Flux-aligned quad mesh generation in magnetohydrodynamic simulation

Meng Wu, Xuhui Wang, Boniface Nkonga, Bernard Mourrain, Gang Xu, Qian Ni, Yuan Liu

Flux-aligned Quad Mesh Generation in Magnetohydrodynamic Simulation

Meng Wu Xuhui Wang Boniface Nkonga Bernard Mourrain
Gang Xu Qian Ni Yuan Liu *
School of Mathematics and Statistics
Nanjing University of Science and Technology, Nanjing 210094, P. R. China
School of Mathematics
Hefei University of Technology, Hefei 230009, P. R. China
AROMATH
Inria, Sophia Antipolis 06902, France
Lab. J. A. Dieudonné
University of Nice Sophia Antipolis, Nice 06103, France
School of Computer Science and Technology
Hangzhou Dianzi University, Hangzhou 310018, P. R. China
School of Physical and Mathematical Sciences
Nanjing Tech. University, Nanjing 211816, P. R. China
College of Science
Hohai University, Nanjing 210098, P. R. China

Abstract

Flux-aligned mesh generation plays an important role in the magnetohydrodynamic (MHD) simulation of Tokamak plasmas.In this paper, we present the existence theory of flux-aligned meshes by generalized Morse theory to the situation in Tokamak simulation. A high-order algorithm is developed to validate the theory by generating flux-aligned quad meshes with the same topologies as the typical flux contours in JOREK for Tokamak configuration MAST.

*Y. Liu (Corresponding author, email: liuyzz@mail.ustc.edu.cn)

1 Introduction

Mesh generation has been investigated for many years and plays a fundamental role in many fields, such as geometric processing and the finite element method. There is a rich literature on two-dimensional mesh generation. A thorough survey can be found in [Bommes et al.(2012)] [Armstrong et al.(2015)] and other recent works [Chen et al.(2019)] [Lei et al.(2020)] [Xiao et al.(2020)]. For quad-mesh generation, there is a specified requirement in Tokamak experiments [Guillard et al.(2018)]. In the following, we briefly present the mesh generation background in Tokamak simulation.

The understanding of magnetohydrodynamic (MHD) instability is essential for the optimization of magnetically confined plasma in Tokamak. Numerical simulation plays an important role in the investigation of the nonlinear behavior of these instabilities and the interpretation of experimental observations [Czarny and Huysmans(2008)]. MHD instability in Tokamak plasma is believed to be the following quasi bidimensional [Guillard et al.(2018)]: the fluctuation scales in the direction parallel to the magnetic field are typically 3 to 4 orders of magnitude larger than those in the transverse plane. It is important to consider a sufficiently fine spatial resolution in the transverse direction that makes the use of grids/meshes, which are adapted to the magnetic flux surfaces in each 2D poloidal plane particularly attractive, i.e., flux-aligned quad mesh generation is important for further MHD simulation in Tokamak plasma. This has led to a number of state-of-the-art numerical codes in various field- or flux-aligned grid techniques [Schneider et al.(2006)] [Dudson and Leddy(2017)] [Guillard et al.(2018)] [Tamain et al.(2016)] [Chang et al.(2009)] [Stegmeir et al.(2018)] [Pamela et al.(2019)]. However, research on the existence of flux-aligned block-structured meshes is rare. Theoretical analysis helps to improve the robustness of existing codes. Classical Morse theory [Spivak and Wells(1963)] is a natural way to build a flux-aligned block-structured mesh generation algorithm [Guillard et al.(2018)]. Regarding the application of MHD simulation, generally, a flux function is not a Morse function.

(1) Regarding the distribution of critical points, Morse functions are functions with special isolated critical points. The shape of the poloidal cross-section of the averaged flux surfaces in the poloidal planes is governed by magnetic equilibrium, i.e., given by solving the Grad-Shafranov equation. The distribution of isolated critical points of the Grad-Shafranov equation solution is more complex than that of the Morse functions [Alessandrini and Magnanini(1992)] [Arango and Gómez(2011)] [Cheeger et al.(2015)].

(2) The smoothness of Morse functions is at least C^2 on physical domains. In MHD simulation, the numerical solutions of the Grad-Shafranov equation are C^1 spline surfaces [JOREK-Team(2008)] [Czarny and Huysmans(2008)]. A flux is a C^1 spline surface defined over a mesh with an irregular topology, such as the structure illustrated in Figure 1.

In this paper, we assume that the input of flux-aligned quad mesh generation is

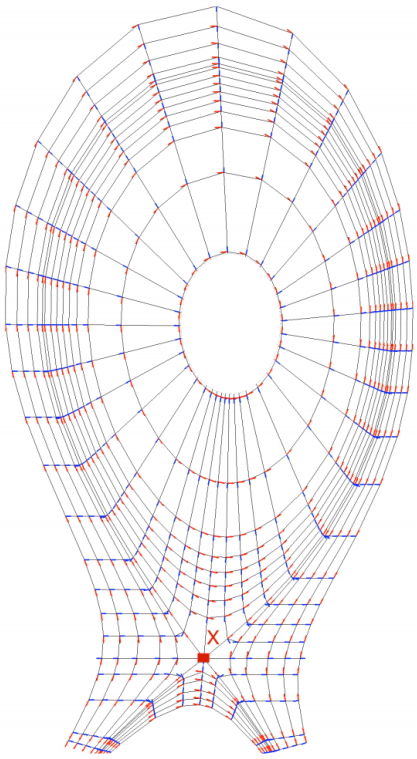


Figure 1: A quad mesh with an irregular topology at the X-point.

a physical domain and a flux surface with a set of isocurves. Flux surfaces $F(x, y)$s are generally C^1 functions with C^2 piecewise on a physical domain Ω. Numerical solutions of the MHD simulation [Czarny and Huysmans(2008)] are in this class of flux functions. In addition, following [Guillard et al.(2018)], we suppose $F(x, y)|_{\partial\Omega}$ is a constant. The contributions of this paper are as follows:

(1) We present the existence theory of a flux-aligned quad mesh with any given isocurves.

(2) We design a high-order algorithm to verify the flux-aligned quad mesh generation theory.

(3) Numerical experiments on generating flux-aligned quad meshes of interest by simulating MHD in the Tokamak are given.

The structure of the remainder of this paper is listed as follows. In Section 2, we discuss how to decompose physical domains such that there is a flux-aligned quad mesh with any given isocurves. In Section 3, we present a framework for a high-order algorithm to generate flux-aligned quad meshes. In Section 4, numerical experiments on the MHD simulation of Tokamak plasma are conducted in Section 2. Then, we conclude this paper in Section 5 by indicating directions for future work, and the proofs of lemmas are collected as an appendix in Section 6.

2 Flux-aligned quad meshes existence theory

In this section, we consider a flux-aligned parameterization, which is the continuous form of flux-aligned quad meshes. This concept contains all the possible of isocurves. It allows us to follow any given level set, which is concretely defined as follows.

Definition 2.1. *$\mathcal{P}$ is called a flux-aligned parameterization with respect to a flux function $F(x, y)$, if*

$$\mathcal{P} : \Sigma \longrightarrow \Omega$$

such that the s-curves of $\mathcal{P}(s, t)$ are iso-curves of $F(x, y)$. s-curves and t-curves are perpendicular to each other, where an s(or t)-curve is $\mathcal{P}(s, t)$, s is a constant and $\Sigma = [s_0, s_1] \times [t_0, t_1]$.

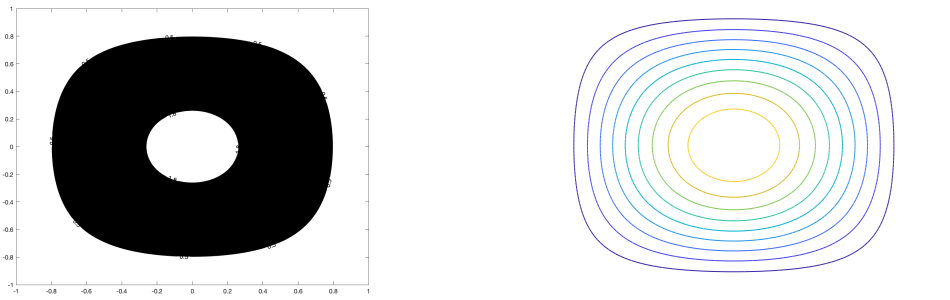
When we generate a flux-aligned quad mesh with a given iso-curve set, suitable s-parameters and t-parameters are formed according to the isocurves. Thus, we consider the existence of a flux-aligned parameterization that is well defined along common edges between adjacent subdomains. Generally, a flux-aligned parameterization does not exist globally. In Section 2.1, we present how to decompose a physical domain into subdomains and demonstrate the existence of a flux-aligned parameterization on each subdomain in Section 2.2.

2.1 Critical point-based domain decomposition

In this section, a domain decomposition approach is presented to generate a flux-aligned parameterization on a physical domain. The necessity of domain decomposition is explained in Example 2.1. Then, the details of the decomposition approach are presented.

Example 2.1.

- *(a)* $F(x,y) = (2-x^2-y^2)(1-x^2)(1-y^2)$ *is a flux function, where* $(x,y) \in \Omega, \Omega = \{(x,y) \in \mathbb{R}^2 | 0.5 \leq F(x,y) \leq 1.8\}$ *shown in Figure 2(a). In Figure 2(b), the isocurves can be contained by a flux-aligned parameterization.*



(a) $\Omega = \{(x,y) \in \mathbb{R}^2 | 0.5 \leq F(x,y) \leq 1.8\}$ (b) Isocurves of $F(x,y)$ on Ω

Figure 2: Isocurves of $F(x,y)$ and Ω in Example 2.1(a)

- *(b)* $F(x,y) = xy(1-x^2)(1-y^2)$ *is a flux function, where* $(x,y) \in [-1,1] \times [-1,1]$. *In Figure 3, there is no flux-aligned parameterization containing isocurves around the central point* $(0,0)$.

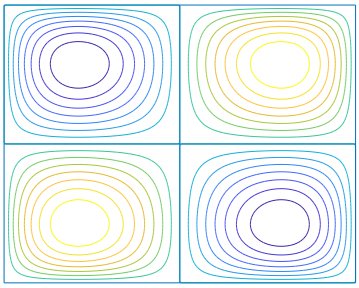


Figure 3: $F(x,y) = xy(1-x^2)(1-y^2)$ in Example 2.1(b)

In the following, the critical point-based domain decomposition method is presented.

Definition 2.2. *Let $F(x,y)$ be a C^1 function. $p \in \Omega$ is called* **a critical point** *of $F(x,y)$ if*

$$\frac{\partial F}{\partial x}|_p = \frac{\partial F}{\partial y}|_p = 0.$$

In Example 2.1 (a), there is no critical point. In Example 2.1 (b), the central point $(0,0)$ is a critical point. We note that there is no flux-aligned parameterization. In the following, flux functions are always supposed to be C^1 and C^2 piecewise, which includes numerical solutions of MHD simulation in Tokamaks by JOREK [JOREK-Team(2008)] [Czarny and Huysmans(2008)].

Definition 2.3. *On a physical domain Ω, let $\mathcal{A} = \{p_i\}_{p_i \in \mathcal{I}}$ be all the critical points of $F(x,y)$ and*

$$\mathcal{V} = \{F(p_i) | p_i \in \mathcal{A}\}.$$

Ω can be decomposed by curves

$$\mathcal{C} = \{(x,y) : F(x,y) = c_i, c_i \in \mathcal{V}\} \cup \{\partial\Omega\},$$

into connected components called **subdomains**. *This type of decomposition is called* **critical point-based domain decomposition**.

Remark 2.1. Finiteness Hypothesis: *Regarding the computation, there is a finiteness hypothesis. We list the structures in the decomposition process, and we assume that their numbers are finite in this paper.*

1. *Connected components in $\mathcal{C}$, i.e., curves and isolated points in $\mathcal{C}$;*
2. *Subdomains of critical point-based domain decomposition;*
3. *Nonsmooth points of the curves in $\mathcal{C}$.*

In the following, we illustrate the critical point-based decompositions of Example 2.1(b) in Example 2.2 and explain the subdomains, isolated points in $\mathcal{C}$ and nonsmooth points of curves in $\mathcal{C}$.

Example 2.2. *Let $F(x,y) = xy(1-x^2)(1-y^2), (x,y) \in [-1,1] \times [-1,1]$.*

$$\begin{aligned}\mathcal{A} =& \{(0,0), (0,1), (0,-1), (\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}), (\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}), (-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}), (-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}),\\ & (0,1), (0,-1), (1,1), (-1,1), (1,-1), (-1,-1)\};\\ \mathcal{V} =& \{0, \frac{4}{27}, -\frac{4}{27}\},\end{aligned}$$

then

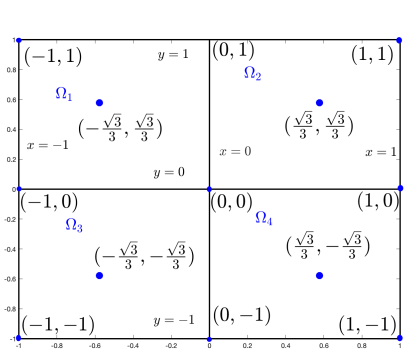
$$\begin{aligned}\mathcal{C} =& \{(x,y) : x = 0, \text{ or } y = 0, \text{ or } x = 1, \text{ or } x = -1, \text{ or } y = 1, \text{ or } y = -1\}\\ & \cup \{(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}), (\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}), (-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}), (-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3})\}.\end{aligned}$$

In Figure 4(a), the critical point-based decomposition is obtained by removing $\mathcal{C}$ from Ω. Notably, there are

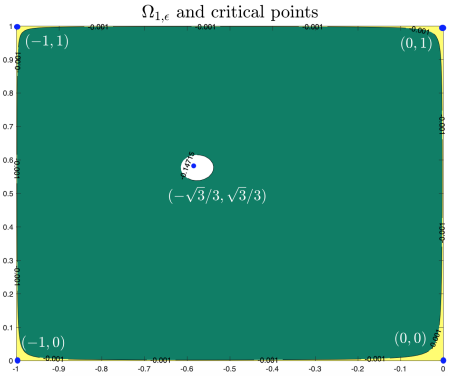
(1) 4 isolated points: $(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}), (-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}), (-\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3})$, *and* $(-\frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3})$;

(2) 9 nonsmooth points: $(1,1), (1,-1), (-1,1), (-1,-1), (0,1), (0,-1), (0,0), (-1,0)$, *and* $(1,0)$;

(3) Ω *is divided into 4 subdomains denoted as* $\Omega_1, \Omega_2, \Omega_3$, *and* Ω_4.



(a) The critical point-based decomposition



(b) $\Omega_{1,\epsilon} = \{(x,y) \in \Omega_1 : -\frac{4}{27} + \epsilon \leq F(x,y) \leq 0 - \epsilon\}, \epsilon = \frac{1}{1000}$

Figure 4: (a): Critical point-based decomposition in Example 2.2; (b): $\Omega_{1,\epsilon}$ (in green)

In the next section, the theoretical analysis of the existence of a flux-aligned parameterization on each subdomain is presented.

2.2 Existence of a flux-aligned parameterization on each subdomain

In the following, we list several lemmas for constructing flux-aligned parameterizations in Theorem 2.6.

(1) Based on the implicit function theorem, we can obtain the following lemma.

Lemma 2.1. *Let c not be a critical value of $F(x,y)$, i.e., $c \notin \mathcal{V}$. Then, $\mathcal{C} = \{(x,y) \in \Omega : F(x,y) = c\}$ consists of a set of C^1 loops without self-intersection.*

(2) For a curve in Lemma 2.1, there is a parameterization.

Lemma 2.2. *Let C be a curve of $\mathcal{C}$ defined in Lemma 2.1. Then, there is a parameterization ϕ of C,*

$$\begin{aligned}\phi :&[s_0, s_1] \longrightarrow C\\ &s \mapsto (x, y),\end{aligned}$$

such that

$$\frac{d\phi}{ds} = \frac{(-F_y, F_x)}{||\nabla F||^2}, \text{ or } \frac{d\phi}{ds} = \frac{(F_y, -F_x)}{||\nabla F||^2}. \tag{1}$$

(3) For the values of $F(x, y)$ on the boundaries of a subdomain Ω_i, we have the following lemma.

Lemma 2.3. *There are only two different values of $F(x, y)$ on $\partial\Omega_i$ and they are a boundary value or a critical value of $F(x, y)$.*

For instance, in Example 2.2, there are four subdomains $\{\Omega_i : i = 1, 2, 3, 4\}$, where

$$F|_{\partial\Omega_1} = \{0, -\frac{4}{27}\},\ F|_{\partial\Omega_2} = \{0, \frac{4}{27}\},\ F|_{\partial\Omega_3} = \{0, \frac{4}{27}\},\ F|_{\partial\Omega_4} = \{0, -\frac{4}{27}\}.$$

Here, $0, \frac{4}{27}$, and $-\frac{4}{27}$ are the critical values of $F(x, y)$; and 0 is also the boundary value of $F(x, y)$. Based on Lemma 2.3, we present the following definition.

Definition 2.4. *Let $c_{i,1}, c_{i,2}$ $(c_{i,1} < c_{i,2})$ be the values of $F(x, y)$ on $\partial\Omega_i$. Define*

$$\Omega_{i,\epsilon} := \{(x, y) \in \Omega_i : c_{i,1} + \epsilon \le F(x, y) \le c_{i,2} - \epsilon\}$$

where ϵ is small enough such that $c_{i,1} + \epsilon < c_{i,2} - \epsilon$.

By Definition 2.4, $\Omega_{i,\epsilon} \subset \Omega_i$. The following gives an example of $\Omega_{i,\epsilon}$.

Example 2.3. *Let us consider $F(x, y)$ in Example 2.2. $F|_{\partial\Omega_1} = \{-4/27, 0\}$.*

$$\Omega_{1,\epsilon} = \{(x, y) \in \Omega_1 : -\frac{4}{27} + \frac{1}{1000} \le F(x, y) \le 0 - \frac{1}{1000}\},$$

where $\epsilon = \frac{1}{1000}$. $\Omega_{1,\epsilon}$ is shown in Figure 4(b), where we can find that
(i) there is no critical point in $\Omega_{1,\epsilon}$;
(ii) $\Omega_{1,\epsilon}$ is homeomorphic to an annulus, and its boundaries consist of two closed curves defined by $F(x, y) = -\frac{4}{27} + \frac{1}{1000}$ and $F(x, y) = 0 - \frac{1}{1000}$.

(4) This phenomenon can be explained by the following lemma.

Lemma 2.4. *$\Omega_{i,\epsilon}$ is homeomorphic to an annulus, and its boundary consists of two closed curves defined by $F(x,y) = c_{i,1} + \epsilon$ and $F(x,y) = c_{i,2} - \epsilon$, where $c_{i,1}, c_{i,2}$ and ϵ are defined in Definition 2.4.*

The proofs of Lemmas 2.2, 2.3 and 2.4 are given in the Appendix. Now, we prepare to prove the existence of flux-aligned parameterizations on a subdomain. First, we prove that there is a flux-aligned parameterization on $\Omega_{i,\epsilon}$ in Lemma 2.5, and then a flux-aligned parameterization is constructed on Ω_i in Theorem 2.6.

Lemma 2.5. *For $\Omega_{i,\epsilon}$, there is a flux-aligned parameterization $\mathcal{P}_{i,\epsilon}$*

$$\mathcal{P}_{i,\epsilon} : \Sigma_{i,\epsilon} \longrightarrow \Omega_{i,\epsilon},$$

where $\Sigma_{i,\epsilon} = [s_0, s_1] \times [0, T_p]$.

Proof. Based on Lemma 2.4, $\partial\Omega_{i,\epsilon} = C_1 \cup C_2$, where C_1 and C_2 are boundary curves and $F(C_1) = c_{i,1} + \epsilon, F(C_2) = c_{i,2} - \epsilon$, where $c_{i,1} + \epsilon < c_{i,2} - \epsilon$. By Lemma 2.2, there is a parameterization of C_1

$$\phi : [s_0, s_1] \longrightarrow C_1,$$

such that

$$\frac{d\phi}{ds} = \frac{(-F_y, F_x)}{||\nabla F||^2}.$$

Moreover, $\forall p_s \in C_1$, consider

$$\begin{cases} \dfrac{d\psi}{dt} = \dfrac{\nabla F}{||\nabla F||^2}, \\ \psi(0) = p_s. \end{cases} \tag{2}$$

According to the Picard-Lindelöf theorem [Teschl(2012)], there is a unique solution

$$\psi_{p_s} : [0, t_{p_s}] \longrightarrow \mathbb{R}^2$$

that satisfies (2), and $\psi_{p_s}(t_{p_s})$ is a boundary point of $\Omega_{i,\epsilon}$, $F(\psi_{p_s}(t_{p_s})) > F(\psi_{p_s}(0))$ because $\nabla F \neq \mathbf{0}$, i.e., $\psi_{p_s}(t_{p_s}) \in C_2$. Define

$$\mathcal{P}_{i,\epsilon}(s,t) = \psi_{\phi(s)}(t), \ (s,t) \in \Sigma_{i,\epsilon} \tag{3}$$

where

$$\Sigma_{i,\epsilon} = \cup_{s \in [s_0, s_1]} \{s\} \times [0, t_{p_s}] \subset \mathbb{R}^2.$$

According to the definition of ψ,

$$\frac{\partial \mathcal{P}_{i,\epsilon}}{\partial t} = \frac{\nabla F}{||\nabla F||^2}.$$

Moreover, for a fixed s,

$$\frac{\partial F(\mathcal{P}_{i,\epsilon}(s,t))}{\partial t} = \nabla F \cdot \frac{\partial \mathcal{P}_{i,\epsilon}}{\partial t} = 1.$$

Thus,

$$F(\mathcal{P}_{i,\epsilon}(s,t)) = t + h(s), \tag{4}$$

where $h(s)$ is a function of s. According to the definition of $\mathcal{P}_{i,\epsilon}(s,t)$,

$$\mathcal{P}_{i,\epsilon}(s,0) = \psi_{\phi(s)}(0) = \phi(s).$$

Compared to (4),

$$F(\mathcal{P}_{i,\epsilon}(s,0)) = F(C_1) = c_{i,1} + \epsilon,$$

that is,

$$h(s) = c_{i,1} + \epsilon,$$

and

$$F(\mathcal{P}_{i,\epsilon}(s,t)) = t + c_{i,1} + \epsilon \tag{5}$$

i.e., for a fixed t_0, $F(\mathcal{P}_{i,\epsilon}(s,t_0))$ is a const $t_0 + c_{i,1} + \epsilon$. According to this result,

$$\frac{\partial F(\mathcal{P}_{i,\epsilon})}{\partial s} = \nabla F \cdot \frac{\partial \mathcal{P}_{i,\epsilon}}{\partial s} = 0.$$

Thus,

$$\frac{\partial \mathcal{P}_{i,\epsilon}}{\partial t} \perp \frac{\partial \mathcal{P}_{i,\epsilon}}{\partial s}. \tag{6}$$

According to (5),

$$F(\mathcal{P}_{i,\epsilon}(s,t_{p_s})) = t_{p_s} + c_{i,1} + \epsilon.$$

Moreover, $\mathcal{P}_{i,\epsilon}(s,t_{p_s}) = \psi_{\phi(s)}(t_{p_s}) \in C_2$, i.e.,

$$F(\mathcal{P}_{i,\epsilon}(s,t_{p_s})) = c_{i,2} - \epsilon.$$

Thus $\forall s \in [s_0, s_1]$,

$$t_{p_s} = c_{i,2} - c_{i,1} - 2\epsilon := T_p,$$

and $\Sigma_{i,\epsilon} = [s_0, s_1] \times [0, T_p]$. Thus, based on (5) and (6), $\mathcal{P}_{i,\epsilon}$ is a flux-aligned parameterization of $\Omega_{i,\epsilon}$. □

Remark 2.2. *In the proof of Lemma 2.5, C_1 can be replaced by any isocurve on $\Omega_{i,\epsilon}$. For instance, take $c = (c_{i,1} + c_{i,2})/2$ and $c_{i,1} + \epsilon < c < c_{i,2} - \epsilon$, where ϵ (> 0) is sufficiently small. Considering the curve $C \subset \Omega_{i,\epsilon}$ such that $F(C) = c$, according to Lemma 2.2, there is a unique parameterization of C such that*

$$\phi : [s_0, s_1] \longrightarrow C$$

and

$$\frac{d\phi}{ds} = \frac{(-F_y, F_x)}{||\nabla F||^2}.$$

ϕ can be instead the one in Lemma 2.5, and there are constants $T_0, T_1 > 0$ such that

$$\mathcal{P}_{i,\epsilon} : [s_0, s_1] \times [-T_0, T_1] \longrightarrow \Omega_{i,\epsilon},$$

and $\mathcal{P}_{i,\epsilon}(s, 0) = \phi(s)$.

We are ready to prove the existence of a flux-aligned parameterization on a subdomain.

Theorem 2.6. *There is a flux-aligned parameterization $\mathcal{P}_i$,*

$$\begin{aligned} \mathcal{P}_i : &\Sigma_i \longrightarrow \Omega_i \\ &(s, t) \mapsto (x, y), \end{aligned}$$

where there are s_0, s_1, t_0, and $t_1 (> 0)$ such that $\Sigma_i = [s_0, s_1] \times [t_0, t_1]$.

Proof. According to the definition of $\Omega_{i,\epsilon}$, for $\epsilon > 0$, $\Omega_{i,\epsilon} \subset \Omega_i$ and

$$\Omega_i = \cup_{\epsilon>0} \Omega_{i,\epsilon}.$$

For $\epsilon_1 > \epsilon_2 > 0$, $\Omega_{i,\epsilon_1} \subset \Omega_{i,\epsilon_2}$. Select $\epsilon_0 > 0$ such that $\emptyset \neq \Omega_{i,\epsilon_0} \subset \Omega_i$. According to Lemma 2.4, $\partial\Omega_{i,\epsilon_0} = C_1 \cup C_2$. Suppose $\partial\Omega_i = C_1 \cup C_2$ and $F(C_i) = c_i$ $(i = 1, 2)$. Take $c = (c_1 + c_2)/2$, where

$$C = \{(x, y) \in \Omega_i : F(x, y) = c\}.$$

Suppose that $\epsilon_n = \epsilon_0/2^n$, where $n = 0, 1, \cdots$. Then, $C \subset \Omega_{i,\epsilon_n}$. Based on Lemma 2.5,

$$\mathcal{P}_{i,\epsilon_n}(s, t) : [s_0, s_1] \times [-t_{1,n}, t_{2,n}] \longrightarrow \Omega_{i,\epsilon_n}.$$

Since $\Omega_{i,\epsilon_n} \subset \Omega_{i,\epsilon_{n+1}}$,

$$t_{1,n+1} \geq t_{1,n}, \ t_{2,n+1} \geq t_{2,n}.$$

Let

$$t_1 = \lim_{n\to\infty} t_{1,n}, \ t_2 = \lim_{n\to\infty} t_{2,n}.$$

Define $\mathcal{P}_i : \Sigma_i \longrightarrow \Omega_i$ as

$$\mathcal{P}_i(s,t) = \mathcal{P}_{i,\epsilon_n}(s,t),$$

when $(s,t) \in [s_0, s_1] \times [-t_{1,n}, t_{2,n}] \subset \Sigma_i$, where $\Sigma_i = [s_0, s_1] \times (-t_1, t_2) \subset \mathbb{R}^2$. According to the uniqueness of Picard-Lindelöf theorem, $\mathcal{P}_{i,\epsilon_{n+1}}|_{\Omega_{i,\epsilon_n}} = \mathcal{P}_{i,\epsilon_n}$. Thus $\mathcal{P}_i$ is well-defined. Moreover, according to Lemma 2.5,

$$\frac{\partial \mathcal{P}_i}{\partial t} = \frac{\nabla F}{||\nabla F||^2}, \frac{\partial \mathcal{P}_i}{\partial s} \perp \frac{\partial \mathcal{P}_i}{\partial t},$$

In addition, for $t \in (-t_1, t_2)$, $F(\mathcal{P}_i(s, t_0))$ is a constant, i.e., $\mathcal{P}_i(s, t_0)$ is a flux-aligned parameterization of Ω_i. □

Remark 2.3. *(1) In Theorem 2.6, the regularity of flux functions is C^1 with C^2 piecewise, which is a weaker requirement of the regularity of Morse functions. This type of flux function includes numerical solutions of the MHD simulation of Tokamak plasma. The flux-aligned quad mesh generation algorithm [Guillard et al.(2018)] is a special case presented here.*

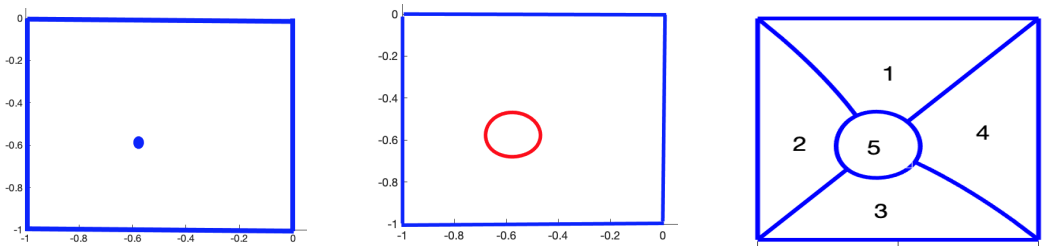
(2) Consider the boundary of Ω_i in Theorem 2.6. Although $\partial\Omega_{i,\epsilon}$ are smooth curves, there may be nonsmooth points and isolated points on $\partial\Omega_i$. For example, $\partial\Omega_2$ has 4 nonsmooth points $(0,0), (1,0), (0,1)$, and $(1,1)$ and an isolated point $(\sqrt{3}/3, \sqrt{3}/3)$ in Figure 4(a). Flux-aligned parameterization degenerates at nonsmooth points and isolated points. This degeneration will cause singularity of the flux-aligned quad-mesh generation. Thus, it is reasonable that there are algorithms to eliminate degeneration. In fact, these codes have been designed. For example, edge plasma codes (SOLPS, FBGKI, TOKAM3X, and SOLEDGE) and GYSELA exclude a central part of the core plasma around the magnetic axis. The boundary of this central part will be considered a constant flux surface.

In the following, we present the methods to eliminate degeneration at isolated points and nonsmooth points based on the present codes.

Isolated Point Elimination: The flux-aligned parameterizations degenerate at isolated points. That is, for this case, following what [Huysmans and Czarny(2007)] has done, we remove an isolated point p_I using an isocurve with $F(x,y) = c + \epsilon$, where $c = F(p_I)$ and ϵ is a small constant. We adopt the approach to eliminate degeneration at isolated points in Figure 5(b).

Nonsmooth Point Elimination: We introduce gradient integral curves by taking nonsmooth points as their initial points to split subdomains into regular subdomians. We adopt the approach to eliminate degeneration at the nonsmooth points shown in Figure 5(c).

Definition 2.5. *The subdomain with $F(x,y) = c + \epsilon$ as its boundary is called **a central subdomain**. In addition to central subdomains, others are called **regular subdomains**.*



(a) (b) (c)

Figure 5: Elimination of isolated points and nonsmooth points on $[-1,0]\times[-1,0]$ with $xy(1-x^2)(1-y^2)$ as the flux function

For instance, in Figure 5(c), the fifth subdomain is a central subdomain, and the others are regular subdomains.

Remark 2.4. *A compatible flux-aligned quad-mesh on a global computational domain can be obtained by choosing appropriate parameters on adjacent subdomains.*

3 Framework of the flux-aligned quad-mesh generation algorithm

In this section, the framework of the flux-aligned quad mesh generation algorithm is presented in Section 3.1, and an example is illustrated in Section 3.2.

3.1 Algorithm framework

The flux-aligned quad mesh generation algorithm is divided into two subalgorithms. The first subalgorithm is used to decompose a given physical domain into subdomians. The second subalgorithm is used to generate an order of these subdomains. Following this order, a conformal quad mesh is obtained. In the following, we present the framework of this algorithm.

Algorithm: Flux-aligned quad-mesh generation algorithm

Input: A physical domain Ω and a flux function $F(x,y)$
Output: A flux-aligned quad-mesh defined on Ω
Subalgorithm 1: Domain decomposition algorithm
Subalgorithm 2: Quad mesh generation algorithm

The flux-aligned quad mesh generation algorithm is presented in the following. To execute this algorithm, we need to generate two basic curves.

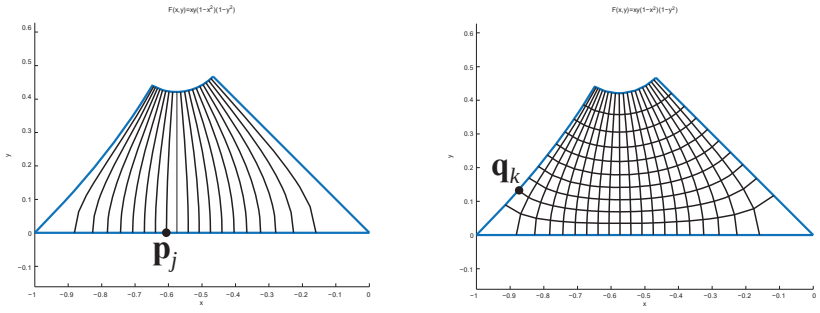
1. Gradient Integral Curve Generation: Computing gradient integral curves, it is defined by

$$\begin{aligned} \frac{d\mathbf{X}}{dt} &= \frac{\nabla F}{||\nabla F||^2}, \\ \mathbf{X}(0) &= \mathbf{p}_0, \end{aligned} \tag{7}$$

where $\mathbf{p}_0 \in \Omega_i$.

For a regular subdomain, choose $\mathbf{p}_j$ on an isocurve boundary as an initial value, where $j = 1, 2, \cdots, n_i$. We can solve (7) and obtain the gradient integral curves $\mathbf{X}_1(t), \mathbf{X}_2(t), \cdots, \mathbf{X}_{n_i}(t)$ shown in Figure 6(a).

2. Isocurve Generation: Choose $\mathbf{q}_k$ on a gradient integral curve boundary $\mathbf{X}_1(t)$ (or $\mathbf{X}_{n_i}(t)$). Based on $\mathbf{X}_1(t)$ (or $\mathbf{X}_{n_i}(t)$) and $\mathbf{q}_k$, we compute the parameter t_k for each $\mathbf{q}_k$, where $k = 1, 2, \cdots, m_i$. Then, $\mathbf{X}_j(t_k)$ and $\mathbf{X}_{j+1}(t_k)$ are connected, where $j = 1, 2, \cdots, n_i - 1$ and $k = 1, 2, \cdots, m_i$ as shown in Figure 6(b).



(a) (b)

Figure 6: The gradient integral curves and a flux-aligned quad-mesh on a regular subdomain Ω_i

The first and second subalgorithms are presented in the following. The details of the first subalgorithm are explained in Section 2.1.

Subalgorithm 1: Domain decomposition algorithm

Input: $F(x, y), \Omega$
Output: regular subdomains: $\Omega_1^r, \Omega_2^r, \cdots, \Omega_n^r$; central subdomains: $\Omega_1^c, \Omega_2^c, \cdots, \Omega_m^c$
1: The critical point-based decomposition
2: Isolated point elimination
3: Nonsmooth point elimination

Remark 3.1. *The mesh generation speed and distribution are important for application in MHD simulation. In this paper, aligned parameterizations are continuous.*

Subalgorithm 2: Quad mesh generation algorithm

Input: $F(x,y)$; regular subdomains: $\Omega_1^r, \Omega_2^r, \cdots, \Omega_n^r$; central subdomains: $\Omega_1^c, \Omega_2^c, \cdots, \Omega_m^c$.
Output: A flux-aligned quad-mesh defined on Ω.
1: Ordering subdomains: Order regular subdomains in two ways (a gradient-induced graph and an iso-curve induced graph), and then order central subdomains.
2: Flux-aligned quad mesh generation on regular subdomains: Based on the gradient-induced graph and isocurves induced graph, we generate gradient integal curves and isocurves on each subdomain.
3: Quad mesh generation on central subdomains: The algorithm was developed in [Wu et al.(2017)].

Thus, it can induce different flux aligned quad meshes. The mesh distribution of flux aligned quad meshes is determined by flux functions and the initial points p_j and q_k.

3.2 Example of generating flux-aligned quad-meshes

In this section, an example is used to explain the flux-aligned quad-mesh generation algorithm.

Example 3.1. *Input:* $\Omega = [-1,1] \times [-1,1]$, $F(x,y) = xy(1-x^2)(1-y^2)$.

Subalgorithm 1. *Domain decomposition algorithm:*

(1) Critical point-based decomposition: Based on the computation in Example 2.2, we present the critical point-based decomposition and obtain the 4 connected open set $\Omega_1, \Omega_2, \Omega_3$ and Ω_4 shown in Figure 4(a).

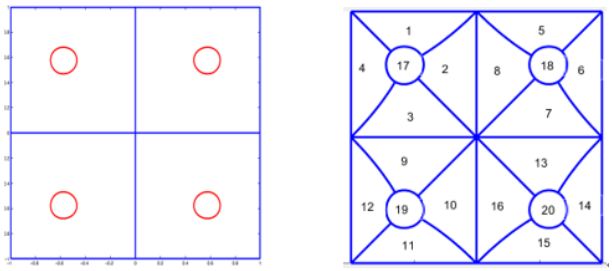


Figure 7: Isolated point elimination (a), nonsmooth point elimination and the indices of subdomains (b)

(2) Isolated point elimination: There are 4 isolated points $(-\sqrt{3}/3, \sqrt{3}/3)$, $(\sqrt{3}/3, \sqrt{3}/3)$, $(-\sqrt{3}/3, -\sqrt{3}/3)$, and $(\sqrt{3}/3, -\sqrt{3}/3)$. For an isolated point, we remove this point

using a central domain whose boundary is an isocurve. The subdomains after removing these isocurves are shown in Figure 7 (a).

(3) Nonsmooth point elimination: For nonsmooth points, gradient integral curves are introduced. The subdomains are shown in Figure 7 (b).

Subalgorithm 2. *Quad mesh generation algorithm:*

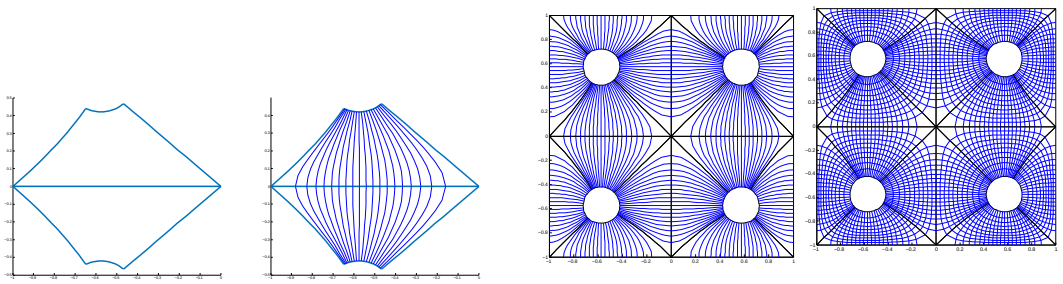
(1) Ordering: In Figure 7 (b), we assign an index to each subdomain.

Gradient induced Graph (G-Graph)*: Following the gradient direction, the relationship graph is presented as*

$$3 \longrightarrow 9; 2 \longrightarrow 8; 7 \longrightarrow 13; 10 \longrightarrow 16;$$
$$1;\ 4;\ 5;\ 6;\ 12;\ 11;\ 15;\ 14.$$

Isocurves induced Graph (I-Graph)*: Following the iso-curve direction, the relationship graph is presented as*

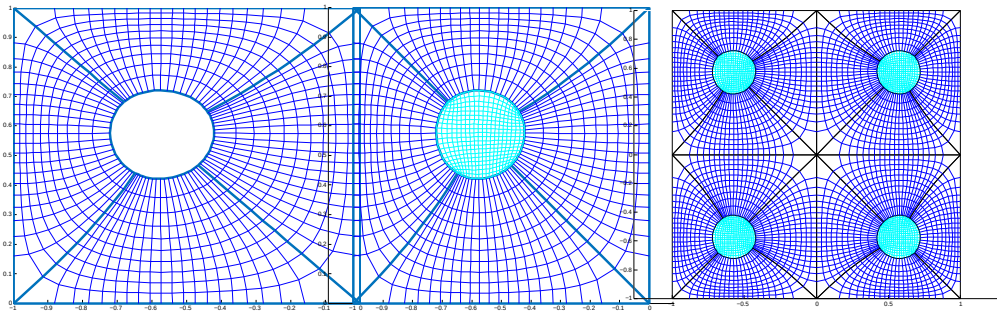
$$1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4; 5 \longrightarrow 6 \longrightarrow 7 \longrightarrow 8;$$
$$9 \longrightarrow 10 \longrightarrow 11 \longrightarrow 12; 13 \longrightarrow 14 \longrightarrow 15 \longrightarrow 16.$$



(a) (b) (c) (d)

Figure 8: (a) 3rd and 9th subdomains; (b) initial points and gradient integral curves; (c) gradient integral curves; (d) for each subgraph in the I-Graph, initial points are given following the isocurves direction. Iso-curves are computed.

(2) Flux-aligned quad-mesh generation on regular subdomains: For each subgraph in the ***G-Graph****, initial points are given following the gradient direction. The gradient integral curves are computed. For instance, with the given initial points on an isocurve edge, curves can be computed, as shown in Figure 8(b), For each subgraph, we use the same method to generate integral curves, as shown in Figure 8(c). Then,*



(a) Boundary points (b) A local conformal quad-mesh (c) A global conformal quad-meshes

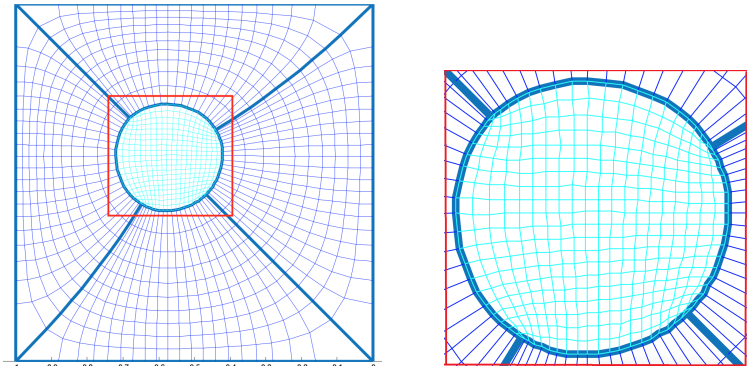
Figure 9: A conformal quad-mesh generated on a central subdomain

isocurves are generated following the ***I-Graph*** *shown in Figure 8(d). The quad-mesh is conformal because the initial points are given following the G-Graph and I-Graph.*

(3) Quad mesh generation on central subdomains: For each central subdomain, initial points are given by the endpoints of the gradient integral curves. Take the 17th subdomain as an example. After generating aligned quad-meshes on regular subdomains, there are several points on its boundary, as shown in Figure 9(a). Using the algorithm in [Wu et al.(2017)], as shown in Figure 9(b), a quad mesh can be generated and it is compatible with boundary points, i.e., the quad-mesh is conformal. Please see an enlarged view of the central subdomain in Figure 10. In Figure 9(c), all the quad mesh generation central subdomains are generated.

4 Runge-Kutta based algorithm and numerical experiments

In this section, to validate the theory in this paper, a specific algorithm is developed to solve gradient integral curves in Section 4.1. This algorithm is based on the high-order Runge-Kutta method to solve (7) numerically. Then, in Section 4.2, this algorithm is applied to generate flux-aligned quad meshes using the generation algorithm framework in Section 3.



(a) The local conformal quad-mesh

(b) An enlarged view of the central subdomain

Figure 10: An enlarged view of the central subdomain of the conformal quad mesh in Figure 9(b)

4.1 Runge-Kutta based gradient integral curve generation algorithm

If $\nabla F|_{\mathbf{p}_0} \neq \mathbf{0}$, the Runge-Kutta method can be used directly. Because of the initial points on $\partial\Omega_i$, $\nabla F|_{\mathbf{p}_0}$ might be zero. There is a theorem obtained by reparameterization to solve this problem.

Theorem 4.1. *Let* $\mathbf{X}_1$ *and* $\mathbf{X}_2$ *be curves in* $\mathbb{R}^3$. *They satisfy*

$$\frac{d\mathbf{X}_1}{ds} = \nabla F, \mathbf{X}_1(0) = \mathbf{p}_0,$$

and

$$\frac{d\mathbf{X}_2}{ds} = \frac{\nabla F}{||\nabla F||^2}, \mathbf{X}_2(0) = \mathbf{p}_0.$$

Then $\mathbf{X}_1$ *and* $\mathbf{X}_2$ *share the same locus with each other.*

The Runge-Kutta based gradient curve generation algorithm is presented in the following.

(1) If $\nabla F|_{\mathbf{p}_0} \neq \mathbf{0}$, solve

$$\frac{d\mathbf{X}}{dt} = \frac{\nabla F}{||\nabla F||^2}, \mathbf{X}(0) = \mathbf{p}_j,$$

using the Runge-Kutta method.

(2) If $\nabla F|_{\mathbf{p}_0} = \mathbf{0}$, solve

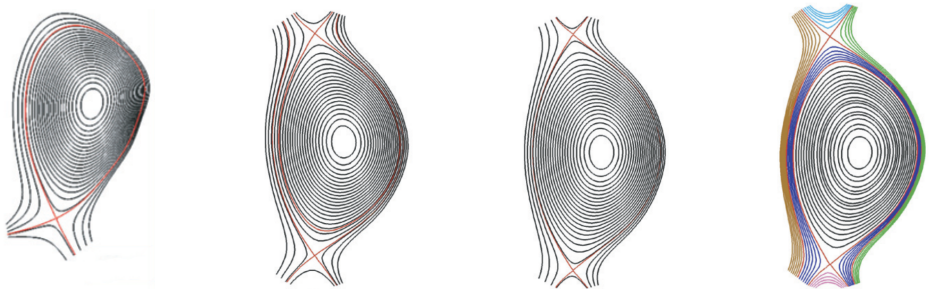
$$\frac{d\mathbf{X}'}{dt} = \nabla F, \mathbf{X}'(0) = \mathbf{p}_0,$$

and obtain a new point $\mathbf{p}_0' \in \Omega_i$ such that $\nabla F|_{\mathbf{p}_0'} \neq \mathbf{0}$. According to Theorem 4.1, $\mathbf{p}_0'$ is on $\mathbf{X}_j$. Thus it is used to instead of $\mathbf{p}_0$ as an initial value and solves

$$\frac{d\mathbf{X}_j}{dt} = \frac{\nabla F}{||\nabla F||^2}, \mathbf{X}_j(0) = \mathbf{p}_0',$$

4.2 Numerical Experiments

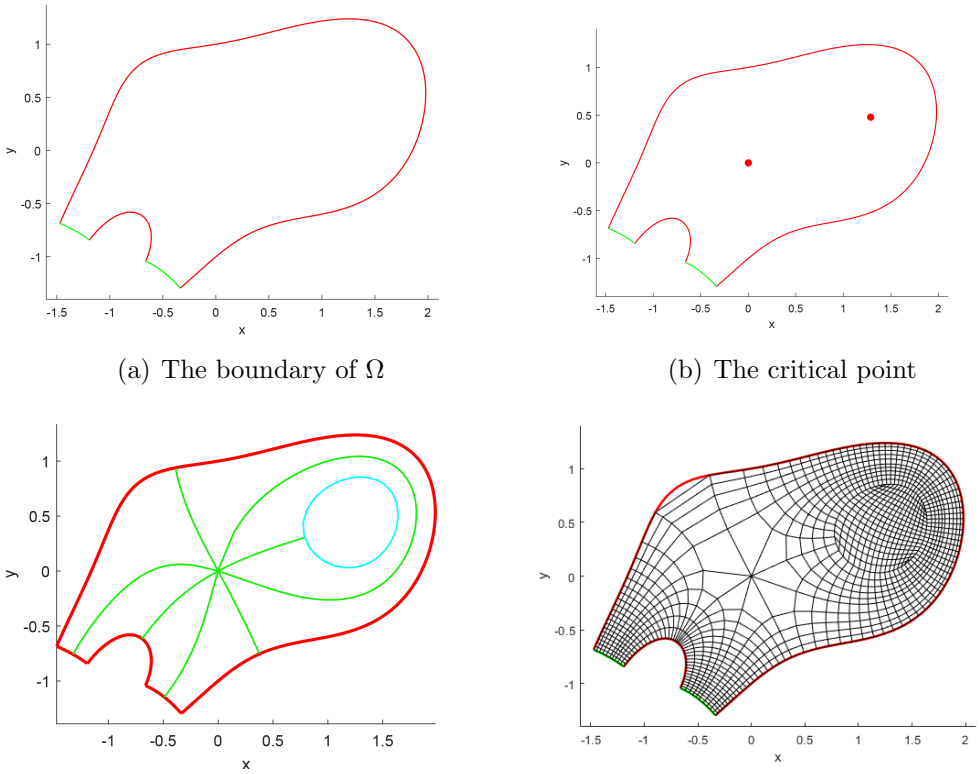
In this section, flux-aligned quad-meshes are generated. Their contours of single-null discharge, symmetric double-null discharge and nonsymmetric double-null discharge have the same topology as the typical flux contours in the MAST device in Figure 11. For simulation, the quad-mesh is aligned to equilibrium flux surfaces for the three regions of the core, the scrape-off layer (SOL) and the private region. Alignment along open flux surfaces in the SOL is important to accurately treat the fast parallel transport of energy along the magnetic field lines. Simulation of single-null discharge (SNDs) in MAST with the single-null contours of JOREK in Figure 11 (a). The double-null quad-meshes of JOREK are generated by poloidal flux contours for the three standard double-null cases: the upper double-null in Figure 11(b), the symmetric double-null in Figure 11(c) and the lower double-null in Figure 11(d).



(a) (b) (c) (d)

Figure 11: Typical flux contours for single-null discharge and double-null discharges in the MAST device [Pamela et al.(2013)].

Based on the algorithm framework in Section 3.1 and the Runge-Kutta based gradient integral curve algorithm in Section 4.1, Figures 12,13 and 14 present the numerical experiments, where the domain boundaries and critical points of the flux are as shown in Figure k(a). Using Subalgorithm 1, the domain is divided into regular and central subdomains, as shown in Figure k(b), and the flux-aligned quad mesh is shown in Figure k (c), where k =12,13 and 14.



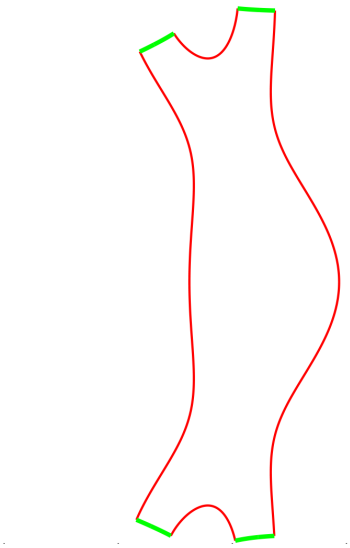
(a) The boundary of Ω

(b) The critical point

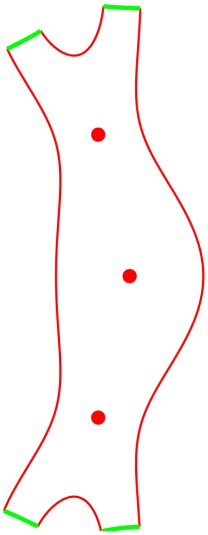
(c) Decomposition

(d) The flux-aligned quad mesh

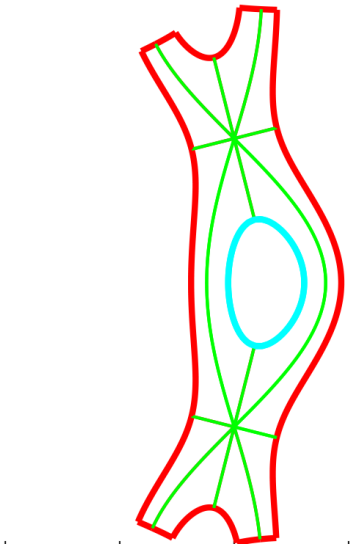
Figure 12: A single-null discharge flux contour experiment



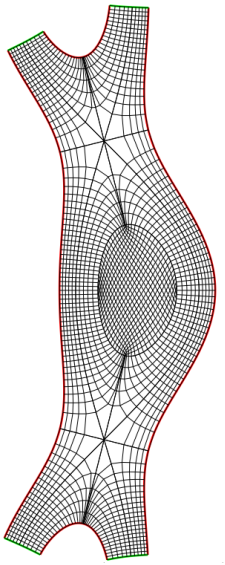
(a) The boundary of Ω



(b) The critical point

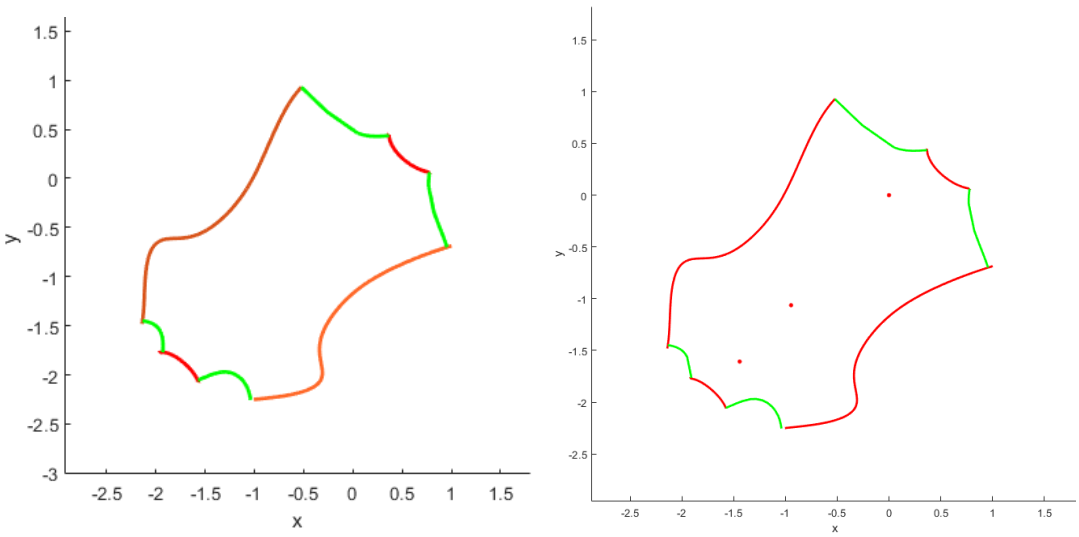


(c) Decomposition



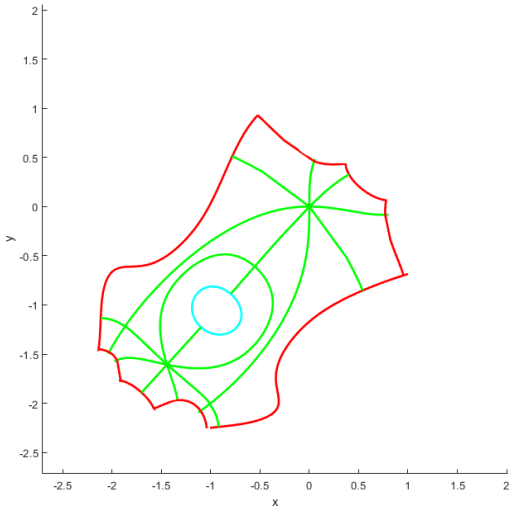
(d) The flux-aligned quad mesh

Figure 13: A symmetric double-null discharge flux contour experiment

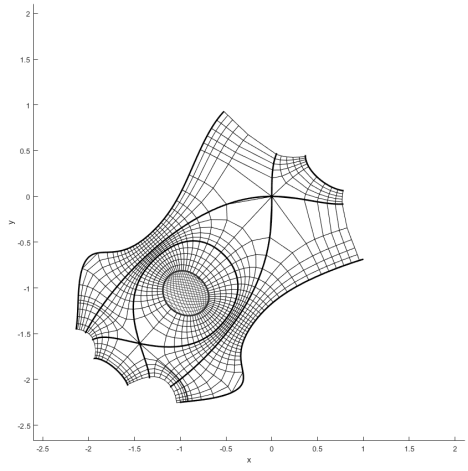


(a) The boundary of Ω

(b) The critical point



(c) Decomposition



(d) The flux-aligned quad mesh

Figure 14: A nonsymmetric double-null discharge flux contour experiment

5 Conclusion and Future Works

In this paper, by introducing flux-aligned parameterizations, we develop a theory to generate a flux-aligned quad mesh with any given isocurves for application to MHD simulation and design a high-order algorithm to test the theory via numerical experiments on the MHD simulation of Tokamak plasma. In this paper, we demonstrate the existence of flux-aligned quad meshes. It is a theoretical supplement for the flux-aligned quad mesh generation algorithm in [Guillard et al.(2018)]. This research will contribute to improving the robustness of existing codes by knowing more information about flux-aligned parameterizations. Using the theory of the flux-aligned quad mesh generation algorithm, eliminating isolated and nonsmooth points cannot be avoided to generate nondegenerate analysis-suitable quad meshes. However, these tricks have effects on the robustness of the current algorithms. In the future, we will research a robust and efficient algorithm for generating flux-aligned quad meshes by modifying a flux function locally to avoid eliminating isolated and nonsmooth points.

6 Appendix

Lemma 2.2 Proof:

Based on Lemma 2.1, C is a regular loop without self-intersection, where $C = C_0 \cup C_1 \cdots \cup C_n$ such that $C_n \cap C_0 = p_{n,0}$ and $C_i \cap C_{i+1} = p_{i,i+1}$. C_i $(i = 1, 2, \cdots, n-1)$ are C^2-curves and C^1-joint at $p_{i,i+1}$ and $p_{n,0}$. On C_i, according to the Picard-Lindelöf Theorem, there is a unique solution

$$\phi_{i,i+1} : [s_i, s_{i+1}] \longrightarrow \mathbb{R}^2$$

such that

$$\frac{d\phi_{i,i+1}}{ds} = \frac{(-F_y, F_x)}{||\nabla F||^2}, \phi_{i,i+1}(s_i) = p_{i,i+1}. \tag{8}$$

Consider

$$\frac{dF(\phi_{i,i+1})}{ds} = \nabla F \cdot \frac{d\phi_{i,i+1}}{ds} = (F_x, F_y) \cdot (-F_y, F_x)/||\nabla F||^2 = 0.$$

Thus $\phi_{i,i+1}([s_i, s_{i+1}])$ is a part of C. Then we can define

$$\phi : [s_0, s_n] \longrightarrow C,$$

by joining $\phi|_{[s_i, s_{i+1}]} = \phi_{i,i+1}$ together. We can consider $\dfrac{d\phi}{ds} = \dfrac{(F_y, -F_x)}{||\nabla F||^2}$ similarly.

Lemma 2.3 Proof:

First, we prove that the number of critical values and the boundary value is more than 2 in (1). Then, in (2), we prove the number is less than 3.

(1). $\bar{\Omega}_i$ is a bounded closed set, and F reaches C^1 continuous on $\bar{\Omega}_i$. Then, on $\bar{\Omega}_i$, F is reached the maximum and minimum values denoted as M and m, respectively, where $M > m$ because F is not a constant function on Ω_i and $\Omega_i \subset \bar{\Omega}_i$. The maximum and minimum values are the critical values or the boundary values. Thus, the number of critical values and the boundary values is more than 2. Moreover, there is no critical value on Ω_i; then, these critical values and boundary values are reached at $\partial\Omega_i$ because $\bar{\Omega}_i = \Omega_i \cup \partial\Omega_i$.

(2). In this part, we will prove that the number is less than 3 by contradiction. Suppose there are at least 3 critical values or boundary values of Ω on $\bar{\Omega}_i$ denoted as c_1, c_2 and c_3. Without loss of generality, $c_1 < c_2 < c_3$. By the analysis in (1), there are $p_1, p_3 \in \partial\Omega_i$ such that $F(p_1) = c_1$ and $F(p_3) = c_3$. Moreover, F is a continuous function on $\bar{\Omega}_i = \Omega_i \cup \partial\Omega_i$. Thus, for $\epsilon > 0$, there are $p_{1,\delta}, p_{3,\delta} \in \Omega_i$ such that

$$F(p_{1,\delta}) < c_1 + \epsilon < c_2 < c_3 - \epsilon < F(p_{3,\delta}). \tag{9}$$

Ω_i is a connected set; then, it is a path-connected set. Thus, there is a curve $\Gamma \subset \Omega_i$ whose endpoints are $p_{1,\delta}$ and $p_{3,\delta}$. Considering $F|_\Gamma$, it is continuous on Γ. Thus, there is a $p_2 \in \Gamma \subset \Omega_i$ such that $F(p_2) = c_2$. There is a contradiction between $p_2 \in \Gamma$ and the definition of $\bar{\Omega}_i$.

Lemma 2.4 Proof:

It is easy to prove that $\Omega_{i,\epsilon}$ is compact. Because there is no critical point in $\Omega_{i,\epsilon}$, based on Theorem 3.1 in [Spivak and Wells(1963)], $\Omega_{i,\epsilon}$ is homeomorphic to an annulus or a cup of some annuluses. Moreover, $\nabla F|_{\Omega_i} \neq 0$, thus $\Omega_{i,\epsilon}$ is homeomorphic to an annulus. $\partial\Omega_{i,\epsilon}$ are consist of two loops with $F(x, y) = c_{i,1} + \epsilon$ and $F(x, y) = c_{i,2} + \epsilon$.

References

[Bommes et al.(2012)] D. Bommes, B. Lévy, N. Pietroni, et. al., Quad-mesh Generation and processing: a survey, Computer Graphics forum 32 (2012) 51–76.

[Armstrong et al.(2015)] C. G. Armstrong, H. J. Fogg, C. M. Tierney, T. T. Robinson, Common themes in multi-block structured quad/hex mesh generation, Procedia Engineering 124 (2015) 70–82.

[Chen et al.(2019)] W. Chen, X. Zheng, J. Ke, N. Lei, Z. Luo, X. Gu, Quadrilateral mesh generation I: Metric based method, Computer Methods in Applied Mechanics and Engineering (2019) 652–668.

[Lei et al.(2020)] N. Lei, X. Zheng, Z. Luo, F. Luo, X. Gu, Quadrilateral mesh generation II: Meromorphic quartic differentials and abel-jacobi condition, Computer Methods in Applied Mechanics and Engineering (2020) 652–668.

[Xiao et al.(2020)] Z. Xiao, S. He, G. Xu, J. Chen, Q. Wu, A boundary element-based automatic domain partitioning approach for semi-structured quad mesh generation, Engineering Analysis with Boundary Elements (2020) 133–144.

[Schneider et al.(2006)] R. Schneider, X. Bonnin, K. Borrass, D. P. Coster, H. Kastelewicz, D. Reiter, V. A. Rozhansky, B. Braams, Plasma edge physics with b2-eirene, Contributions to Plasma Physics 46 (2006).

[Dudson and Leddy(2017)] B. D. Dudson, J. Leddy, Hermes: global plasma edge fluid turbulence simulations, Plasma Physics and Controlled Fusion 59 (2017).

[Guillard et al.(2018)] H. Guillard, J. Lakhlili, A. Loseille, B. Nkonga, et. al., Tokamesh:a software for mesh generation in tokamaks, Research Report RR-9230, CASTOR (2018).

[Tamain et al.(2016)] P. Tamain, H. Bufferand, G. Ciraolo, C. Colin, D. Galassi, P. Ghendrih, F. Schwander, E. Serre, The tokam3x code for edge turbulence fluid simulations of tokamak plasmas in versatile magnetic geometries, Journal of Computational Physics 321 (2016).

[Chang et al.(2009)] C. S. Chang, S. Ku, P. Diamond, et.al., Whole-volume integrated gyrokinetic simulation of plasma turbulence in realistic diverted-tokamak geometry, Journal of Physics: Conference Series 180 (2009).

[Stegmeir et al.(2018)] A. Stegmeir, D. Coster, A. Ross, O. Maj, K. Lackner, E. Poli, Grillix: a 3d turbulence code based on the flux-coordinate independent approach, Plasma Physics and Controlled Fusion 60 (2018).

[Pamela et al.(2019)] S. Pamela, G. Huijsmans, A. J. Thornton, A. Kirk, S. F. Smith, M. Hoelzl, T. Eich, J. Contributors, the MAST Team, the JOREK Team, A wall-aligned grid generator for non-linear simulations of mhd instabilities in tokamak plasmas, Computer Physics Communications 243 (2019).

[Spivak and Wells(1963)] M. Spivak, R. Wells, Morse Theory, Princeton University Press, 1963.

[Alessandrini and Magnanini(1992)] G. Alessandrini, R. Magnanini, The index of isolated critical points and solutions of elliptic equations in the plane, Ann. Scuola Norm. Sup. Pisa Cl. Sci. Ser. 4, 19 (1992) 567–589.

[Arango and Gómez(2011)] J. Arango, A. Gómez, Critical points of solutions to elliptic problems in planar domains, Commun. Pure Appl. Anal. 10(1) (2011) 327–338.

[Cheeger et al.(2015)] J. Cheeger, A. Naber, D. Valtorta, Critical sets of elliptic equations, Commun. Pure Appl. Anal. 68 (2015) 173–209.

[JOREK-Team(2008)] JOREK-Team, Jorek is a non-linear extended mhd code for toroidal x-point geometries., 2008. URL: https://www.jorek.eu/.

[Czarny and Huysmans(2008)] O. Czarny, G. Huysmans, Bézier surfaces and finite elements for mhd simulations, Journal of Computational Physics 227 (2008) 7423–7445.

[Teschl(2012)] G. Teschl, Ordinary Differential Equations and Dynamical Systems, American Mathematical Society, 2012.

[Huysmans and Czarny(2007)] G. T. A. Huysmans, O. Czarny, MHD stability in X-point geometry: simulation of ELMs, Nuclear Fusion 47 (2007) 659–666.

[Wu et al.(2017)] M. Wu, B. Mourrain, A. Galligo, B. Nkonga, H1-parameterizations of planar physical domains with complex topology in iso-geometric analysis, Computer Methods in Applied Mechanics and Engineering 381 (2017) 296–318.

[Pamela et al.(2013)] S. J. P. Pamela, G. T. A. Huysmans, et. al., Resistive mhd simulation of edge-localized-modes for double-null discharges in the mast device, Plasma Physics and Controlled Fusion 55 (2013) 095001(13pp).