line intersecting 3 successive Poincaré plots (the plots shown here have been triangulated). For a typical inner mesh point, shown in blue, the number of 'toroidal' neighbors is exactly 2. These lie on the same field line but on the neighboring Poincar'e plots, forwards and backwards. This mesh point also has a number of 'poloidal' neighbors lying scattered around it on the same Poincaré plot. The exact number of poloidal neighbors varies, depending on the local distribution of points, but is usually ∼6. In our discretization scheme we also consider the poloidal neighbors of the toroidal neighbors and so we have a stencil with 21 points on average. The stencil shown here has only 19 points.*

port equation on the mesh much more difficult than on a structured rectilinear uniform mesh.

6 The Transport Equation

The only transport equation considered in this work is the Braginski fluid equation for electron energy, or electron heat conduction [40]. It is a parabolic partial differential equation, that is, it is time dependent but it settles to a steady state. In its simplest form it may be written as a flux divergence equal to a source term:

$$\nabla \cdot Q \quad = \quad S \tag{6.1}$$

Flux, Q, is equal to a transport coefficient times a temperature derivative. The source term, S, may contain sources, sinks, a coupling term, a time derivative and boundary conditions. In 1D, equation 6.1 may be written as:

$$\frac{\partial}{\partial x}\left(-\kappa\frac{\partial T_e}{\partial x}\right) \quad = \quad -\frac{3n_e}{2}\frac{\partial T_e}{\partial t} \tag{6.2}$$

where,
κ : *conduction coefficient*, $m^{-1}s^{-1}$. $\kappa = \chi n_e$
χ : *thermal diffusivity*, m^2s^{-1}
T_e : *electron temperature*, eV
n_e : *electron density*, m^{-3}

6.1 3D Rectangular form of the transport equation.

In a 3D rectilinear mesh the points lie on mutually orthogonal axes and equation (6.2) may be written as:

$$-\frac{\partial}{\partial x_i}\left(\kappa_{\perp_1}\frac{\partial T}{\partial x_1}+\kappa_{\perp_2}\frac{\partial T}{\partial x_2}+\kappa_{\parallel}\frac{\partial T}{\partial x_3}\right) \quad = \quad -\frac{3n}{2}\frac{\partial T}{\partial t} \tag{6.3}$$

The 3 terms on the left hand side are mutually orthogonal flux components. To model an anisotropic system such as a magnetized plasma we set the x_3 term tangential to the field lines. This is the parallel term. The x_1 and x_2 terms are the perpendicular, or radial, terms. The conduction coefficients, $\kappa_{\perp_1}$, $\kappa_{\perp_2}$ and $\kappa_{\parallel}$, are dependent on temperature in the following way:

$$\kappa_{\perp_1} \propto T^{-1/2} \quad , \quad \kappa_{\perp_2} \propto T \quad , \quad \kappa_{\parallel} \propto T^{5/2} \tag{6.4}$$

$\kappa_{\perp_1}$ and $\kappa_{\perp_2}$ are similar in magnitude and both are much smaller than $\kappa_{\parallel}$ so we regard the radial transport (on the plane of each Poincaré plot) as quasi-isotropic due to electrostatic micro-turbulence. The analytical expressions for these conduction coefficients are taken from the plasma formulary [41].

Discretizing equation 6.3 on a rectlinear mesh is straightforward, using standard techniques found in textbooks. The derivatives may be discretized in each direction uniquely, and there is no mixing of derivatives. The metric tensor is simply a 3×3 identity matrix.

The mesh generated for W7-X is non-rectilinear. The points do *not* lie on mutually orthogonal axes and so there *are* mixed derivatives and the metric tensor is 'full'. Discretizing equation 6.3 on such a mesh is more challenging, and so we turn to a vector form of the transport equation.

6.2 Vector form of the transport equation

Below is the heat balance equation for an electron-ion fluid in vector form [40, 42, 33]:

$$\frac{\partial u_\alpha}{\partial t}+\nabla\cdot\left[u_\alpha(\mathbf{V}_{\perp\alpha}+\mathbf{h}V_{\|\alpha})-\frac{2}{3}\chi_{\perp\alpha}\nabla u_\alpha-\frac{2}{3}(\chi_{\|\alpha}-\chi_{\perp\alpha})\mathbf{hh}\cdot\nabla u_\alpha\right]=-\nu_\alpha u_\alpha+S_\alpha^{(u)} \quad (6.5)$$

where,
α : *particle species, e or i*
u_α : *internal energy,* eVm^2s^{-1}
$\mathbf{h}$: *unit contravariant vector quantity,* $\mathbf{h}=\mathbf{B}/\|B\|$
$\mathbf{V}_{\perp\alpha}, V_{\|\alpha}$: *Perpendicular and parallel velocities,* ms^{-1}
$\chi_{\perp\alpha}, \chi_{\|\alpha}$: *Perpendicular and parallel diffusivities,* m^2s^{-1}
ν_α : *heat loss rate, function of Coulomb energy exchange between electrons and ions*
$S_\alpha^{(u)}$: *heat source term*

If we assume no net current in the plasma, then $V_i = V_e$. Neglecting velocity contributions ($\mathbf{V}_{\perp\alpha} = V_{\|\alpha} = 0$), and source terms ($S_\alpha^{(u)} = 0$), and substituting in:

$$u_e=\frac{3n_e}{2}T, \qquad \chi_{\perp e}=\frac{\kappa_{\perp e}}{n_e}, \qquad \chi_{\|e}=\frac{\kappa_{\|e}}{n_e}$$

The heat balance equation (6.5) simplifies to the same form as the flux divergence equation 6.2, except that the transport coefficient contains more information.

$$\nabla\cdot\left[\frac{2}{3}\frac{\kappa_{\perp e}}{n_e}\nabla\left(\frac{3n_e}{2}T\right)+\frac{2}{3}\left(\frac{\kappa_{\|e}}{n_e}-\frac{\kappa_{\perp e}}{n_e}\right)\mathbf{hh}\cdot\nabla\left(\frac{3n_e}{2}T\right)\right]=-\frac{\partial}{\partial t}\left(\frac{3n_eT}{2}\right)$$

$$\Rightarrow \qquad \nabla\cdot\left[\{\kappa_\perp+(\kappa_\|-\kappa_\perp)\mathbf{h^2}\}\cdot\nabla T\right]=-\frac{3n_e}{2}\frac{\partial T}{\partial t} \quad (6.6)$$

7 Local Magnetic Coordinate Systems

In the physical system we are modeling, the ergodicity determines the choice of coordinates, and these in turn determine the structure of the mesh. To handle the ergodicity we use local magnetic coordinate systems [33]. This means that each mesh point has a unique curvilinear coordinate system, x_i, which is valid locally and in which, the x_3 coordinate is aligned with the magnetic field. Fig.7.1 shows 3 consecutive cuts in a simplified toroidal geometry.

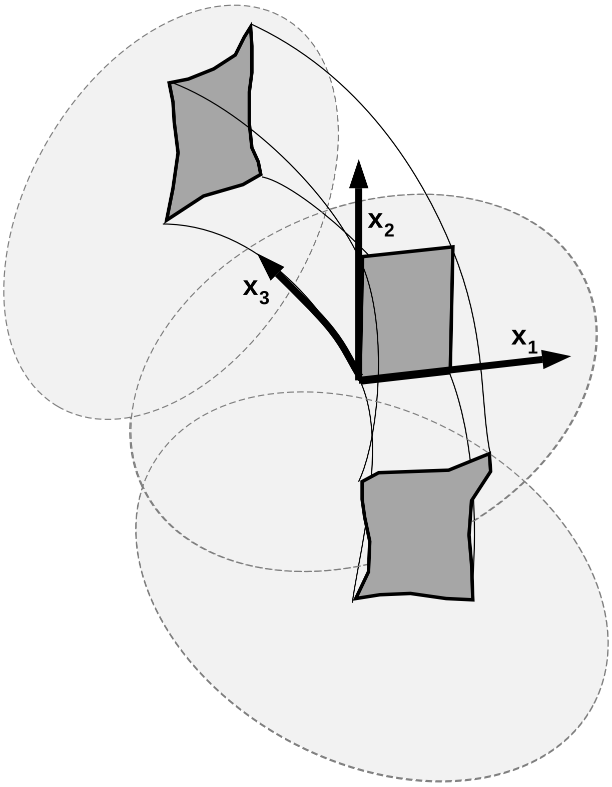

Figure 7.1: *A local coordinate system aligned with the magnetic field in a section of a simplified toroidal geometry. Note that when the polygonal area on the central cut is mapped to the 2 neighboring cuts its shape is deformed by the field line trajectories but the area is conserved.*

On the central cut is a polygonal area defined by mutually orthogonally poloidal coordinates, x_1 and x_2. The vertices are defined by magnetic field lines. When the area is mapped to the neighboring cuts, forward and backward, it becomes deformed due to the field line trajectories. This happens in ergodic and non-ergodic regions and causes a mixing of the poloidal coordinates.

The third coordinate, x_3 is toroidal and is tangential to the magnetic field lines. These never intersect the cuts orthogonally, so there is a mixing of the toroidal coordinate with the poloidal coordinates. The overall result is a complete mixing of all coordinates and a full metric tensor to describe the coordinate system, that is, all the tensor elements are non-zero.

The backward and forward cuts are separated by a toroidal angle such that a field line segment extending across the 3 cuts is much shorter than the local Kolmogorov length. In this way, the coordinate system determines the structure of the mesh because it limits the toroidal separation between the cuts.

A coordinate system which is shorter then the Kolmogorov length is unable to 'see' ergodicity, that is, ergodic effects are not transmitted from one end of the coordinate system to the other. The advantage is that we are not constrained by ergodicity in our discretization of the domain. In this way, the ergodicity determines the choice of coordinates. With a global coordinate system we would have to follow flux tubes around the torus in order to guarantee flux conservation. This would be impossible because the ergodicity and shear of the field line trajectories would deform the volumes too much.

By aligning the local coordinate system with the magnetic field we can discretize the transport equation along the field lines. This allows us to evaluate the parallel transport to high accuracy and separate it from the radial transport. In this way we minimize the numerical diffusion which results from the strong anisotropy.

7.1 The Metric Tensor

The metric tensor, g^{ij}, is a geometric quantity describing the curvature of the mesh at each mesh point. It is a set of coefficients associated with a transformation from one point to another. It is a second rank tensor (2 indices, 2 points), and for a 3-dimensional system each index has 3 possible values. The metric tensor has the form:

$$g^{ij} \quad = \quad \begin{bmatrix} g^{11} & g^{12} & g^{13} \\ g^{21} & g^{22} & g^{23} \\ g^{31} & g^{32} & g^{33} \end{bmatrix} \qquad \begin{array}{ll} i = j & \text{unmixed terms} \\ i \neq j & \text{mixed terms} \end{array} \tag{7.1}$$

The diagonal elements, $i = j$, represent a simple transformation in one coordinate. The off-diagonal terms, $i \neq j$, represent non-scalar transformations. When the metric tensor is included in the transport equation it makes possible the discretization of the equation on an unstructured, non-rectilinear mesh.

7.1.1 Coordinate surfaces and coordinate curves

In toroidal geometry we can imagine 3 families of coordinate surfaces; a set of poloidal cross-sections, a set of nested toroidal surfaces, and a set of radial surfaces. Within this geometry, any position i may be specified with 3 parameters, ϕ, r, θ. A coordinate surface may be obtained by holding one parameter constant and varying the other two. For example, a Poincaré plot lies on a plane (a poloidal cross-section) on which r and θ vary, but ϕ is held constant. Therefore, a set of discrete values of ϕ defines a unique family of coordinate surfaces. Any position, i, may be defined as the intersection of 3 coordinate surfaces, one from each family.

A coordinate curve is the intersection line between any two coordinate surfaces. Therefore, there are 3 families of coordinate curves. A coordinate curve is obtained by holding any two parameters constant and varying the remaining parameter. For example, in the case of a coordinate system aligned with the magnetic field, a field line may be thought of as coordinate curve along which r and θ are held constant and ϕ is constantly changing.

The W7-X plasma has a complicated toroidal geometry. In a W7-X mesh the Poincaré plots form one family of coordinate surfaces. On each plot, ϕ is some fixed unique constant. Each Poincaré plot has non-ergodic regions (core and islands) where the field lines form flux surfaces. These are another family of coordinate surfaces in which, for each flux surface, r varies within a finite range of values. This family is undefined in the ergodic region where there are no flux surfaces. Throughout the mesh the points do not lie on fixed radial axes, so the family of radial coordinate surfaces, each with constant θ is completely undefined. However, in any region of the plasma we can still use 3 parameters to describe the position of each mesh point. Therefore we can still obtain coordinate curves.

7.1.2 Basis Vectors

Consider a point i in the plasma, determined by the position vector $\mathbf{R}$. The *tangent basis* at i is the set of vectors, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, pointing along the coordinate curves at p. If u^1, u^2 and u^3 are generalized curvilinear coordinates for the system, the tangent basis vectors, or *covariant basis vectors* are:

$$\mathbf{e}_1 = \frac{\partial \mathbf{R}}{\partial u^1}, \qquad \mathbf{e}_2 = \frac{\partial \mathbf{R}}{\partial u^2}, \qquad \mathbf{e}_3 = \frac{\partial \mathbf{R}}{\partial u^3}, \qquad \Rightarrow \qquad \mathbf{e}_i = \partial \mathbf{R}/\partial u^i$$

This definition is consistent with the requirement that a set of basis vectors must be linearly independent, that is, the triple product $\mathbf{e}_1 \cdot \mathbf{e}_2 \times \mathbf{e}_3 \neq 0$.

Similarly, the *reciprocal basis* at p is the set of unit vectors, $\mathbf{e}^1$, $\mathbf{e}^2$, $\mathbf{e}^3$, which are perpendicular to the coordinate curves at p. The reciprocal basis vectors, or *contravariant basis vectors* are:

$$\mathbf{e}^1 = \nabla u^1, \qquad \mathbf{e}^2 = \nabla u^2, \qquad \mathbf{e}^3 = \nabla u^3, \qquad \Rightarrow \quad \mathbf{e}^j = \nabla u^j$$

$\mathbf{e}_i$ and $\mathbf{e}^j$ are reciprocal sets. Any vector of one set can be calculated if all three vectors of the other set are known. Any vector, $\mathbf{R}$, can be written as a linear combination of the vectors from either set, that is, with covariant and contravariant components.

7.1.3 Metric coefficients, g_{ij} and g^{ij}

A metric is a non-negative function, g(x,y) describing the distance, ds, between two points. In Euclidean space the square of this distance is given by Pythagoras:

$$ds^2 = dx^2 + dy^2 + dz^2 \tag{7.2}$$

In curvilinear coordinate space we have the general quadratic form:

$$\begin{aligned} ds^2 = & g_{11}du^1du^1 + g_{12}du^1du^2 + g_{13}du^1du^3 \\ + & g_{21}du^2du^1 + g_{22}du^2du^2 + g_{23}du^2du^3 \\ + & g_{31}du^3du^1 + g_{32}du^3du^2 + g_{33}du^3du^3 \\ = & \sum_{ij} g_{ij}du^idu^j \end{aligned} \tag{7.3}$$

For a simple Cartesian system the metric coefficient, g_{ij}, is a 3×3 identity matrix and equation (7.3) reduces to Pythagoras. Spatial domains for which equation (7.3) is valid are called *metric* or *Riemannian.* For more complicated systems the metric may be used to evaluate the distance between two points by integration.

The tangential metric coefficient g_{ij} may be defined as the dot product of the tangent basis vectors:

$$g_{ij} \quad = \quad \mathbf{e}_i \cdot \mathbf{e}_j \quad = \quad \frac{\partial \mathbf{R}}{\partial u^i} . \frac{\partial \mathbf{R}}{\partial u^j} \quad = \quad \frac{\partial x}{\partial u^i}\frac{\partial x}{\partial u^j} + \frac{\partial y}{\partial u^i}\frac{\partial y}{\partial u^j} + \frac{\partial z}{\partial u^i}\frac{\partial z}{\partial u^j} \tag{7.4}$$

where $\mathbf{R} = \mathbf{R}(x, y, z)$, and $\mathbf{e}_i$ and $\mathbf{e}_j$ are the tangent basis vectors at points p_1 and p_2 respectively. From this definition it follows that g_{ij} is symmetric: $g_{ij} = g_{ji}$. The metric coefficient matrix g_{ij} is fundamental to general curvilinear coordinate systems. It can be used to determine the differential arc length along a curve. It also allows us to change between covariant and contravariant vector components.

Similarly, the reciprocal or conjugate metric coefficient, g^{ij} results from the dot product of the reciprocal basis vectors:

$$g^{ij} \quad = \quad \mathbf{e}^i \cdot \mathbf{e}^j \quad = \quad \nabla u^i . \nabla u^j \tag{7.5}$$

where $\mathbf{e}^i$ and $\mathbf{e}^j$ are the reciprocal basis vectors at p_1 and p_2. g^{ij} is also symmetric: $g^{ij} = g^{ji}$

Details of the algorithm for evaluating the metric coefficient can be found in Appendix 13.1.

7.1.4 The metric tensor in the transport equation

We now introduce the metric tensor into the vector form of the transport equation (6.6) so that we can write the equation as a 9-element tensor containing mixed and unmixed derivatives. This will allow us to discretize the equation on an unstructured non-rectilinear mesh because the individual elements can be discretized separately. Equation (6.6) can be written as:

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_i}\left(\sqrt{g}\kappa^{ij}\frac{\partial T}{\partial x_j}\right) = -\frac{3n}{2}\frac{\partial T}{\partial t} \qquad i,j = 1,2,3 \tag{7.6}$$

Recall that in our curvilinear coordinate system, x_1 and x_2 are poloidal coordinates on the plane of each Poincaré plot, and x_3 is a toroidal coordinate which is aligned with the magnetic field, that is, x_3 remains constant as you move along a field line. $\sqrt{g}$ is evaluated as the determinant of the metric tensor and is equivalent to a volume around each mesh point. κ^{ij} is the conduction tensor which can be written as:

$$\kappa^{ij} \quad = \quad \kappa_\perp g^{ij} + (\kappa_\parallel - \kappa_\perp)(h_3)^2\delta_{i,3}\delta_{j,3} \tag{7.7}$$

Kronecker delta, $\delta_{i,k} = 0$ for $i \neq k$, and $\delta_{i,k} = 1$ for $i = k$. Therefore, with $k = 3$, as in (7.7), only the $ij = 33$ term contains the full expression shown in equation (7.7). All the other terms contain the shortened form; $\kappa_\perp g^{ij}$. In tensor form we have:

$$\kappa^{ij} \quad \Rightarrow \quad \begin{bmatrix} \kappa_\perp g^{11} & \kappa_\perp g^{12} & \kappa_\perp g^{13} \\ \kappa_\perp g^{21} & \kappa_\perp g^{22} & \kappa_\perp g^{23} \\ \kappa_\perp g^{31} & \kappa_\perp g^{32} & \kappa_\perp g^{33} + (\kappa_\parallel - \kappa_\perp)(h_3)^2 \end{bmatrix} \tag{7.8}$$

We can now write the full 9-element tensor form of the transport equation.

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_i}\left(\sqrt{g}\kappa^{ij}\frac{\partial T}{\partial x_j}\right) = -\frac{3n}{2}\frac{\partial T}{\partial t} \qquad \Rightarrow$$

$$\begin{aligned}
&\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_1}\left(\sqrt{g}\kappa_\perp g^{11}\frac{\partial T}{\partial x_1}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_1}\left(\sqrt{g}\kappa_\perp g^{12}\frac{\partial T}{\partial x_2}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_1}\left(\sqrt{g}\kappa_\perp g^{13}\frac{\partial T}{\partial x_3}\right)+\\
&\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_2}\left(\sqrt{g}\kappa_\perp g^{21}\frac{\partial T}{\partial x_1}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_2}\left(\sqrt{g}\kappa_\perp g^{22}\frac{\partial T}{\partial x_2}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_2}\left(\sqrt{g}\kappa_\perp g^{23}\frac{\partial T}{\partial x_3}\right)+\\
&\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(\sqrt{g}\kappa_\perp g^{31}\frac{\partial T}{\partial x_1}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(\sqrt{g}\kappa_\perp g^{32}\frac{\partial T}{\partial x_2}\right)+\ldots..\\
&\qquad\qquad\qquad\ldots..+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(\sqrt{g}\left(\kappa_\perp g^{33}+(\kappa_\parallel-\kappa_\perp)(h_3)^2\right)\frac{\partial T}{\partial x_3}\right)\\
&\qquad\qquad= \quad -\frac{3n}{2}\frac{\partial T}{\partial t} \qquad (7.9)
\end{aligned}$$

The diagonal terms, $(i = j)$, are unmixed derivatives. Of these, the purely toroidal term, $ij = 33$, is discretized along the field line. This term includes a $\kappa_\perp$ in the conduction coefficient, but this is negligible compared to $\kappa_\parallel$. Most importantly, there is no contamination (numerical diffusion) of the strong parallel conduction in any term which is not purely toroidal. In this way we obtain a satisfactory separation of parallel and radial transport, which is crucial for modeling in a strongly anisotropic system.

The off-diagonal terms, $(i \neq j)$, are mixed derivatives. Of these, the four terms with $i = 3$ or $j = 3$ combine radial and parallel temperature derivatives, but they contain only radial conduction coefficients. The four terms with $i, j = 1, 2$ include mixed and unmixed, purely radial derivatives. They lie on the planes of the Poincaré plots and represent purely poloidal flux divergences.

Our basic ansatz has the advantage of separating the large anisotropy in the system which simplifies the numerical task. The very large parallel terms appear only in the $ij = 33$ term and can be easily constructed from central differences [43]. All the other terms show much smaller anisotropy and are therefore much less demanding.

In the following chapter we will discuss the special treatment of the $i, j = 1, 2$ subsystem to develop the necessary tool to handle the purely poloidal flux divergences on the cuts.

8 Triangulation of Poincaré plots

The mesh consists of a series of 2D scatter plots called Poincaré plots. The transport equation is discretized, using an implicit method, so that it can be solved at each mesh point. In the discretized form of this equation the temperature at any point, i, is solved as a function of the temperature at the neighboring points. Therefore, we must determine exactly which of the points near i are 'neighbors' of i.

Each mesh point has a unique index number. The toroidal neighbors of i are trivial to find. They are simply the next and previous points on the same field line. The poloidal neighbors of i are not so trivial to find because they lie scattered around i on the same Poincaré plot, mostly on different field lines. The poloidal neighbors are found by triangulating each Poincaré plot. We use a triangulation algorithm to connect the points on each plot in order to form a network of triangles. In this way we determine the connectivities, or neighborhood relationships, between the points, and therefore the neighborhood array of each point. A neighborhood array for a mesh point i is a list of the indices of the neighboring points.

8.1 Delaunay triangulation.

In Delaunay triangulation a special rule called the *Delaunay criterion* is used to determine how the points on a scatter plot should be connected. According to this criterion the circumcircle through any triplet of points contains no points within its circumference. This criterion has the effect of favoring small fat triangles and excluding large and elongated triangles [44]. A Delaunay triangulation is isotropic, and this matches the quasi-isotropic radial transport on the plane of a Poincaré plot. Each Poincaré plot is triangulated independently of the other plots in the mesh, and each triangulated plot shows graphically the neighborhood relationships between the points. The information describing the triangulation is stored in the neighborhood array for each point.

The algorithm for triangulating a Poincaré plot begins by generating every possible triangle, that is, by connecting every point to every other point. This would form a dense mass of overlapping triangles. Each triangle is then tested with the Delaunay criterion. Most triangles fail the test and are discarded. Those that pass the test are kept and therefore contribute to the information stored in the neighborhood arrays. A problem is that the initial number of triangles is very large and this slows down the algorithm.

Consider a W7-X mesh with 40 Poincaré plots, or cuts. The total number of possible triangles on each cut is given by the Binomial theorem, where n is the number of points on a cut (~ 1270), and $k = 3$, the number of vertices on a triangle:

$$C_k^n = \frac{n!}{k!(n-k)!} \quad = \quad \frac{n \times (n-1) \times (n-2)}{k!} \quad = \quad 340,591,140$$

Multiply this by 40, and you have the total number of triangles to be tested with the Delaunay criterion; over 13.6 billion! and each test involves a large number of calculations. In fact, more than 99.999% of these triangles will fail the criterion and be discarded. Therefore, a brute force method of triangulation is very slow and we must find ways to optimize the triangulation algorithm.

8.2 Optimizing the triangulation algorithm.

We can use the geometric features of the Poincaré plots to optimize the triangulation algorithm. We begin, as before, by generating every possible triangle and then we apply a series of filters (rules) which eliminate most triangles. The Delaunay criterion is then applied to the remaining triangles to generate the neighborhood arrays.

Consider the Poincaré plot in Fig.8.1. Each mesh point is assigned a flux surface number. The points on the outermost flux surface in the plasma core have flux surface(i) = 1, whwere i is the point index. Points on flux surfaces further inside the core have successively higher numbers, up to flux surface(i) = 4 for points on the innermost flux surface. All points in the edge region, including the islands, have flux surface(i) = 0. Points on the outer boundary have flux surface(i) = -1. These flux surface numbers are used to greatly reduce the initial large number of triangles.

Filter 1. Points can be connected only to other points which have the same or neighoring

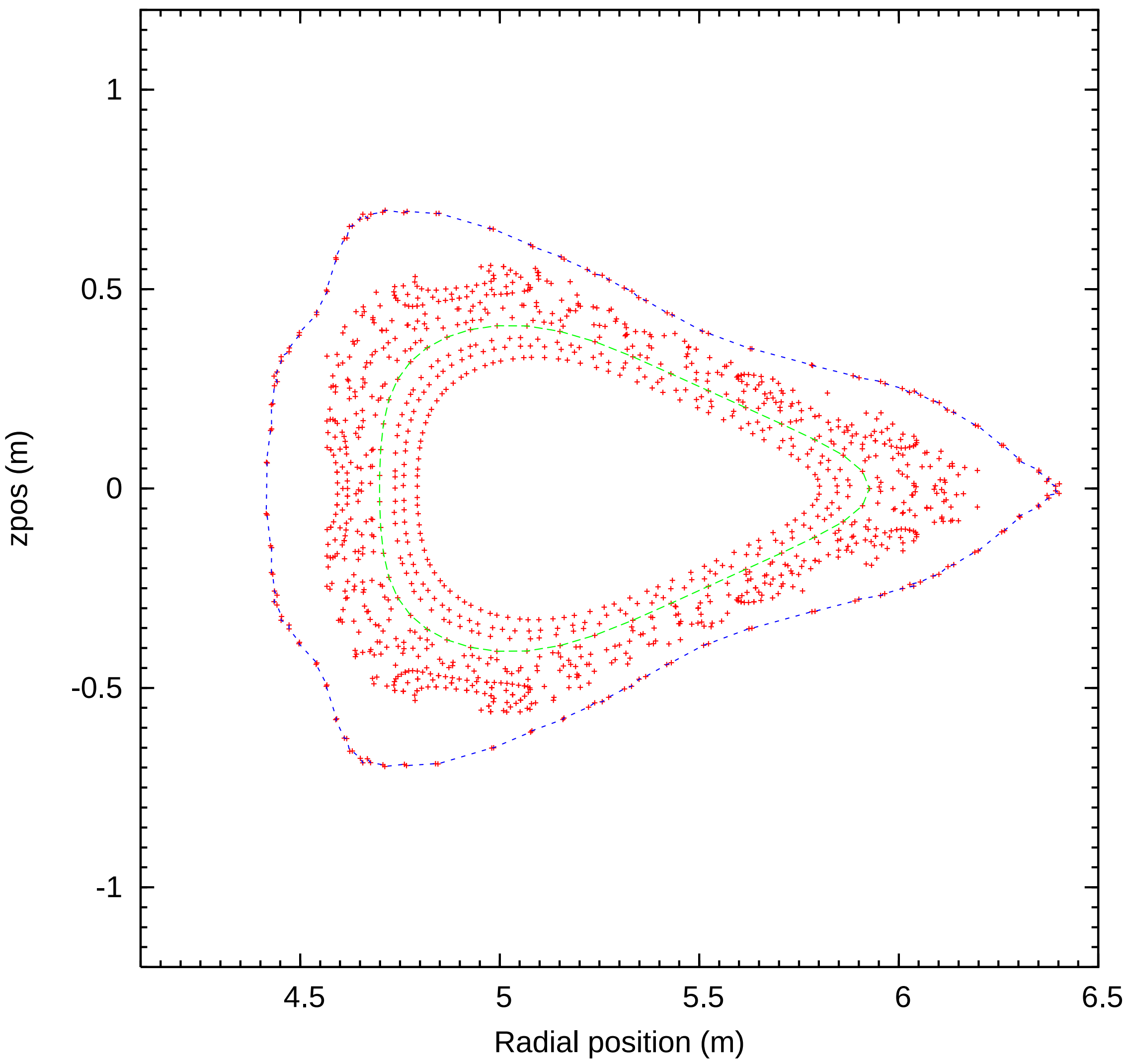

Figure 8.1: *Poincaré plot at* $\phi = 36°$ *from a W7-X mesh. This plot has* ~ 1270 *points. The points on the outermost flux surface of the plasma core (shown as a green line) have flux surface number 1. The points on the outer boundary (shown as a blue line) have flux surface number -1. The points between these 2 curves have flux surface number 0. These designations are used to optimize the triangulation process.*

flux surface numbers. That is, given 2 points, if the difference in their flux surface numbers is greater than 1 then any triangle containing those 2 points is discarded. This has the additional effect of preserving the flux surface structures in the plasma core and thus maintaining realism in the numerical model.

Filter 2. If any triangle has all three points which lie on the innermost plasma core flux surface or on the outer boundary, then that triangle is discarded. This has the additional effect of eliminating triangles which go across the hole in the center of the plasma core and across any concave regions on the outer boundary.

Filter 3. Points can be connected only to other points which lie within a specified search radius. This has the most dramatic effect on reducing the initial number of triangles and speeding up the algorithm, but there is a danger. The search radius should be as small as possible to maximize the optimizing effect, but if it made too small then we risk eliminating valid neighbors. This will result in holes in the neighborhood arrays which will affect the final temperature solution on the mesh. The best search radius is found by estimating the largest separation distance between 2 points on a Poincaré plots and then rounding up.

Fig.8.2 shows a network of triangles formed on a Poincaré plot. A full set of triangulated Poincaré plots in one period of W7-X is shown in Appendix 13.3.

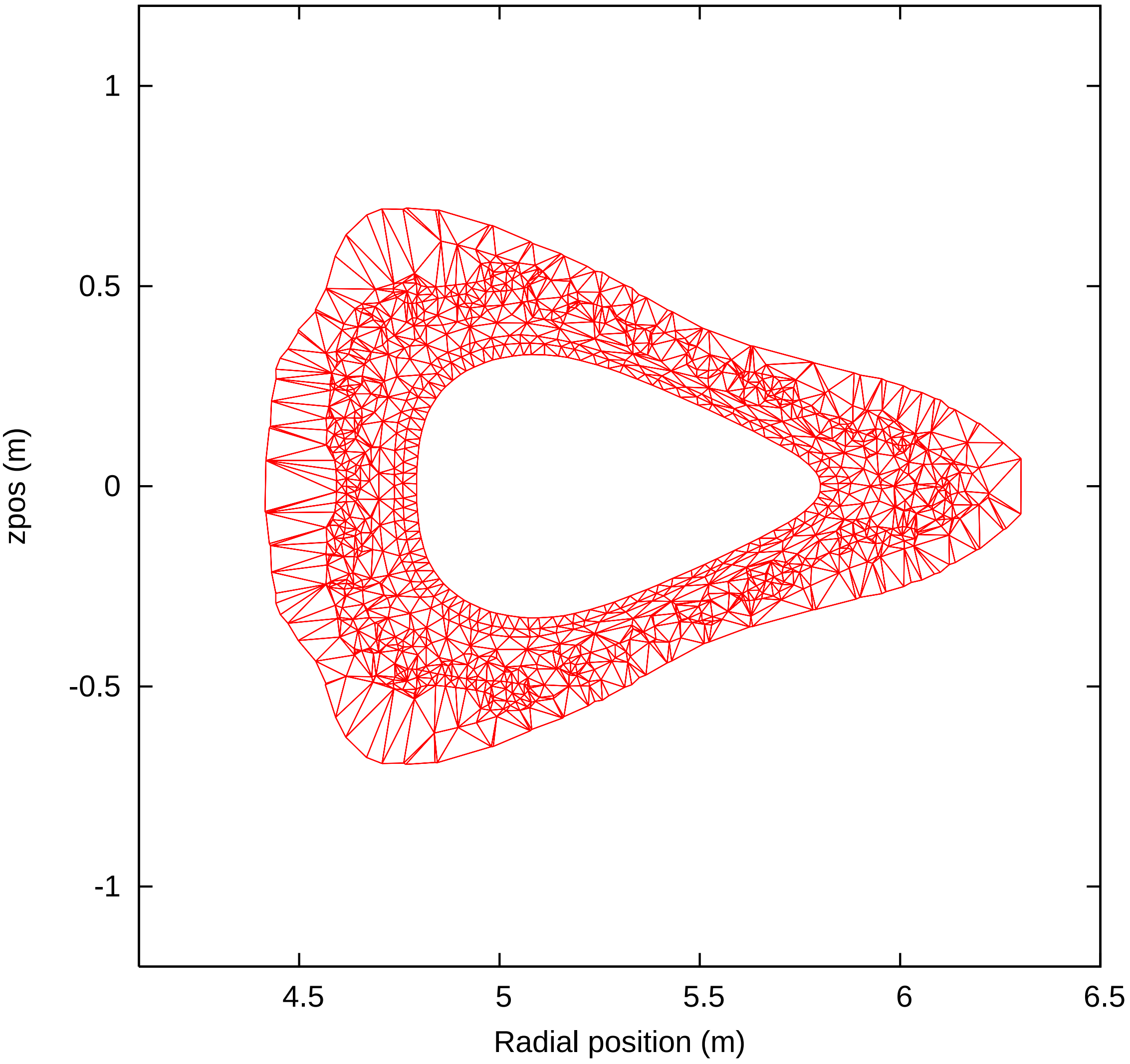

Figure 8.2: *Triangulated Poincaré plot at* $\phi = 36°$ *from a W7-X mesh. There are* ~ 1270 *points and* ~ 2100 *triangles.*

In the present work, the points on the island flux surfaces all have flux surface(i) = 0, like all other edge points. This means that no effort is made to properly preserve the island flux surfaces. In reality they are partially preserved as a result of the close proximity of the points on these structures. It would be possible, with a suitable flux surface numbering system and more complex filters, to properly preserve these structures.

Below is an outline of the triangulation algorithm.

1. Generate all possible triangles on each Poincaré plot using 3 nested DO loops.
2. Apply filters to reduce the large number of initial triangles.
3. Evaluation of circumcenters of remaining triangles.
4. Use Delaunay criterion to determine valid triangles.
5. Build neighborhood arrays from valid triangles.

8.3 Secondary points.

Primary points are the points generated by field line tracing. The transport equation is solved at these points. Most primary points lie on the Poincaré plots. They are the vertices of the triangles. The start and end points of the open field lines are also primary points but they do not lie on the Poincaré plots so they do not play any role in triangulation (but they each have exactly one toroidal neighbor).

Secondary points are the centers of gravity of the triangles, therefore they exist only on the Poincaré plots. Each secondary point also has a unique index number and for each primary point we also store the list of neighboring secondary points. For a given primary point i, if the secondary neighbors are connected to each other, they form a polygonal cell around that point i on the plane of the Poincaré plot. By connecting all the secondary points in this way we can generate a corresponding cell diagram for a triangulated Poincaré plot, as shown in Fig.8.3. These cells are used to evaluate poloidal flux divergences.

The cell diagram is a geometric dual of the triangulation. Normally, the dual of a Delaunay triangulation is the Voronoi diagram but the cells used in this work are not Voronoi cells. The vertices of Voronoi cells are the circumcenters of the triangles. Such points sometimes lie outside a triangle, if the triangle is very narrow. In this work, the cell vertices are the centers of gravity of the triangles. Such points always lie inside a triangle. This property is useful when we generate the poloidal fluxes.

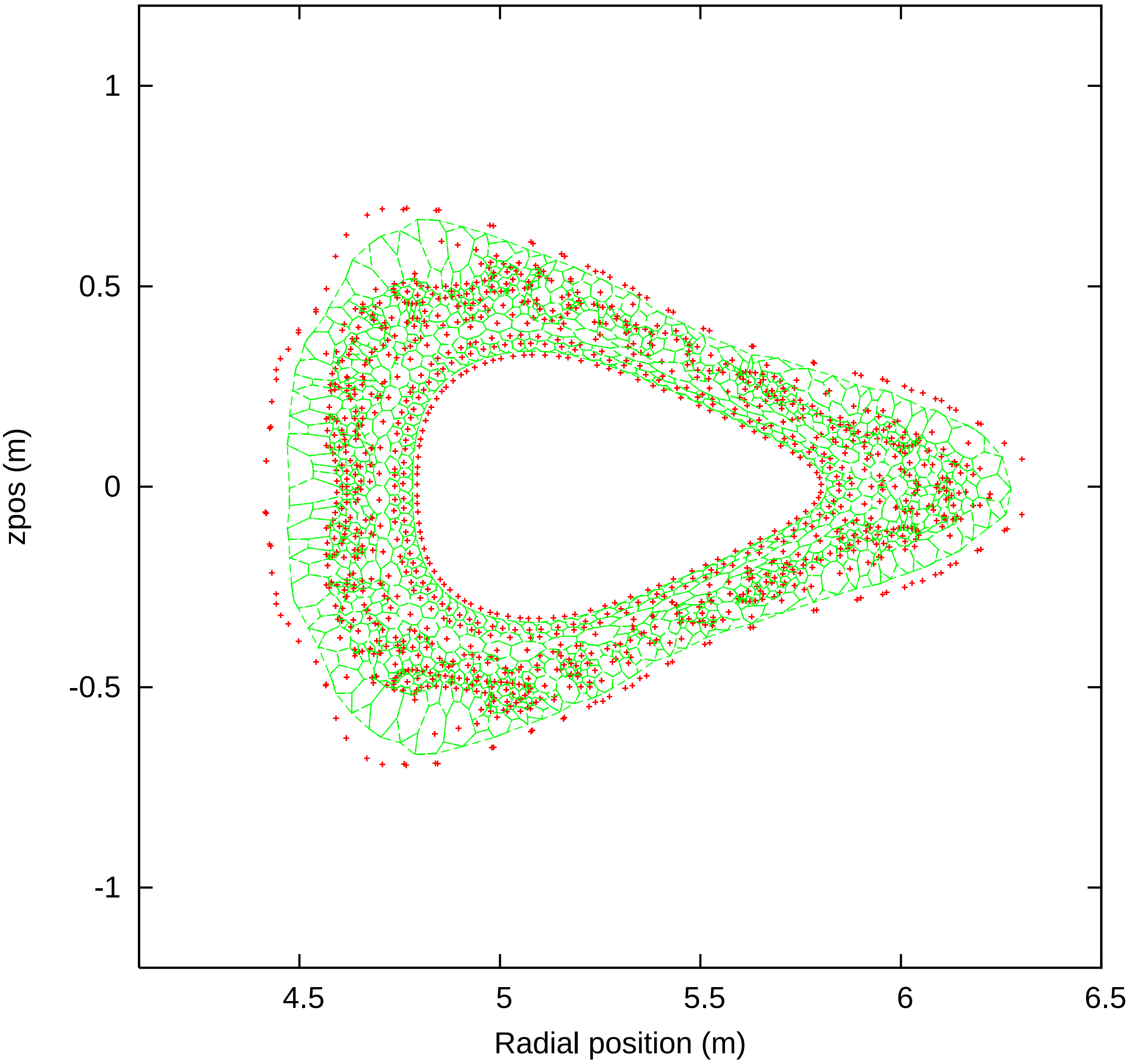

Figure 8.3: *Cell diagram of a Poincaré plot at* $\phi = 0°$ *from a W7-X mesh. The red points are primary points. The polygonal vertices are secondary points. Primary points on the innermost core flux surface and on the outer boundary are poloidal boundary points and have only partial cells.*

Fig.8.4 shows a close-up view of part of a triangulated Poincaré plot. The corresponding cell diagram has been superimposed and the mesh points (primary points) are shown in blue. Each mesh point is connected to 6 poloidal neighbors, on average, and therefore, each polygonal cell has 6 faces, on average. This average number of poloidal neighbors is indicated on the stencil in Fig.5.8. It is possible to quantify the uniformity of the points on a Poincaré plot by calculating the proportion of points which have exactly 6 neighbors.

In the bottom right hand quarter of Fig.8.4 it is possible to see that the 4 flux surface structures in the plasma core have been preserved. In the top right hand quarter it is possible to see that the flux surfaces in one of the islands have been partially preserved. The triangulation code generates a complete neighborhood array for each point, that is, poloidal and toroidal neighbors, and also secondary neighbors, cell areas and a list of the point triplets which form the valid triangles. All this information is used in the transport code.

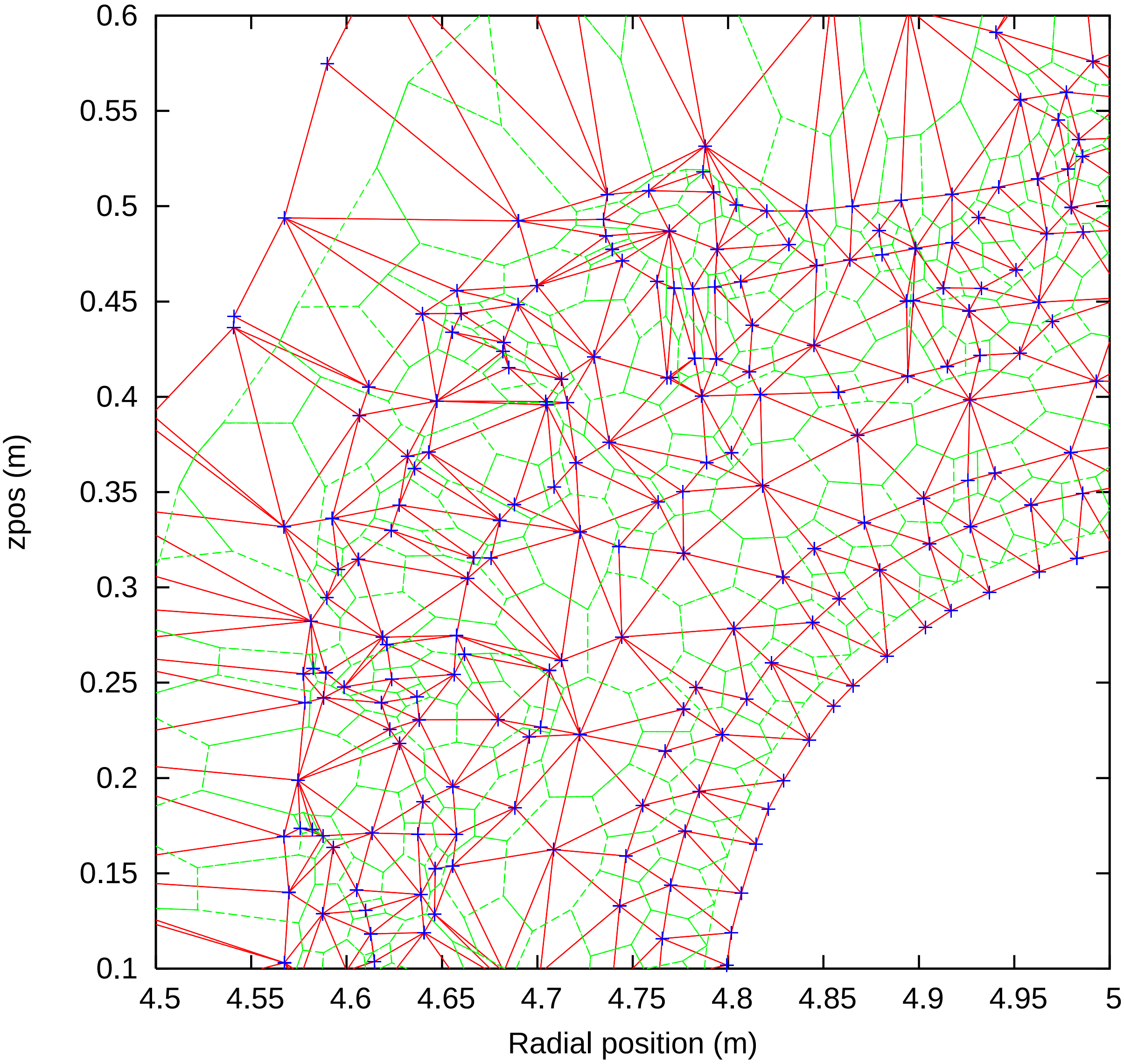

Figure 8.4: *Close-up of a triangulation, with cells. The mesh points, in blue, are connected to each other by triangulation edges, in red. The green lines are the cell faces and each cell has one mesh point.*

9 Constructing the Coefficient Matrix

We are solving a transport equation simultaneously at every point in a mesh. The resulting system of simultaneous equations is a matrix equation of the form $Ax = b$, where A is the matrix of coefficients. The left hand side of the transport equation, shown below, is a flux divergence which can be expanded into a 9-element tensor:

$$\sum_{i,j=1,2,3} \frac{1}{\sqrt{g}} \frac{\partial}{\partial x_i} \left(-\sqrt{g}\kappa_\perp g^{ij} \frac{\partial T}{\partial x_j} \right) \quad \Rightarrow \quad \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} \tag{9.1}$$

Equation 7.9 shows the full form. Each element of the tensor is a separate flux divergence with a unique combination of derivatives. Our task is to discretize these tensor elements to generate a series of 'component' matrices. These are combined by addition to form the coefficient matrix, A, for the mesh.

All the component matrices, and the coefficient matrix, are very sparse, $\sim 99.99\%$ zeros. However, they are very large, so there are still many thousands of non-zero elements in each matrix. We use special methods for storing and manipulating these matrices. This process is described in Appendix 13.5.

The table below summarizes the differences in discretizing the transport equation on two different meshes; a simple 3D mesh, and a realistic mesh used for the W7-X plasma.

Rectilinear structured 3D meshes	**Non-rectilinear unstructured W7-X mesh**
The mesh points lie on mutually orthogonal axes and the number of neighbors for inner points is a constant.	The mesh points do *not* lie on mutually orthogonal axes and the number of neighbors for inner points is *not* a constant.
7-point stencil; each inner point has exactly 6 neighbors; 2 in each coordinate direction (positive and negative)	∼21-point stencil; each inner point has exactly 2 toroidal neighbors and ∼18 poloidal neighbors, including the poloidal neighbors of the toroidal neighbors.
The rectilinear form of the transport equation is split into 3 mutually orthogonal components, each representing an unmixed derivative, ie purely in one coordinate direction, there are no mixed derivatives.	The vector form of the transport equation is re-written as a 9-element tensor, the diagonal elements represent unmixed derivatives, the off-diagonal elements represent mixed derivatives, a non-rectilinear mesh yields non-zero unmixed derivatives.
The metric tensor, g^{ij} is a 3×3 identity matrix.	The metric tensor is 'full', ie all elements are non-zero.
'Single matrix method'; the mesh structure and stencil are simple, the coefficient matrix may be generated in a single loop over all the mesh points using standard discretization techniques. This method can be applied only to structured meshes.	'Multi-matrix method'; the mesh structure and stencil are complex, loop over the mesh points many times, each time generate a separate 'component' matrix for one or more of the 9 elements, combine the component matrices to form the coefficient matrix. This method can be applied to any mesh.
A 7-point stencil gives 7 diagonal bands of non-zero elements in the coefficient matrix.	A ∼21-point stencil gives multiple non-zero elements on each row of the coefficient matrix, but the distribution of the non-zeros is a complex unstructured pattern.

9.1 Poloidal Flux Divergences

The poloidal coordinates, x_1 and x_2, are orthogonal coordinates on the plane of each Poincaré plot. The four poloidal flux divergences, shown below, contain derivatives only in these coordinates. They are treated as a group.

$$\begin{bmatrix} \bullet & \bullet & \circ \\ \bullet & \bullet & \circ \\ \circ & \circ & \circ \end{bmatrix} \qquad i,j = 1,2$$

$$\begin{array}{cccc} \frac{1}{\sqrt{g}}\frac{\partial}{\partial x_1}\left(-\sqrt{g}\kappa_\perp g^{11}\frac{\partial T}{\partial x_1}\right) & , & \frac{1}{\sqrt{g}}\frac{\partial}{\partial x_1}\left(-\sqrt{g}\kappa_\perp g^{12}\frac{\partial T}{\partial x_2}\right) & , \\ \frac{1}{\sqrt{g}}\frac{\partial}{\partial x_2}\left(-\sqrt{g}\kappa_\perp g^{21}\frac{\partial T}{\partial x_1}\right) & , & \frac{1}{\sqrt{g}}\frac{\partial}{\partial x_2}\left(-\sqrt{g}\kappa_\perp g^{22}\frac{\partial T}{\partial x_2}\right) & , \end{array}$$

$$\Rightarrow \qquad L_\perp f \quad = \quad \frac{1}{\sqrt{g}}\frac{\partial}{\partial x^i}\sqrt{g}\left(\kappa^{ij}\frac{\partial f}{\partial x^j}\right) \quad , \quad i,j = 1,2 \tag{9.2}$$

In order to describe the $L_\perp f$ term we use a 2D finite volume ansatz limited to the Poincaré plots. Fig.9.1 shows a small representative part of a Poincaré plot.

The operator for the poloidal flux divergences can be written as:

$$L_\perp f \quad = \quad \sqrt{\frac{g^{ij}_{2D}}{g^{ij}}}\, div_{2D}\mathbf{F} \tag{9.3}$$

where div_{2D} is the 2D divergence operator on the Poincaré plot and the radial fluxes $\mathbf{F}$ are redefined as:

$$F^i \quad = \quad \sqrt{\frac{g}{g_{2D}}}\left(\kappa^{ij}\frac{\partial f}{\partial x^j}\right) \tag{9.4}$$

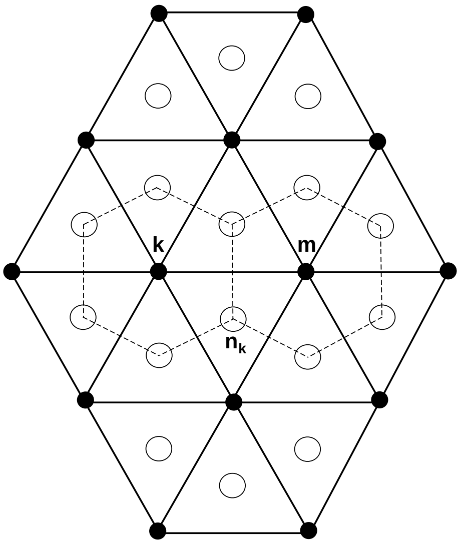

Figure 9.1: *Generalized part of a trianglated Poincaré plot. Full circles are primary points. Open circles are secondary points.*

where g_{2D} is the determinant of the submatrix $[g^{ij}]$, i,j=1,2. Now we can make use of Gauss' theorem on the 2D Poincaré cut and write the flux divergence at the point k:

$$(div_{2D}\mathbf{F})_k = \frac{1}{\Delta S_k}\sum_{n_k}\mathbf{F_{n_k}}\cdot\mathbf{dl_{n_k}} \tag{9.5}$$

as the sum of the fluxes over the cell faces, n_k, divided by the cell volume ΔS_k. Here, $\mathbf{dl_{n_k}}$ is the vector outward normal to the face, the length of which is equal to the corresponding face length. In order to approximate the conductive term we use the analog of the central difference scheme. This means that for every pair of secondary neighbors we take the average of the flux vectors at the cell face mid point.

The group of 4 purely poloidal flux divergences in 9.2 contains 2 unmixed derivatives ($ij = 11, 22$), and 2 mixed derivatives ($ij = 12, 21$). For simplicity, we assume quasi-isotropic transport on the plane of each Poincaré plot, that is, we do not distinguish between radial conductions in the x_1 and x_2 coordinate directions. This group may be represented as a product of matrices.

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_i}\left(-\sqrt{g}\kappa_\perp g^{ij}\frac{\partial T}{\partial x_j}\right) \quad \Rightarrow \quad \frac{1}{\sqrt{g}}\left[\frac{\partial}{\partial x_i}\right]\left[\frac{-\sqrt{g}\kappa_\perp g^{ij}}{1}\right]\left[\frac{\partial}{\partial x_j}\right]T^{\Delta t}$$

In the iteration process, $T^{\Delta t}$ is the array of unknown temperatures across the mesh at the current iteration. It is implicit and so it has no role in the construction of the coefficient matrix. From the expression above, we can write:

$$\Rightarrow \qquad \frac{1}{\sqrt{g}}[A2][A1][A0] \quad = \quad \frac{1}{\sqrt{g}}[A6]$$

Elements of matrix	*represent*
$[A0]$	poloidal gradients : $\partial/\partial x_{11}$, $\partial/\partial x_{12}$
$[A2]$	poloidal gradients : $\partial/\partial x_{21}$, $\partial/\partial x_{22}$
$[A1]$	poloidal transport coefficients : $-\sqrt{g}\kappa_\perp g^{11}$, $-\sqrt{g}\kappa_\perp g^{12}$, $-\sqrt{g}\kappa_\perp g^{21}$, $-\sqrt{g}\kappa_\perp g^{22}$

These are called 'fundamental' matrices. They are used to construct the component matrix [A6]. Combining these matrices successively gives:

$$\begin{aligned}
[A0]T^{\Delta t} &= \text{poloidal temperature gradients} \\
[A1][A0]T^{\Delta t} &= \text{poloidal temperature fluxes} \\
\tfrac{1}{\sqrt{g}}[A2][A1][A0]T^{\Delta t} &= \tfrac{1}{\sqrt{g}}[A6]T^{\Delta t}, \text{ poloidal flux divergences}
\end{aligned}$$

In this way, the component matrix [A6] contains the coefficients for the poloidal flux divergences. Later, [A6] is combined with other component matrices to form the final coefficient matrix for the mesh.

Matrix [A0]

$$\underset{2NS \,\times\, NP}{\begin{bmatrix} \quad A0 \quad \end{bmatrix}}$$

The elements in [A0] are poloidal gradients, or spatial derivatives in x_1 and x_2, evaluated at the secondary points. [A0] has 2NS rows and NP columns. NS is the number of secondary points in the mesh. NP is the number of primary points in the mesh. Therefore, the rows correspond to secondary points and the columns correspond to primary points.

Each secondary point has exactly three primary point neighbors. So each row in [A0] has 3 non-zero elements. [A0] has 2NS rows because the rows occur in pairs. Each pair corresponds to one secondary point in the mesh. In each pair, the first row contains the spatial derivatives in x_1 and the second row contains the spatial derivatives in x_2. [A0] has an unstructured distribution of non-zero elements.

Matrix [A1]

$$\begin{array}{c} \left[\quad A1 \quad \right] \\ 2NS \times 2NS \end{array}$$

The elements in [A1] are the poloidal transport coefficients evaluated at each secondary point. [A1] has a block diagonal structure where each block has the form:

$$\begin{bmatrix} \kappa_\perp g^{11} & \kappa_\perp g^{12} \\ \kappa_\perp g^{21} & \kappa_\perp g^{22} \end{bmatrix}$$

Matrix [A2]

$$\begin{array}{c} \left[\quad A2 \quad \right] \\ NP \times 2NS \end{array} \tag{9.6}$$

The elements in [A2] are spatial derivatives in x_1 and x_2 evaluated at the primary points. The rows correspond to primary points, and the columns correspond to the secondary neighbors of the primary points. The columns in [A2] occur in pairs. Each pair corresponds to a secondary point. In each pair, the first column contains the derivatives in x_1, and the second column contains the derivatives in x_2. Each secondary point has exactly three primary neighbors. So each column in [A2] has three non-zero elements. Conversely, each primary point has ∼6 secondary neighbors. So each row in [A2] has ∼12 non-zero elements.

Structurally, [A2] is the transpose of [A0] but the elements are evaluated in the following way. Consider 2 adjacent secondary neighbors, a and b, of primary point, i. a and b have known coordinates on the Poincaré plane. Also, a and b each have a flux vector with known x_1 and x_2 components. These components are the poloidal gradients evaluated for matrix [A0]. First, the vector joining these points, ie the cell face $\overline{a,b}$, and the normal to this cell face, are given by:

$$\overline{a,b} \quad = \quad (x_a - x_b,\, y_a - y_b) \tag{9.7}$$

$$\vec{n} \quad = \quad \overline{-b,a} \quad = \quad (-(y_a - y_b),\, (x_a - x_b)) \tag{9.8}$$

The flux vector for this cell face is the average of the flux vector components at a and b:

$$flux(\overline{ab}) \quad = \quad \left(\frac{1}{2}\left(x_{flux(a)} + x_{flux(b)}\right),\, \frac{1}{2}\left(y_{flux(a)} + y_{flux(b)}\right)\right) \tag{9.9}$$

The total flux through the cell face is:

$$\begin{gathered} flux(\overline{ab}) \cdot \vec{n} \quad = \\ -(y_a - y_b)\frac{1}{2}(x_{flux(a)} + x_{flux(b)}) \quad + \quad (x_a - x_b)\frac{1}{2}(y_{flux(a)} + y_{flux(b)}) \end{gathered} \tag{9.10}$$

Conceptually, we sum the fluxes through all the cell faces for each primary point and then divide by the cell volume to get the flux divergence at that primary point. This means that the flux vector components at each secondary point will enter twice; once when the secondary point is the first member of an adjacent pair, and once when it is the second member of an adjacent pair. This leads to a cancelling out.

Now consider a sequence of three adjacent secondary neighbors, a, b, c, of primary point i. That is, a is followed by b is followed by c. The flux components at the middle point, b, enter into the divergence as follows:

$$-(y_c - y_a)\frac{1}{2}x_{flux(b)} \quad + \quad (x_c - x_a)\frac{1}{2}y_{flux(b)}$$

$$\Rightarrow \quad \frac{1}{2}(x_c - x_a)y_{flux(b)} \quad - \quad \frac{1}{2}(y_c - y_a)x_{flux(b)} \tag{9.11}$$

Finally, dividing through by the cell volume, we have two terms for each secondary neighbor of primary point i. The even numbered columns have elements evaluated as:

$$\frac{1}{2}\,\frac{1}{\text{cell volume(i)}}\,(x_c - x_a) \tag{9.12}$$

and the odd numbered columns have elements evaluated as:

$$\frac{1}{2}\,\frac{-1}{\text{cell volume(i)}}\,(y_c - y_a) \tag{9.13}$$

9.2 Toroidal flux divergence

The remaining flux divergences in the transport equation are treated individually. First, we consider the purely toroidal term.

$$\begin{bmatrix} \circ & \circ & \circ \\ \circ & \circ & \circ \\ \circ & \circ & \bullet \end{bmatrix} \qquad ij = 33$$

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(-\sqrt{g}\left(\kappa_\perp g^{33} + \left(\kappa_\parallel - \kappa_\perp\right)(h_3)^2\right)\frac{\partial T}{\partial x_3}\right) \tag{9.14}$$

x_3 is a toroidal coordinate. At any mesh point, x_3 is tangential to the field line at that point. The poloidal coordinates, x_1 and x_2, lie on the plane of each Poincaré plot. The field lines do *not* intersect the Poincaré plots at right angles, therefore the toroidal coordinate is *not* orthogonal to the poloidal coordinates. The purely toroidal term, (9.14), is discretized along the field line to evaluate the parallel transport to high accuracy. This is a key concept in handling the strong anisotropy of the plasma.

Recall that for each mesh point i, the local coordinate system extends over 3 consecutive Poincaré plots, or cuts. These are indicated by the indices $(i,-1)$, $(i,0)$ and $(i,1)$ for the behind, central and forward cuts, respectively. Thus, each point i has 3 sets of metric coefficients. For example, for g^{33} we have $g^{33}_{i,-1}$, $g^{33}_{i,0}$ and $g^{33}_{i,1}$. These enable us to take derivatives of the transport coefficients in the x_3 coordinate direction. For simplicity, we set:

$$\begin{aligned}
\psi_{i,-1} &= -\sqrt{g}\left(\kappa_\perp g^{33}_{i,-1} + \left(\kappa_\parallel - \kappa_\perp\right)(h_3)^2\right)\\
\psi_{i,0} &= -\sqrt{g}\left(\kappa_\perp g^{33}_{i,0} + \left(\kappa_\parallel - \kappa_\perp\right)(h_3)^2\right)\\
\psi_{i,1} &= -\sqrt{g}\left(\kappa_\perp g^{33}_{i,1} + \left(\kappa_\parallel - \kappa_\perp\right)(h_3)^2\right)
\end{aligned}$$

In the discretization of the toroidal term, we use ψ^+ and ψ^- which are evaluated as averages at the halfway points between i and the forward and behind neighbors. Recall that in the toroidal transport term there is some small contribution from radial conduction, $\kappa_\perp$, but it is negligible compared to the parallel conduction, $\kappa_\parallel$ The toroidal term is discretized in the following way:

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial z}\left(-\psi\frac{\partial T}{\partial z}\right) \quad\Rightarrow\quad \frac{1}{\sqrt{g}}\frac{2}{\delta x_3^+ + \delta x_3^-}\left((\psi^+)\frac{T_{i+1}-T_i}{\delta x_3^+} - (\psi^-)\frac{T_i - T_{i-1}}{\delta x_3^-}\right) \tag{9.15}$$

From this we obtain the coefficients for the component matrix [A11]. This matrix has a tri-diagonal structure.

The non-zero elements on the central diagonal are evaluated as:

$$-\frac{1}{\sqrt{g}}\frac{2}{\delta x_3^+ + \delta x_3^-}\left(\frac{\psi^+}{\partial x_3^+} + \frac{\psi^-}{\partial x_3^-}\right) \tag{9.16}$$

The non-zero elements on the upper diagonal are evaluated as:

$$\frac{1}{\sqrt{g}}\frac{2}{\delta x_3^+ + \delta x_3^-}\frac{\psi^+}{\partial x_3^+} \tag{9.17}$$

The non-zero elements on the lower diagonal are evaluated as:

$$\frac{1}{\sqrt{g}}\frac{2}{\delta x_3^+ + \delta x_3^-}\frac{\psi^-}{\partial x_3^-} \tag{9.18}$$

The distances, δx_3^+ and δx_3^-, between the points are known from the field line tracing process.

9.3 Mixed flux divergences, $ij = 13, 23$

$$\begin{bmatrix} \circ & \circ & \bullet \\ \circ & \circ & \bullet \\ \circ & \circ & \circ \end{bmatrix} \quad ij = 13, 23$$

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial x_1} \left(-\sqrt{g} \kappa_\perp g^{13} \frac{\partial T}{\partial x_3} \right) \quad , \quad \frac{1}{\sqrt{g}} \frac{\partial}{\partial x_2} \left(-\sqrt{g} \kappa_\perp g^{23} \frac{\partial T}{\partial x_3} \right) \tag{9.19}$$

These are the flux divergences for the mixed terms $ij = 13$ and $ij = 23$. They are handled separately, but with the same method. We are not taking derivatives of the transport terms along the field line. Therefore we use the metric coefficients at the central cuts, that is, we have:

$$\begin{aligned} \psi_{i,0} &= -\sqrt{g} \kappa_\perp g^{13}_{i,0} \quad \text{for term } ij = 13 \\ \psi_{i,0} &= -\sqrt{g} \kappa_\perp g^{23}_{i,0} \quad \text{for term } ij = 23 \end{aligned}$$

The 2 mixed flux divergences can be written as:

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial x_1} \left(\psi_{i,0} \frac{\partial T}{\partial x_3} \right) \quad \Rightarrow \quad \frac{1}{\sqrt{g}} [A3][A4]T \quad \Rightarrow \quad [A9]T \quad , \quad \text{for term ij} = 13 \tag{9.20}$$

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial x_2} \left(\psi_{i,0} \frac{\partial T}{\partial x_3} \right) \quad \Rightarrow \quad \frac{1}{\sqrt{g}} [A5][A4]T \quad \Rightarrow \quad [A10]T \quad , \quad \text{for term ij} = 23 \tag{9.21}$$

Matrices [A3], [A5] and [A4] are 'fundamental' matrices. They are used to construct the component matrices [A9] and [A10] which contain the flux divergence coefficients for the mixed terms.

Matrix [A3] contains the spatial derivatives in the poloidal coordinate x_1. The elements of [A3] are evaluated by looping over the primary point neighbors of each primary point and using a least squares method to find the poloidal gradient in x_1 at each primary point. Each primary point has, on average, 6 primary point neighbors. Therefore each row in [A3] has, on average, 6 non-zero elements. The least squares method is explained in Appendix 13.4.

Similarly, [A5] contains spatial derivatives in the poloidal coordinate x_2. They are evaluated with the same method as for [A3]. The structures of [A5] and [A3] are identical.

Matrix [A4] contains first order spatial derivatives along the field lines. It has a bi-diagonal structure. The discretization is as follows:

$$\psi_{i,0}\frac{\partial T}{\partial x_3} \quad \Rightarrow \quad \psi_{i,0}\left(\frac{T_{i+1} - T_{i-1}}{\delta x_3^+ + \delta x_3^-}\right) \tag{9.22}$$

From this we obtain the coefficients for the fundamental matrix [A4], which is evaluated separately for $ij = 13$ and $ij = 23$.

The non-zero elements on the upper diagonal are evaluated as:

$$\frac{\psi_{i,0}}{\partial x_3^+ + \partial x_3^-} \tag{9.23}$$

The non-zero elements on the lower diagonal are evaluated as:

$$\frac{-\psi_{i,0}}{\partial x_3^+ + \partial x_3^-} \tag{9.24}$$

The coefficients for the component matrices [A9] and [A10] are obtained by matrix multiplication as shown in expressions (9.20) and (9.21).

9.4 Mixed flux divergences, $ij = 31, 32$

$$\begin{bmatrix} \circ & \circ & \circ \\ \circ & \circ & \circ \\ \bullet & \bullet & \circ \end{bmatrix} \quad ij = 31, 32$$

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(-\sqrt{g}\kappa_\perp g^{31}\frac{\partial T}{\partial x_1}\right) \quad , \quad \frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(-\sqrt{g}\kappa_\perp g^{32}\frac{\partial T}{\partial x_2}\right) \tag{9.25}$$

These are the flux divergences for the mixed terms $ij = 31$ and $ij = 32$. They are handled separately, but with the same method. Unlike with the previous mixed flux divergences, this time we *are* taking derivatives of the transport terms along the field lines. Therefore we will use:

$$\begin{aligned}
\psi_{i,1} &= -\sqrt{g}\kappa_\perp g^{13}_{i,1} && \text{for term } ij = 13 \\
\psi_{i,-1} &= -\sqrt{g}\kappa_\perp g^{13}_{i,-1} && \text{for term } ij = 13 \\
\\
\psi_{i,1} &= -\sqrt{g}\kappa_\perp g^{23}_{i,1} && \text{for term } ij = 23 \\
\psi_{i,-1} &= -\sqrt{g}\kappa_\perp g^{23}_{i,-1} && \text{for term } ij = 23
\end{aligned}$$

The 2 mixed flux divergences can be written as:

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(\psi\frac{\partial T}{\partial x_1}\right) \quad \Rightarrow \quad \frac{1}{\sqrt{g}}[A4][A3]T \quad \Rightarrow \quad [A7]T \quad , \quad \text{for term ij} = 31 \tag{9.26}$$

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x_3}\left(\psi\frac{\partial T}{\partial x_1}\right) \quad \Rightarrow \quad \frac{1}{\sqrt{g}}[A4][A5]T \quad \Rightarrow \quad [A8]T \quad , \quad \text{for term ij} = 32 \tag{9.27}$$

Matrices [A3] and [A5] are the same as for the previous 2 mixed flux divergences.

Matrix [A4] contains the first order spatial derivatives of the transport coefficient along the field line. It has a bi-diagonal structure. The discretization is as follows:

$$\frac{\partial\psi}{\partial x_3} \quad = \quad \frac{\psi_{i,1} - \psi_{i,-1}}{\partial x_3^+ + \partial x_3^-} \tag{9.28}$$

From this we obtain the coefficients for [A4], which is evaluated separately for $ij = 31$ and $ij = 32$.

The non-zero elements on the upper diagonal are evaluated as:

$$\frac{\psi_{i,1}}{\partial x_3^+ + \partial x_3^-} \tag{9.29}$$

The non-zero elements on the lower diagonal are evaluated as:

$$\frac{-\psi_{i,-1}}{\partial x_3^+ + \partial x_3^-} \tag{9.30}$$

The coefficients for the component matrices [A7] and [A8] are obtained by matrix multiplication as shown in expressions (9.26) and (9.27).

9.5 Fundamental matrices

The fundamental matrices are used to construct the component matrices.

fund. matrix	*size*	*temperature dependent*	*description of elements*	*used to construct,*
[$A0$]	2NS×NP	no	poloidal gradients in the x_1 and x_2 coordinate directions, evaluated at the secondary points.	[$A6$]
[$A1$]	2NS×2NS	yes	poloidal transport coefficients, evaluated at the secondary points.	[$A6$]
[$A2$]	NP×2NS	no	poloidal gradients in the x_1 and x_2 coordinate directions, evaluated at the primary points.	[$A6$]
[$A3$]	NP×NP	no	poloidal gradients in the x_1 coordinate direction, evaluated at the primary points.	[$A7$], [$A9$]
[$A4$]	NP×NP	no	toroidal gradients in the x_3 coordinate direction, evaluated at the primary points.	[$A7$], [$A8$], [$A9$], [$A10$].
[$A5$]	NP×NP	no	poloidal gradients in the x_2 coordinate direction, evaluated at the primary points.	[$A8$], [$A10$]

• NP : number of primary mesh points. These are points generated by the field line tracing process. The transport equation is solved at these points.

• 2NS : number of secondary mesh points. These are the centers of gravity of the triangles on the triangulated Poincaré plots.

• Matrices which are *not* temperature dependent are strictly geometric and are evaluated outside the iteration loop in the transport code. They are computed only once for a given mesh.

• Matrices which *are* temperature dependent are evaluated inside the iteration loop in the transport code, that is, [A1] is re-computed at every iteration.

9.6 Component matrices

The component matrices are used to construct the final matrix of coefficients.

comp. matrix	*size*	*temperature dependent*	*description of elements*
[$A6$]	NP×NP	yes	purely poloidal flux divergences, on the plane of each Poincaré plot.
[$A7$]	NP×NP	yes	mixed toroidal-poloidal flux divergences
[$A8$]	NP×NP	yes	mixed toroidal-poloidal flux divergences
[$A9$]	NP×NP	yes	mixed poloidal-toroidal flux divergences
[$A10$]	NP×NP	yes	mixed poloidal-toroidal flux divergences
[$A11$]	NP×NP	yes	purely toroidal flux divergences, along the field lines.

The component matrices contain the coefficients of the flux divergences from the 9-element tensor form of the transport equation. They correspond to the metric coefficients in the following way:

$$\begin{bmatrix} g^{11} & g^{12} & g^{13} \\ g^{21} & g^{22} & g^{23} \\ g^{31} & g^{32} & g^{33} \end{bmatrix} \Rightarrow \begin{bmatrix} A6 & A6 & A9 \\ A6 & A6 & A10 \\ A7 & A8 & A11 \end{bmatrix}$$

9.7 Time derivative

The right hand side of the transport equation contains the time derivative. In the matrix equation ($Ax = b$, b is a vector containing the time derivative at each mesh point, i. The time derivative is discretized using a forward difference approximation.

$$-\frac{3n}{2}\frac{\partial T}{\partial t} \quad = \quad -\frac{3n}{2}\frac{T_i^{\Delta t}}{\Delta t} \quad + \quad \frac{3n}{2}\frac{T_i^t}{\Delta t} \tag{9.31}$$

$T_i^{\Delta t}$ is the unknown temperature at mesh point i at the current iteration or time step. Its coefficient, $-3n/2\Delta t$, goes to the left hand side of the transport equation, and it will contribute to the final coefficient matrix. Thus, we have one more component matrix to consider. [At] has a single diagonal structure and its elements are evaluated as $3n/2\Delta t$.

comp. matrix	*size*	*temperature dependent*	*description of elements*
$[At]$	NP×NP	yes	time derivatives

T_i^t is the known temperature at mesh point i at the previous iteration or time step. The elements of vector b are evaluated as $3nT_i^t/2\Delta t$. The elements in b may also include boundary conditions and source terms,

9.8 Constructing the final coefficient matrix

To construct the coefficient matrix for the mesh, we sum all the component matrices.

$$[A6] + [A7] + [A8] + [A9] + [A10] + [A11] + [At] \quad = \quad [A]$$

[A] and b are sent to a matrix solver subroutine. The matrix equation is solved iteratively until the average residual across the mesh falls below a specified convergence criterion.

Fig.9.2 shows the structure of the final coefficient matrix [A].

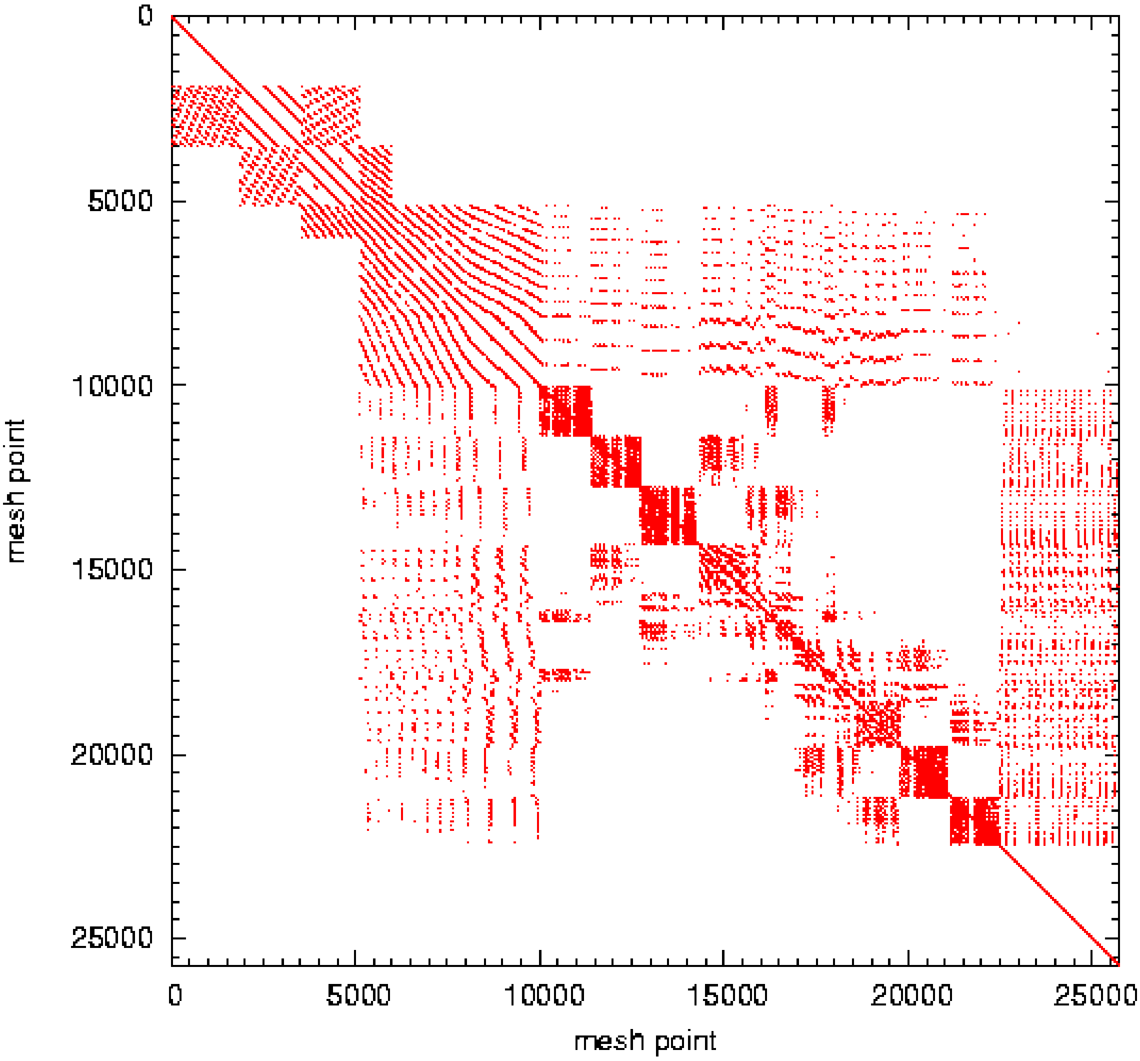

Figure 9.2: *Structure of the final matrix of coefficients [A] for a 'standard' W7-X mesh 11.1.. Total points* ~ 25000. *Note that the matrix is symmetrical about the diagonal. This shows that if any mesh point a is a neighbor of b, then b is also a neighbor of a.*

At the top of Fig.9.2, the first ~2000 mesh points lie on the innermost plasma core flux surface and they are fixed at 200 eV. Therefore the transport equation is not solved at these points. Below this, a series of square structures results from the remaining closed field lines in the core region. Further down the diagonal can be seen 5 smaller square structures resulting from the islands. Elsewhwere, the matrix is relatively unstructured, resulting from the many open field lines in the edge region, some of which are ergodic. At the bottom the last ~3000 mesh points lie on the outer boundary and are fixed at 10 eV, ie the transport equation is not solved at these points.

10 General Ansatz

Here we summarize the general approach to model electron heat transport in the plasma edge. First we recall the characteristics of the physical system:

1. ***Strong anisotropy***. In a magnetized plasma, heat transport is dominated by very strong electron heat conduction along the field lines. Typically, this parallel transport is $10^4 \rightarrow 10^7$ times stronger than the radial transport across the field lines. Because of this strong anisotropy it is very important to separate the parallel and radial transport in order to minimize numerical diffusion in the solution.

2. ***Strong non-linearity***. The parallel conduction is proportional to $T_e^{5/2}$ This leads to strong gradients and added complexity in the solution of the transport equation.

3. ***Field line ergodicity***. Ergodic effects, either by additional coils, as in TEXTOR-DED, or by intrinsic plasma effects, as in W7-X, cause a mixing of parallel and radial transport in the plasma edge region. The result is a flattening of the radial temperature profile across the edge region. We are interested in modeling the edge region of a plasma because this has direct relevance to the performance of fusion experiments.

4. ***Complex geometry***. The present work is focused on the W7-X experiment in which the plasma has a 5-fold symmetry. There are five large island structures in the edge region which wind helically around the core as they go around the torus. These islands are composed of flux surfaces which intersect a series of divertor plates which are themselves complex curved 3D surfaces. In our numerical model all the essential geometrical features in the plasma must be represented.

To handle the strong anisotropy and ensure a separation of the parallel and radial transports we use a finite difference method in which we solve the discretized transport equation at points generated along the field lines. This allows us to evaluate the parallel transport to high accuracy.

The ergodicity in the edge region makes it impossible to follow flux tubes around the torus of the device and excludes the use of a global coordinate system. Therefore we turn to a Multiple Coordinate Systems Approach in which each point has a unique and local coordinate system which extends only as far as the neighboring points, forward and backward, on the same field line. Because the points are generated along the field lines, these local coordinates systems are aligned with the magnetic field, and they are called local magnetic coordinates.

The computational process has 4 main stages:

1. ***Mesh generation***. Starting from a magnetic field configuration data file we use a field line tracing code to trace field lines around the torus and generate points at regular toroidal intervals. Therefore the mesh is a 3D array of points filling the volume of the plasma but with the points lying on a set of discrete toroidal positions. Each mesh point is stored as an index number with x, y, z coordinates.
2. ***Triangulation***. Using a modified Delaunay algorithm we triangulate the mesh to determine the connectivities between the points, that is, the neighborhood array for each point. This information is needed by the discretized form of the transport equation. Each point is stored with a list of the indices of its neighbors.
3. ***Metric Coefficients***. Using another kind of field line tracing code we generate the metric coefficients for each mesh point. These are geometric quantities which describe the local curvature of the mesh. The use of non-orthogonal coordinates at each point means that we have a full metric tensor, that is, the off-diagonal elements are non-zero. Each point is stored with a set of unique metric coefficients.
4. ***Transport code***. The transport code contains the discretized form of the transport equation. It reads the mesh data file, the neighborhood arrays and the metric coefficients. In an input file we specify various operating parameters such as initial and boundary conditions, time step and radial diffusivity. Then the transport code generates the matrix of coefficients (Fig.9.2). It then solves the transport equation as a system of simultaneous equations, by iteration, in order to obtain a steady state electron temperature solution across the mesh. The output may be presented as a series of 2D temperature solutions, one for each Poincaré plot.

The computational process is presented in Fig.10.1. Generating large meshes is time consuming but is done only once for a given magnetic configuration. Then the transport code can be run many times, with different boundary conditions.

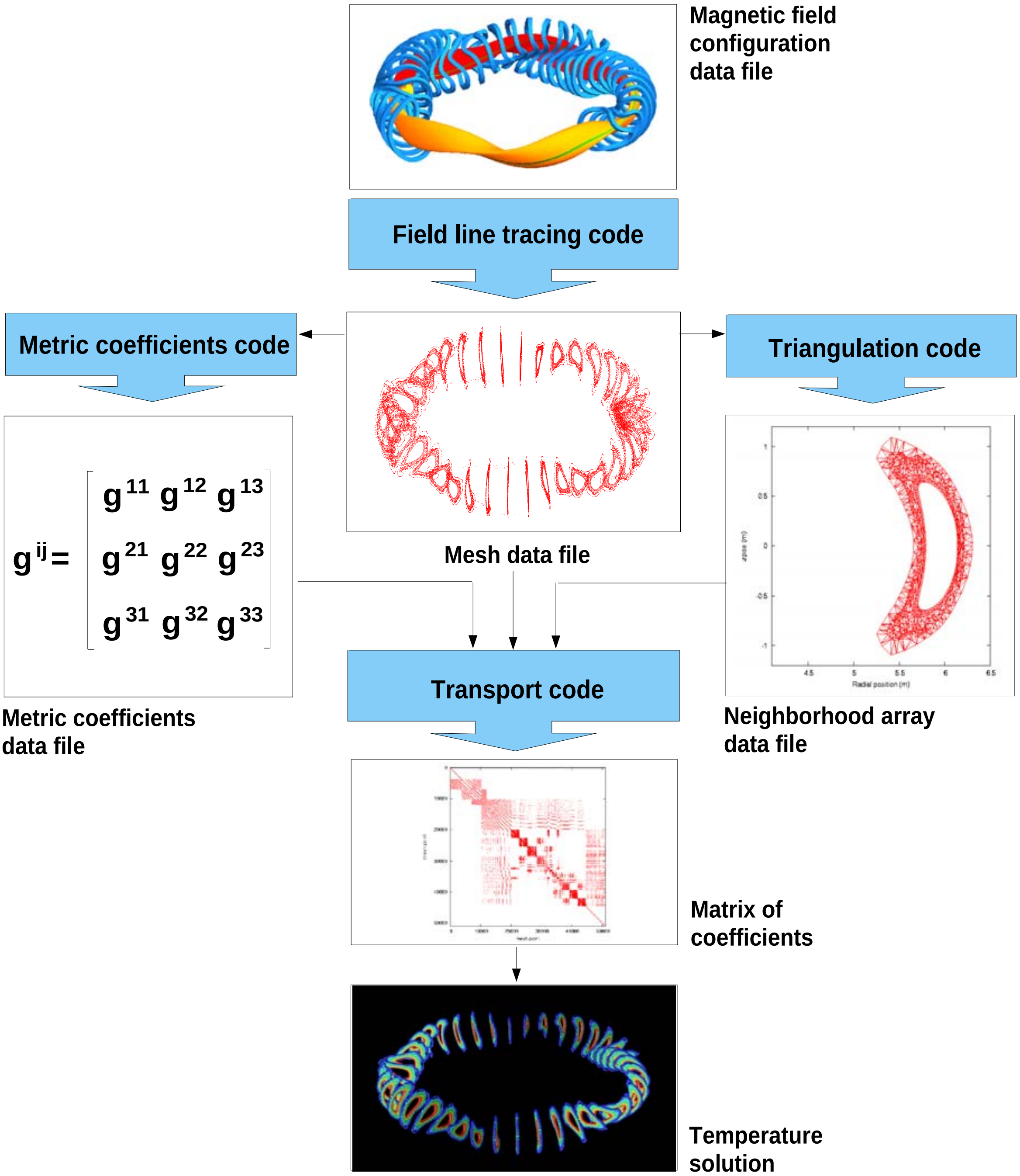

Figure 10.1: *Summary of computational process.*

11 Results and Discussion

In this chapter we discuss results for numerics and physics.

11.1 Numerics

First, it is necessary to validate the numerical concept of using an optimized grid to minimize the numerical diffusion. We also want to investigate the sensitivity of the results to the choice of metrics, that is, a full metric versus a local orthogonal assumption.

11.1.1 Numerical diffusion

Numerical diffusion is numerical error which can creep into the solution as a result of numerical innacuracy in the calculations. In a magnetized plasma this is a potentially serious problem because of the extreme anisotropy, so we need a strict separation of the parallel and radial transport processes. To investigate this problem we can estimate the numerical diffusion in our discretization scheme, compare it to the case where all physical diffusion is switched off, and also investigate the effects of point sources in the mesh.

Estimate of numerical diffusion

To estimate the numerical diffusion in the discretization scheme, a series of studies were made on a 'standard' W7-X mesh shown in Fig 11.1. A Poincaré plot from this mesh is shown in Fig. 11.2. It shows the mesh structure and the different types of field lines which make up the mesh. A sequence of 4 Poincaré plots from the standard mesh is shown in Appendix 13.2.

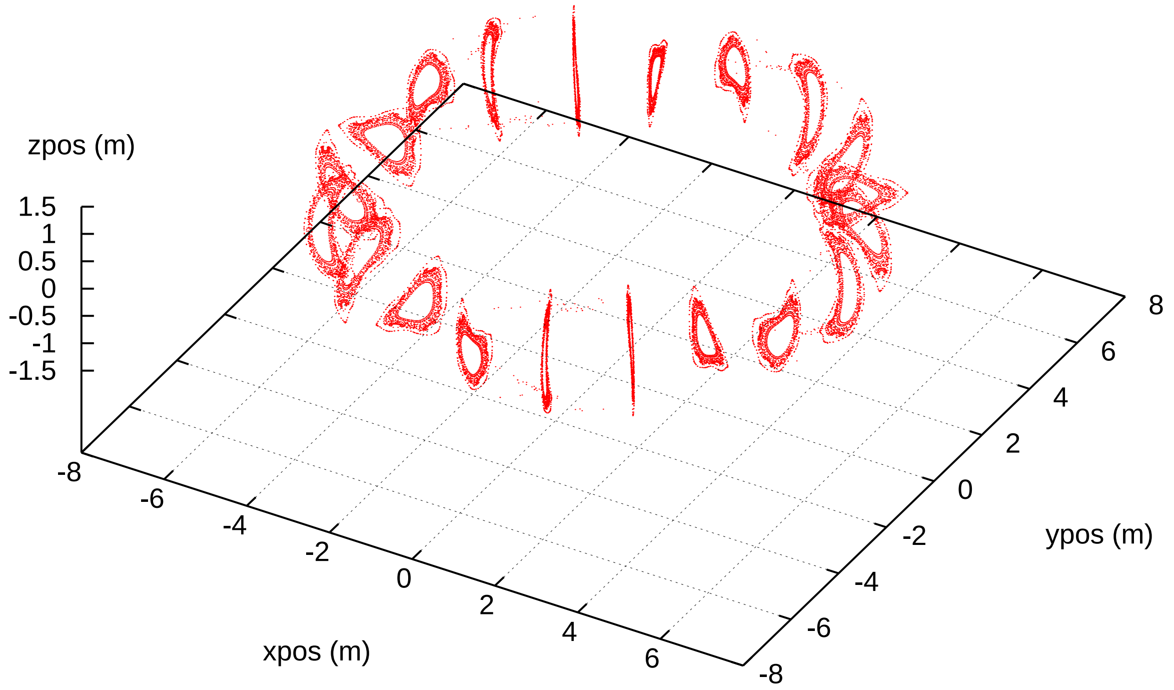

Figure 11.1: *A W7-X edge mesh generated in a finite beta field. This mesh has 20 cuts and contains points in the core and in the edge, plus a field line forming an outer boundary. Total points:* 25719.

In the series of studies the radial diffusivity was varied from $\chi_\perp = 10 \rightarrow$ 1.0e-6 $m^2 s^{-1}$. The temperature solutions are shown in Fig.11.3 and have been staggered vertically to aid in comparison. Note that mesh points are numbered along field lines. In this mesh there are 157 field lines. Most of these are open field lines in the edge. Mesh point 1 is the first point on the first field line (the innermost flux surface in the core), and mesh point 25719 is the last point on the last field line (the outer boundary).

Recall that the radial flux is $Q_\perp = \chi_\perp n_e$, where n_e is the electron density, held constant at 1.0e20m^{-3}. So by reducing the radial diffusivity, we are progressively switching off the mechanism of radial heat transfer. In the core region, the only possibility of heat transfer across the flux surfaces, apart from numerical diffusion, is by radial transport. In the edge region there is an additional parallel transport in the radial direction due to ergodicity.

In Fig.11.3 the step-like structures at the left hand side (mesh points 1 → 10000) correspond to the closed field lines in the plasma core region. This includes the 4 flux surfaces in the core and the 3 closed ergodic field lines surrounding the core. Mesh points

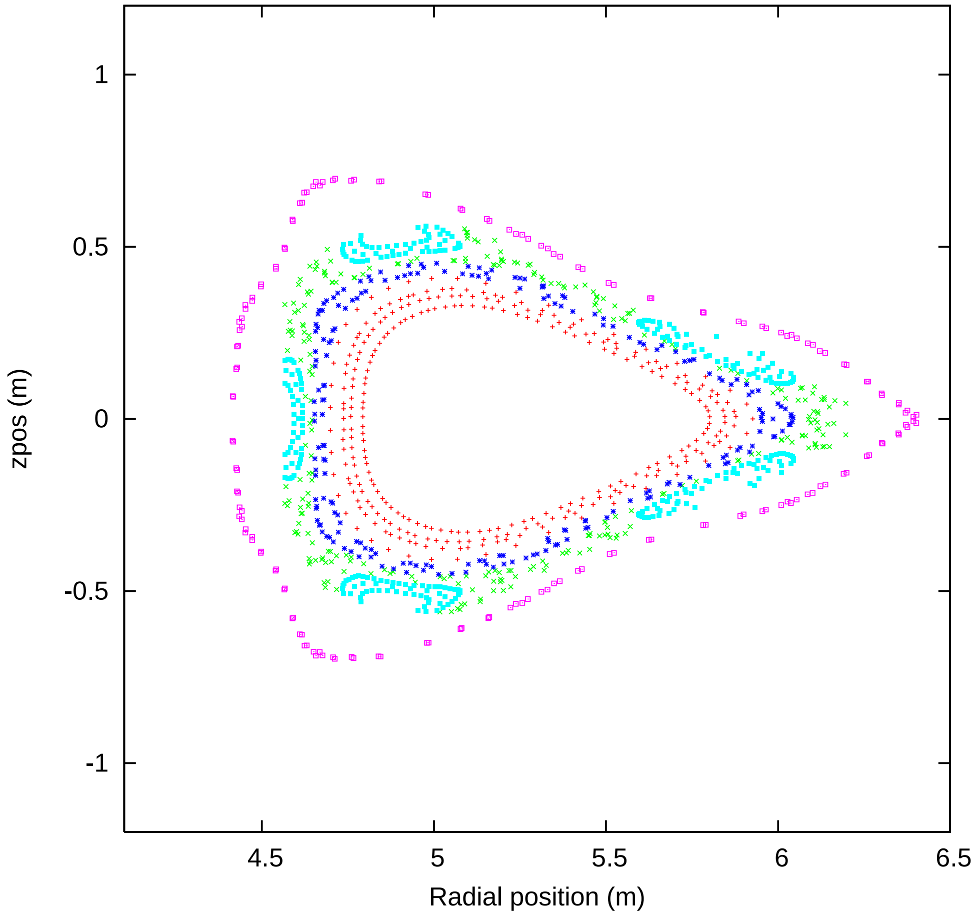

Figure 11.2: *Poincaré plot at* $\phi = 36°$ *from the W7-X mesh in Fig.11.1. Red points lie on field lines which form closed flux surfaces in the plasma core. Dark blue points lie on closed ergodic field lines which form a thin ergodic layer around the plasma core. Green points lie on open field lines in the edge. Light blue points lie on field lines which form flux surfaces in the islands. These flux surfaces are 'open' because they are intersected by the divertor plates. Pink points lie on a closed field line which forms an outer boundary. Total points on this plot* ~ 1270.

10000 → 22000 are points on open field lines in the edge region and the islands. At the right hand side (mesh points 22000+) is the outer boundary. The results show that as radial diffusivity is reduced there is a flattening of the temperature profile. In the core region, the step-like structures converge on the initial condition of 100 eV. There is also a greater decoupling of transport between the core region and the edge region. This decoupling appears as an increasingly large discontinuity at mesh point $\sim$ 10000 which is the last point on the outermost closed field line.

In the core region, there is also a smoothing of fluctuations along the closed field lines. This is most clearly visible in the outermost closed ergodic field line (mesh points 8000 → 10000. These fluctuations are caused by geometric effects related to the 3-dimensional trajectories of the closed field lines, and the close proxmity of the open field lines in the edge. The

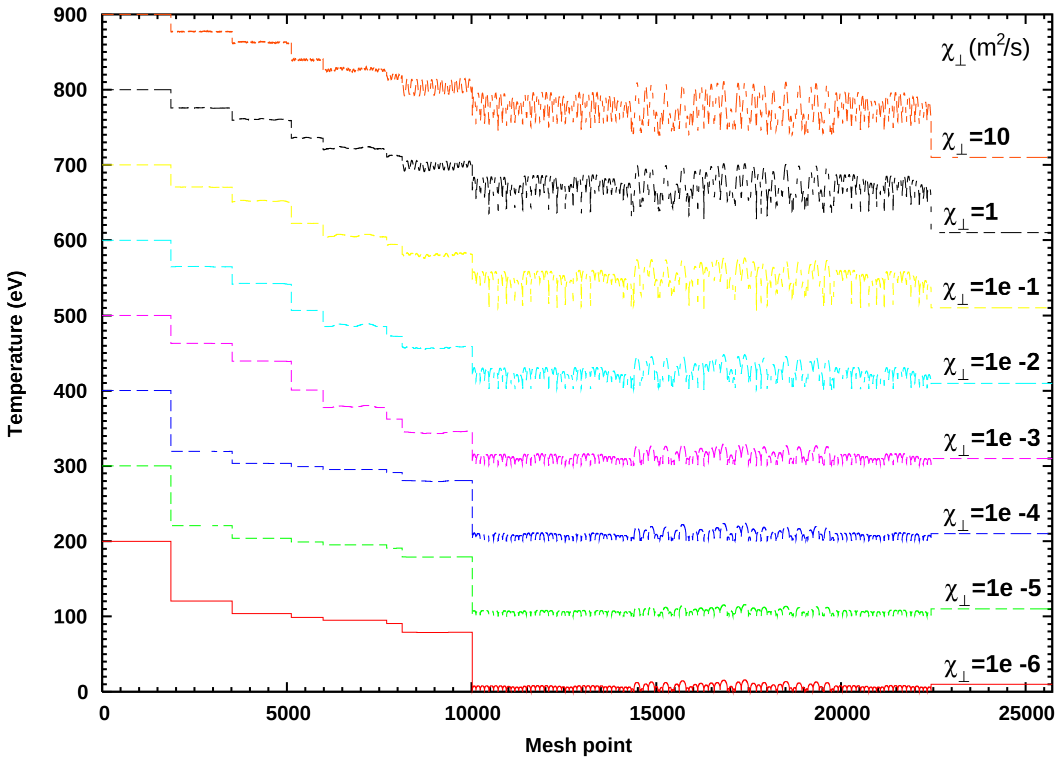

Figure 11.3: *A series of solutions in a W7-X mesh with varying radial diffusivity. All the solutions have the same boundary conditions: 200 eV in the core, 10 eV on the outer boundary, and sheath condition at the ends of the open field lines. The solutions have been shifted vertically to aid in comparison. The colors in this plot do NOT correspond to the colors in Fig.11.2.*

smoothing of these fluctuations is due to the increasing dominance of parallel transport over radial transport as the radial diffusivity is reduced. The bottom two solutions (green and red curves) show that as $\chi_\perp$ is reduced from 1.0e-5 to 1.0e-6 there is almost no change in the solution. This suggests that the numerical diffusion is about 1.0e-5m^2s^{-1}. In the generation of the optimized grid the estimated numerical diffusion was approximately 1.0e-4m^2s^{-1}.

Switching off the radial transport

A further investigation of numerical diffusion can be made by switching off the radial transport completely. The solution shown in Fig.11.4 may be compared with the bottom curve in Fig.11.3.

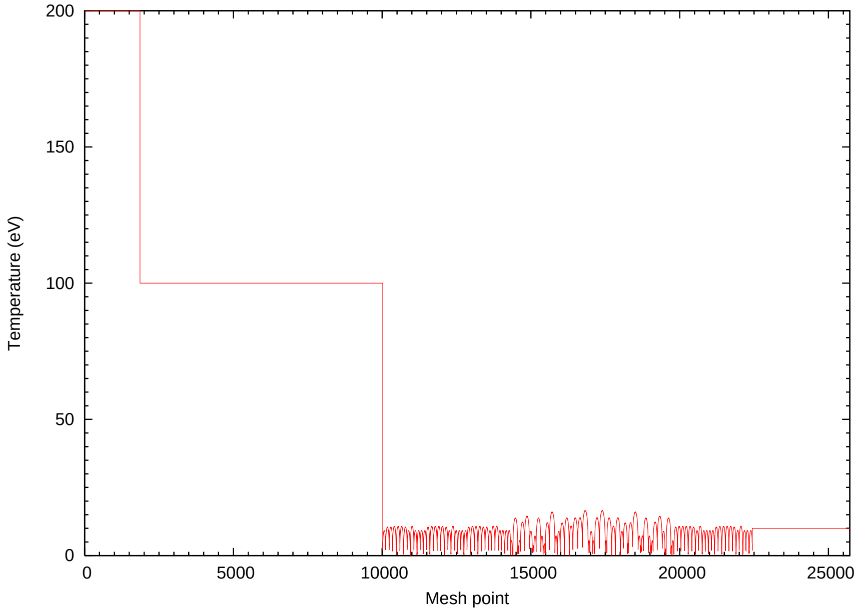

Figure 11.4: *Temperature solution on a W7-X mesh with the radial diffusivity* $\chi_\perp = 0$.

In the core region the innermost flux surface is held constant at 200 eV and there is no way for heat to travel across the flux surfaces, they are completely decoupled, and so the temperature remains at the initial condition of 100 eV and there are no visible effects from numerical diffusion. In the edge region (mesh points 10000 → 22000) the structure of the solution is due to numerical diffusion coming from the points on the outer boundary which are held constant at 10 eV.

Testing with localilized sources

Tests were also done with source terms in the plasma. A source can result from external heating methods such as Radio Frequency heating, Ion and Electron Cyclotron Resonance Heating (ICRH, ECRH) and Neutral Beam Injection (NBI). By comparison, nuclear reactions represent a very broad source because they can happen everywhere in the core. Sources can be simulated in the transport code by specifying a fixed temperature at a selected mesh point. It is then possible to see how the effect of the source term propagates in the mesh, toroidally and poloidally. Fig.11.5 shows a mesh generated in the core of W7-X.

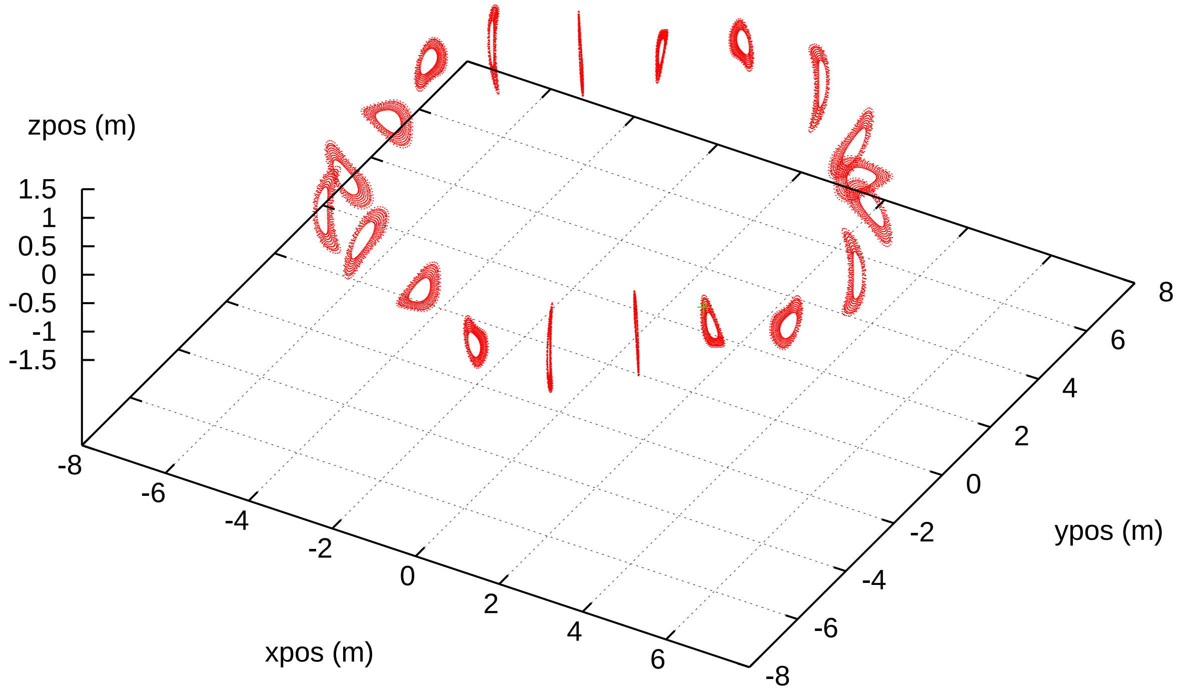

Figure 11.5: *A W7-X core mesh with 20 cuts. The mesh contains 9 flux surfaces in the core and no points in the edge. The outermost flux surface serves as an outer boundary. Total points:* 13180*. The point source is shown as a green point on one of the triangular-shaped cuts at lower right.*

We use a core mesh because it is useful to see the propagation effect of the source term along and across ordered flux surfaces in the plasma core. The field lines in the edge region are not required. Also, smaller meshes, such as the core mesh used here, run much faster than large edge meshes in the transport code.

Fig. 11.6 shows a Poincaré plot from the mesh, with the location of the point source in the upper left hand part of the middle flux surface.

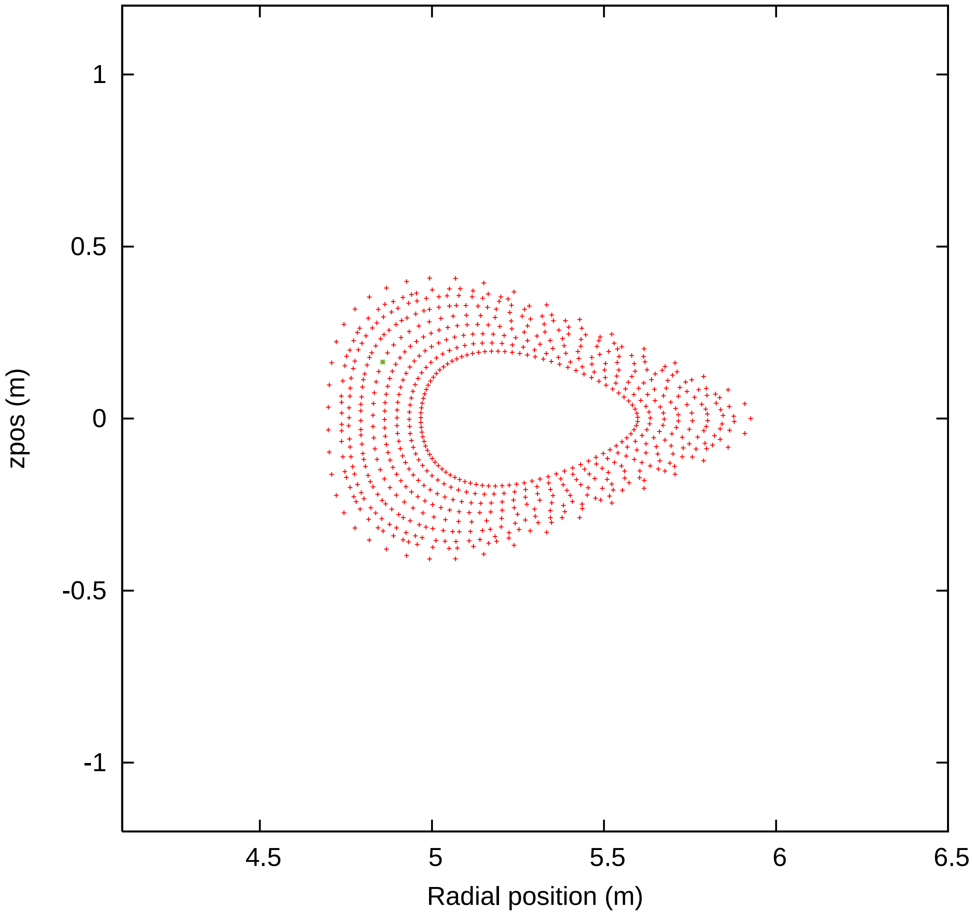

Figure 11.6: *A Poincaré plot, at* $\phi = 324°$*, from the W7-X core mesh. The 4 outermost flux surfaces in this mesh are the same core flux surfaces used in the W7-X edge mesh in Fig.11.1. The location of a point source is shown in green.*

First, Fig.11.7 shows a temperature solution on the mesh with no sources. Fig.11.8 shows a solution on the same mesh but with a point source set to 100 eV. The source appears as a spike in the middle of the 5th flux surface. The source causes fluctuations along the same flux surface and on the neighboring flux surfaces. The outer flux surfaces (lower right) are more strongly affected than the inner flux surfaces (upper left) where the higher temperature and stronger parallel transport damps out the effects of the source. The boundary conditions were 200 eV on the innermost flux surface, and 10 eV on the outermost flux surface. $n_e = 1.0e20m^{-3}$, $\chi_{\perp} = 1.0m^2s^{-1}$.

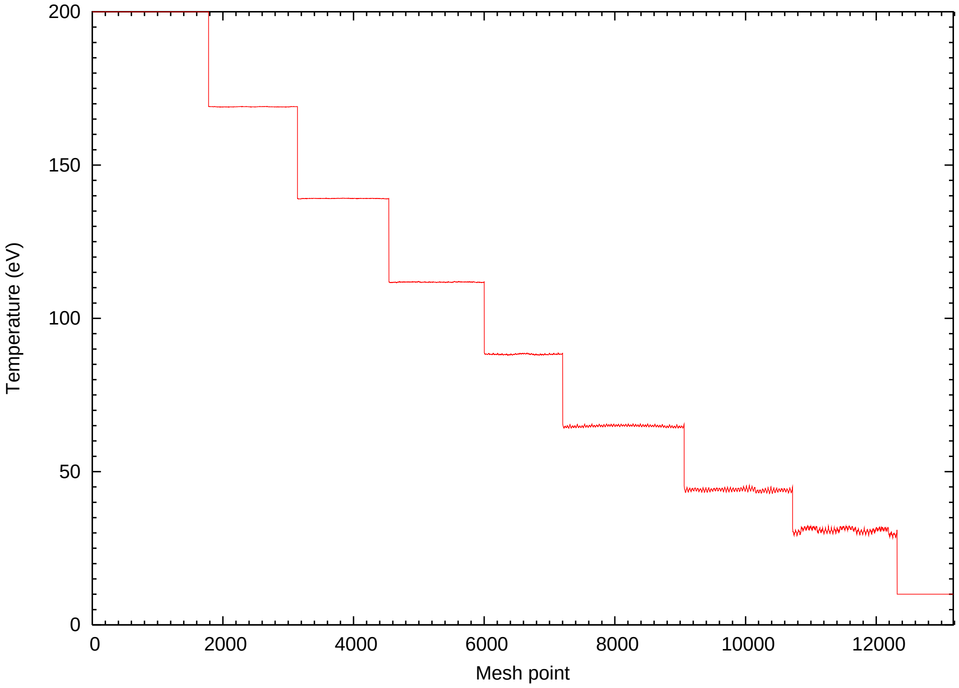

Figure 11.7: *Temperature solution on a W7-X core mesh. No point sources.*

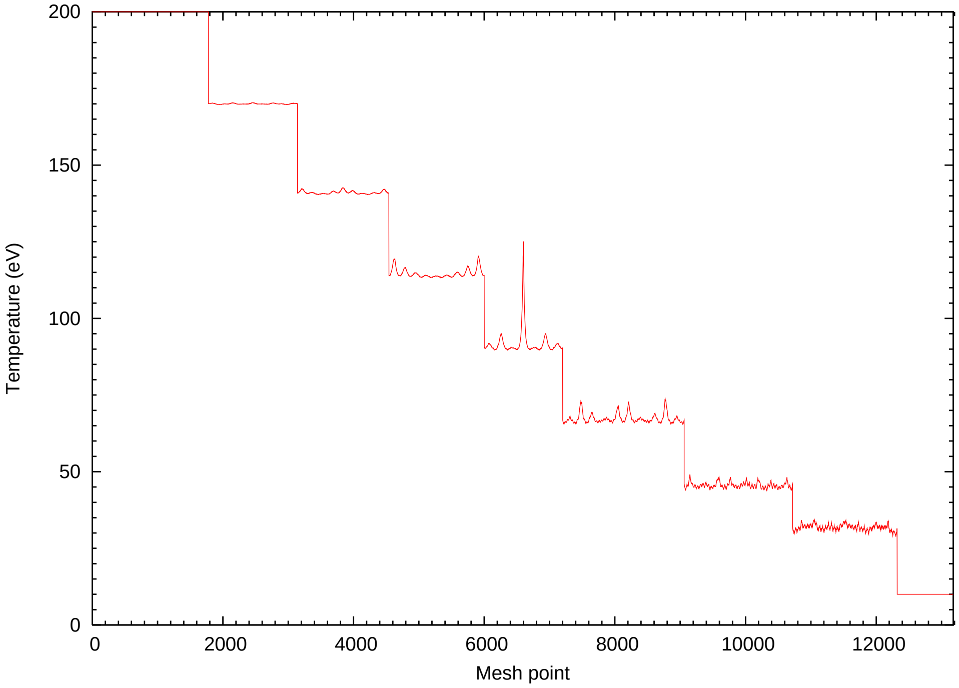

Figure 11.8: *Temperature solution on a W7-X core mesh, with a source term on the middle flux surface. The effect of the point source on the neighboring flux surfaces can be seen.*

11.1.2 Testing the sensitivity to the metrics

Finally, in this section, we test the sensitivity of the discretization scheme to the choice of metrics. That is, we compare temperature solutions in which we used a full metric (all elements of g^{ij} are non-zero), and a diagonal metric (only diagonal elements of g^{ij} are non-zero). Fig.11.9 shows 2 solutions super-imposed.

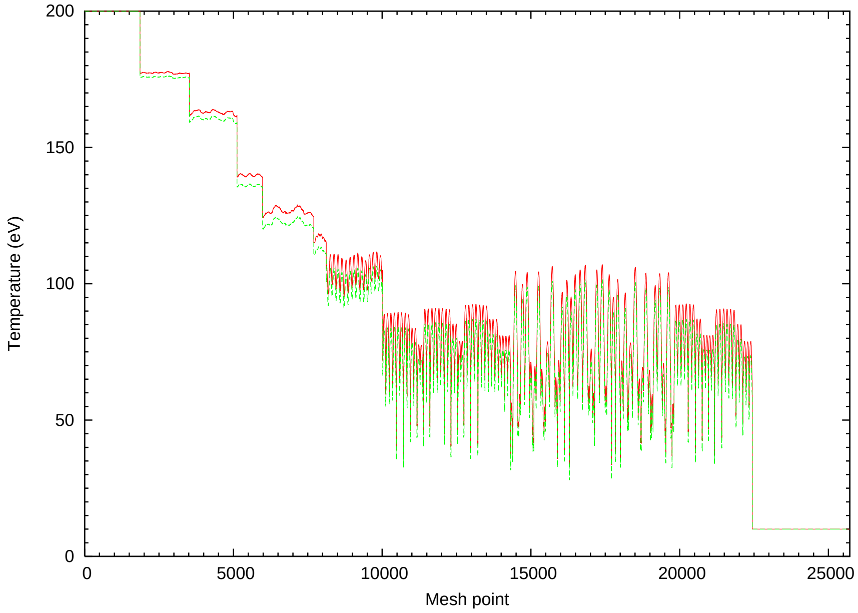

Figure 11.9: *Temperature solutions on the standard W7-X edge mesh (Fig.11.1). The red curve is for the diagonal metric, and the green curve is for the full metric. Boundary conditions were 200 eV in the core, 10 eV at the outer boundary, sheath condition at the ends of the open field lines, and* $\chi_\perp = 1.0 m^2 s^{-1}$.

The average percentage difference between corresponding points in the two solutions was ~4%. When radial diffusivity is reduced by a factor of 10, and further, the average difference is ~10%. The reason for this is the particular geometry of W7-X where the field line pitch (the ratio of poloidal and total magnetic field) is very low, ~1/500 (for comparison, a typical tokamak value is ~1/10) and therefore the effect of the off-diagonal metric elements representing a situation which is close to local orthogonality is extremely small. In terms of field line trajectories, this means that the field lines intersect the Poincaré cuts at almost right angles, and therefore the local coordinate system at any mesh point is close to orthogonal.

11.2 Physics

In this section we discuss the effects of ergodicity on edge transport in the W7-X and the NCSX experiments.

11.2.1 W7-X

Studies were done on W7-X meshes generated in a vacuum field, where there is small ergodicity, and in a finite beta field where there is large ergodicity. Studies were also done to investigate the power loading on the target plates.

Solutions in the vacuum magnetic field

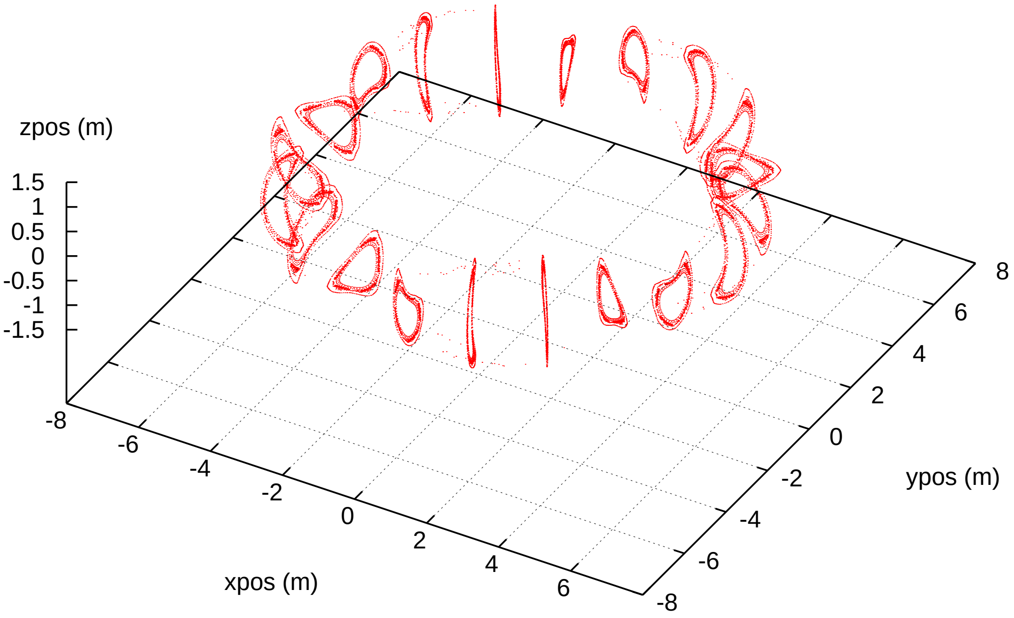

Figure 11.10: *A W7-X edge mesh generated in a vacuum field. This mesh has 20 cuts and contains points in the core and in the edge, plus an outer boundary. Total points:* 21205.

Fig.11.11 shows a Poincaré plot from the vacuum mesh. Note that the distribution of points has, in general, a smoother structure compared to the corresponding plot in the finite beta mesh (Fig.11.2). This is due to the relatively small ergodicity in the vacuum field. The separatrix x-point lies between the islands in the plasma edge. In 3D the x-point is a line of singularity where even small numerical errors due to interpolation can establish a shallow ergodic layer. The low ergodicity in the vacuum mesh means that there is relativiely little excursion of parallel transport in the radial direction, as compared to the finite beta case.

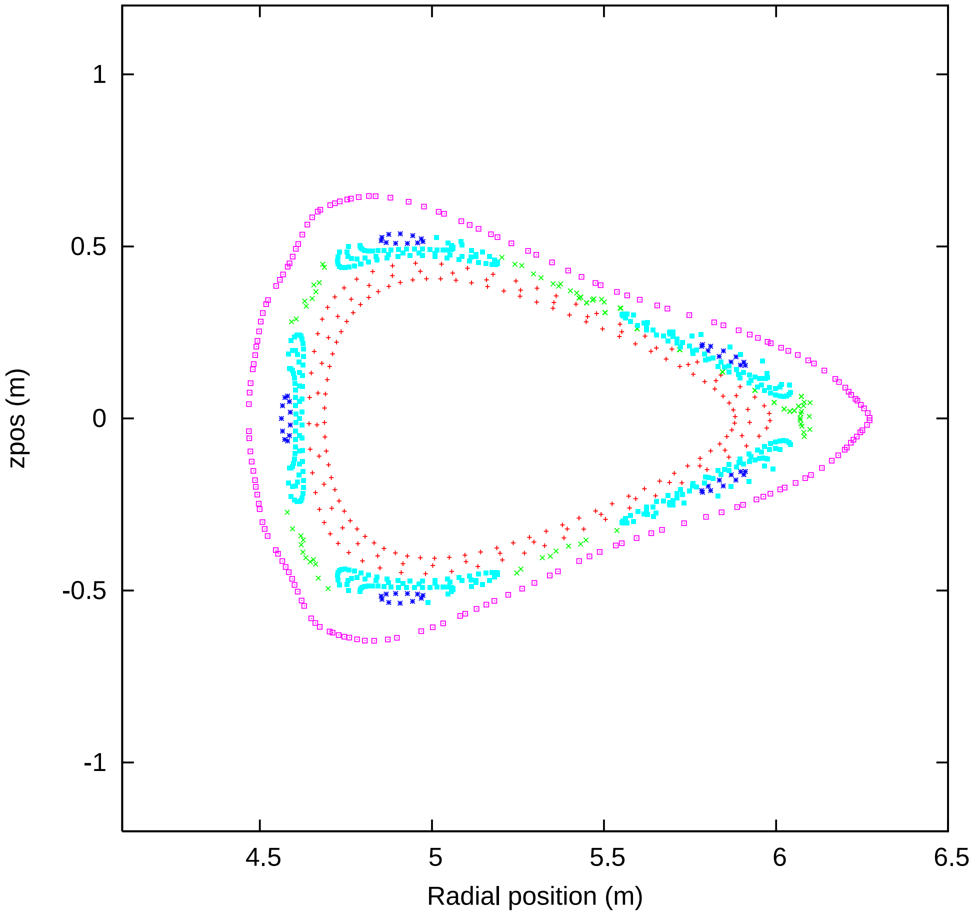

Figure 11.11: *Poincaré plot at* $\phi = 36^\circ$ *from the W7-X vacuum mesh. Red points lie on field lines which form closed flux surfaces in the plasma core. Green points lie on open field lines in the edge. Light blue points lie on field lines which form open flux surfaces in the islands. Dark blue points lie on field lines which form closed flux surfaces in the islands (one in each island). Pink points lie on a closed field line which forms an outer boundary. Total points on this plot* ~ 1050*.*

Fig.11.12 shows a temperature solution on the vacuum mesh. The light blue curves are the open flux surfaces in the islands. Within each island the solution shows an ordered temperature gradient across the flux surfaces. The dark blue curves are the closed flux surfaces in the islands (one in each island). The green curves are the open field lines elsewhere in the edge region. The 2 high green peaks come from 2 long open field lines which come close to the plasma core.

Also noteworthy are the field lines forming the outermost flux surfaces of the islands.

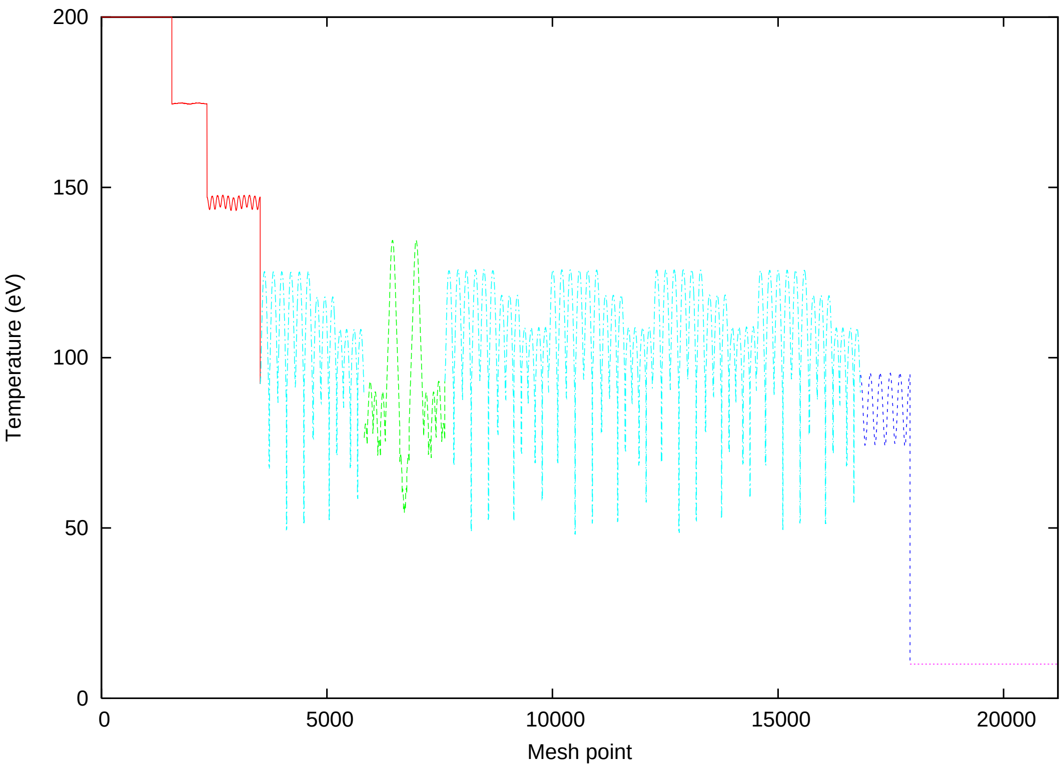

Figure 11.12: *Temperature solution on a W7-X vacuum mesh. Points on the innermost core flux surface are fixed at 200 eV. Points on the outer boundary are fixed at 10 eV. Points at the ends of the open field lines have a sheath boundary condition. The color scheme here corresponds to the color scheme in the Poincaré plot in Fig.11.2. Particle density is* $1.0e20m^{-3}$. *Radial diffusivity is* $1m^2s^{-1}$.

Inspection of the Poincaré plot in Fig.11.11 shows that they come quite close to the last closed flux surface in the core and in Fig.11.12 the temperature peaks of these field lines reach up to ∼125 eV. But on the plane of the Poincaré plot the ends of these flux surfaces are also in direct contact with the wall. Consequently, in the temperature solution the lowest points on these open field lines reach down to ∼50 eV. Therefore the outermost flux surface on each island acts as a direct bridge between the core and the wall. Note that the positioning of structures in the plot is due to the ordering of field lines in the mesh data file, and has no effect on the heat transport in the system.

Fig.11.13 shows the same information as in Fig.11.12 but as temperatures along field lines which have been normalized with respect to their own lengths. The red curves are the 3 core flux surfaces of which the outermost flux surface is the wavy line at ∼150 eV. Two green curves peak at ∼135 eV. These correspond to the 2 high green peaks in Fig.11.12.

The open flux surfaces in the islands contain many field lines of different lengths. The triple grouping of light blue contours which peak in the range 100→125 eV correspond to field lines on successive flux surfaces, eg the highest group (peaking at ∼125 eV) lie on the outermost flux surfaces of the islands. The flatter light blue contours in the range

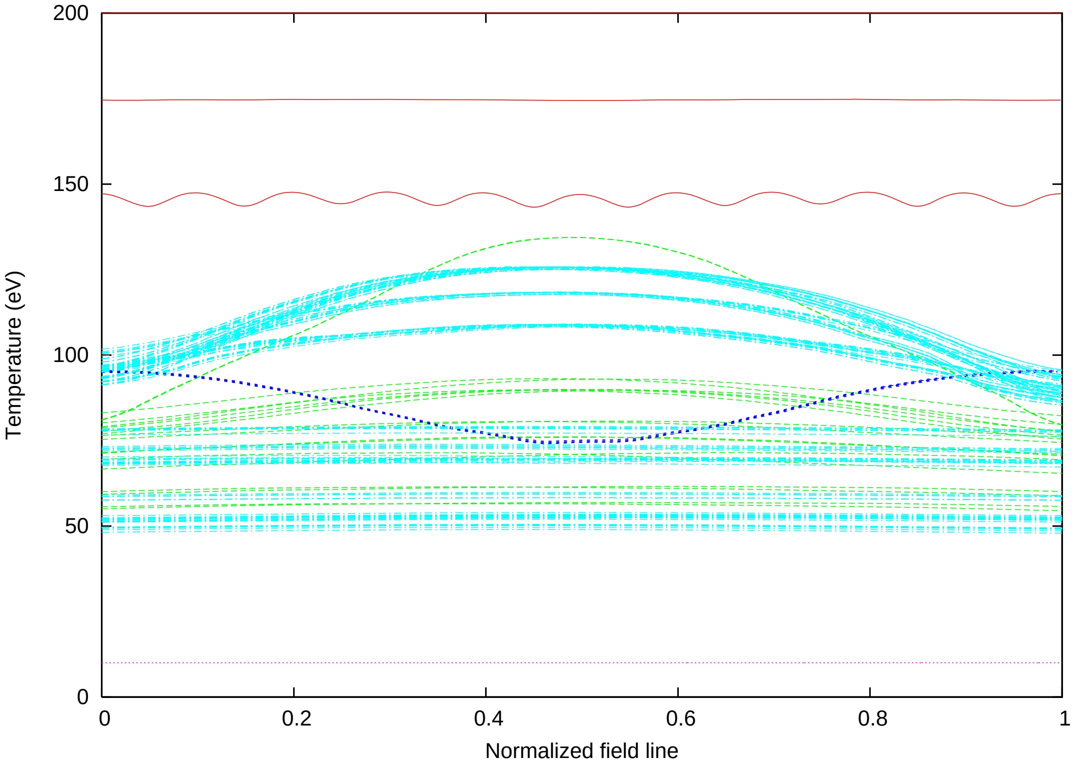

Figure 11.13: *Temperature contours along normalized field lines in W7-X vacuum field. The color scheme here corresponds to the color scheme in the Poincaré plot in Fig.11.11 and the temperature solution in Fig.11.12.*

50→80 eV are much shorter open field lines in the islands.

The dark blue contours are the 5 field lines forming which each form one closed flux surface in each island. The Poincaré plot in Fig.11.11 shows that the closed flux surface inside each island is exposed to the outer boundary which is held constant at 10 eV, and this is the cause of the dip.

Figs.11.14 and 11.15 show temperature solutions on 2 Poincaré plots, with each mesh point colored according to its steady state temperature.

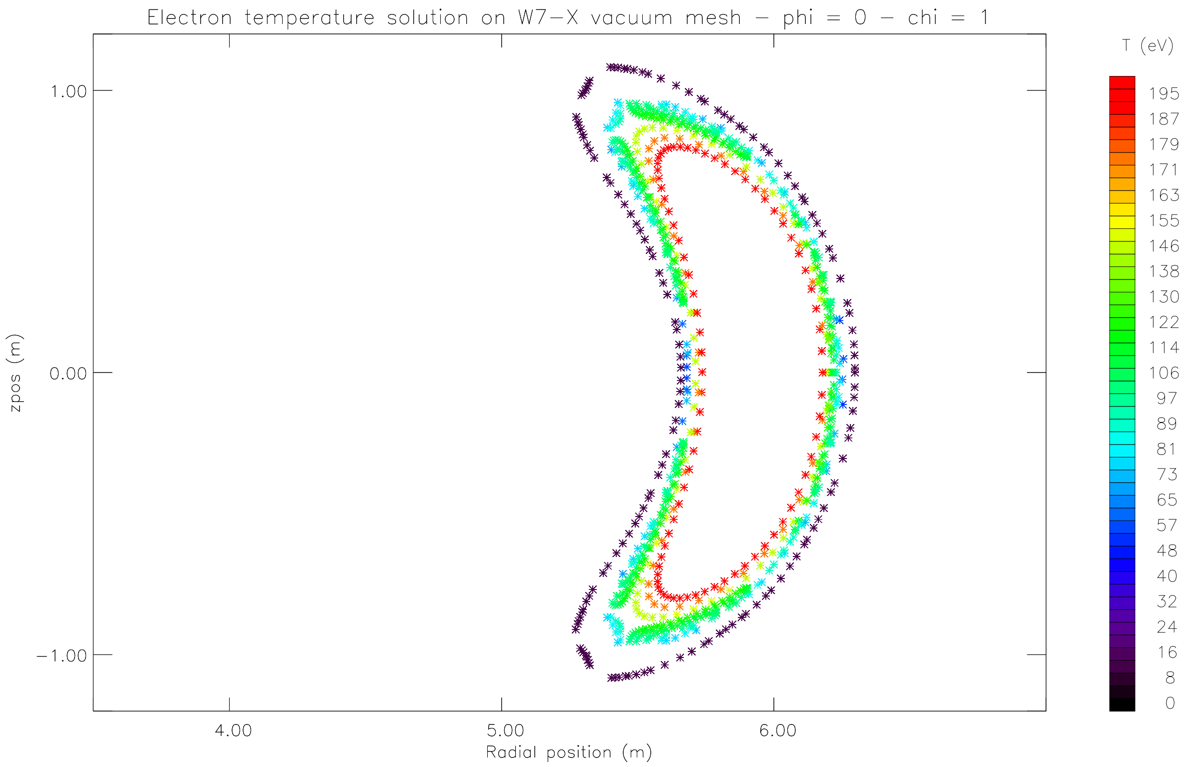

Figure 11.14: *Temperature solution on a vacuum mesh Poincaré plot at $\phi = 0°$.*

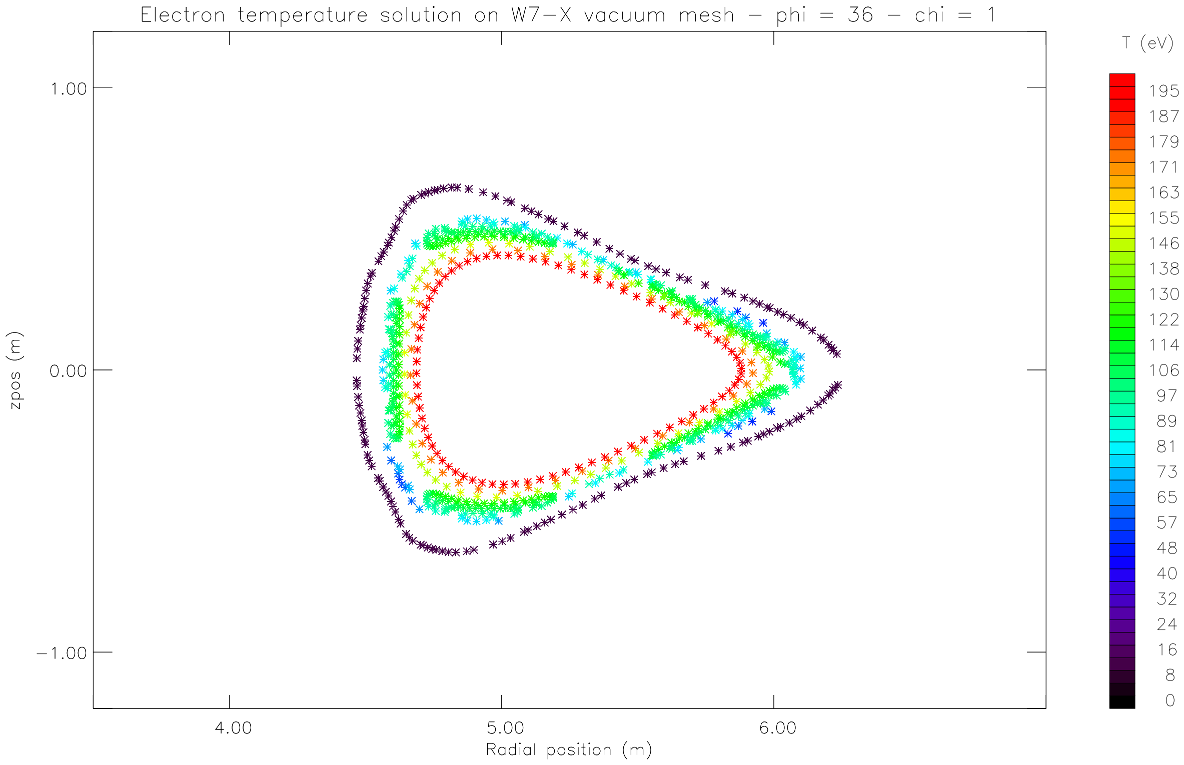

Figure 11.15: *Temperature solution on a vacuum mesh Poincaré plot at $\phi = 36°$.*

Solutions in the finite beta magnetic field

Next, studies were done for a finite beta case in which there are strong ergodic effects around the separatrix. Here we present solutions on a 20-cut mesh similar to that shown in Fig.11.1 but with greatly increased poloidal density in order to reveal more detail in the radial transport, that is, many more open field lines were traced in the edge region. A Poincaré plot is shown in Fig.11.16.

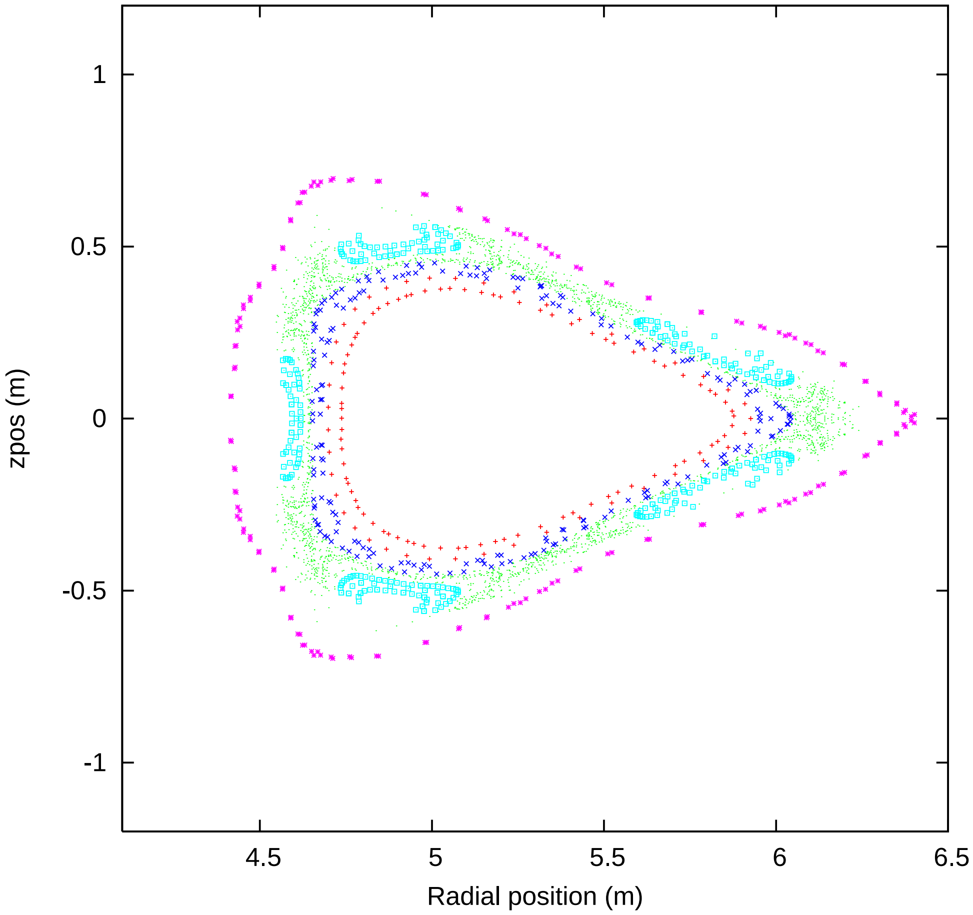

Figure 11.16: *Poincaré plot at* $\phi = 36°$ *from a W7-X finite beta mesh with enhanced poloidal density. Red points lie on closed flux surfaces in the plasma core. Dark blue points lie on closed ergodic field lines. Green points lie on open field lines in the edge. Light blue points lie on open flux surfaces in the islands. Pink points lie on a closed field line which forms an outer boundary. Total points on this plot* ~ 3400.

Compared to the vacuum mesh, the finite beta mesh has a coarser general structure due to the presence of the plasma (large ergodicity) and there is a thicker ergodic region between the islands and the core.

Fig.11.17 shows a temperature solution on the poloidally dense finite beta mesh.

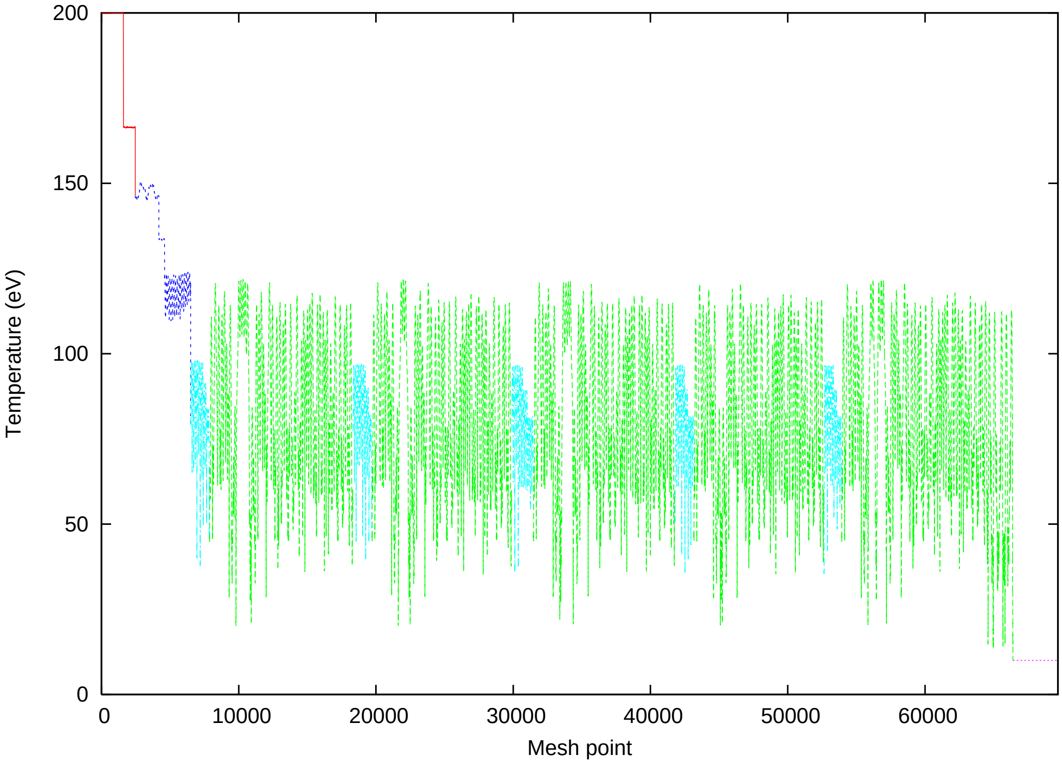

Figure 11.17: *Temperature solution on a W7-X finite beta mesh. Points on the innermost core flux surface are fixed at 200 eV. This corresponds to a core flux of* $\sim 8MW/m^2$*. Points on the outer boundary are fixed at 10 eV. Points on the ends of open field lines (green and light blue) have a sheath boundary condition. The color scheme here corresponds the color scheme used in the Poincare plot in Fig.11.16. Particle density,* $n_e = 1.0e20m^{-3}$*. Radial diffusivity,* $\chi_\perp = 1m^2s^{-1}$*.*

At the left hand side in Fig.11.17 the red and dark blue structures are the closed field lines in the core region. The green structures are open field lines in the edge region. Note that the peaks on these field lines reach up to the same temperature, ~120 eV, as the outermost closed ergodic field line (the lowest dark blue structure). This indicates enhanced transport from the core into the edge. The outermost flux surface in each island (light blue) peaks at ~95 eV. As before, the ordering of field lines in the solution does not affect the transport in the system.

Fig.11.18 shows the normalized temperatures along the field lines corresponding to the temperature solution in Fig.11.17. In the top half of the plot can be seen the closed field lines in the core region. The outermost closed ergodic field line surrounding the core is a blue wavy line at ~120 eV. Below this is the dense mass of open field lines in the edge region (green), and the islands (light blue). Most of these form roughly parabolic curves

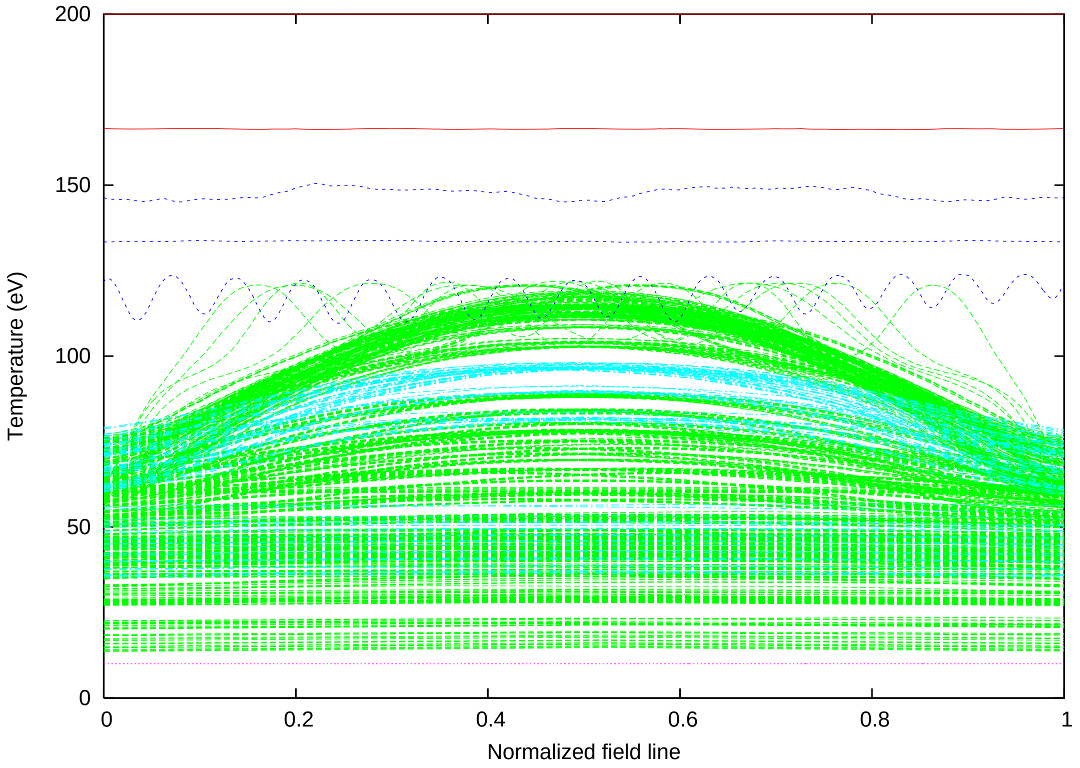

Figure 11.18: *Temperature contours along 788 normalized field lines in the W7-X poloidally dense finite beta mesh. The color scheme here corresponds to the color scheme in Figs.11.16 and 11.17 Particle density,* $n_e = 1.0e20m^{-3}$. *Radial diffusivity,* $\chi_\perp = 1m^2s^{-1}$.

but a few form oscillating curves because they are long enough to see the ergodicity in the system. Note that many of the temperature contours on these open field lines overlap the outermost closed ergodic field line.

Figs.11.19 and 11.20 show temperature solutions on Poincaré plots at $\phi = 0^\circ$ and $\phi = 36^\circ$. The mesh points are colored according to their steady state temperature as found by the transport code. Both have particle density $n_e = 1.0e20m^{-3}$, and radial diffusivity $\chi_\perp = 1.0m^2s^{-1}$. They show a smoothing of the temperature solution in the islands. These figures may be compared with Figs.11.14 and 11.15 from the vacuum case.

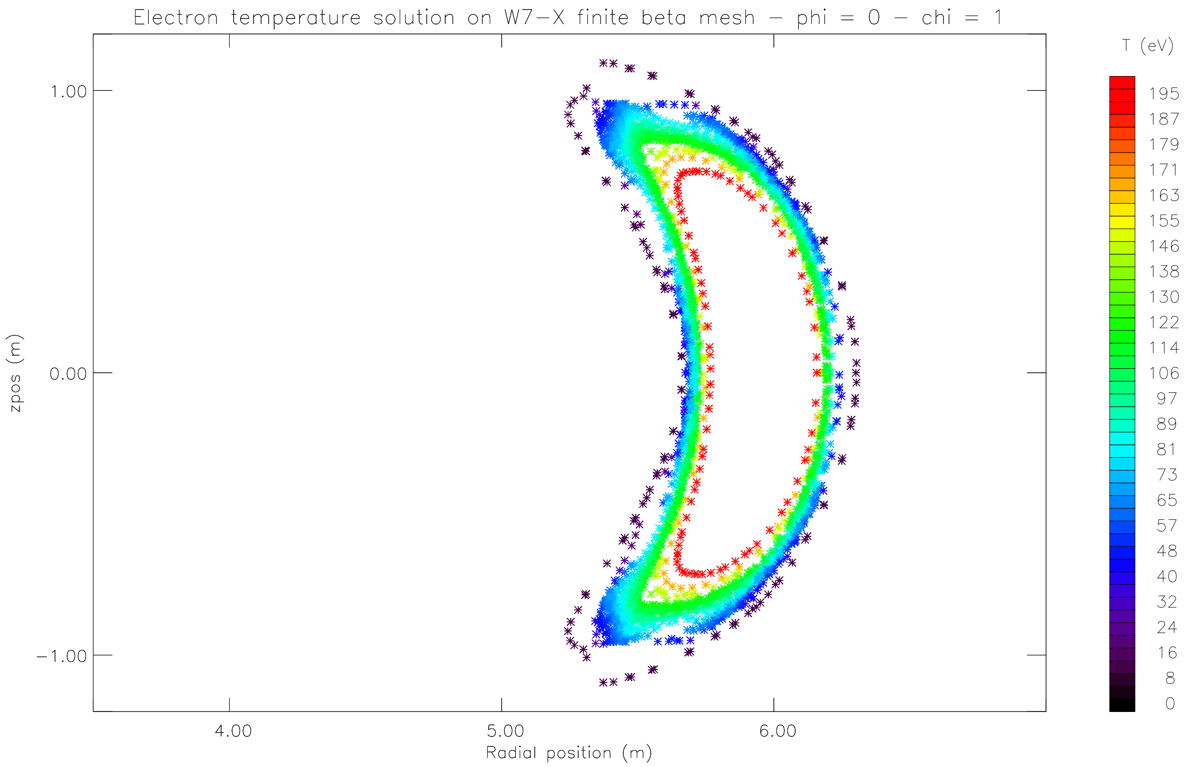

Figure 11.19: *Temperature solution on a finite beta mesh Poincaré plot at* $\phi = 0°$.

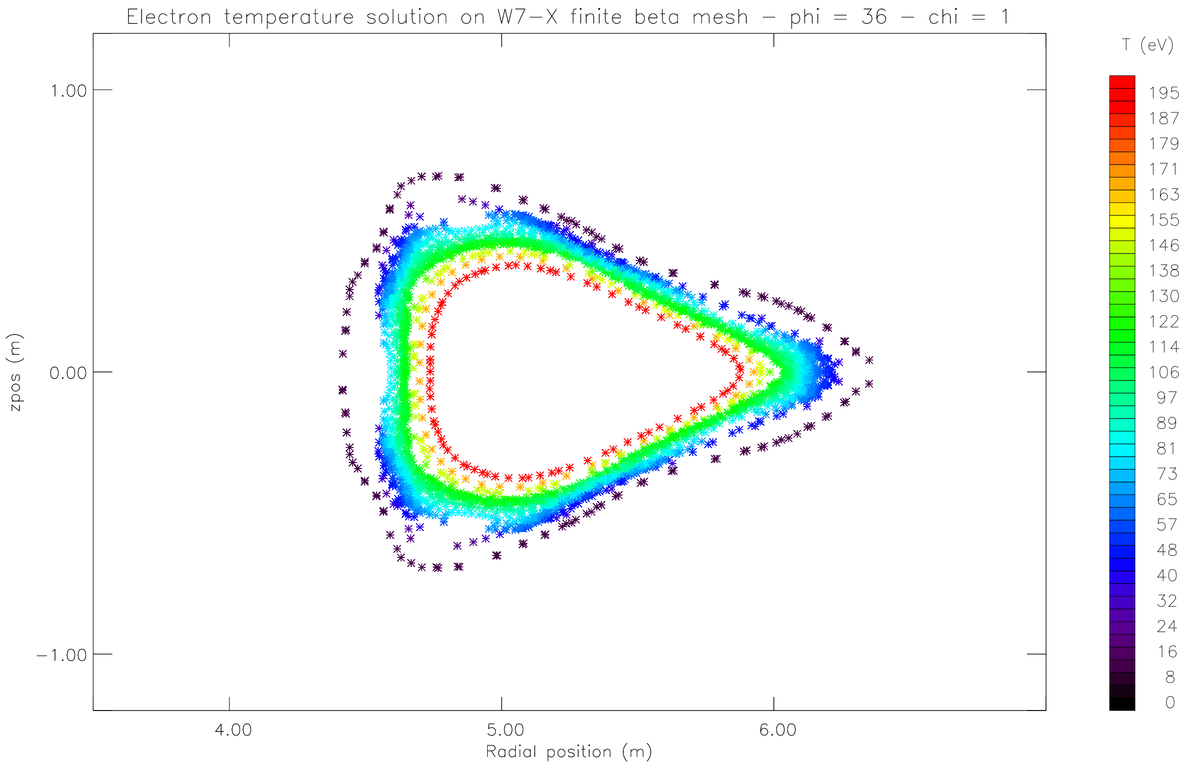

Figure 11.20: *Temperature solution on a finite beta mesh Poincaré plot at* $\phi = 36°$.

Comparison of vacuum and finite beta solutions

Fig.11.21 compares normalized temperature solutions on the W7-X vacuum and finite beta meshes. It is difficult to precisely compare these 2 meshes because they have different structures but the most notable difference is the gap, at ∼140 eV in the vacuum solution, between the closed core field lines and the open edge field lines. This represents a decoupling of transport from the core to the edge in the vacuum case. In the finite beta case there is considerable overlapping of the temperature contours of the open field lines with the outermost closed ergodic field line. This represents a feeding of the open field lines by contact with the ergodic layer, but in only temperature space, not in geometric space.

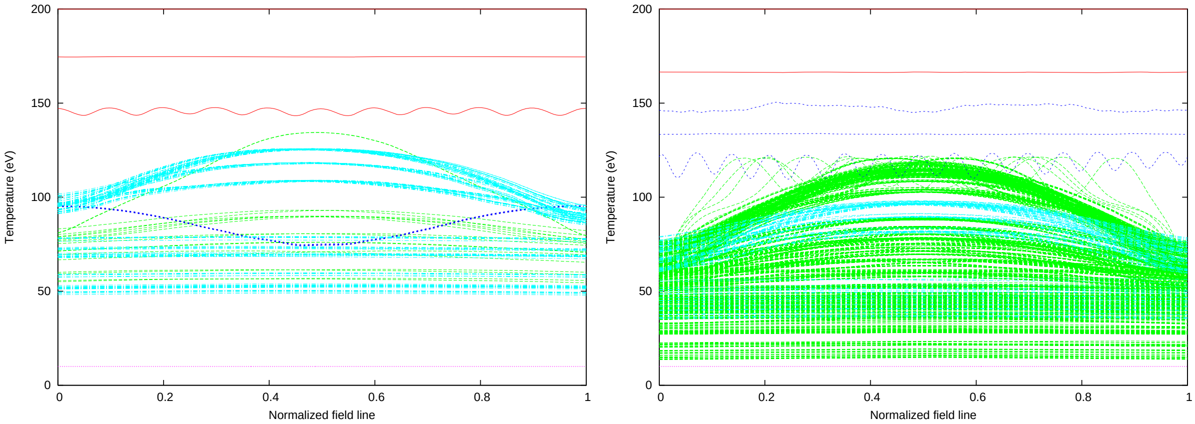

Figure 11.21: *(left) Solution on the vacuum mesh. This mesh has no closed ergodic field lines surrounding the plasma core, and there is a closed flux surface in the center of each island (dark blue curves) (right) Solution on the finite beta mesh. This mesh has 3 closed ergodic field lines surrounding the plasma core (dark blue curves), and no closed flux surfaces in the islands.*

Fig.11.22 shows close-up views of the Poincaré plots from the vacuum and finite beta meshes, with radial trajectories marked at equivalent positions extending from the innermost flux surface in the plasma core, directly to the outer boundary. At these positions the 2 meshes most closely resemble each other. Fig.11.23 shows the temperature profiles along these trajectories evaluated by interpolation from the temperatures at nearby mesh points. At the central part of both profiles there is a 'flattening' or broadening where the trajectory passes through an island. This broadening is much larger in the finite beta mesh due to the greater ergodicity. As discussed later, this is not a direct effect of ergodicity (parallel transport enhancing radial transport due to field line geometry.)

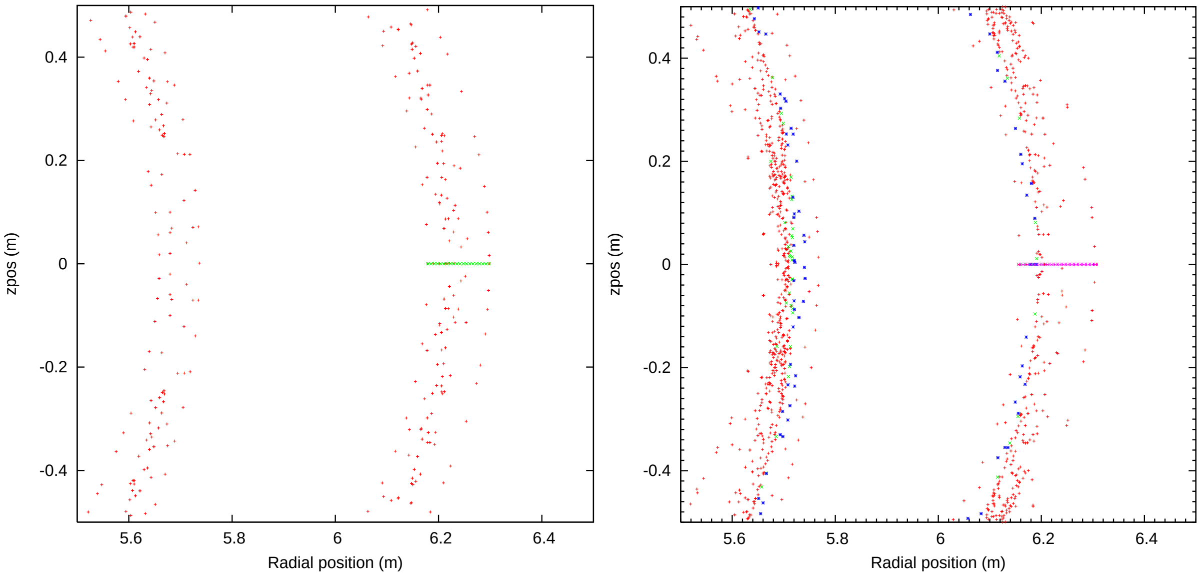

Figure 11.22: *(left) Position of radial trajectory (green line) on a Poincaré plot at $\phi = 0°$ from the vacuum mesh. (right) Position of radial trajectory (purple line) on a Poincaré plot at $\phi = 0°$ from the finite beta mesh. Blue points are mesh points on closed ergodic field lines which surround the plasma core.*

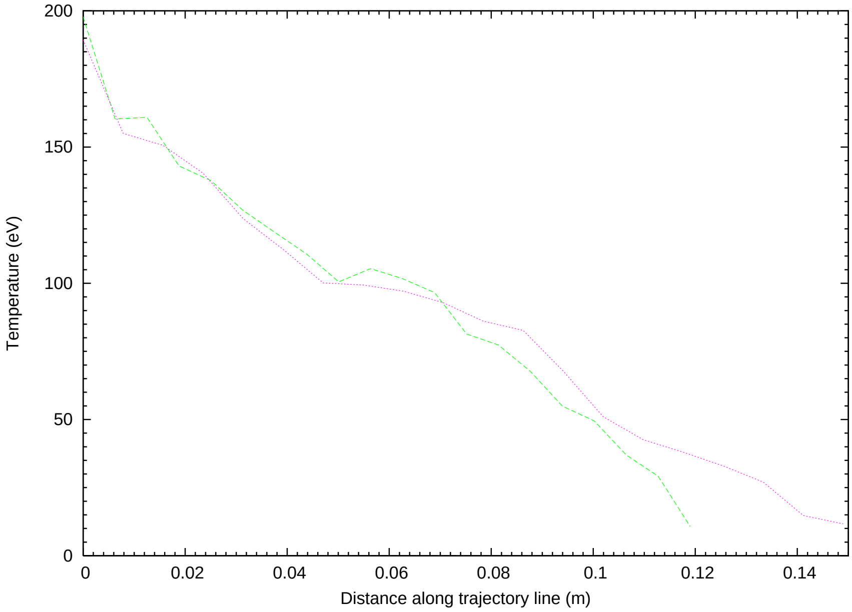

Figure 11.23: *Radial temperature profiles along the trajectories shown in Fig.11.22. The green curve is for the vacuum mesh. The pink curve is for the finite beta mesh. From the temperature solutions the temperature is known only at the mesh points, so at each of 20 positions on a trajectory the temperature is interpolated from the nearest 10 mesh points.*

Further analysis of transport in the plasma edge

In this section we continue the analysis of finite beta solutions in order to gain further insight into the mechanism of transport in the plasma edge.

Another parameter that can be varied in the transport code is the boundary condition in the core, that is, the temperature at the points on the innermost flux surface. Fig.11.24 shows 2 temperature solutions on the standard finite beta mesh introduced in Fig.11.1, and with the core temperature fixed at 2 different values. (This mesh has 4 flux surfaces in the plasma core, 3 closed ergodic field lines surrounding the core, but fewer field lines traced in the edge region. Therefore it runs faster in the transport code.)

Changing the core boundary condition represents a variation in the parallel transport ($\chi_{\parallel}$) but not in the radial transport ($\chi_{\perp}$). The result is a scaling of the temperature solutions, but the relative positions of the structures in the core and edge regions remain unchanged. That is, the solutions are self-similar.

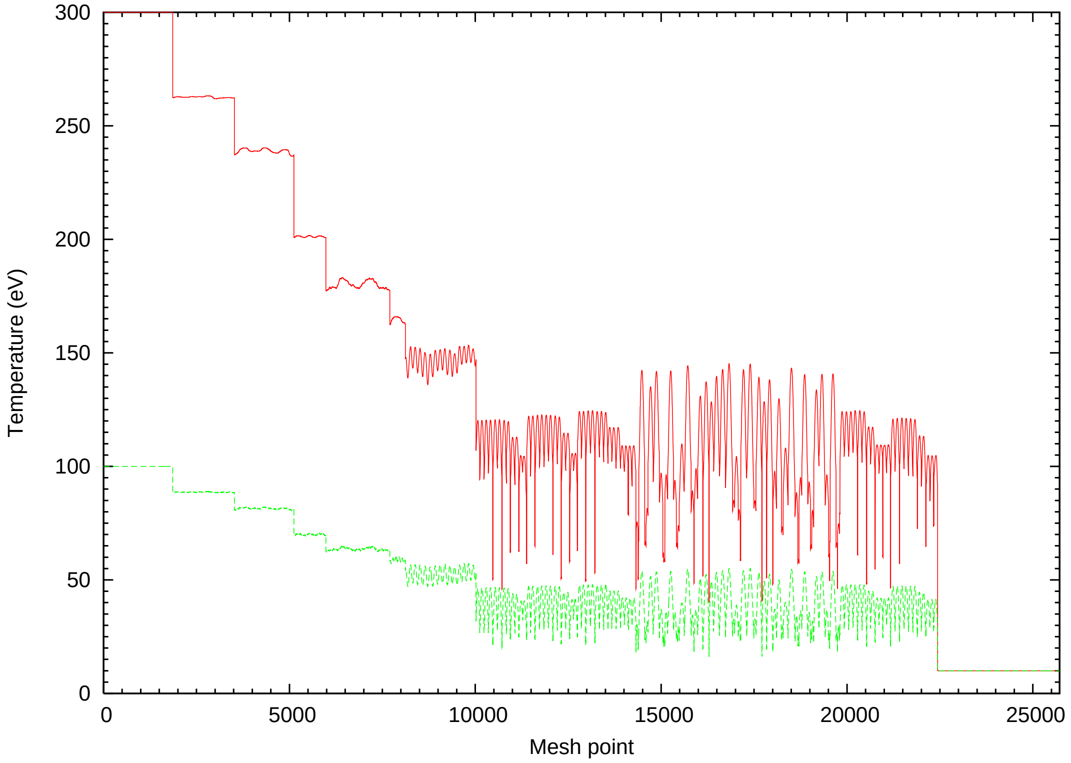

Figure 11.24: *Temperature solutions on the standard W7-X finite beta mesh. For the red curve the core boundary condition was 300 eV ($\sim 13MW/m^2$). For the green curve the core boundary condition was 100 eV ($\sim 4MW/m^2$). Particle density, $n_e = 1.0e20m^{-3}$. Radial diffusivity is $\chi_\perp = 1m^2s^{-1}$.*

Fig.11.25 shows temperatures along normalized field lines for the solutions in Fig.11.24. There are 157 field lines in this mesh of which 149 are open field lines in the edge region.

In the left hand plot the outermost closed ergodic field line is visible as a single wavy line at ∼150 eV. The wavy structure of this field line is a result of its physical trajectory as it makes excursions in the radial direction, moving closer and further away from the hot plasma core. This is characteristic of ergodic field lines because they do not define flux surfaces but fill up volumes.

The overlapping of this field line with the open field line contours, and thus the transport from the core to the edge, is not affected by the reduction in the core temperature, that is, there is still feeding of open field lines through the ergodic layer and broadening of transport in the edge region.

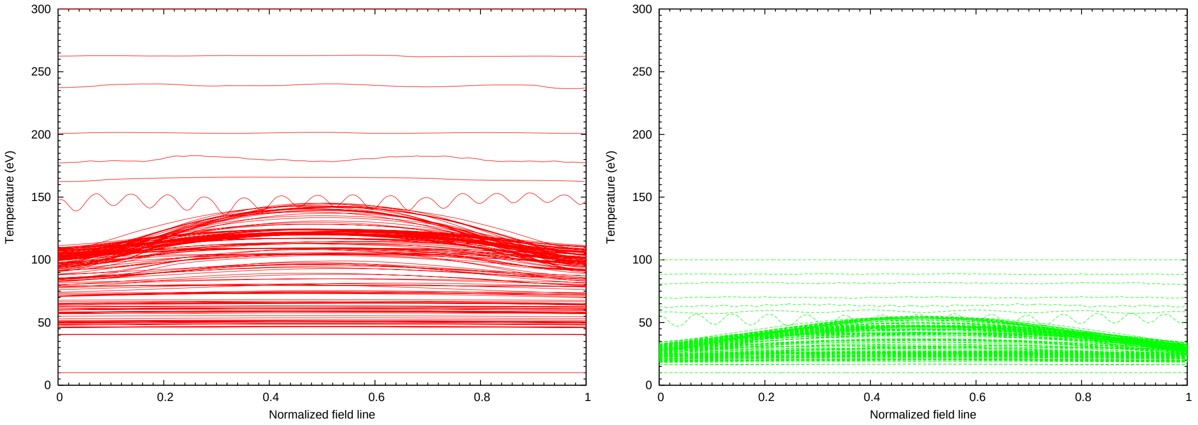

Figure 11.25: *Temperature contours along normalized field lines in the W7-X standard finite beta mesh (Fig.11.1). (left) Core temperature at 300 eV. (right) Core temperature at 100 eV. The solutions are self-similar and even when the core temperature condition is reduced there is still an overlapping of the closed and open field lines in temperature space.*

Figs.11.26 and 11.27 show a variation in the radial transport ($\chi_\perp$), in the standard W7-X finite beta mesh. The core boundary condition was held constant at 200 eV. These figures show that as the radial transport is reduced there is a progressive decoupling of the transport from the ergodic zone surrounding the core to the edge. This proves that the broadening effect is driven by radial transport and not by geometrical interweaving which would be unaffected by varying the transport.

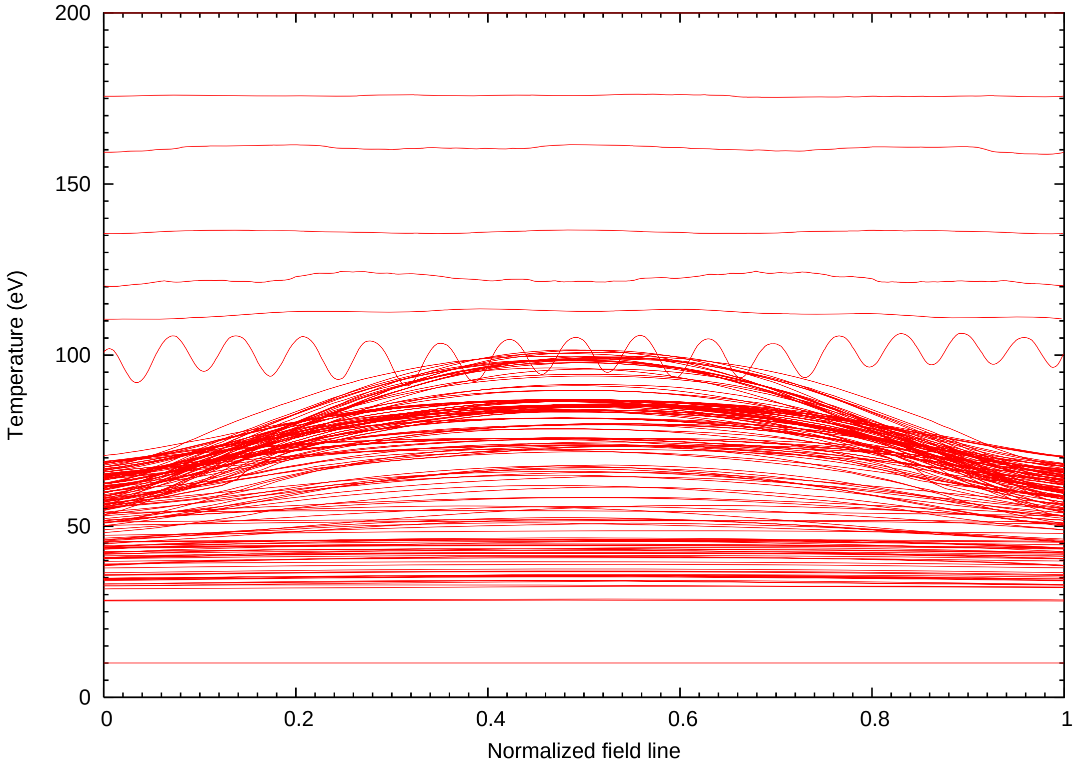

Figure 11.26: *Temperature solution on normalized field lines in the standard W7-X finite beta mesh, with radial diffusivity,* $\chi_\perp = 1.0 m^2 s^{-1}$.

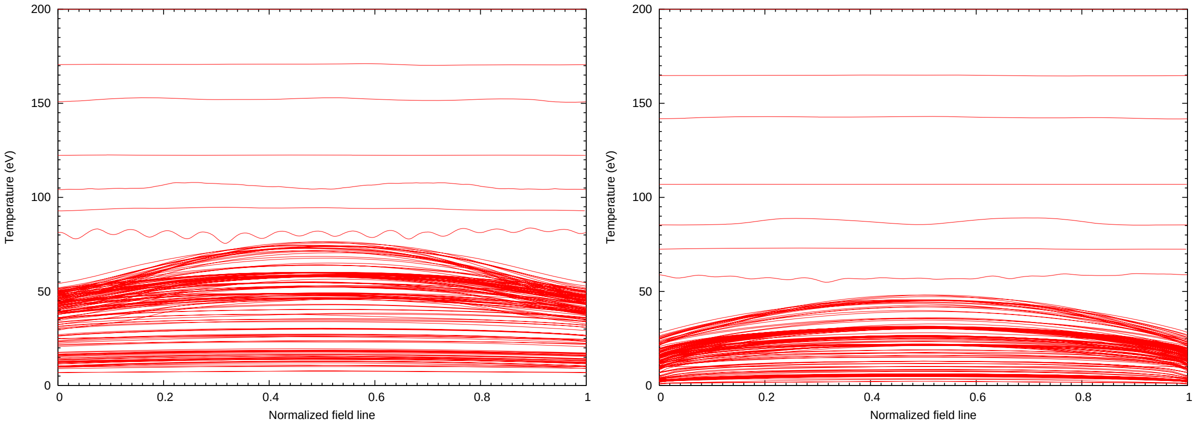

Figure 11.27: *(left) Solution with* $\chi_\perp = 0.1 m^2 s^{-1}$. *(right) Solution with* $\chi_\perp = 0.01 m^2 s^{-1}$.

Fig.11.28 shows temperatures along all 788 normalized field lines in the poloidally dense W7-X finite beta mesh. It is the same data as in Fig.11.18 but only the closed and open ergodic field lines have been highlighted in order to show the interaction between the core and the edge. Only those field lines which are much longer than the Kolmogorov length can 'see' the ergodicity in the system and these field lines exhibit oscillations. The closed field line at ~135 eV is almost flat because it is relatively short.

The general trend of open field lines is that for longer field lines the temperature profiles are more parabolic, and if a field line is long enough to 'see' the ergodicity then it is 'ergodic' and exhibits oscillations. Some of the red parabolic open field lines are in transition between a parabolic structure and a truly ergodic wavy structure.

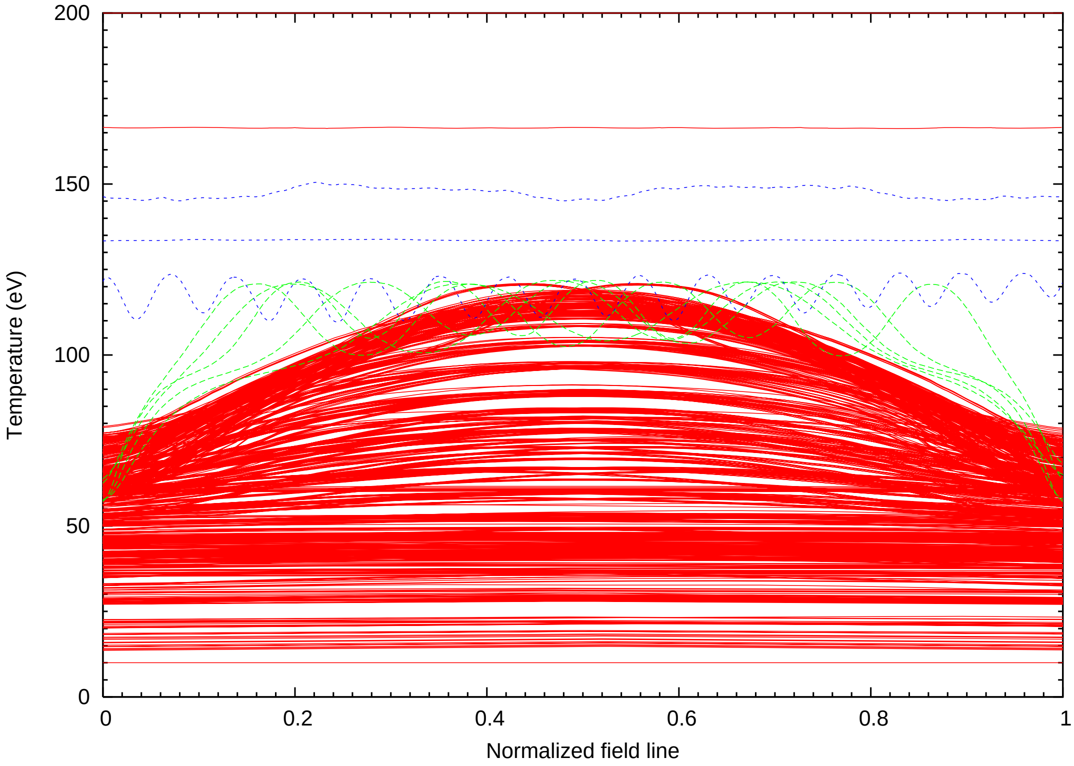

Figure 11.28: *Temperature solution on normalized field lines in the poloidally dense W7-X finite beta mesh. The 2 flat red curves at the top are closed field lines forming flux surfaces in the plasma core. Blue curves are closed field lines forming a thin ergodic region around the core. Green curves are long open ergodic field lines in the edge region. All other red curves are shorter open field lines in the edge and islands. At the bottom, also red, is the outer boundary held constant at 10 eV.*

Fig.11.29 shows the maximum temperature on open field lines in the same mesh. Field lines $\gtrsim$400 m are long enough to reach the same temperature, ~120 eV, as the outermost closed ergodic field line which surrounds the core.

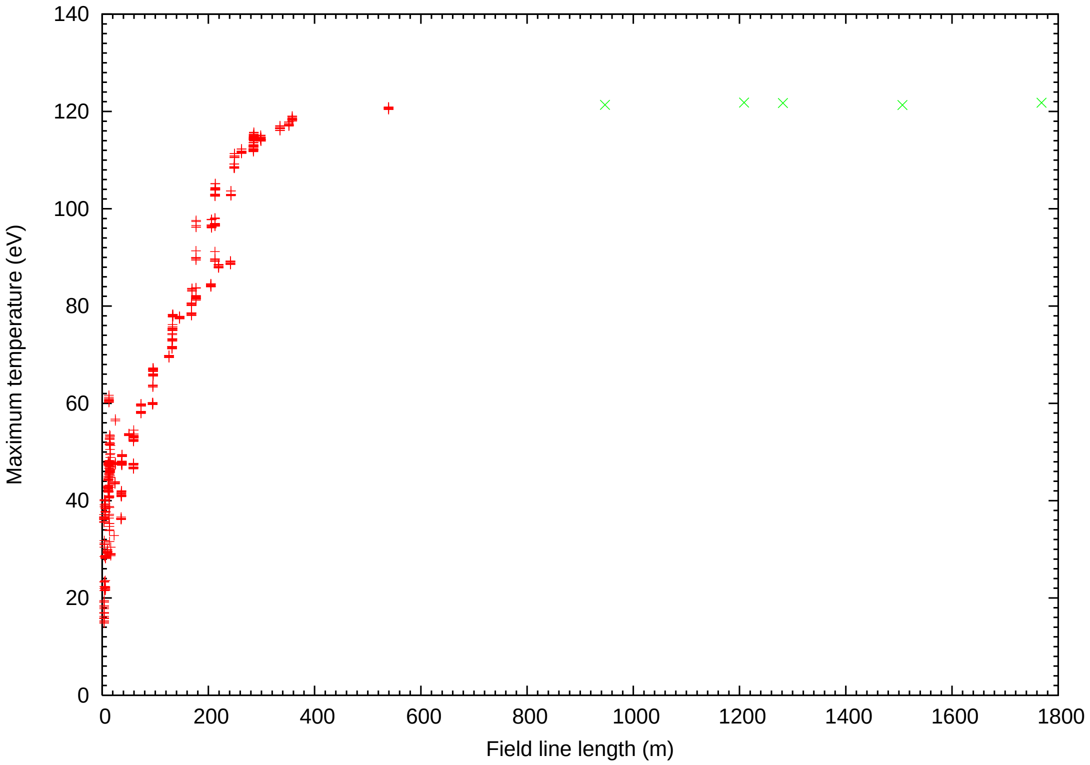

Figure 11.29: *Maximum temperature on open field lines in the poloidally dense W7-X finite beta mesh. Green points are on long open field lines which show oscillatory behavior as shown in Fig.11.28. Red points are on shorter open field lines, though some of these are still long enough to peak at ∼120 eV.*

Fig.11.30 shows a Poincaré plot at $\phi = 0°$ from the poloidally dense W7-X finite beta mesh (left) compared to the infinite solution. In the left hand plot, the closed and open ergodic field lines have been highlighted. The blue points, which lie on the closed ergodic field lines, reveal the thin ergodic layer which surrounds the plasma core. The green points reveal the position of the separatrix and, especially at the top and bottom, show an x-point like formation. These points lie on the long open field lines which show oscillatory behavior as shown in Fig.11.28.

The infinite solution is a Poincaré plot generated without taking into account intersection with any boundary structure. It is therefore an over-estimate of the ergodicity in the system because it neglects over-shadowing effects (laminar zone). In fact, one can see a smaller ergodic region in the bounded case (left).

The summary of these studies of the finite beta case is that the island structures in the edge region have been smoothed out. This broadening of the solution in the islands in the finite beta case is due to an indirect ergodic effect. The ergodicity triggers the cascading of energy into regions which were inaccessible in the vacuum case. Ergodicity does not directly enhance transport on short field lines because the vast majority of these field lines are much shorter than the Kolmogorov length.

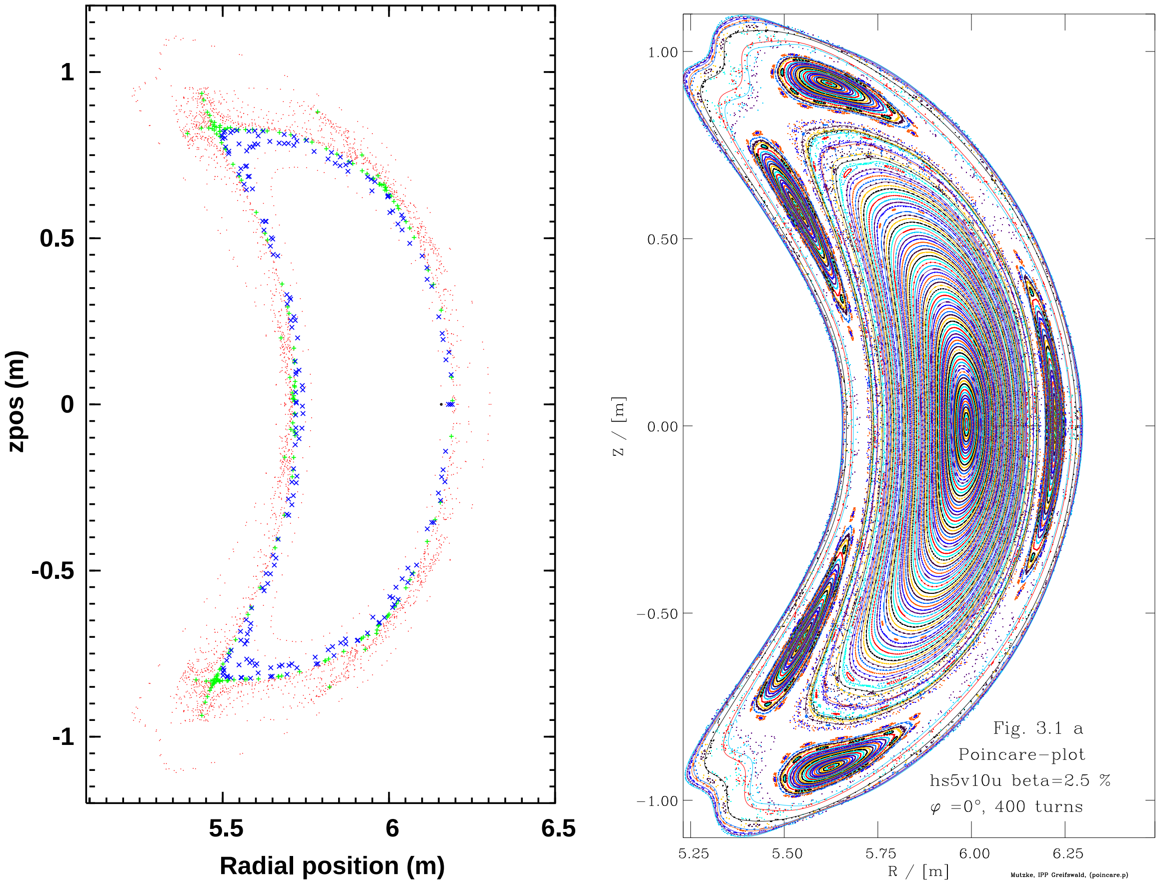

Figure 11.30: *(left) Poincaré plot at* $\phi = 0°$ *from the poloidally dense W7-X finite beta mesh. Blue points are on the closed ergodic field lines. Green points are on long open ergodic field lines in the edge. (right) Infinite field solution.*

Power loading on the target plates

Evaluating the heat flux distribution on the target plates in a fusion experiment is of great interest to designers of those devices, because the heat flux must be controlled and fall within engineering constraints. Fig.11.31 shows feeding fluxes at the ends of the open field lines in the W7-X poloidally dense finite beta mesh. These values may be compared with the simple estimate given in equation 2.1.

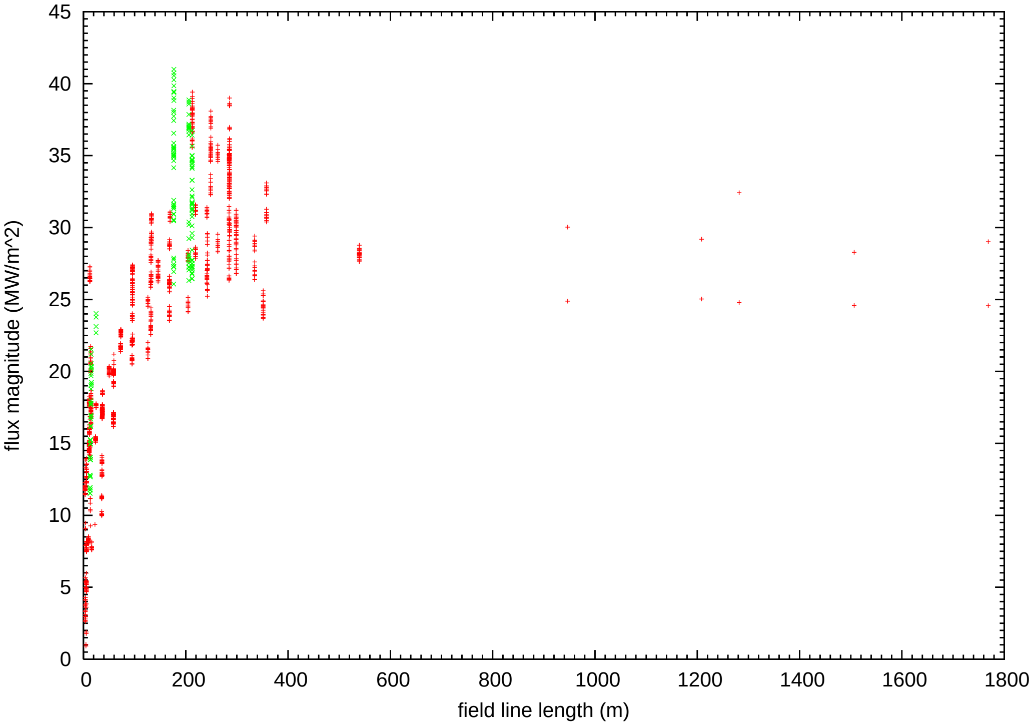

Figure 11.31: *Feeding flux densities on open field lines in the W7-X poloidally dense mesh. The red points are ends of open field lines in the edge region between the islands and the core. The green points are the ends of open field lines which form the open flux surfaces on the islands. No correction was made for the angle of intersection between the field lines and the target plates. Radial diffusivity,* $\chi_\perp = 1.0em^2s^{-1}$.

Each point in Fig.11.31 belongs to a field line of a known length and has a feeding flux density evaluated as:

$$Q_{feeding} \quad = \quad \kappa_\| \frac{T_2 - T_1}{dx} \tag{11.1}$$

where T_1 is the end point of an open field line which has intersected a target plate, and T_2 is the next point along the same field line at a distance dx. No correction is made for the angles of intersection between the field lines and the target plate. In general the longer open field lines have the greater feeding flux density but geometric factors (spatial position of field line trajectories) also play an important role in determing the flux on an open field line.

Fig.11.32 shows the same data as in Fig.11.31 but with the flux densities corrected by the intersection angles of the field lines. These values may be compared with the simple estimate given in equation 2.1.

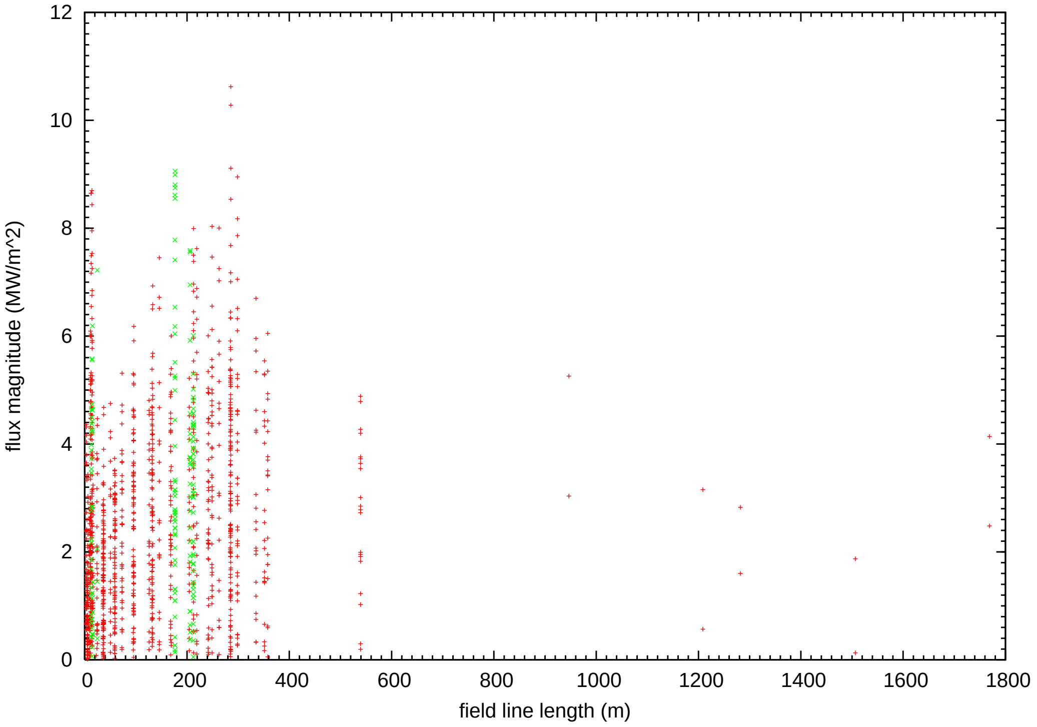

Figure 11.32: *Angled fluxes on open field lines in the W7-X poloidally dense mesh. The red points are ends of open field lines in the edge region between the islands and the core. The green points are the ends of open field lines which form the open flux surfaces on the islands. Radial diffusivity,* $\chi_\perp = 1.0em^2s^{-1}$.

The 'angle-corrected' flux densities are evaluated as:

$$Q_{angled} = \kappa_\parallel \frac{T_2 - T_1}{dx} sin(\theta) \tag{11.2}$$

where θ is the angle of intersection between an open field line and a target plate.

The points in Fig.11.32 lie grouped in columns because the open field lines are traced from similar sets of start points on similar cuts in the device, and the multiple symmetry of W7-X causes field line lengths to fall into groups.

Note that when the angle of intersection is included in the flux density calculations, the fluxes are greatly reduced. This is because the target plates are angled so as to reduce the power loading to within engineering limits ($\sim 15 MW/m^2$). The average angle of intersection is $\sim 7°$. The average flux on an open field line end point, allowing for intersection angle, is $\sim 2.6 MW/m^2$.

There are various possibilities for the driving mechanisms of edge transport. The first is an interweaving of the open field lines within in the edge region. A long open field line in the edge region between the islands and the core can approach closely to the core at some points and then closely to islands at other points, thus acting as a transport bridge between the core and islands. The second possibility is the ergodic effect in the edge by which the temperature profile is broadened by a mixing of parallel and radial transport processes. This was demonstrated by varying the radial diffusivity and the amount by which the strong parallel transport dominated the radial transport. The third possibility was trivial radial transport (effects scaling with the radial transport)

Fig.11.33 shows a complete layout of one possible design for the target plates in W7-X. Fig.11.34 shows the distribution of open field line impact points on a subset of divertor plates (see [45]).

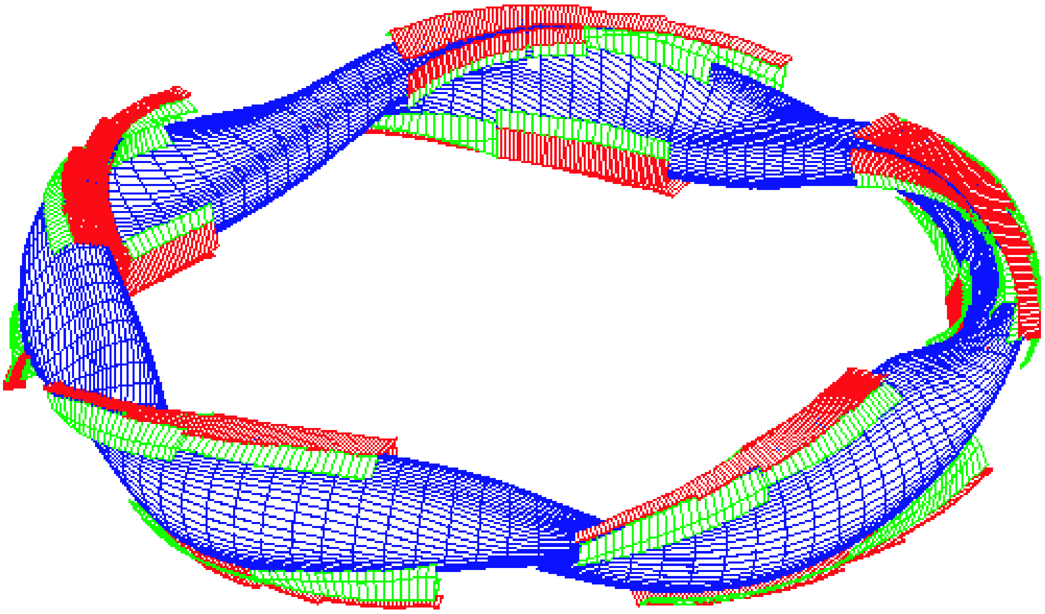

Figure 11.33: *Divertor plates (red) and baffles (green) in W7-X.*

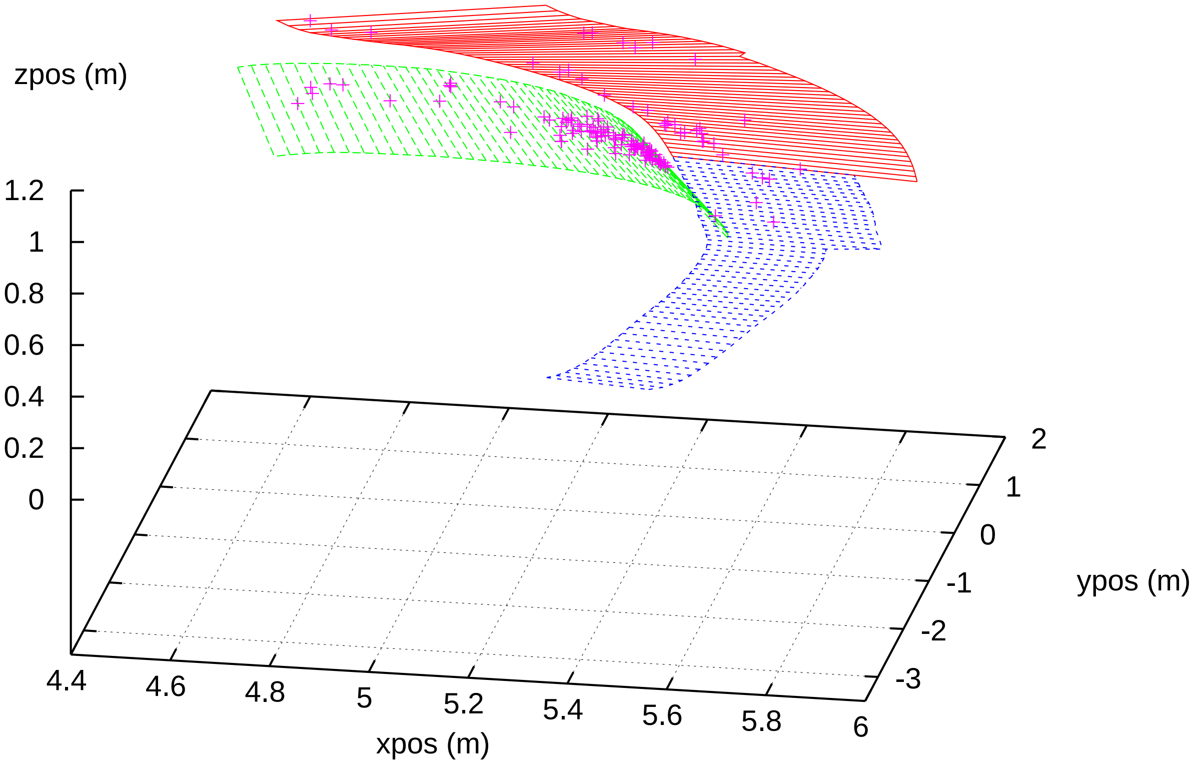

Figure 11.34: *Open field line impact points on a set of 2 horizontal divertor plates (red and blue) and a vertical divertor plate (green).*

11.2.2 NCSX

Experiments were done on an NCSX mesh generated in a finite beta field to demonstrate the applicability of the transport code to other devices.

Solutions in the finite beta magnetic field

In NCSX [46] the plasma major radius is $\sim 2m$ (compared to $\sim 5.5m$ in W7-X) and the magnetic field has a 3-fold symmetry (compared to 5-fold in W7-X). Also, no target plates were considered in these experiments. Fig.11.35 shows a mesh generated in a finite beta field for NCSX.

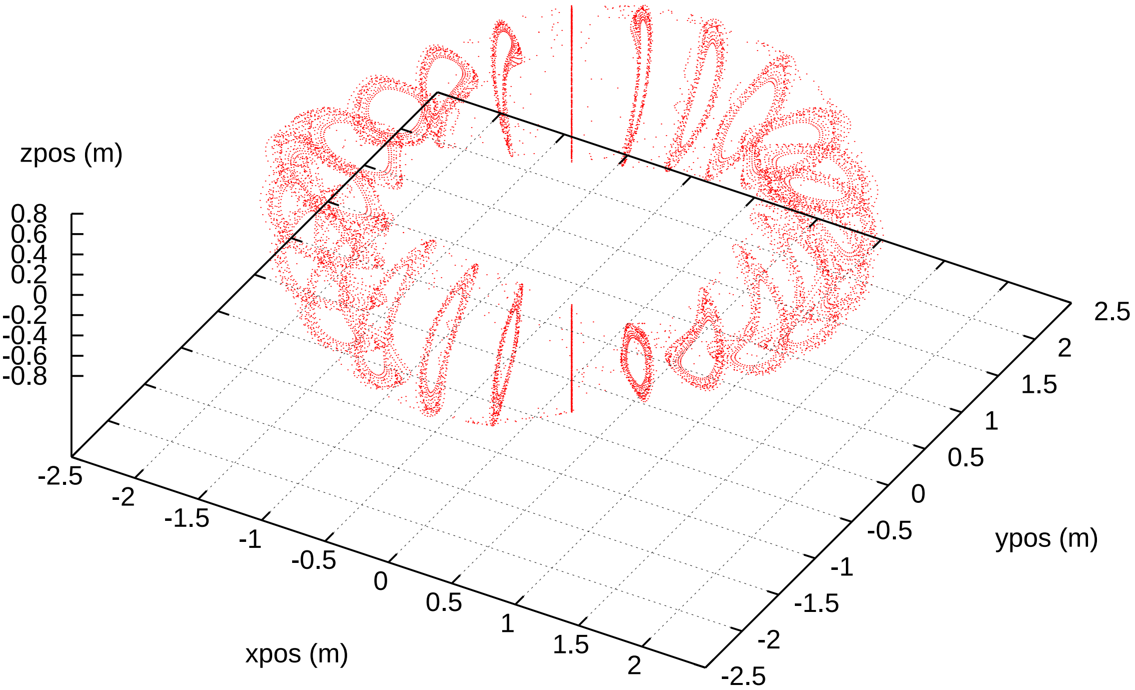

Figure 11.35: *An NCSX edge mesh generated in a finite beta field. This mesh has 24 cuts and contains points in the core and in the edge, plus an outer boundary. Total field lines; 624. Total points:* 16668*.*

The Poincaré plot in Fig.11.36 shows the different types of field lines which make up the NCSX mesh. Unlike in the W7-X meshes, there are no large islands structures in the edge region, and the points on the outer boundary represent the actual position of one possible design of the vacuum vessel wall (rather than an approximation as in W7-X).

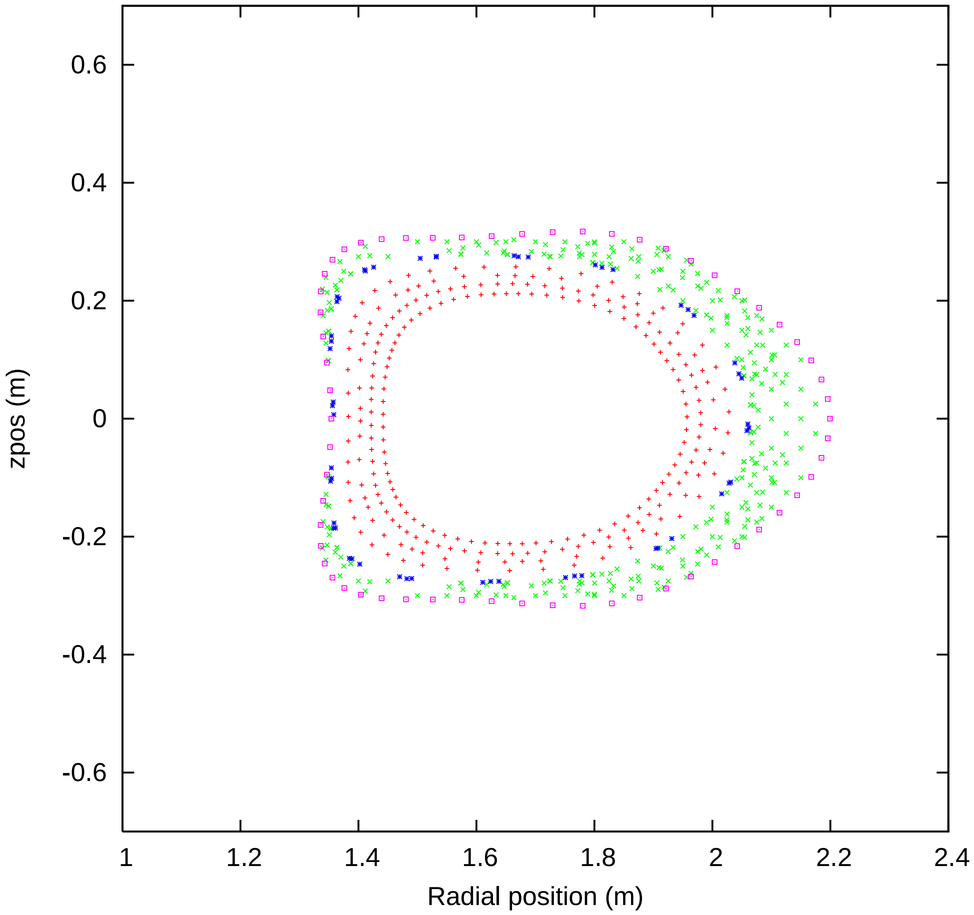

Figure 11.36: *Poincaré plot at* $\phi = 60^\circ$ *from the NCSX finite beta mesh. Red points lie on closed flux surfaces in the plasma core. Dark blue points lie on a closed ergodic field line. Green points lie on open field lines in the plasma edge. Pink points lie on a closed field line which represents the vacuum vessel wall. Total points on this plot* ~ 420.

Fig.11.37 shows a temperature solution on the NCSX mesh. A characteristic of the magnetic field in NCSX is that the open field lines in the edge region are generally shorter than those in W7-X. The closed ergodic field line (dark blue) has 3 large fluctuations along its length due to geometric effects. There is also a notable fluctuation along the 3rd flux surface ($\sim$ 125 eV). This is probably due to a relatively large change in the number of points from the 2nd to the 3rd flux surfaces.

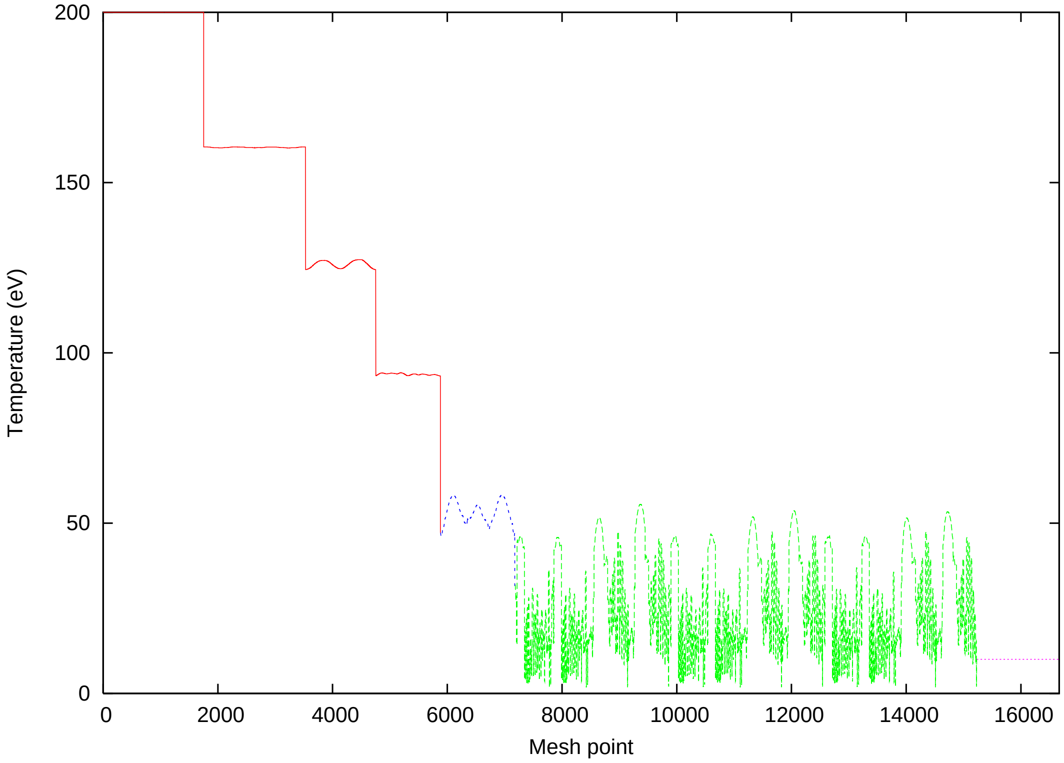

Figure 11.37: *Temperature solution on an NCSX mesh. The color scheme here corresponds to the color scheme in the Poincaré plot in Fig.11.36. Points on the innermost plasma core flux surface were held constant at 200 eV. Points on the wall were held constant at 10 eV. Points on the ends of open field lines had a sheath boundary condition. Particle density,* $n_e = 1.0e20m^{-3}$. *Radial diffusivity,* $\chi_\perp = 1m^2s^{-1}$.

In the edge region (green curves) there is a repeating pattern because the open field lines are traced from similar start points on similar cuts around the torus. As in the W7-X mesh, the ordering of field lines has no effect on the heat transport in the system. There are at least 6 high green peaks which reach up to the temperature of the closed ergodic field line ($\sim$ 50 eV).

Fig.11.38 shows the temperatures along the normalized field lines. At the bottom of the plot the large number of open field lines (green) are flattened due to their short length but some have a parabolic shape and overlap the closed ergodic field line (dark blue). This interweaving occurs only in temperature space. There is no physical interweaving of the open field lines with the closed ergodic field line.

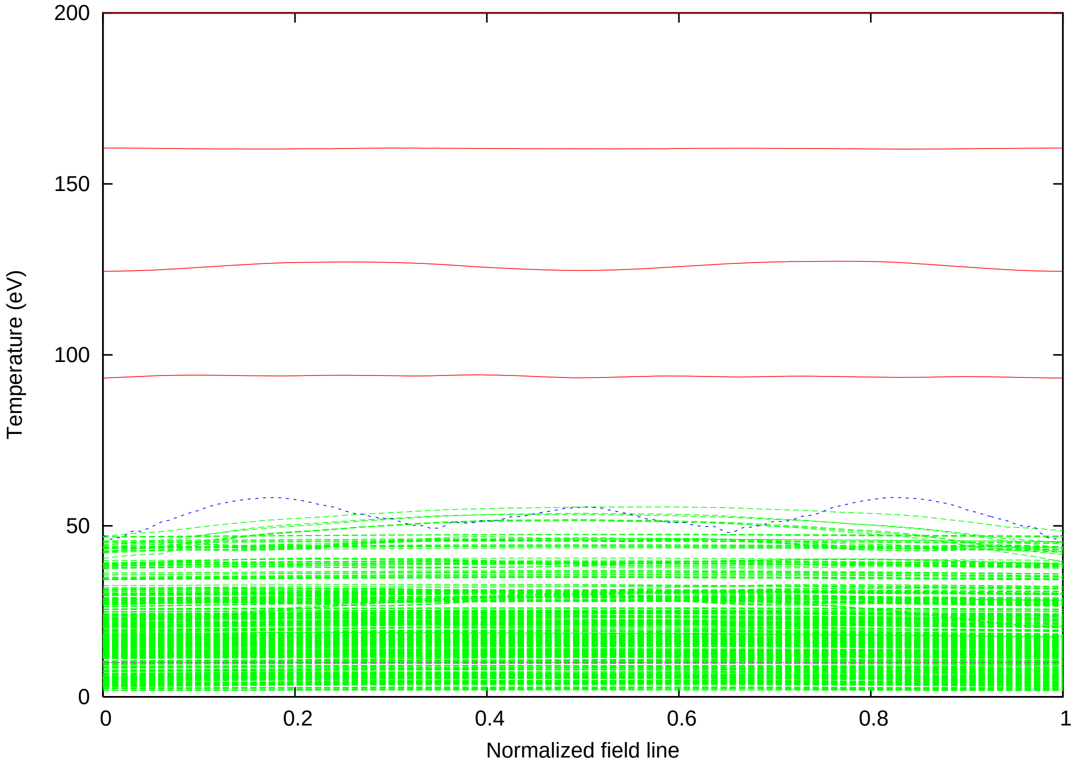

Figure 11.38: *Temperature along normalized field lines in an NCSX solution. The color scheme here corresponds the the color scheme in Figs.11.36 and 11.37. Particle density,* $n_e = 1.0e20m^{-3}$. *Radial diffusivity,* $\chi_\perp = 1m^2s^{-1}$.

Figs.11.39 and 11.40 show temperature solutions on 2 different Poincaré plots. The mesh points have been colored according to the steady state temperature as found by the transport code. They also show how the plasma cross-section changes with toroidal angle. Note that the purpose of these solutions is mainly to demonstrate the flexibility of the finite difference code developed in the present work. The magnetic field and the wall design in NCSX have undergone extensive modification since these results were generated.

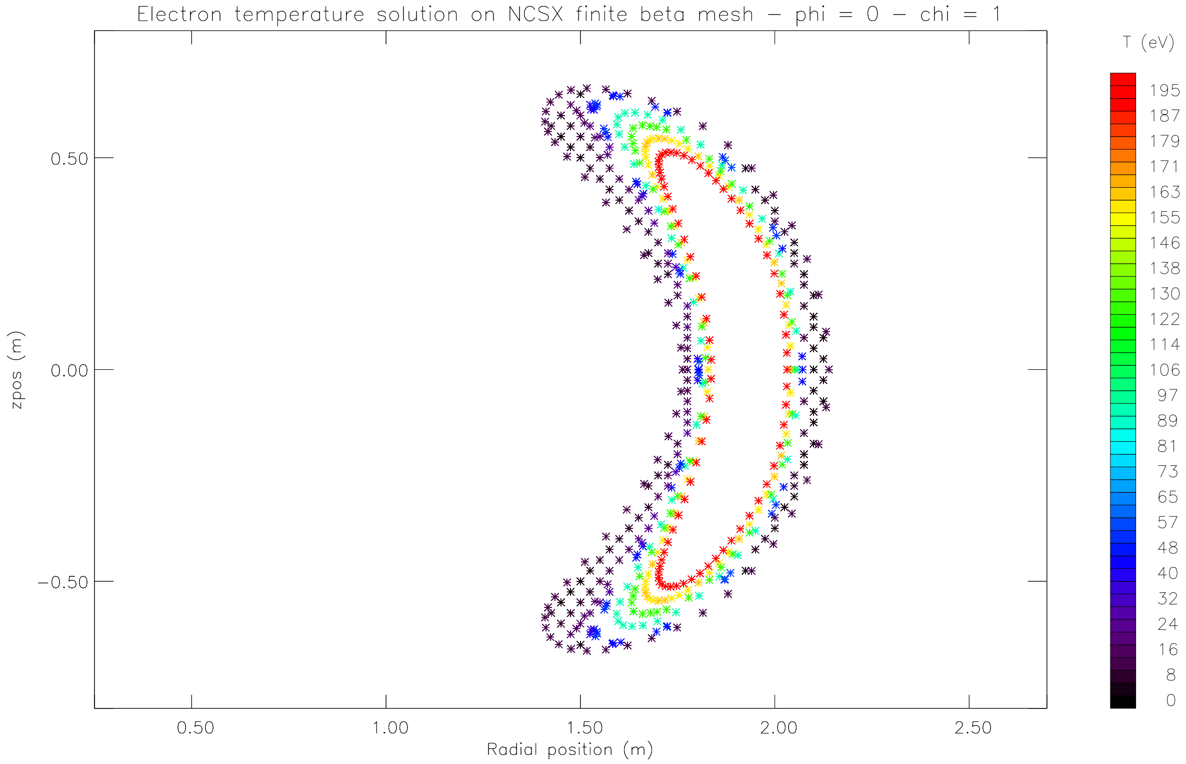

Figure 11.39: *Temperature solution on a finite beta mesh Poincaré plot at* $\phi = 0°$.

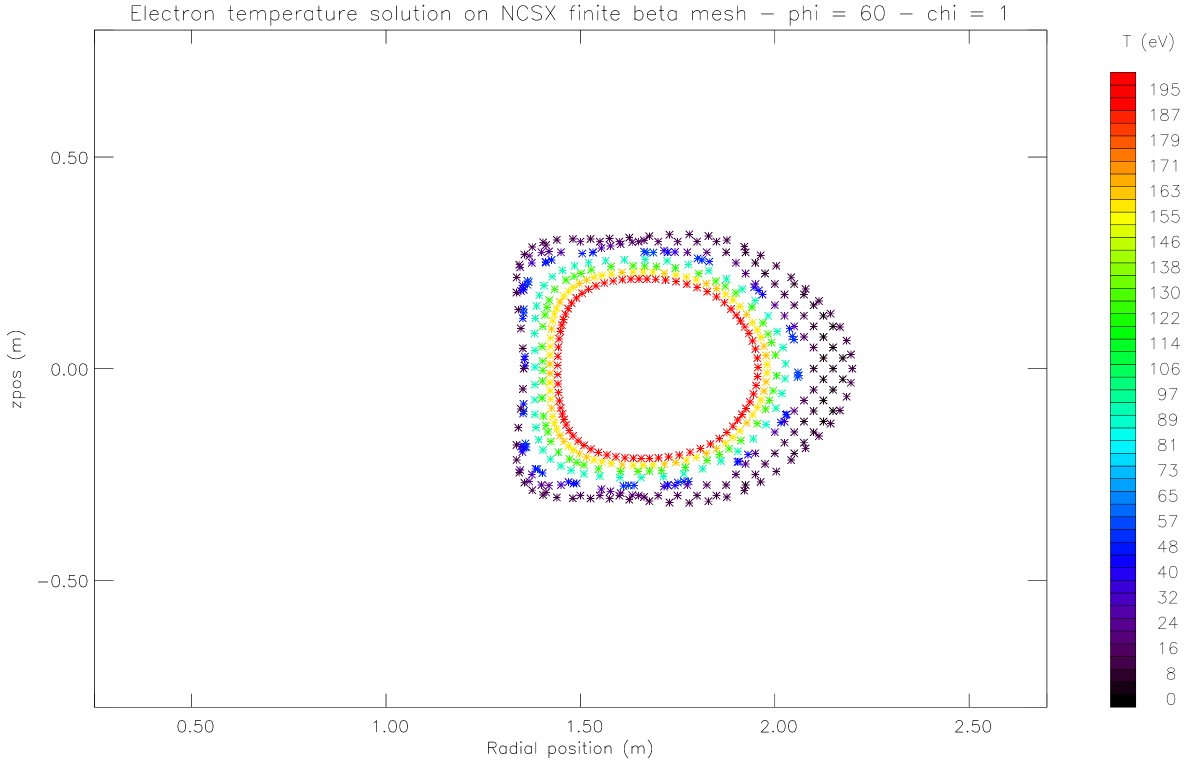

Figure 11.40: *Temperature solution on a finite beta mesh Poincaré plot at* $\phi = 60°$.

Fig.11.41 shows a promotional picture of a temperature solution on an NCSX mesh obtained by the finite difference code developed in the present work.

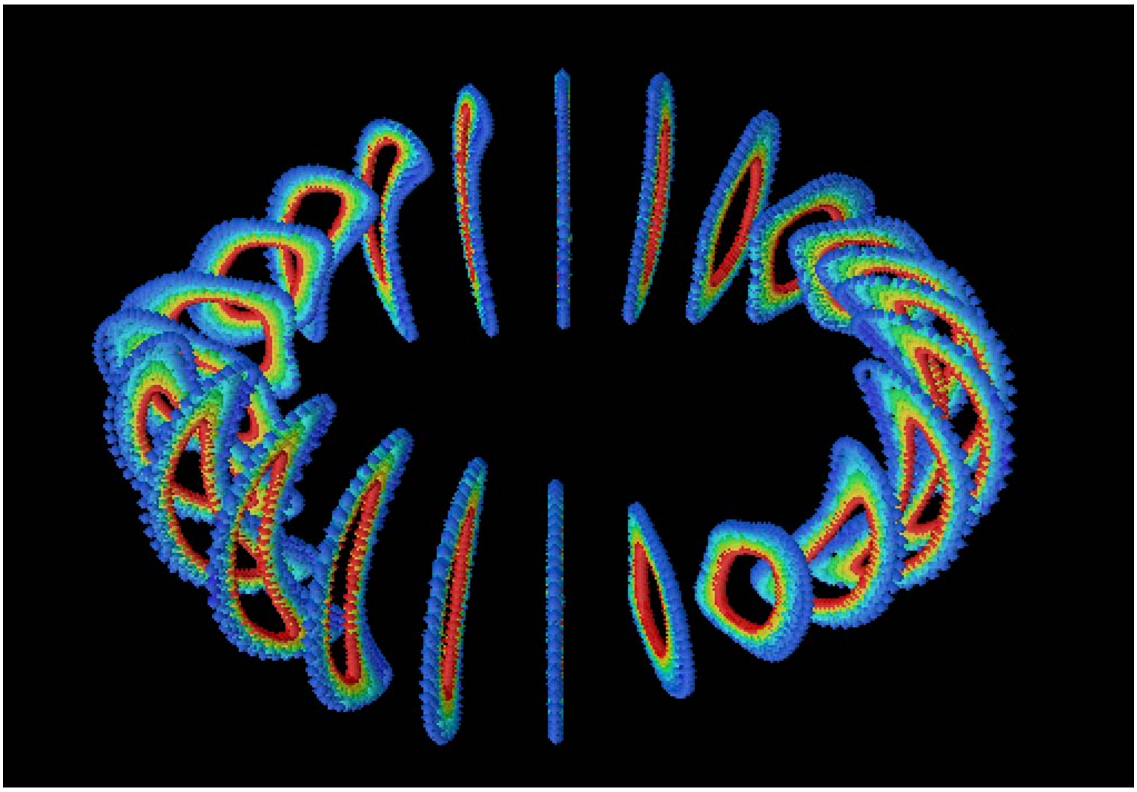

Figure 11.41: *A temperature solution on a W7-X mesh containing 24 cuts (total points* $\sim$ 17000*. Each mesh point is colored according to temperature. The red core points are at 200 eV. The purple outer boundary points are at 10 eV.*

12 Conclusions and outlook

Modeling in ergodic regions in magnetized plasmas offers special difficulties due to the strong anisotropy of the transport and the complex geometry of the computational domain. In the present work a method is developed for finite difference discretization of a transport equation in order to model electron heat conduction in the plasma edge of fusion experiments. This is to gain insight into the physics and to optimize the design and performance of such devices.

Using field line tracing, meshes are generated to represent plasmas in W7-X and NCSX. The transport equation is discretized along the field lines to allow accurate evaluation of the extreme anisotropic transport with an anisotropy factor between parallel and radial transport of $\sim 10^4 \rightarrow 10^7$. This very large anisotropy poses a real challenge to avoid numerical diffusion. Using a suitable optimization criterion in the mesh generation it was possible to reduce numerical diffusion to $\sim 1.0e - 5m^2s^{-1}$. This is much less than the physical diffusion, $\sim 1.0m^2s^{-1}$.

The problem of ergodicity is handled by using local magnetic coordinates. This allows a discretization of the transport equation in finite difference form as long as the distance between the mesh points is well below the typical length where neighboring field lines diverge exponentially (Kolmogorov length). The use of local magnetic coordinates allows a complete description of the system without additional approximation. However, the price to pay for this complete model is a full metric tensor including non-diagonal terms.

A variety of studies are done to investigate the physics in the plasma edge in the presence of ergodicity. The results show that the W7-X finite beta case allowed cascading of energy into regions which were inaccessible in the vacuum case, resulting in a smoothing of the island structures in the edge region. Thus, the broadening of the solution in the islands is due to an indirect ergodic effect. The island structures are not prominent in the solution, being masked by the general flattening of the temperature profile in the edge region.

Holding the core boundary condition constant and reducing the radial transport coefficient resulted in a decoupling, in temperature space, of the open field lines in the edge region from the closed field lines in the core region, thus inhibiting transport from the core. By comparison, reducing the core boundary condition, while holding the radial transport constant resulted only in a scaling down of the temperature solution while *not* inhibiting transport from the core.

By investigation of flux densities at the ends of the open field lines it was found that the longer open field lines played a greater role in removing heat from the plasma core because they had more contact with the ergodic region surrounding the plasma core.

The future development of the code will include additional boundary condition options, the inclusion of a convective term in the transport equation, and eventually, additional transport equations (ion heat conduction) and the possibility to handle impurities.

13 Appendices

13.1 Evaluating the metric coefficient

In the present work a separate code developed in [47] reads the mesh data file and the magnetic field configuration file, and generates the metric coefficients for each mesh point. The algorithm is outlined below.

1. Cylindrical coordinates are transformed to local magnetic coordinates using the tensorial rule for a transformation between coordinate systems, (1) and (2):

$$g^{ij}_{(2)} = g^{kl}_{(1)} \frac{\partial u^i_{(2)}}{\partial u^k_{(1)}} \frac{\partial u^j_{(2)}}{\partial u^l_{(1)}} \tag{13.1}$$

2. We start with a cylindrical system with a well known metric tensor:

$$g^{kl}_{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/R^2 \end{bmatrix} \tag{13.2}$$

3. For the coordinates change we need the following matrix of transformation:

$$C^i_j \equiv \frac{\partial u^i_{(2)}}{\partial u^j_{(1)}} = \begin{bmatrix} C^1_1 & C^1_2 & -C^1_1 \cdot f^1 - C^1_2 \cdot f^2 \\ C^2_1 & C^2_2 & -C^2_1 \cdot f^1 - C^2_2 \cdot f^2 \\ 0 & 0 & 1 \end{bmatrix} \tag{13.3}$$

   Here components C^i_j are defined per system of ODE in a way similar to Nemov's algorithm [48]

$$\frac{\partial}{\partial \phi} C^i_j = -C^i_j \frac{\partial f^l}{\partial y^j}, \quad \text{where} \quad f^i(R, z, \phi) = \frac{B^i(R, z, \phi)}{B^3(R, z, \phi)}, \quad i = 1, 2 \tag{13.4}$$

We integrate this system of ordinary differential equations along the field line with initial conditions on a section $\phi = \phi_m$

$$C^i_j \mid_{\phi=\phi_m} = \begin{bmatrix} 1 & 0 & -f^1 \\ 0 & 1 & -f^2 \\ 0 & 0 & 1 \end{bmatrix} \tag{13.5}$$

13.2 Poincaré plots in W7-X

Here we present a series of Poincaré plots covering 1 period (72°) of a W7-X 20-cut finite beta mesh, with 'standard' poloidal density.

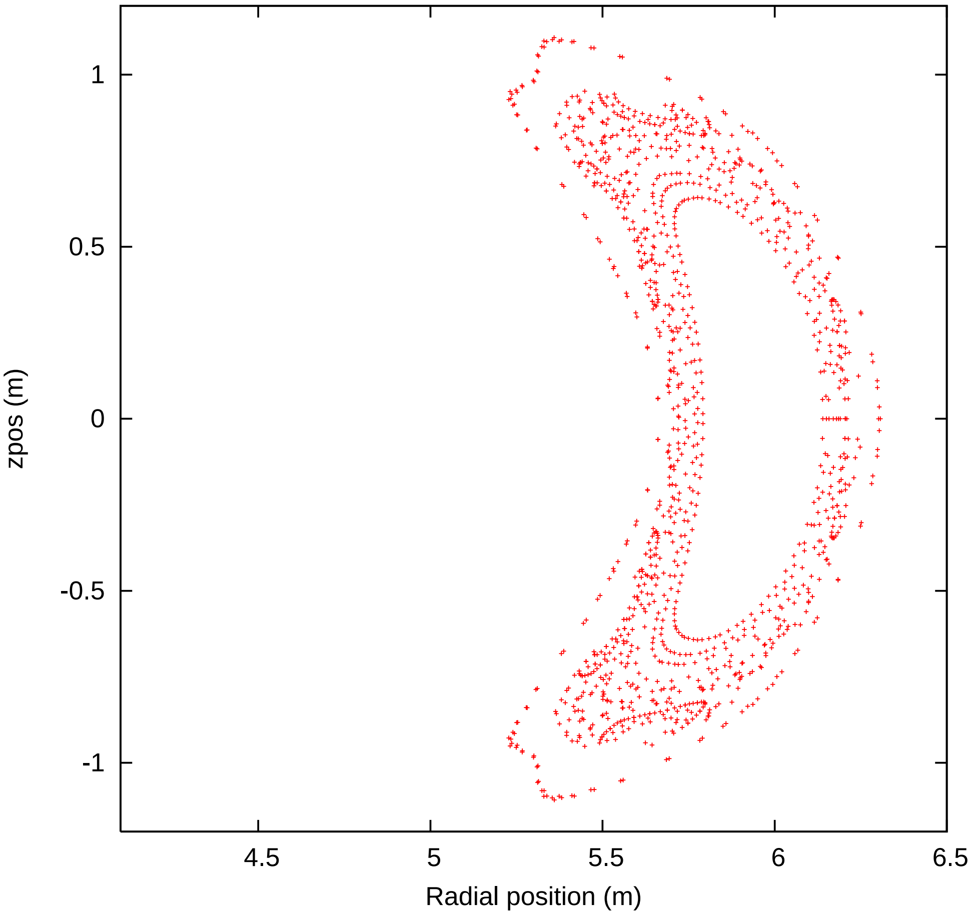

Figure 13.1: *Poincaré plot at* $\phi = 0°$. *1281 points.*

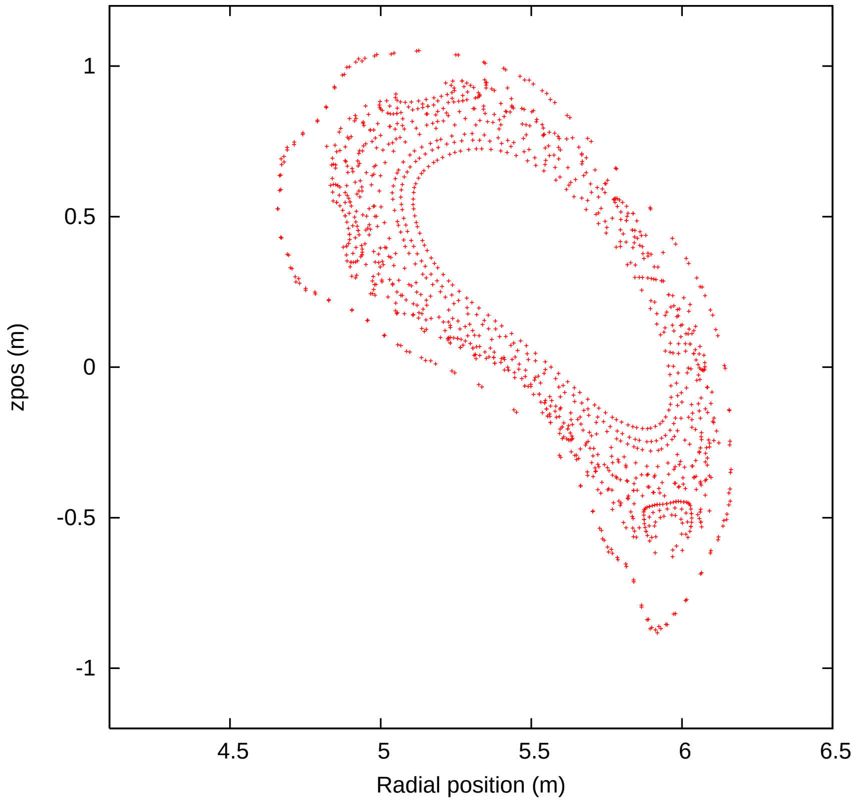

Figure 13.2: *Poincaré plot at* $\phi = 18°$. *1268 points.*

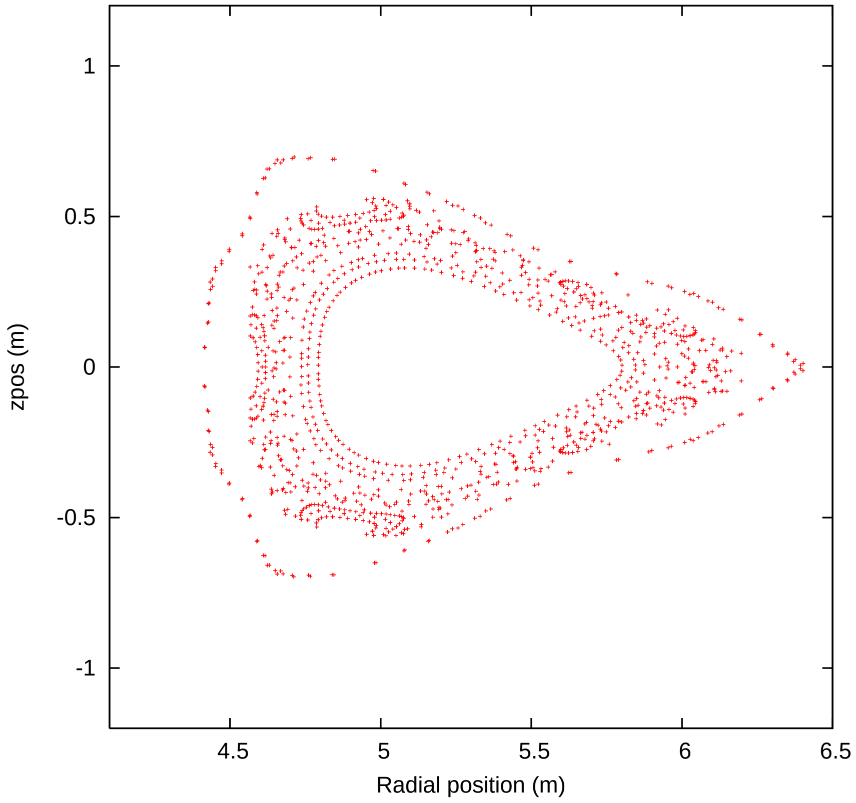

Figure 13.3: *Poincaré plot at* $\phi = 36°$. *1268 points.*

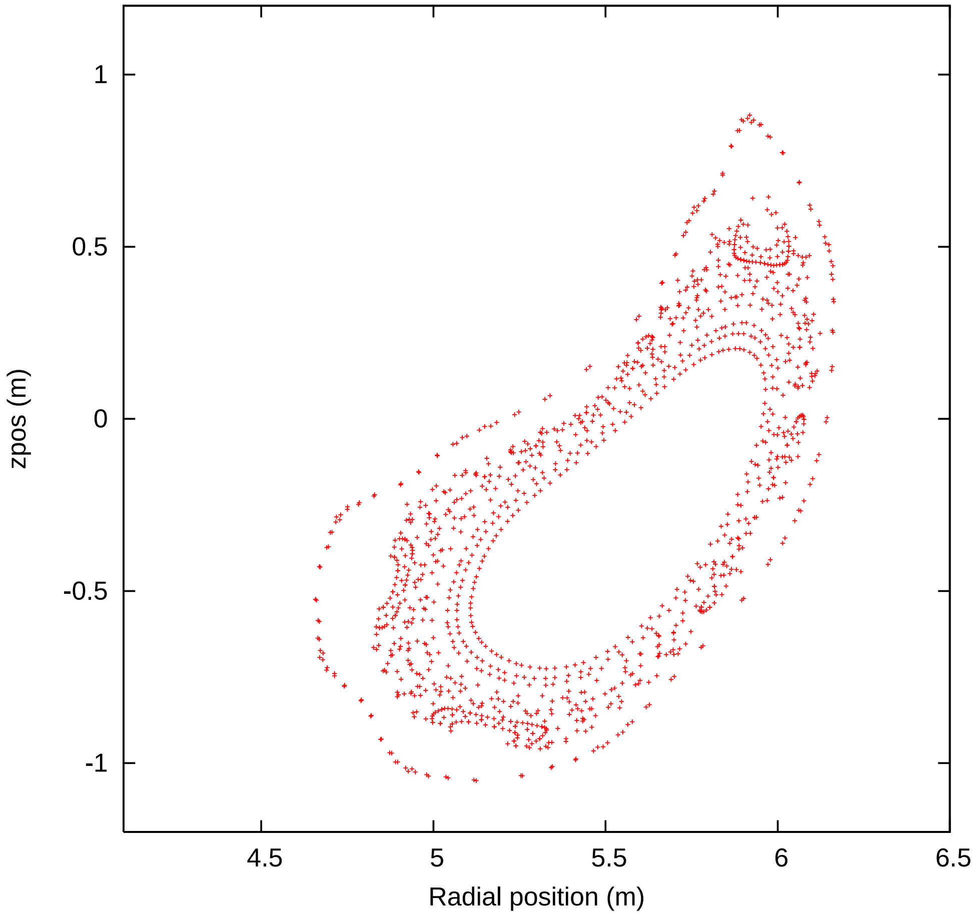

Figure 13.4: *Poincaré plot at* $\phi = 54°$. *1268 points.*

13.3 Triangulations in W7-X

Here we present a series of triangulated Poincaré plots covering 1 period (72°) of a W7-X 20-cut finite beta mesh, with 'standard' poloidal density.

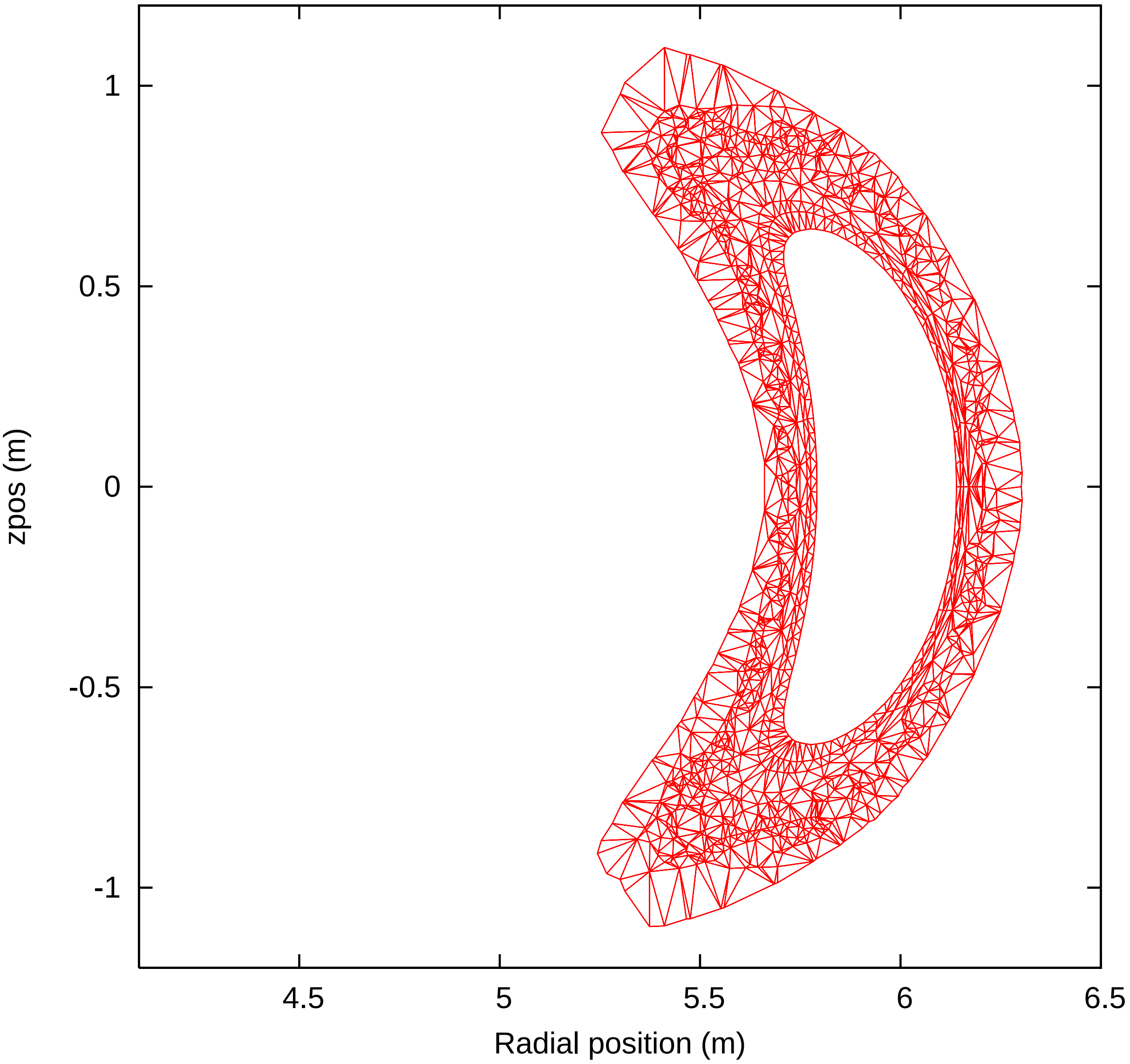

Figure 13.5: *Triangulated Poincaré plot at* $\phi = 0°$.

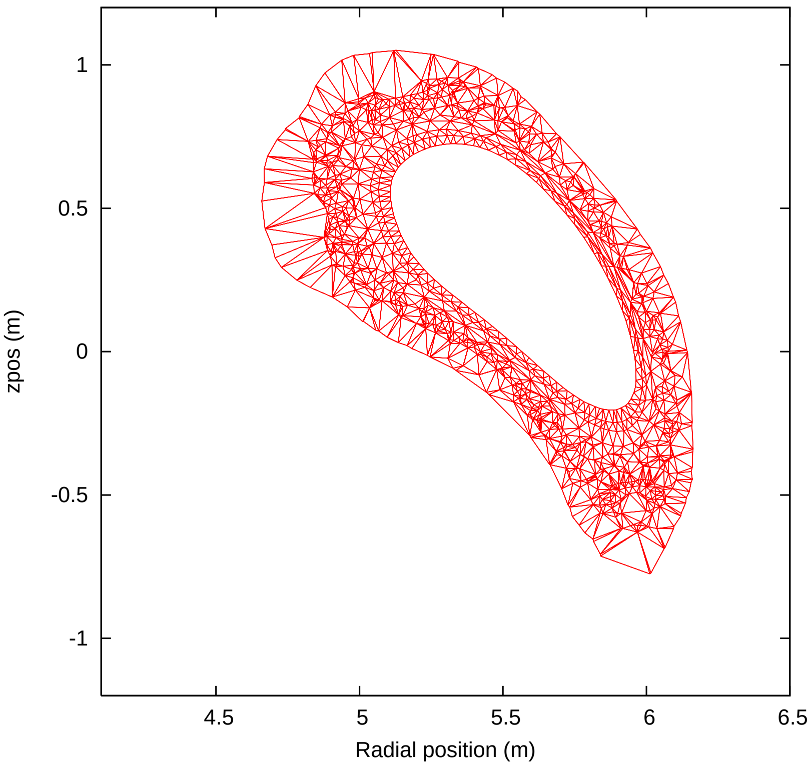

Figure 13.6: *Triangulated Poincaré plot at* $\phi = 18°$.

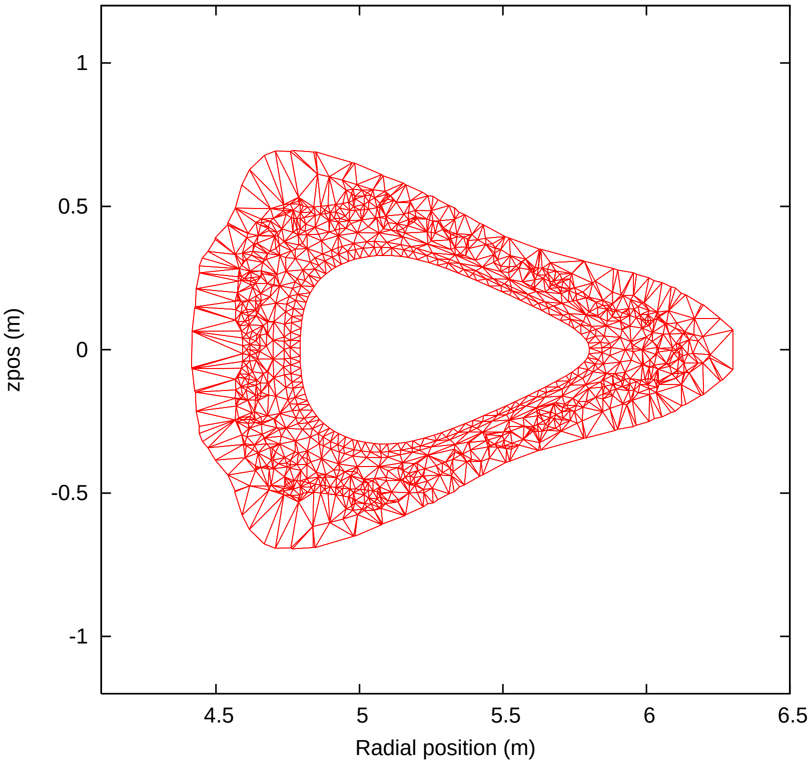

Figure 13.7: *Triangulated Poincaré plot at* $\phi = 36°$.

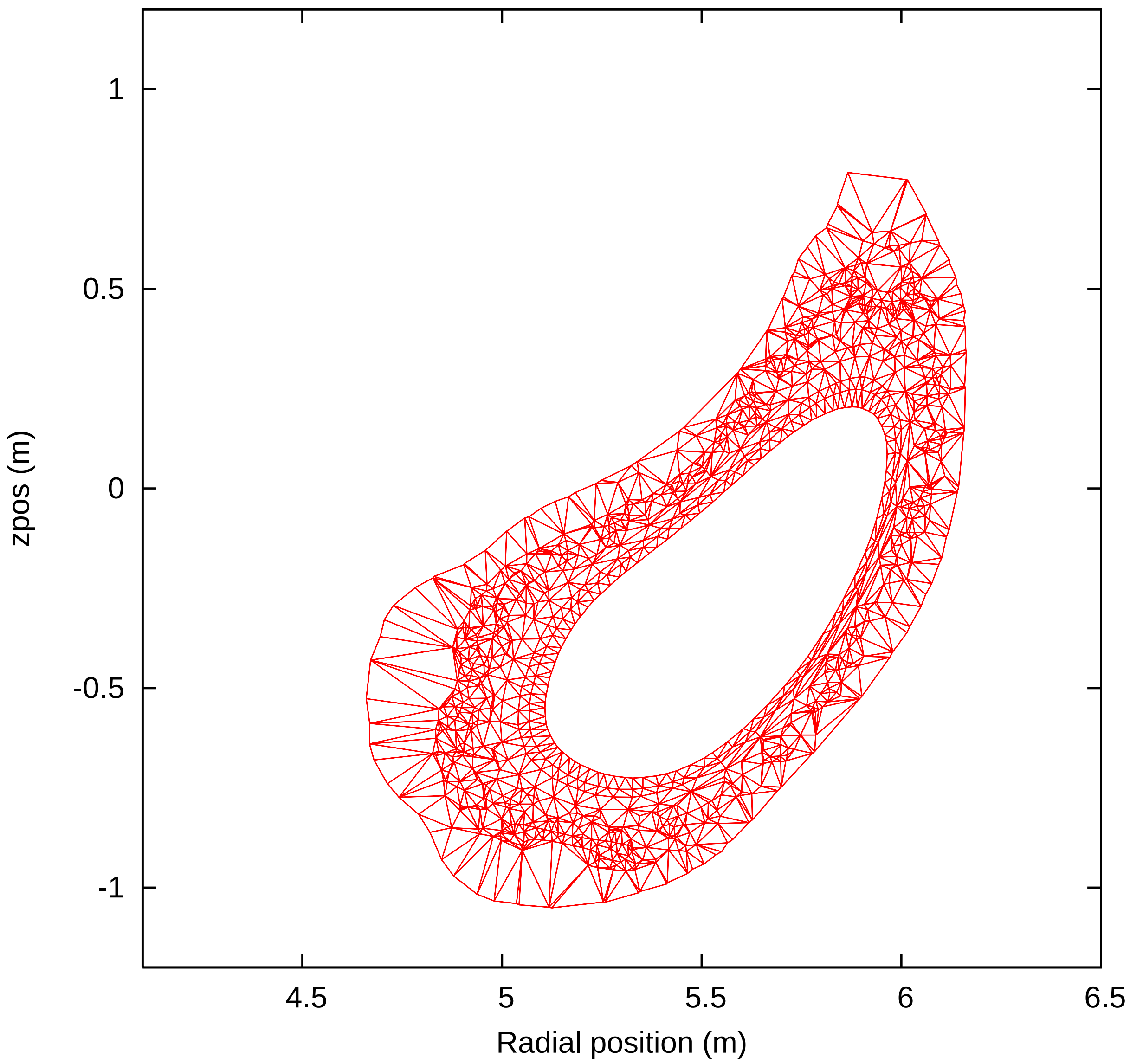

Figure 13.8: *Triangulated Poincaré plot at* $\phi = 54°$.

13.4 Least squares method

The Least Squares method is a method for determining a function to fit a set of data. For our purposes, the data is the set of points on a Poincaré plot. We use the Least Squares method to find the poloidal spatial derivatives, $\partial/\partial x$ and $\partial/\partial y$ at the primary and secondary points. See Fig.13.9.

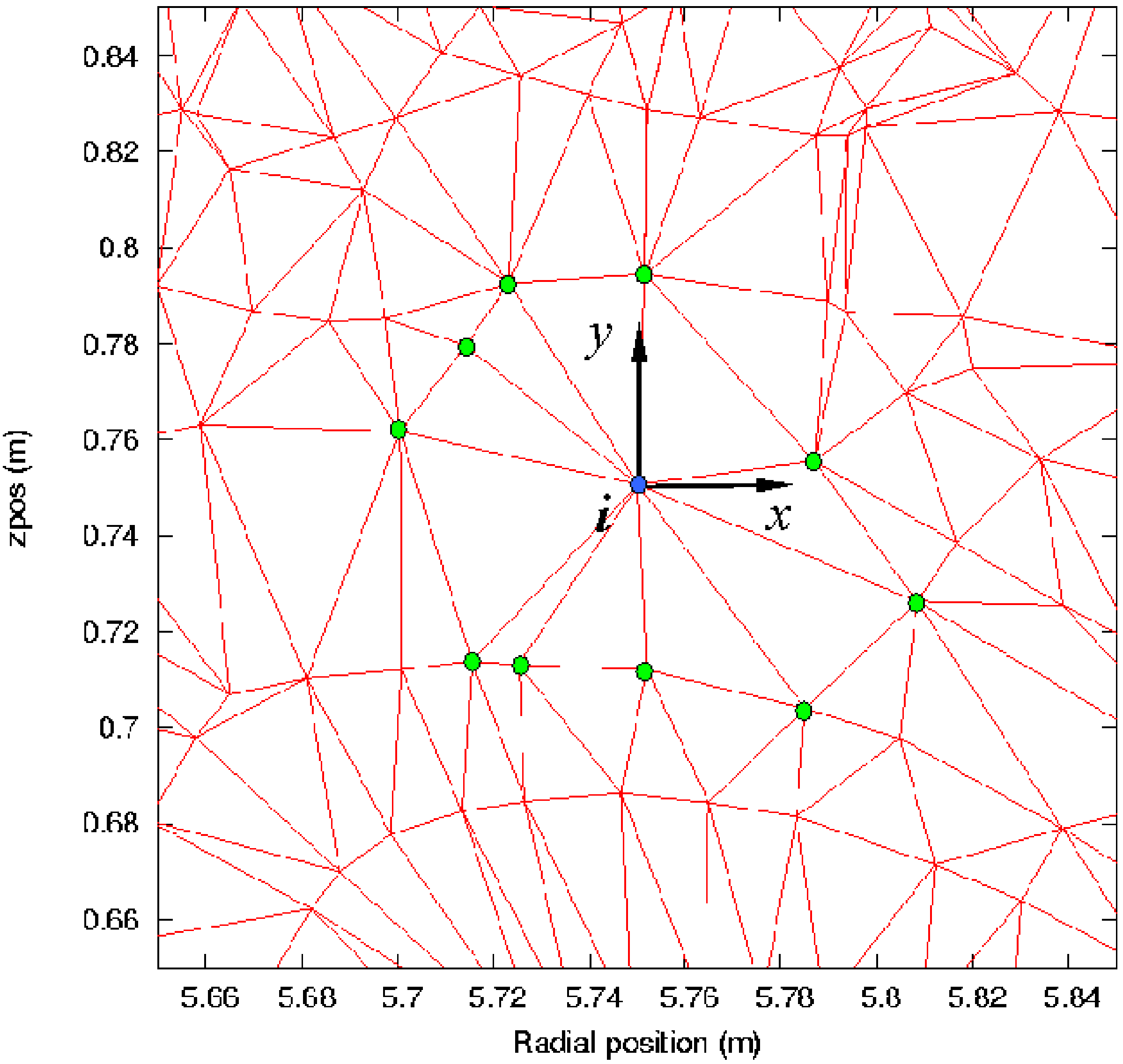

Figure 13.9: *Zoom-in view of a Poincaré plot. A typical mesh point, i, is shown as a blue dot, and its poloidal neighbors are shown as green dots. Each mesh point is the center of its own local coordinate system in which the poloidal coordinates, here shown as x and y, are Cartesian coordinates on the plane of the Poincaré plot.*

13.4.1 The Least Squares fit function

On each Poincaré plot each point has a set of poloidal primary neighbors scattered around it on the plane. These neighbors may be numbered $i = 1, 2, ...n$, where n is the number of neighbors. For secondary points, $n = 3$. For primary points, $n \approx 6$. Given the function value (temperature) at the poloidal neighbors we want to evaluate the derivatives on the plane. We solve a Least Squares fit using the model:

$$F(x, y) = f_o + f_x (x_i - \overline{x}) + f_y (y_i - \overline{y}) \tag{13.6}$$

where,
f_o : *the average temperature of the poloidal neighbors*
f_x, f_y : *gradients* $\partial/\partial x$, $\partial/\partial y$
x_i, y_i : x *and* y *coordinates of poloidal neighbor* i
$\overline{x}, \overline{y}$: *average* x *and* y *coordinates of the poloidal neighbors*

Equation (13.6) is a standard linear approximation in x and y. It provides a least squares fit to the data for a linear model. It is suitable because we want to evaluate only the gradients. To solve for the unknown coefficients (f_o, f_x, f_y) we minimize a least squares fit $\sum_i (F(x_i, y_i) - f_i)^2$ where f_i is the temperature at poloidal neighbor i.

Set

$$\begin{aligned} I &= \sum_i (F(x_i, y_i) - f_i)^2 \\ &= \sum_i (f_o + f_x (x - \overline{x}) + f_y (y - \overline{y}) - f_i)^2 \end{aligned} \tag{13.7}$$

13.4.2 Differentiating the Least Squares fit function

To solve for the coefficients (f_o, f_x, f_y) we obtain the system of 3 equations by differentiating the least squares fit with respect to each coefficient.

- Differentiate I with respect to f_o

$$\frac{\partial I}{\partial f_o} = 2 \left(n f_o - \sum_i f_i \right)$$

$$\Rightarrow \quad 0 = nf_o - \sum_i f_i \qquad \therefore \qquad f_o = \frac{1}{n}\sum_i f_i \tag{13.8}$$

Note that f_o, the average temperature of the poloidal neighbors, is not necessarily the same as the temperature at p. Equation (13.8) is a trivial result. Therefore, instead of solving a system of 3 equations for 3 unknowns, we need only to solve a system of 2 equations for 2 unknowns: f_x and f_y. To obtain the remaining 2 equations:

- Differentiate I with respect to f_x

$$\begin{aligned}\frac{\partial I}{\partial f_x} &= 2\sum_i \left(f_o + f_x\left(x-\overline{x}\right) + f_y\left(y-\overline{y}\right) - f_i\right)\left(x-\overline{x}\right)\\ \Rightarrow \quad 0 &= \sum_i \left(x-\overline{x}\right)^2 f_x + \sum_i \left(y-\overline{y}\right)\left(x-\overline{x}\right) f_y - \sum_i \left(x-\overline{x}\right) f_i\\ \Rightarrow \quad \sum_i \left(x-\overline{x}\right)^2 f_x + \sum_i \left(y-\overline{y}\right)\left(x-\overline{x}\right) f_y &= \sum_i \left(x-\overline{x}\right) f_i \qquad (13.9)\end{aligned}$$

- Differentiate I with respect to f_y

$$\begin{aligned}\frac{\partial I}{\partial f_y} &= 2\sum_i \left(f_o + f_x\left(x-\overline{x}\right) + f_y\left(y-\overline{y}\right) - f_i\right)\left(y-\overline{y}\right)\\ \Rightarrow \quad 0 &= \sum_i \left(x-\overline{x}\right)\left(y-\overline{y}\right) f_x + \sum_i \left(y-\overline{y}\right)^2 f_y - \sum_i \left(y-\overline{y}\right) f_i\\ \Rightarrow \quad \sum_i \left(x-\overline{x}\right)\left(y-\overline{y}\right) f_x + \sum_i \left(y-\overline{y}\right)^2 f_y &= \sum_i \left(y-\overline{y}\right) f_i \qquad (13.10)\end{aligned}$$

13.4.3 The resulting system of equations

Equations (13.9) and (13.10) are a system of 2 equations which we can use to solve for f_x and f_y. In matrix form the system of equations is:

$$\begin{bmatrix} \sum_i \left(x-\overline{x}\right) & \sum_i \left(x-\overline{x}\right)\left(y-\overline{y}\right) \\ \sum_i \left(x-\overline{x}\right)\left(y-\overline{y}\right) & \sum_i \left(y-\overline{y}\right) \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \sum_i \left(x-\overline{x}\right) f_i \\ \sum_i \left(y-\overline{y}\right) f_i \end{bmatrix}$$

$$\Rightarrow \quad \begin{bmatrix} \alpha_{xx} & \alpha_{xy} \\ \alpha_{yx} & \alpha_{yy} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \sum_i \left(x-\overline{x}\right) f_i \\ \sum_i \left(y-\overline{y}\right) f_i \end{bmatrix}$$

$$\Rightarrow \quad \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \beta_{xx} & \beta_{xy} \\ \beta_{yx} & \beta_{yy} \end{bmatrix} \begin{bmatrix} \sum_i (x-\overline{x}) f_i \\ \sum_i (y-\overline{y}) f_i \end{bmatrix} \tag{13.11}$$

$$where, \quad [\alpha]^{-1} \quad = \quad [\beta]$$

From equation (13.11) the poloidal gradients at any primary or secondary point, p, are:

$$\begin{aligned} f_x &= \beta_{xx}\sum_i (x-\overline{x}) f_i \quad + \quad \beta_{xy}\sum_i (y-\overline{y}) f_i \\ \frac{\partial T}{\partial x} &= \left(\beta_{xx}\sum_i (x-\overline{x}) + \beta_{xy}\sum_i (y-\overline{y})\right) T \end{aligned} \tag{13.12}$$

$$\begin{aligned} f_y &= \beta_{yx}\sum_i (x-\overline{x}) f_i \quad + \quad \beta_{yy}\sum_i (y-\overline{y}) f_i \\ \frac{\partial T}{\partial y} &= \left(\beta_{yx}\sum_i (x-\overline{x}) + \beta_{yy}\sum_i (y-\overline{y})\right) T \end{aligned} \tag{13.13}$$

13.4.4 Inverting the least squares coefficient matrix $[\alpha]$

Recall that we invert a 2×2 matrix in the following way:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{13.14}$$

The least squares coefficient matrix $[\alpha]$ is:

$$\begin{bmatrix} \alpha_{xx} & \alpha_{xy} \\ \alpha_{yx} & \alpha_{yy} \end{bmatrix} = \begin{bmatrix} \sum_i (x-\overline{x}) & \sum_i (x-\overline{x})(y-\overline{y}) \\ \sum_i (x-\overline{x})(y-\overline{y}) & \sum_i (y-\overline{y}) \end{bmatrix} \tag{13.15}$$

The determinant, $(ad-bc)$, of matrix $[\alpha]$ is:

$$\det[\alpha] \quad = \quad \left(\sum_i (x_i - \overline{x})\right)\left(\sum_i (y_i - \overline{y})\right) - \left(\sum_i (x_i - \overline{x})(y_i - \overline{y})\right)^2 \qquad (13.16)$$

Therefore, the elements of the inverse matrix $[\beta]$ are:

$$\beta_{xx} \quad = \quad \frac{\sum_i (y_i - \overline{y})}{\det[\alpha]}, \qquad \beta_{xy} \quad = \quad \frac{-\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\det[\alpha]} \qquad (13.17)$$

$$\beta_{yx} \quad = \quad \frac{-\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\det[\alpha]}, \qquad \beta_{yy} \quad = \quad \frac{\sum_i (x_i - \overline{x})}{\det[\alpha]} \qquad (13.18)$$

The sum quantities are evaluated by looping over the poloidal neighbors of p.

13.4.5 Matrix [A0], poloidal gradients at the secondary points

$$\underset{2NS \times NP}{\begin{bmatrix} \quad A0 \quad \end{bmatrix}}$$

Recall that matrix [A0] contains the poloidal gradients at the secondary points. It is used to construct the component matrix [A6] which contains the coefficients for the poloidal flux divergences.

A secondary point is the center of gravity of a triangle. Therefore it has 3 primary point neighbors, and therefore each row in [A0] has 3 non-zero elements. (The Least Squares method is not strictly necessary in this case because, with only 3 points, it is possible to find exact solutions.) The rows occur in pairs. Each pair corresponds to one secondary point. In each pair, the odd numbered row contains the x gradients, and the even numbered row contains the y gradients.

Recall from equation (13.12) that the x poloidal gradient at any point is evaluated as:

$$\frac{\partial T}{\partial x} = \left(\beta_{xx}\sum_i (x-\overline{x}) + \beta_{xy}\sum_i (y-\overline{y})\right) T$$

The expression in brackets is the poloidal spatial derivative. For a secondary point it has 3 components, one for each primary neighbor. Each component is evaluated as:

$$\frac{\partial}{\partial x_i} = \beta_{xx}\,(x_i-\overline{x}) + \beta_{xy}\,(y_i-\overline{y}) \tag{13.19}$$

where the index i refers to the particular primary neighbor of the secondary point, and therefore a particular non-zero element in [A0].

The y poloidal spatial derivatives are evaluated in the same way. From equation (13.13) the poloidal gradient is:

$$\frac{\partial T}{\partial y} = \left(\beta_{yx}\sum_i (x-\overline{x}) + \beta_{yy}\sum_i (y-\overline{y})\right) T$$

and the component spatial derivatives are:

$$\frac{\partial}{\partial y_i} = \beta_{yx}\,(x_i-\overline{x}) + \beta_{yy}\,(y_i-\overline{y}) \tag{13.20}$$

13.4.6 Matrices [A3] and [A5], poloidal gradients at the primary points

$$\underset{NP\times NP}{\begin{bmatrix} & & \\ & A3 & \\ & & \end{bmatrix}} \quad \underset{NP\times NP}{\begin{bmatrix} & & \\ & A5 & \\ & & \end{bmatrix}}$$

Matrices [A3] and [A5] contain the x and y spatial derivatives respectively at the primary points. These matrices are used in the construction of the component matrices [A9] and [A10]. These contain the coefficients for the mixed poloidal-toroidal flux divergences.

The non-zero elements in these matrices are evaluated in exactly the same way as in matrix [A0]. The only difference is that the number of primary poloidal neighbors for each primary point is not 3, as for each secondary point, but 6 on average. That is, each row in [A3] and [A5] corresponds to one primary point and has ∼6 non-zero elements. The elements in [A3] are evaluated using equation (13.19), and the elements in [A5] use equation (13.20). As before, the index i refers to the particular neighbor.

13.5 Sparse matrix storage and manipulation

The process of constructing the final coefficient matrix requires the efficient storage and manipulation of large sparse matrices [49]. Recall that we construct a series of fundamental matrices which contain gradients and transport coefficients. These are combined by multiplication into a set of component matrices which represent the 9 flux divergences from the vector form of the transport equation. The component matrices are then combined by addition into the final coefficient matrix.

A standard W7-X mesh used in the present work is ∼25000 points, so the corresponding coefficient matrix size is 25000×25000, with a sparsity of $> 99.99\%$. For matrices of this size and sparsity there are still hundreds of thousands of non-zero elements. It is impossible to use the normal Fortran functions for matrix operations, so we turn to specialized methods.

13.5.1 Coordinate format

When we construct the fundamental matrices, the non-zero elements are stored in coordinate format (COO) [49]. That is, for each matrix we generate 3 arrays. Each array has length nnz, which is the number of non-zeros in the matrix.

1. A real array, A, for the values of the non-zero elements.
2. An integer array, $Arow$, for the row indices of the non-zero elements.

3. An integer array, $Acol$, for the column indices of the non-zero elements.

COO format is a common and simple method for storing large sparse matrices of any structure. For each mesh point, the neighborhood array gives us the row and column indices for each non-zero element. For example, matrix A3 contains poloidal gradients. Each row and each column corresponds to a mesh point. If we look along the row for mesh point p, there are non-zero elements lying at the column positions which represent the known poloidal neighbors of p.

13.5.2 Compressed Sparse Row format

The information for each matrix is converted from COO format to compressed sparse row format (CSR) [49]. Each matrix is represented by 3 arrays:

1. A real array, A, containing the non-zero element values. (length nnz)
2. An integer array, JA, for the column indices. (length nnz)
3. An integer array, IA, containing the pointers to the beginning of each row in arrays A and JA. The $i-$th element in IA is the position in arrays A and JA where the $i-$th row starts. The length of IA is $nrows + 1$. $nrows$ is the number of rows in the matrix. The last element, $IA(nrows + 1)$, contains the number $IA(1) + nnz$, that is, the address in A and JA of the start of a fictitious row $nrows + 1$.

CSR format is convenient for storing matrices which have complex distributions of non-zero elements. Like the COO format, it is applicable to any matrix structure, but it is more efficient in its memory requirements. After the conversion to CSR format, the manipulation of the matrices is performed. That is, the fundamental matrices are combined by multiplication to form the component matrices, and the component matrices are combined by addition to form the final coefficent matrix.

After all the matrix operations have been performed, the final coefficient matrix, is converted from CSR into COO format. This information is fed into the matrix solver subroutine in the transport code. Recall that the matrix equation $Ax = b$ is solved iteratively. The matrix of coefficients, A, is usually a function of the unknown variable T_e and therefore the system is non-linear. To account for the non-linearity we apply the subsequent substitutions iterative method [50] (also known as Picard iteration, functional iteration, and successive approximation). Most of the fundamental matrices are purely geometric, ie the elements are *not* temperature dependent. They are constructed only once for a given mesh, and they are evaluated outside the iteration loop. Some of the fundamental matrices, and all of the component matrices, have elements which *are* temperature dependent. They are evaluated inside the iteration loop. In summary, all of the sparse matrix conversion, manipulation, and deconversion occurs at every iteration in the solution process.

13.5.3 Summary of matrix operations in the transport code

The procedure (in fact, the entire code) is summarized as follows:

1. Read mesh data file, neighborhood arrays and metric coefficients.
2. Generate the purely geometric fundamental matrices. These are all in COO format.
    a) poloidal gradients: A0, A2, A3, A5
    b) toroidal gradients: A4
    c) toroidal second order derivatives: $A11_0$
3. Start iteration loop.
4. Construct temperature dependent fundamental matrices, also in COO format.
    a) transport coefficients for purely poloidal flux divergence: A1
    b) transport coefficients for mixed and purely toroidal flux divergences: Ψ
    c) matrix, T^t, containing the temperatures at the previous iteration (or at the initial condition, in the case of the first iteration).
    d) matrix, At, containing the time derivatives.
5. Combine fundamental matrices into component matrices. Subroutines in 'Sparse' module convert fundamental matrices to CSR format and do the matrix multiplications. Resulting component matrices are in CSR format; A6, A7, A8, A9, A10, A11.
6. Combine component matrices into final coefficient matrix, A12. Subroutines in the 'Sparse' module do the addition. A12 is in CSR format.
7. Convert A12 to COO format.
8. Construct the right hand side vector, *rhs*, containing the time derivatives and boundary conditions.

9. Send A12 and *rhs* to the matrix solver subroutine.
10. Iterate toward temperature solution across the mesh.
11. End iteration loop.
12. Send final temperature solution to the Output module.

Bibliography

[1] G. Grieger, W. Lotz, P. Merkel, J. Nührenberg, J. Sapper, and E. Strumberger. Physics optimization of stellarators. *Physics of Fluids*, B4:2081–2091, 1992.

[2] R. SCHNEIDER. *Plasma Edge Physics for Tokamaks.* Max-Planck-Institut für Plasmaphysik, Greifswald, Germany, 2001.

[3] P.C. Stangeby. The Bohm–Chodura plasma sheath criterion. *Physics of Plasmas*, 2(3):702–706, 1995.

[4] P.C. Liewer. Measurements of microturbulence in tokamaks and comparisons with theories of turbulence and anomalous transport. *Nuclear Fusion*, 25(5):543–621, 1985.

[5] J.A. Wesson. *Tokamaks.* Clarendon Press, Oxford, 1987.

[6] D. Reiter, G.H. Wolf, and H. Kever. Burn condition, helium particle confinement and exhaust efficiency. *Nuclear Fusion*, 30(10):2141–2155, 1990.

[7] D.E. Post and K. Lackner. Plasma models for impurity control experiments. In D.E. Post and R. Behrisch, editors, *Physics of Plasma Wall Interaction in Controlled Fusion (Proc. NATO Workshop ValMorin, Quebec, Canada)*, pages 627–693, New York, 1984. Plenum Press.

[8] R. Behrisch and V. Prozesky. Particle and power exhaust for a fusion plasma. *Nuclear Fusion*, 30(10):2166–2169, 1990.

[9] J. Neuhauser. The physics of power exhaust from toroidal magnetoplasmas. *Plasma Physics and Controlled Fusion*, 34(13):2015–2022, 1992.

[10] A. Herrmann. Overview on stationary and transient divertor heat loads. *Plasma Physics and Controlled Fusion*, 44(6):883–903, 2002. IAEA TCM meeting 2001 Aix–en–Provence.

[11] G.F. Matthews, S.J. Fielding, G.M. McCracken, C.S. Pitcher, P.C. Stangeby, and M. Ulrickson. Investigation of the fluxes to a surface at grazing angles of incidence in the tokamak boundary. *Plasma Physics and Controlled Fusion*, 32:1301–1320, 1990.

[12] J.D. Lawson. Some criteria for a power producing thermonuclear reactor. In *Proceedings of the Physical Society*, volume B70, page 6, 1957.

[13] K. Itoh and S.-I. Itoh. The role of electric field in confinement. *Plasma Physics and Controlled Fusion*, 38(1):1–49, 1996.

[14] K. Ida. Experimental studies of the physical mechanism determining the radial electrical field and its radial structure in a toroidal plasma. *Plasma Physics and Controlled Fusion*, 40(8):1429–1488, 1998.

[15] D.P. Coster. Divertor design: issues raised by steady state and advanced tokamak operation, 2000. Invited paper for the 2000 Toki Conference.

[16] A. Field, C. García–Rosales, D. Naujoks, C.S. Pitcher, G. Lieder, G. Fussmann, S. Hirsch, R. Radtke, U. Wenzel, and R. Neu. Studies of divertor target plate erosion in the ASDEX Upgrade tokamak. *Journal of Nuclear Materials*, 220–222:553–557, 1995. 11th PSI Mito, Japan, 23.–27. May 1994.

[17] C.S. Pitcher and P.C. Stangeby. Experimental divertor physics. *Plasma Physics and Controlled Fusion*, 39(6):779–930, 1997.

[18] H. Renner, J. Boscary, V. Erckmann, H. Greuner, H. Grote, J. Sapper, E. Speth, F. Wesner, M. Wanner, and W7–X team. The capabilities of steady state operation at the stellarator W7–X with emphasis on divertor design. *Nuclear Fusion*, 40:1083–1093, 2000.

[19] R. Chodura. Plasma flow in the sheath and presheath of a scarpe–off layer. In D.E. Post and R. Behrisch, editors, *Physics of Plasma Wall Interaction in Controlled Fusion (Proc. NATO Workshop Val–Morin, Quebec, Canada)*, pages 99–134, New York, 1984. Plenum Press.

[20] R. Behrisch. Surface erosion by electrical arcs. In D.E. Post and R. Behrisch, editors, *Physics of Plasma Wall Interaction in Controlled Fusion (Proc. NATO Workshop ValMorin, Quebec, Canada)*, New York, 1984. Plenum Press.

[21] R. Chodura. Plasma flow in the sheath and presheath of a scrape–off layer. volume 131 of *NATO Advanced Science Institute*, pages 99–134, New York, 1986. NATO Science Committee, Plenum Press.

[22] P.C. Stangeby and A.V. Chankin. The ion velocity (Bohm–Chodura) boundary condition at the entrance to the magnetic presheath in the presence of diamagnetic and $\vec{E} \times \vec{B}$ drifts in the scrape–off layer. *Physics of Plasmas*, 2(3):707–715, 1995.

[23] P.C. Stangeby. *The Plasma Boundary of Magnetic Fusion Devices*. IOP, Bristol, 2000.

[24] F.F. Chen. *Introduction to Plasma Physics*. Plenum Press, New York, 1984.

[25] E. Strumberger. The stochastic edge of W7–X. *Journal of Nuclear Materials*, 266–269:1207–1211, 1999.

[26] S.S. Abdullaev, K.H. Finken, Jakubowski M.W., S.V. Kasilov, M. Kobayashi, D. Reiser, D. Reiter, A.M. Runov, and R. Wolf. Overview of magnetic structure induced by the textor-ded and the related transport. *Nuclear Fusion*, 43:299–313, 2003.

[27] K.H. Finken, T. Eich, S.S. Abdullaev, A. Klaeck, G. Mank, D. Reiser, A.M. Runov, and M. Tokar. Modeling of the dynamic ergodic divertor of TEXTOR–94. *Journal of Nuclear Materials*, 266–269:495–500, 1999.

[28] M.N. Rosenbluth, R.D. Hazeltine, and F.L. Hinton. Plasma transport in toroidal confinement systems. *Physics of Fluids*, 15(1):116–140, 1972.

[29] P. Ghendrih, A. Grosman, and H. Capes. Theoretical and experimental investigations of stochastic boundaries in tokamaks. *Plasma Physics and Controlled Fusion*, 38(10):1653–1724, 1996.

[30] M. Borchardt, J. Riemann, R. Schneider, and X. Bonnin. W7–X edge modelling with the 3D SOL fluid code BoRiS. *Journal of Nuclear Materials*, 290–293(3):546–550, 2001.

[31] Y. Feng, F. Sardei, and J. Kisslinger. 3D fluid modelling of the edge plasma by means of a Monte Carlo technique. *Journal of Nuclear Materials*, 266–269:812–818, 1999.

[32] R. Schneider, M. Borchardt, J. Riemann, A. Mutzke, and S. Weber. Concept and status of a 3D SOL fluid code. *Contributions to Plasma Physics*, 40(3–4):340–345, 2000. 7th PET Conference, Tajimi, Gifu, Japan, 4–6 October, 1999.

[33] A.M. Runov, S. Kasilov, J. Riemann, M. Borchardt, D. Reiter, and R. Schneider. Benchmark of the 3–dimensional plasma transport codes E3D and BoRiS. *Contributions to Plasma Physics*, 42(2–4):169–174, 2002.

[34] A.M. Runov, S. Kasilov, D. Reiter, N. McTaggart, X. Bonnin, and R. Schneider. Transport in complex magnetic geometries: 3D modelling of ergodic edge plasmas in fusion experiments. *Journal of Nuclear Materials*, 313(3):1292–1297, 2003.

[35] R. Schneider, D.P. Coster, B.J. Braams, P. Xantopoulos, V.A. Rozhansky, S.P. Voskoboynikov, L. Kovaltsova, and H. Bürbaumer. B2–SOLPS5.0: SOL transport code with drifts and currents. *Contributions to Plasma Physics*, 40(3–4):328–333, 2000. 7th PET Conference, Tajimi, Gifu, Japan, 4–6 October, 1999.

[36] J. Riemann, M. Borchardt, R. Schneider, X. Bonnin, A. Mutzke, and T.D. Rognlien. Hierarchy tests of edge transport models (BoRiS, UEDGE). *Journal of Nuclear Materials*, 313(3):1030–1035, 2003.

[37] T.D. Rognlien, J.L. Milovich, M.E. Rensink, and G.D. Porter. A fully implicit, time dependent 2–D fluid code for modeling tokamak edge plasmas. *Journal of Nuclear Materials*, 196–198:347–351, 1992.

[38] R. Zagorski, H. Gerhauser, and H.A. Claassen. Numerical simulation of the TEXTOR edge plasma including drifts and impurities. *Contributions to Plasma Physics*, 38(1–2):61–66, 1998.

[39] M.E. Rensink, G.D. Porter, T.D. Rognlien, D.P. Coster, R. Schneider, C.F.F. Karney, and D.P. Stotler. Edge plasma simulations with coupled fluid and Monte Carlo models. Technical Report UCRL–JC–124364–ABS, Lawrence Livermore National Laboratory, Livermore, CA, 1996. Annual Meeting, American Physical Society (APS), 38th, Division of Plasma Physics, Denver, CO, November 11.–15., 1996.

[40] S.I. Braginskii. Transport processes in a plasma. In *Reviews of Plasma Physics*, volume 1, pages 205–311. Consultants Bureau, New York, New York, 1965.

[41] D.L. Book. *NRL Plasma Formulary.* Office of Naval Research, Naval Research Laboratory, Washington, D.C. 20375, 1980 (Revised).

[42] A.M. Runov, D Reiter, S.V. Kasilov, M.F. Heyn, and W. Kernbichler. Monte carlo study of heat conductivity in stochastic boundaries: Application to the textor ergodic divertor. *Physics of Plasmas*, 8(3):916–930, 2001.

[43] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, New York, 1980.

[44] J. O'Rourke. *Computational Geometry in C (2nd edition).* Cambridge University Press, Cambridge, 1998.

[45] E. Strumberger. Deposition patterns of fast ions on plasma facing components in W7–X. *Nuclear Fusion*, 40(10):1697–1713, 2000.

[46] Transport Chapter of the documentation for the NCSX Conceptual Design Review, http://www.pppl.gov/ncsx/Meetings/CDR/CDRfinal/.

[47] A.M. Runov. private communication, to be published.

[48] V.V. Nemov. *Nuclear Fusion*, 28:1727–1736, 1988.

[49] Y. Saad. *SPARSKIT: a basic tool kit for sparse matrix computations, Version 2.* University of Minnesota, Minneapolis, 1994.

[50] J.M. Ortega and W.C. Rheinholdt. *Iterative Solution of Non-linear Equations in Several Variables.* Academic Press, New York, 1980.

14 Acknowledgements

Many talented people contributed to this thesis. I would like to thank my supervisor Dr. Ralf Schneider for offering me the possibility of doing a PhD in fusion physics, for his endless patience and for sharing his encyclopaedic knowledge of edge physics. Xavier Bonnin helped greatly in explaining some of the finer points of plasma physics and taught me an enormous amount of coding in Fortran 90, which I have learned to enjoy.

Andreas Mutzke supplied the field line tracing code, without which I would still be fooling around with artifical meshes, and also the plotting routines for visualizing my solutions. Alex Runov supplied the code for generating the metric coefficients and offered sound advice on various aspects of the thesis.

Henri Majaniemi contributed a great deal to the triangulation code and offered endless mathematical distractions and stimulations. He also developed a radical new triangulation algorithm which, in the end, was perhaps too clever. Matthias Borchhardt supplied the matrix solver subroutines for inverting the increasingly enormous matrices generated by the finite difference code.

Dr. Roman Zagorski took the code to a new level of sophistication and paved the way to the code's future development. I look forward to visiting him in Poland. Dr. Bas Braams developed the fundamental principles of discretizing the transport equation on an unstructured mesh and made me welcome in New York.

Jutta Gauger, the secretary, was invaluable in shielding me from the German beaurocracy. It has been my pleasure to work in the Stellarator Theory department under Professor Nührenberg.

15 Curriculum Vitae

Neil McTaggart

Nationality: British

Marital state: single

Address: Bahnhofstraße 2/3. 17489 Greifswald, Germany

Email: taggart@ipp.mpg.de

1981 - 1985	Lynfield College, Auckland, New Zealand
1995 - 2000	Mechanical Engineering degree (Masters) at Aberdeen University, Scotland
2000 - 2003	Doctorate in Plasma Physics in the Stellarator Theory Department, IPP, Greifswald, Germany