
**Parameterizations for Mesh Generation:
Discrete Atlases, Crossfields and 3D Frames**

Pierre-Alexandre Beaufort

Thesis submitted in partial fulfillment of the requirements for the Ph.D
Degree in Engineering Sciences

THESIS COMMITTEE

Advisors	Jean-François Remacle	<i>UCLouvain</i>
	Christophe Geuzaine	<i>ULiège</i>
Jury	Vincent Legat	<i>UCLouvain</i>
	Franck Ledoux	<i>CEA</i>
	Jonathan Lambrechts	<i>UCLouvain</i>
	Eric Béchet	<i>ULiège</i>
Chairperson	Sandra Soares-Frazão	<i>UCLouvain</i>

Louvain-la-Neuve, December 2019

MERCI

Juillet 2015. Je suis fraîchement diplômé et mon téléphone ne cesse de sonner. On veut me recruter dans l'industrie et la consultance. Oui c'est vrai, un ingénieur civil n'a pas de mal à trouver un job. Cela ne veut pas dire pour autant que le job lui plaît : c'est mon cas. Le doctorat me paraît plus intéressant. C'est une belle promesse : apprendre et innover pendant 4 ans.

Je tiens donc à remercier Jean-François Remacle non seulement pour m'avoir proposé ce beau projet de recherche, mais aussi pour m'avoir fait confiance dès le début. Le monde du maillage m'était complètement étranger. Je pensais naïvement que le maillage était trivial par rapport à la simulation numérique. Jean-François a su être disponible et surtout patient pour m'écouter et me soutenir.

Je pense que l'écoute et le soutien sont les qualités les plus importantes d'un promoteur. J'ai bien de la chance, car j'ai deux promoteurs qui sont comme ça : Jean-François Remacle et Christophe Geuzaine. En plus, ils sont complémentaires. Là où Jean-François encourage la créativité et l'innovation, Christophe inspire la rigueur et la précision. Ai-je mentionné le fait qu'ils sont des pointures du maillage avec entre autres leur logiciel Gmsh ?

Je remercie Christophe pour m'avoir permis d'effectuer mon doctorat en partie à l'Université de Liège, dans l'institut Montefiore. Bien que j'étais le seul à travailler dans le maillage au sein du groupe *Applied and Computational Electromagnetics* (ACE) à Liège, j'ai été bien accueilli par les autres chercheurs. Je salue plus particulièrement Alexandre, Anthony, Erin, Kevin, Maxime, Steven, Vanessa et Yannick.

J'ai effectué majoritairement ma thèse à l'Université catholique de Louvain, dans le pôle de mécanique et mathématiques appliquées (MEMA). Le pôle MEMA est localisé dans le bâtiment Euler qui héberge également le pôle d'ingénierie mathématique (INMA). Les MEMA et INMA, appelés Eulériens, rendent le bâtiment un lieu agréable pour le travail. Merci donc à tous mes collègues, avec une attention spéciale à Alexandre, Amaury, Astrid, Benjamin, Célestin, Christos, David, François, Jovana, Kilian, Maxence, Ruili et Ruiyang.

Je remercie aussi des anciens Eulériens. Merci Philippe pour toutes ces sorties sportives. Merci Matthew, Matthieu et Romain de m'avoir accueilli au sein de l'Euler. Merci à Corentin et Pierre-Yves pour leurs tours de magie spectaculaires.

Je tiens à remercier plus particulièrement mon co-bureau, Matthieu. Pendant 3 ans, tu m'as bien supporté – dans tous les sens du terme. C'est toi qui m'as transmis le goût de réaliser des images et présentations soignées. Tu as élargi le regard que je porte sur le monde d'aujourd'hui. C'est certain qu'à ton

contact j'ai évolué, progressé aussi bien sur le plan humain que scientifique. Je ne peux que te remercier.

Je m'adresse maintenant à mon jury. Je souhaite remercier respectueusement Sandra Soares-Frazão pour avoir assuré la présidence de ma défense de thèse. Elle a su guider de manière remarquable les discussions. Je remercie sincèrement Eric Béchet pour ses critiques constructives lors de la défense. Celles-ci m'ont permis d'avancer des arguments précis et objectifs. Je remercie Vincent Legat pour ses conseils judicieux, aussi bien à la défense qu'au quotidien. Ses conseils m'ont guidé tout au long du doctorat. Je remercie Franck Ledoux qui travaille au Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA) en France. C'est un honneur de le compter comme membre de mon jury. Enfin, je remercie Jonathan Lambrechts pour son aide précieuse tout au long de mon doctorat. Il est sans aucun doute la colonne vertébrale de MEMA. C'est simple, il soutient tous les projets du pôle. Je tiens à le remercier en particulier pour son enthousiasme scientifique. Sans lui, le chapitre 4 n'existerait pas.

Mes amis sont une bouffée d'air frais, qui fut bienvenue durant le doctorat. Les côtoyer m'épanouit. Je remercie mes amis de la DT, avec qui je traverse les étapes de la vie. Je dis donc merci à Adrien, Benoît, Charles, Guillaume, Hadrien, Jordan, Maël, Patrick, Philippe, Tish, Tony et Xavier, ainsi qu'aux Ratelières Anne-Claire, Delphine, Evelyne, Kelly et Nine. Merci à mes amies Aude et Caroline ; on s'est mutuellement soutenus au cours de nos thèses. Un grand merci à Audrey et Greg qui me font l'honneur d'être Parrain, pour la seconde fois. Je remercie Maud et Laurent, pour nos soirées jeux de société.

Je tiens à exprimer mon amour et ma gratitude à mon Frère, mon Papa et ma Maman. Mon Frère demeure un modèle pour moi. Je n'exagère rien en disant qu'il excelle en tout. Il m'incite à faire mieux, à me surpasser. Papa et Maman ont certainement tout fait pour que je puisse en arriver là. Il est vrai que la santé de Maman les a tous les deux fortement préoccupés. Papa, Maman, je tiens à vous demander pardon d'avoir été absent lors de cette terrible épreuve. Je suis convaincu que cette thèse vous comble de fierté tous les deux, et j'en suis heureux.

Enfin, je remercie ma moitié. Coralie, merci d'avoir été si présente pour moi au quotidien. Je te demande pardon d'avoir trop souvent été absent, physiquement et mentalement. Sache que je suis heureux de vivre avec toi. En fait, tu fais mon bonheur. Bref, je t'aime.

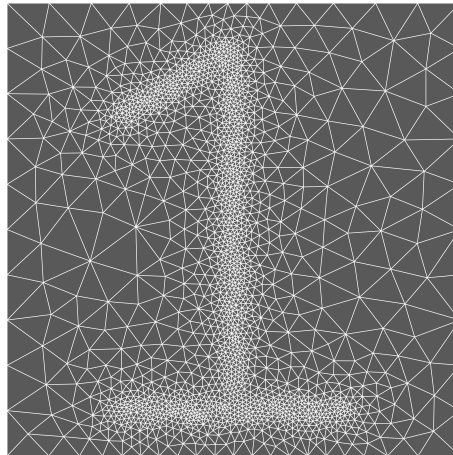
Pierre-Alexandre Beaufort

Contents

Contents	i
1 Introduction	1
1 Context & Motivation	4
2 Some Related Works	8
3 Contributions	21
2 Triangulation remeshing based on parameterization	25
1 Introduction	26
2 Mappings	27
3 Discrete Parametrizations	31
4 Gmsh's Pipeline for Discrete Surface Meshing	41
5 Improving Parametrization on Coarse Discrete Surfaces	44
6 Examples	50
7 Conclusion	51
3 Crossfield computation based on Ginzburg-Landau	53
1 Introduction	54
2 Topology of Triangular and Quadrilateral Meshes	56
3 Why Crossfields?	59
4 Crossfield Computation: the Planar Case	62
5 Computation of Crossfields: Nonplanar Generalization	65
6 Numerical Validation: the Unit Disk	68
7 A Surprising Result: the Sphere	70
8 Weak Boundary Conditions	73
9 Application of our FEM Scheme to the Torus	74
10 Conclusion	76
4 Quaternionic parameterization of 3D frames	85
1 Introduction	86
2 Rotation Representations	89
3 Frame Field Space	97

4	Numerical Insights	103
5	Conclusion	114
5	Conclusion	115
1	Remeshing with Winslow Smoothing	117
2	Size Matters for Directions	119
3	Frames are about Variety Optimization	122
	Bibliography	125
	Appendix A Discretizations of mappings	133
	A.A Derivation of the FEM scheme for harmonic mapping	133
	A.B MVC difference scheme on a structured mesh is not a Laplacian	134
	Appendix B Computing renormalized energy on a unit disk	137
	B.A Solving the Neumann Problem i.e. Computing $\Phi(x)$ of Equation (3.24)	137
	B.B $\int_{\partial S} \Phi f \times f_\tau ds$ is Zero along a Circle	141

CHAPTER



Introduction

Numerical simulation has become a jewel for industrial companies and research programs. It has radically changed most of design processes during the last decades. Aggressive use of numerical simulation has drastically reduced experimental tests, and it has enabled trials that are lacking of experimental facilities. Numerical simulations often provide further physical insight and superior designs at reduced cost and risk.

Physical phenomena are most of the time governed by differential equations. Those equations describe how a phenomenon evolves through space and time. Numerical simulation consists in solving such equations. There exist several classes of methods performing numerical simulation. However, the kind of differential equations constrains the possibilities. Integral methods are not practical for differential equations whose coefficients are not constant. Finite difference methods build very accurate solutions, but their framework is limited to simple geometries and they are hardly able to handle local adaptivity.

The Finite Element Method (FEM) yields a versatile framework for nu-

merical simulation. Actually, FEM is broadly used by industrial companies. There are at least one million finite element analyses carried out a day in engineering offices, according to Hughes et al. (2004). A finite element consists of a geometrical shape of reference, an associated function space and the corresponding linear forms. The associated function space is a subspace of functions whose the domain corresponds to the shape of reference. The linear forms define which features of a function are preserved within the associated function space; in other words, the linear forms define the degrees of freedom. Those keys make FEM able to model a large range of physics over any geometries.

The main and seemingly only drawback is that FEM comes with the tremendous price of mesh generation. Meshing generally accounts for 80% of the human time devoted to finite element analysis. Mesh generation consists in dividing a geometry into simple shapes – called elements – related to the shape of reference. FEM thus relies on mesh generation to handle geometry, which is usually achieved with unstructured¹ meshes.

A mesh can be conforming, or nonconforming. In the former case, it means that whenever two elements intersect, it must be in a vertex, along a complete edge or on a complete facet. Otherwise, the mesh is nonconforming. Despite the fact that it is easier to produce a nonconforming mesh, conforming ones are preferred. In comparison, conforming meshes ease the definition of function spaces, which facilitates the computations and enables a lightweight automated framework. Besides, industrial companies have solvers that usually require conforming meshes.

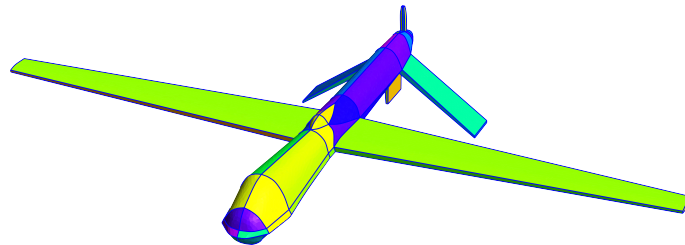


Figure 1.1: Computer aided design of a predator. The surface is defined by patches which are built from splines and nurbs.

Most of the time, mesh generation relies on computer aided design (CAD) systems. CAD systems enable to build a blueprint, which represents exactly the geometry up to manufacturing tolerances (see Fig.1.1). That blueprint allows to spawn vertices with a density depending on the geometry and possibly on a user-defined density called size field. Elements are built from those vertices.

¹The local topology is not constant.

The shape of each element has to be suitable for numerical simulation. This latter statement is roughly what makes mesh generation hard.

In the planar case, the theory of triangulations (see [Edelsbrunner \(2001\)](#)) is well-established. Thanks to theory, any planar geometries can be meshed with triangles of high quality for numerical simulation. Since the seventies with [Lawson \(1972\)](#), there are efficient techniques to produce two-dimensional meshes with triangles. However, meshing capabilities for curved surfaces can be improved. For example, mesh adaptation becomes complicated since the mesh has to preserve the geometry.

Three-dimensional mesh generation is considered as a problem that is extraordinary complicated. The theoretical background is insufficient to guarantee a suitable tetrahedrization of a given region. The crucial key is to have a meshing algorithm which terminates. The Delaunay tetrahedrization is a meshing algorithm which terminates, but it does not avoid the creation of nearly flat tetrahedra called slivers. Meshing is essentially a combinatorial problem, whose complexity can be nonpolynomial.

Meshes are not only made of triangles or tetrahedra, they are not all simplicial. Other elements exist: quadrangles, prisms, pyramids and hexahedra. The type of element is strongly related to the goal of the numerical simulation. For example, prisms are useful for anisotropic physics (e.g. [Delandmeter et al. \(2018\)](#)). Quadrangles and hexahedra are considered by FEM community to be superior to simplices (see [Sarrate Ramos et al. \(2014\)](#)). Even if it is possible to provide a full quadrangular mesh (e.g. [Remacle et al. \(2012\)](#)), full hexahedral mesh is still an open question; David Bommès – professor at the institute of computer science in the University of Bern – just started in September 2019 an ERC project about hexahedral mesh generation (AlgoHex). Nowadays it is only possible to produce a hexahedral dominant mesh, i.e. a mesh whose the volume is mostly composed of hexahedra. A mesh which is composed of different types of element is said hybrid. In a hybrid mesh, pyramids allow to connect conformally quadrangles with triangles.

Simplicial mesh generation is considered as mature, thanks to quality measures such as those of [Shewchuk \(2002\)](#) allowing to remove slivers. In that case, the mesh is optimized in order to remove the slivers and to improve the shape of simplices (e.g. [Freitag and Knupp \(2002\)](#)). However, the task may become delicate if the mesh is constrained on the boundary of the volume. Regarding hexahedral dominant mesh generation, some progress has recently been made by [Pellerin et al. \(2018\)](#). However, even if this latter method efficiently combine vertices into hexahedra, the shape/quality of hexahedra is very sensitive to the initial vertex location. Pyramids and possibly prisms are side products of hexahedral dominant mesh generation. Full prismatic meshes are usually produced by extruding a triangular mesh.

Mesh generation is thus an active research field. This thesis deals with tools which are auxiliary for mesh generation. More precisely, it enables mesh generation on surfaces lacking of CAD, it gives information about how spawning vertices to merge them into quadrangles and it derives analogous information for hexahedra. Actually, it defines a parameterization of surfaces, quadrangles and hexhedra.

In what follows, the project that funded this thesis is described. Afterwards, an overview of auxiliary tools related to this thesis is then given. Finally, the achieved works of the thesis are explicitly explained.

1 | Context & Motivation

A summary about the funding ARC WAVES is first presented. The connections of the thesis with the ERC Hextreme are then emphasized. The software Gmsh where developments have been carried out is then briefly introduced. The objectives of projects defining the work of this thesis are given. The methodology to reach those objectives is finally detailed.

ARC WAVES

The present thesis has been funded by the ARC² WAVES project 15/19-03, whose the title is "Large Scale Simulation of Waves in Complex Media". This project was coordinated by professors Eric Béchet (ULiège), Christophe Geuzaine (ULiège) and Jean-François Remacle (UCLouvain).

This research project aimed to simulate geophysical wave phenomena, which is the propagation of waves in the underground soil. This simulation allows to identify media which are crossed by waves. For example, the result can be interpreted to locate reservoirs containing oil or gas. There are other perspectives of usage, such as studying the controversial effect of hydraulic fracking.

Direct problems consist in solving partial differential equations on a domain whose all the required characteristics are given and with known initial conditions. For example, a wave propagation can be computed by using the wave equation which describes the amplitude of the wave in some location at a specific time. The propagation is parameterized by the wave velocities, which depends on local composition of the soil. The direct problem corresponds to find the amplitude of the wave. In the case of a direct problem, the wave velocities are known.

In the case of the WAVES project, numerical simulation purpose is to solve an inverse problem, i.e. from solution to initial conditions and characteristics of

²"Action de recherche concertée"

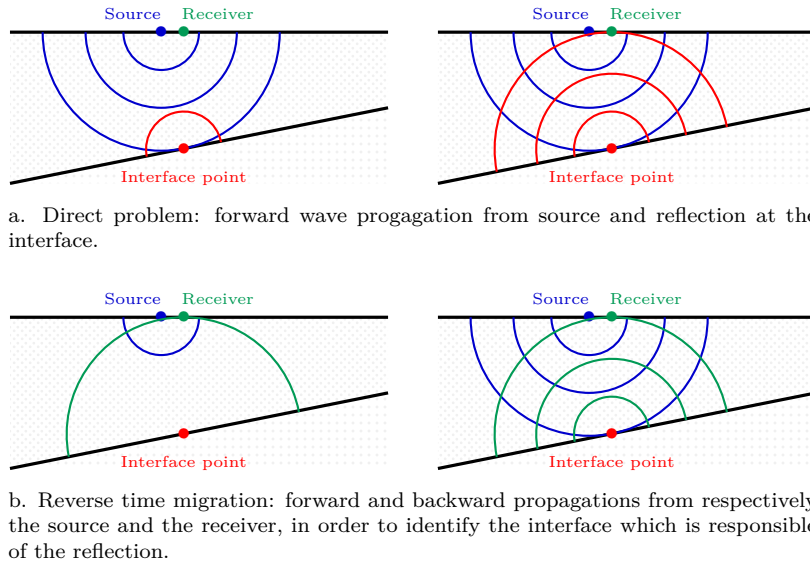


Figure 1.2: Basic idea of solving an inverse problem through reverse time migration. The (refracted) wave propagation in the second medium is not shown for the sake of simplicity.

the domain. From measurements of wave propagation, the nature of the media is determined. In regards of the previous example, it means that the amplitude of the wave is known for some times and positions, but the wave velocities are unknown. Solving an inverse problem relies on optimization algorithms, which requires several simulations of the related direct problem. A way to solve an inverse problems related to WAVES project is the reverse time migration, Fig.1.2. In few words, it is based on the simulation of forward and backward wave propagations from respectively a source and a receiver. The source triggers a wave, and the receiver detects some echo of that same wave, Fig.1.2a. When the forward and backward propagation concurrently coincide in a same location, a point belonging to an interface has been identified, Fig.1.2b.

The goal of the WAVES project is twofold. Mesh generation has to be enabled for geophysical models which are described in a non-standard way, i.e. there is no CAD design of the domain. As usual, the generated meshes have to be suitable for finite element analysis. Such meshes would allow to derive a numerical method which would accurately simulate the wave equations among those complex and anisotropic media.

ERC Hextreme

The thesis has also taken part in the ERC³ Hextreme project, entitled "Hexahedral Mesh Generation in Real Time". Professor Jean-François Remacle (UCLouvain) is leading the project. There have been strong connections with Hextreme members: [Jezdimirovic et al. \(2017\)](#), [Georgiadis et al. \(2017\)](#).

The research aims to produce unstructured conforming hexahedral meshes in general three-dimensional domains. An indirect approach is favored. Boundary layer meshes are considered. The main challenge is to speed up the hexahedral mesh generation by intensively using the multi-threaded architecture of modern computers.

Software Gmsh

WAVES and Hextreme projects aim to carry all developments within Gmsh, [Geuzaine and Remacle \(2009\)](#). Gmsh is a three-dimensional mesh generator, with built-in CAD engine and post-processing facilities. Gmsh is an open source software, which is broadly used by research projects such as GetDP ([Boubendir et al. \(2012\)](#)), SLIM ([Vincent et al. \(2016\)](#)) and MigFlow ([Constant et al. \(2019\)](#)). It is composed of four modules dealing respectively with geometry, mesh, solver and post-processing.

It has been decided to retrieve meshing facilities into a separate and brand new library called HXT, which stands for Hextreme.

Related objectives

Here are developed the objectives of the ARC WAVES project which have guided the work of the thesis.

The underground is represented by data set which consists of point clouds, voxels or stereolithography (stl) triangulation, [Fig.1.3](#). It means there is not CAD design available. There is a lack of information about the topology and geometry of the domain. There is then a need to *enable mesh generation* by building suitable and relevant information from data set.

The underground soil is made of geophysical layers which are potentially crossed by faults, and whose layers may be very close to each others. Wave propagation strongly depends on the distribution of rock types of the underground and the nature of boundaries between them. The mesh has thus to preserve those features.

³"European research council"

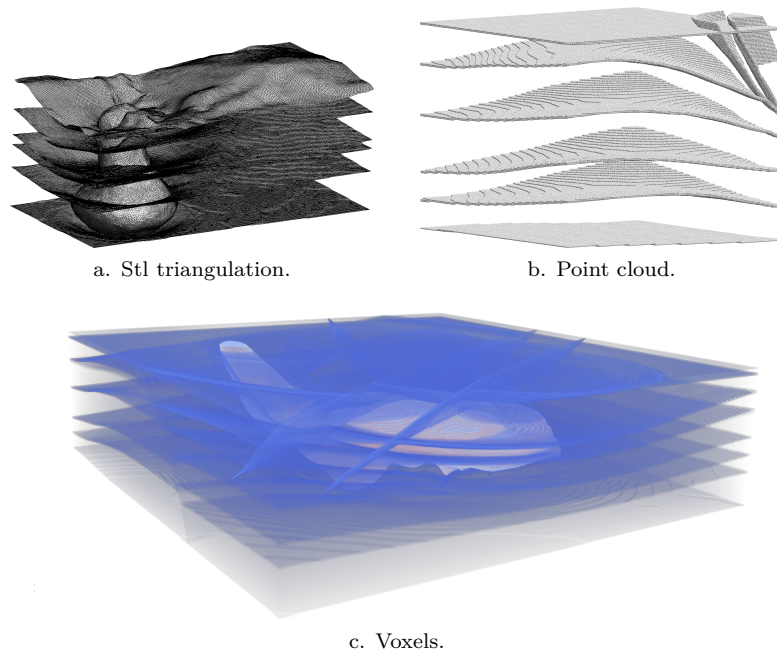


Figure 1.3: Nonstandard geometries of the underground.

Since the reverse time migration is performed by solving many times the wave propagation, there is a need to speed up the computation. *Hexahedral meshes* decrease the computation time, by lowering significantly the number of elements by node. Besides, hexahedra hold better numerical features for wave propagation.

However, the boundaries between the layers have to be preserved. The best way to produce a hexahedral dominant mesh is to use a frontal approach, by meshing first the boundaries and then the volume. A *quadrangular mesh* has thus to be produced first on those boundaries, using ad hoc methods for thin layers that correspond to a pair of almost facing boundaries that are very close to each other.

Several strategies to reach those objectives have been adopted. They are described in the next section.

Methodology

Since there are tools such as [Turner \(2012\)](#) and [Sitnik and Karaszewski \(2008\)](#) respectively converting voxels and point clouds into stl triangulations representing surfaces, only enabling mesh generation for such triangulations has been considered. A CAD design enables mesh generation by giving a parameterization of an object. It allows the meshers to spawn points on an object. It is then necessary to parameterize those triangulated surfaces; this process is called *surface reparameterization* ([chapter 2](#)).

There is plenty of ways to mesh conforming quadrangles ([Johnen, 2016](#), see §2.1.2). In the case of WAVES and Hextreme projects, quadrangulation relies either on an indirect method like [Remacle et al. \(2012\)](#), or on a global parameterization like [Jezdimirovic et al. \(2019\)](#). Even if the former method combines effectively triangles into quadrangles, the shape of quadrangles strongly depends on the point location. In order to ensure a consistent location of vertices to later form well shaped quadrangles, a *crossfield* is needed ([chapter 3](#)). A crossfield gives in each point of a surface the vertex connectivity of a grid of squares. The global parameterization also requires a crossfield.

Hexahedrization is addressed in the same approach by the research team. Indirect methods rely on combination of tetrahedra, or deal with vertex combination such as [Pellerin et al. \(2018\)](#). Again, the shape of hexahedra produced by indirect methods is sensitive to vertex location. A *frame field* is required ([chapter 4](#)); it gives the vertex connectivity of a grid of cubes. It is also possible to build a block structure decomposition from a frame field.

2 | Some Related Works

The strategies sketching the methodology are not specially original. Yet, their implementation is still an active field of research. In order to situate the contributions of the present thesis, a selection of related articles is intensively described. This has been preferred instead of a large set of few words summarizing numerous articles.

Surface remeshing is addressed, focusing partly on parameterization techniques. Quadrangulation methods are then presented, which are mainly based on the computation of a crossfield. Few ways to generate hexahedral meshes are developed, using all a frame field that is possibly computed.

Remeshing & Parameterization

There are essentially two ways to remesh a triangulated surface: local mesh modifications and parameterization. Mesh adaptation techniques work with

local mesh modifications. Those modifications may be subsequent to a refinement of the initial triangulation.

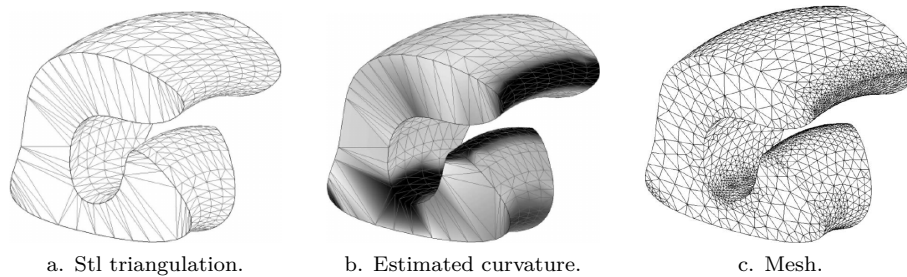


Figure 1.4: Remeshing with local mesh modifications. Figures from (Béchet et al., 2002, Fig. 5, 6 & 32).

Béchet et al. (2002) use local mesh modifications to produce a mesh suitable for FEM analysis. They first identify sharp edges which describe geometrical features: black edges on Fig.1.4a. Those edges have to be preserved. They compute then an estimation of the curvature: grayscale on Fig.1.4b. Each triangle whose the size is beyond a user-defined size field is refined by edge bisection. The bisection is performed from the worst triangle to the least bad one. After each bisection, a modified Delaunay criterion is checked by using a metric which is built from the estimated curvature. Edge swap occurs if the current edge is not part of a sharp edge. If a new triangle is still too large, it is appropriately added to the bisection process. Locally, the vertices that are free to move (i.e. not on a sharp edge) are smoothed using a kind of laplacian with few iterations.

Wang et al. (2007) perform an analogous process. The sharp edges can be detected automatically from an estimation of the principal curvatures. However, some sharp edges can be missed. They use then a second criterion, which consists in identifying an edge as sharp if it is the smallest one in a triangle whose the aspect ratio is high. This criterion can over-identify sharp edges. The curvature is computed by approximating locally the triangulation with a quadratic surface (cf. Hamann (1993)). Their approximated curvature enables to produce either isotropic or anisotropic meshes. Edges can be contracted, i.e. an edge can be replaced by a vertex, Fig.1.5. It allows to coarsen the mesh in areas it is too fine.

Parameterization techniques use the initial triangulation to build a (set of) continuous and bijective mapping(s) between the triangulated surface and a target parameter plane. Those techniques were first mainly used by the com-

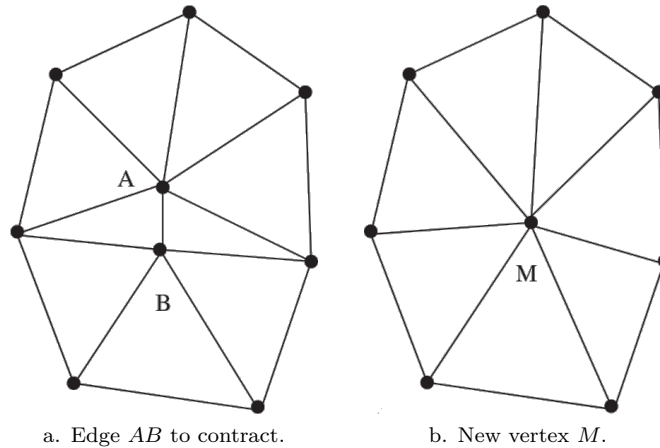


Figure 1.5: Contraction of edges. Figures from (Wang et al., 2007, Fig. 3a & 3b).

puter graphic community for texture mapping (cf. Bennis et al. (1991)). Later the meshing community starts to use such techniques for surface remeshing.

Floater (2003) proposes to build bijective mappings based on the mean value coordinates. It means that each point may be expressed as a convex combination of its neighbors. If the target planar shape is convex, it guarantees that the underlined parameterization is one-to-one. The scheme building those coordinates is a linear approximation of the mean value theorem.

Remacle et al. (2010) use harmonic mappings for parameterization. Harmonic mappings correspond to functions minimizing their Dirichlet energy. In other words, harmonic mappings build parameterization with the least distortion. Unlike the mean value coordinates, discrete harmonics mappings are not ensured to provide one-to-one parameterizations. For this reason, they perform a cavity check once the parameterization is computed. In the parameter plane, they check for each interior vertex that the connected triangles share the same orientation. If not, the interior vertex is moved at the center of the polygon formed by the connected triangles. The parameterization maps the triangulation onto a unit disk, Fig.1.6b. There are then topological constraints. If a triangulation has several boundaries, Dirichlet conditions are only set on one to map it onto the circle, and homogeneous Neumann conditions are set on the other ones. They introduce the issue of large aspect ratios arising with harmonics maps. In short, some triangles have parametric coordinates which become numerically undistinguishable. Marchandise et al. (2011) deal with this matter by using a multiscale laplacian, Fig.1.7. The triangles whose aspect

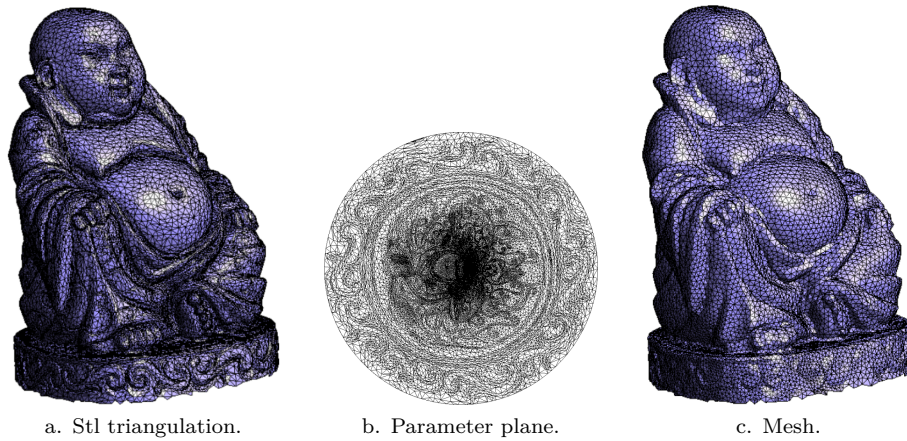


Figure 1.6: Remeshing with harmonic maps. Figures from (Remacle et al., 2010, Fig. 14 up).

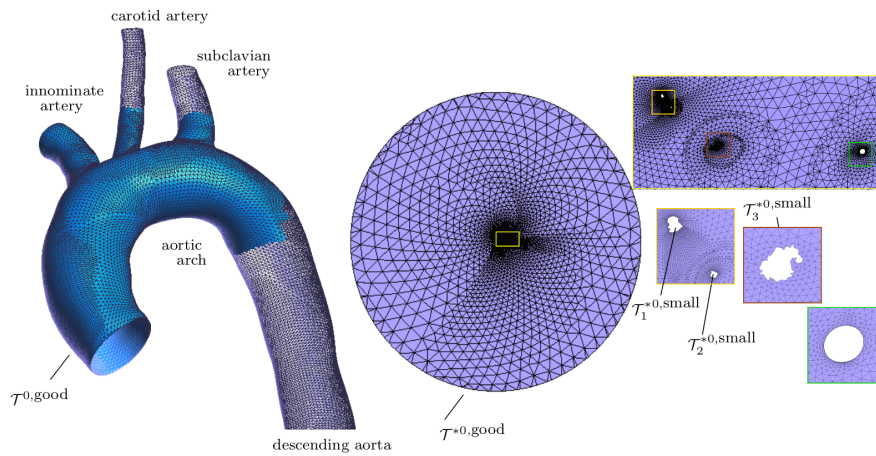


Figure 1.7: Multiscale partitioning to avoid large aspect ratio issues. Figure from (Marchandise et al., 2011, Fig. 4).

ratio is high from connected sets. Each set is reparameterized with Dirichlet conditions corresponding to a scaling of the neighbourhood of triangles whose aspect ratio is correct.

Sawhney and Crane (2018) compute a conformal mapping for parameteri-

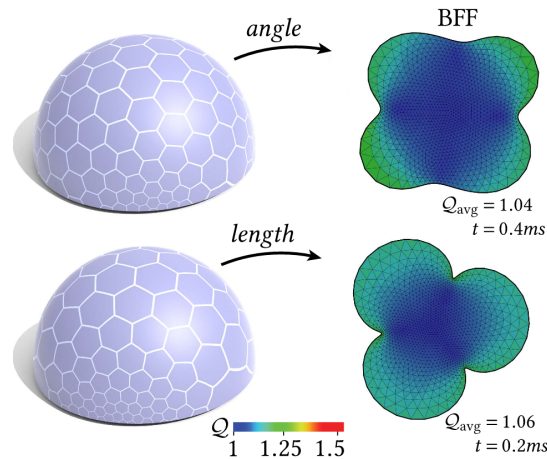


Figure 1.8: Parameterization is computed by flattening first the boundary. Figure from (Sawhney and Crane, 2018, Fig. 4).

zation. They express conformal mappings as pairs of conjugate harmonic functions. It means that their gradient are orthogonal within the tangent space of the underlined surface. This is a constraint to build a conformal mapping from a target planar shape. But the Riemann mapping theorem states there always exists a conformal mapping between a unit disk and a topologically equivalent surface. Actually, a boundary point cannot be pinned to an arbitrary location of the planar shape. Therefore, the boundary has to be flattened properly, before extending the mapping to the whole surface. This is done by solving the Cherrier formula, which correspond to a Poisson problem with a Neumann condition. This gives a planar curve that approximates either the length or curvature of the target planar shape, and which defines consistent boundary conditions for a conformal mapping, Fig.1.8.

Quadrangles & Crossfield

There exist diverse methods to generate quadrangles. Recent methods rely on a crossfield. The crossfield may be used to produce in a direct way a quadrangular mesh on a surface. Ray et al. (2006) does such an approach. They build a parameterization from a crossfield. The crossfield is either user-defined or given by the principal curvatures, Fig.1.9(B). This directional field is smoothed in order to extrapolate it in areas where it is ill-defined, which are in red on Fig.1.9(C,D). Nevertheless, the crossfield has to be scaled properly in order to be curl-free. This latter feature is required to get charts of the underlined sur-

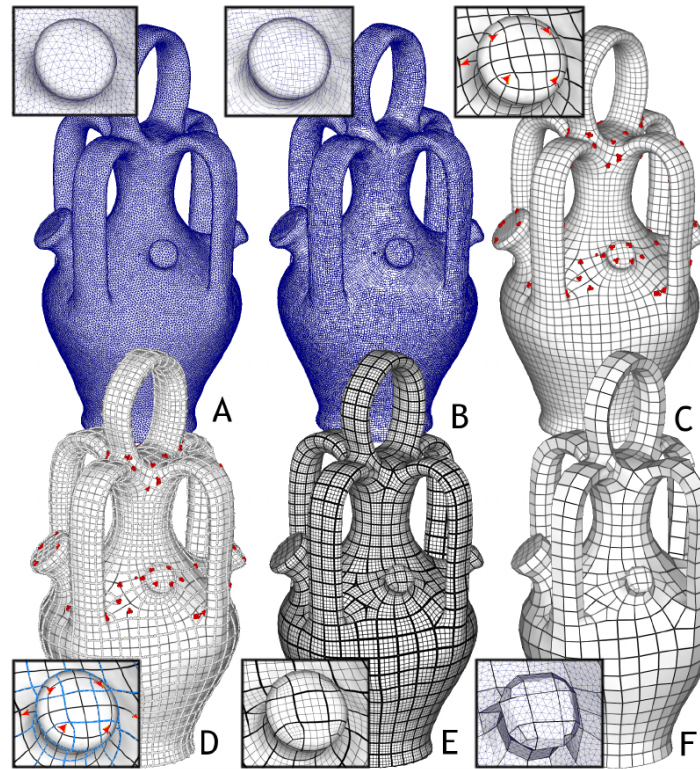


Figure 1.9: Global parameterization (D) from a crossfield (B). Figure from (Ray et al., 2006, Fig. 1).

face, Fig.1.9(E). Indeed, a global parameterization of the surface is computed by penalizing the isovalues of the charts that are not aligned with the crossfield. In some areas, the parameterization has a singular gradient. The corresponding patches are splitted and reparameterized. The global parameterization allows to build implicit meshes such as quadrangular dominant meshes, Fig.1.9(F).

Bommes et al. (2009) follow a comparable approach, but they have singular points instead of singular areas. They use the principal curvatures to measure the local relative anisotropy, which allows to detect the feature (sharp) edges. They then compute a crossfield that is aligned with those feature edges. The crossfield is expressed with an angle field, as it is done by Ray et al. (2008). Due to the symmetries of a crossfield, the angle field is designed with integer jumps of 90 degrees. The singular points correspond to a nonzero index, which depends on the jump and the local angle defect of the triangulation. From

angles and jumps, an energy may be defined. Since the angles are real values and the jumps are integer values, it defines a mixed-integer problem. The linear equations corresponding to partial derivatives with respect to jumps are solved using iterative solvers that round their result after each iteration. A global parametrization is built from the crossfield, where singular points have integer values. The corresponding charts eventually produce a full quadrangular mesh.

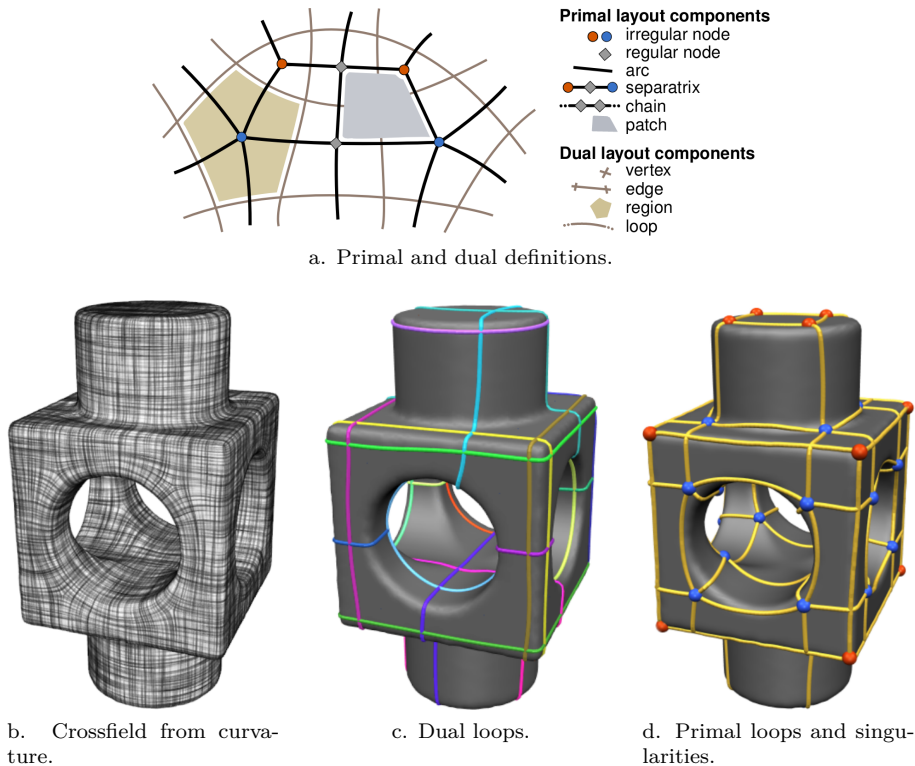


Figure 1.10: Global parameterization based on dual loops from a crossfield guided by the principal curvature directions. Figures from (Campen et al., 2012, Fig. 3, 1a, 1b & 1d)

Campen et al. (2012) create full quadrangular meshes from global parameterizations. The global parameterization is built from a crossfield which is derived from the principal curvatures, Fig.1.10b. A primal approach of global parameterization consists in tracing the separatrices whose the vertices have the required connectivity, i.e. four if it is not an irregular vertex, Fig.1.10a. They however observe that ensuring the correct connectivity is a tricky task which may require to add new irregular vertices implying new separatrices.

Instead, they build dual loops that are aligned with the crossfield, within areas that are away from any singularities, Fig.1.10c. Those loops always intersect in a vertex which is connected to four curves (i.e. two loops), Fig.1.10a. Each loop is generated by following one of the four direction of a cross, each time the closest direction to the previous one. In order to ensure that such a curve loops, a measure is defined to rate their direction fidelity and shortness. Besides, the tangent of the loop must describe an angle with the curve that is bounded to 45 degrees. It limits the tangential deviation of a loop from the principal curvatures. Loops are generated until each singularity is separated from the other ones. The index of a singularity corresponds to the number of vertices (or curves) forming the patch that contains it. If it misses loops to separate singularities, the corresponding area may be easily split with additional loops without adding irregular vertices. Afterwards, loops are selected in a greedy way such that for each pair of singularities, both singularities are separated by a set of smallest loops maximizing their direction fidelity. Once the dual loops are complete, the primal global parameterization is built by connecting the singularities in respect of the dual polypatches, Fig.1.10d. Meshing quadrangles is then trivial.

Remacle et al. (2013) proceed with a frontal approach, which allows to deal with a size field map. They build a triangulation that is suitable for a recombination of a triangulation into a quadrangulation, using the Blossom algorithm (see Remacle et al. (2012)). The triangulation is performed by a frontal Delaunay approach using a L^∞ -norm instead of the usual L^2 -norm. Since the L^∞ -norm is anisotropic, local frames have to be defined to guide the Delaunay algorithm. A crossfield aligned with the boundaries is computed within the domain through two Laplace equations, one equation for each component of the field, Fig.1.11a. Starting from the boundaries, triangles are added to the geometry. Those triangles respect the Delaunay criterion. Following a frontal strategy (Fig.1.11c), new vertices are added on the circumsquare (circumcircle equipped with the L^∞ -norm, Fig.1.11b) defined by an existing edge. The size of the side of the circumsquare corresponds to the local size prescribed by the size field, Fig.1.11d. In doing so, the triangulation is composed of almost right triangles (Fig.1.11e), which is convenient for combination into quadrangles.

High quality quadrangulations strongly rely on a crossfield, whatever the approach. It is then crucial to define a consistent computation of crossfield. The computation has to give an understanding of the relationship between the regularity and the singular points of crossfield. Knöppel et al. (2013) provide such a computation. They model a crossfield as a field of complex numbers, whose the fourth roots correspond to the four directions given by a cross. The associated energy is the Dirichlet energy, which is composed of two terms: the norm and phase of the complex field. The crossfield should correspond to complex numbers of unit-norm. However, this constraint makes the energy

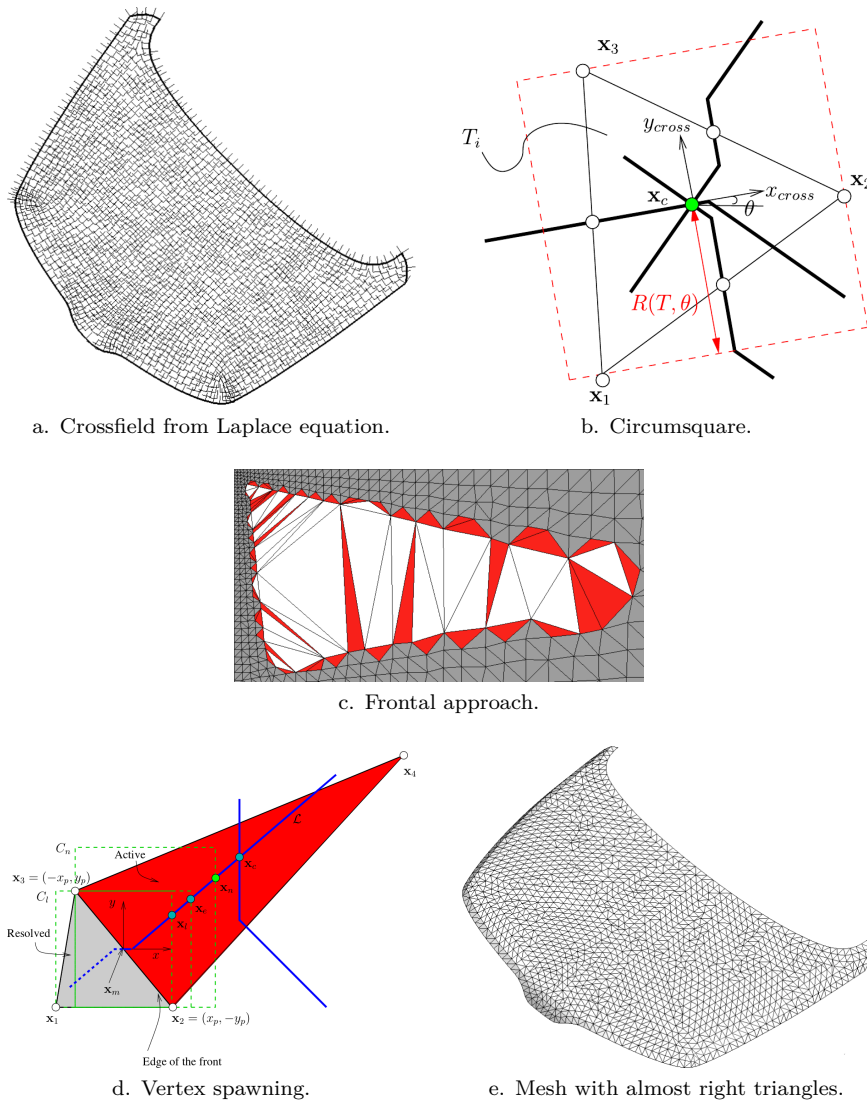


Figure 1.11: Overview of a frontal Delaunay approach for quadrangulation. Figures from (Remacle et al., 2013, Fig. 14 (up left), 10, 12, 13 and 14 (down right)).

ill-defined for singular points. From this observation, they derive the same energy but for a field which is a scaled version of the unit-norm one. The minimization of this energy accounts of finding the eigenvector associated to

the smallest eigen value of the corresponding Laplacian matrix. The aimed crossfield is finally obtained by normalizing the eigenvector.

Hexahedra & Frames

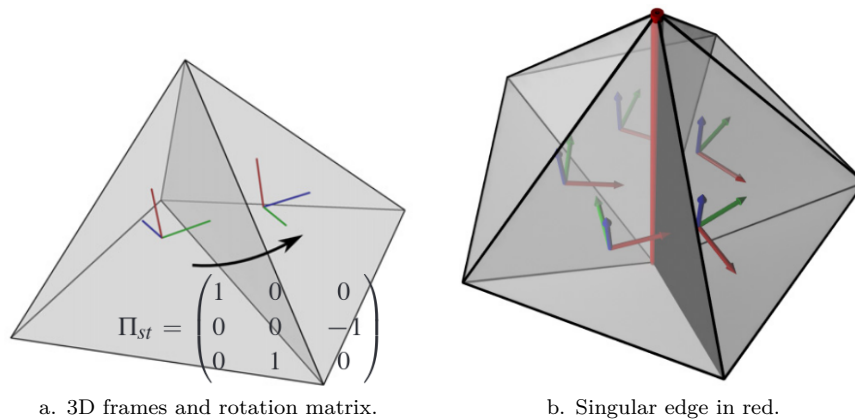


Figure 1.12: 3D frames defined by a set of 3 vectors. 3D frames are related to each others with rotation matrices. Figures from (Nieser et al., 2011, Fig. 4).

The dedicated techniques producing hexahedral meshes are quite similar to the ones for quadrangular meshes. Most of recent approaches require a frame field, which is a direct extension of crossfields. It has to be noted that mathematically, it is not shown that frame fields correspond to hexahedral meshes⁴.

Nieser et al. (2011) generate hexahedral meshes from a parameterization which is guided by a frame field. Their method is an extension of Kälberer et al. (2007) for hexahedrization. The frame field corresponds to a set of three vectors that are mutually orthogonal, Fig.1.12a. This set of vectors defines in each tetrahedron a local basis. Those nearby bases are connected together combinatorially, with rotation matrices. If the combination of those connections around an inner edge gives a matrix which is not the identity, this latter edge is singular. It means that the corresponding chart has been rotated by 90 degrees around that edge, Fig.1.12b. The parameterization consists of a triplet which is defined at the vertices of tetrahedra. The gradients of those triplets have to be aligned with the directions of frame field. The nearby triplets are

⁴while it is the case for crossfields and quadrangular meshes

connected together in the same way that the corresponding bases given by the frame field. Once the parameterization is computed, the hexahedrization is performed.

Bernard et al. (2016) produce hexahedral dominant meshes following an indirect approach. They merge tetrahedra into hexahedra by using the recombination patterns of Yamakawa and Shimada (2003). The tetrahedral mesh is built from vertices which have been inserted by following a smooth frame field. They compute first a crossfield on the boundary with the method of Remacle et al. (2013). They initialize the frame field such that each inner node has the nearest frame laying on the boundary. Afterwards, they iteratively smooth the frame field within the volume by averaging the rotations leading one frame to its neighbor; the average is weighted such that singularities do not average their neighbourhood. The new vertices are spawn with a frontal approach. They start with the smoothest frames on the boundary, and sort the insertion in respect of the local smoothness of frames. After tetrahedrization and combination, they get a mesh which is mostly made of hexahedra.

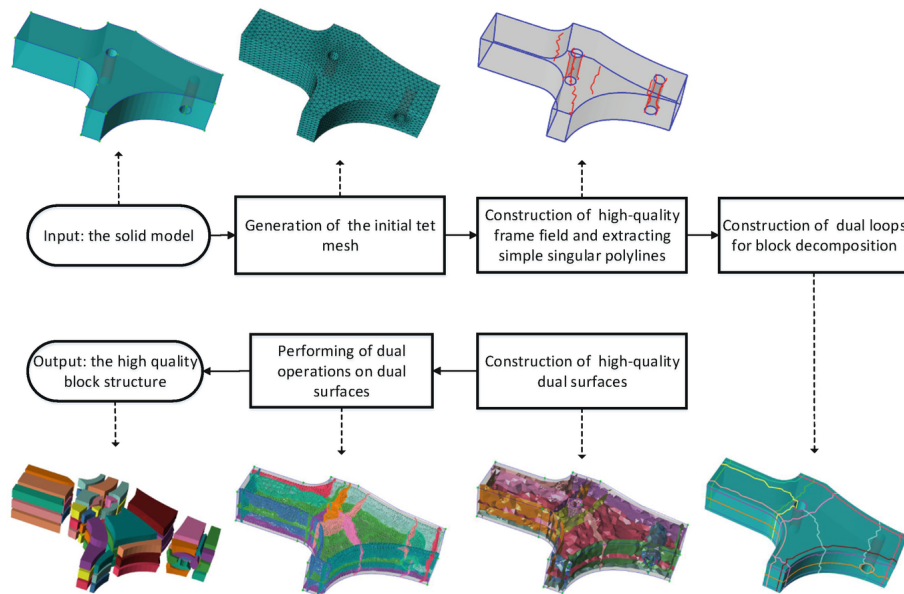
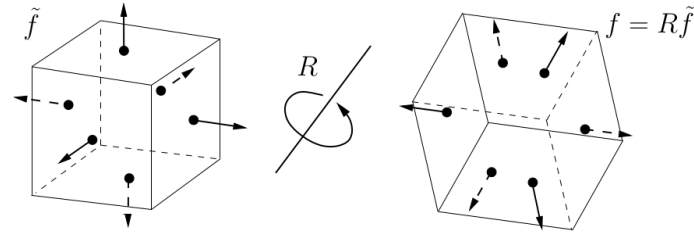


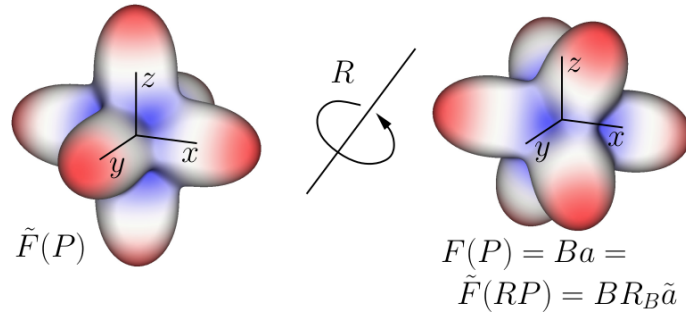
Figure 1.13: Block structure decomposition based on dual surface decomposition. Figure from (Zheng et al., 2018, Fig. 2).

Zheng et al. (2018) follow a procedure which is inspired by the work of Campen et al. (2012) (cf. dual loops, Fig.1.10) in order to produce a block structure decomposition, Fig.1.13. They start by generating a 3D frame field

using the technique of [Li et al. \(2012\)](#). As 3D frame field do not always correspond to valid block decomposition, they correct some invalid frame field singularities by tracing streamlines, [Fig.1.13\(3rd step\)](#). Then, they produce dual loops on the boundary as [Campen et al. \(2012\)](#), using a surface crossfield built by taking the trace of the 3D frame field on the boundary, [Fig.1.13\(4th step\)](#). They classify the dual loops in two categories which are based on the fact that a single dual loop spans or not a dual surface by itself. Therefore, some dual surfaces are directly identified thanks to the first kind of dual loops. Those dual surfaces intersect each others, which results in inner dual loops. Those new dual loops help to identify new dual surfaces. The dual surfaces are then inflated to cover the whole volume, [Fig.1.13\(5th step\)](#). Finally, dual operations are applied to produce the block decomposition, where each block has the topology of a cube, [Fig.1.13\(6th&7th steps\)](#).



a. Cubes related to 3D frames.



b. Real spherical harmonics corresponding to 3D frames.

Figure 1.14: 3D frames defined by real spherical harmonics. Figures from [\(Ray et al., 2016, Fig. 4\)](#).

The previous articles represent the frames with three orthogonal vectors. Such a representation is not convenient for interpolation or smoothing. Other representations exist and are suitable for those goals. [Ray et al. \(2016\)](#) generate frame field by using spherical harmonics of fourth order, whose the use for frames was initially brought by [Huang et al. \(2011\)](#). That use comes from

the fact that the sum of fourth power of each cartesian component yields the same symmetries than a frame. That sum results into a polynomial which may be expressed with the fourth order real spherical harmonics. It is possible to express that polynomial as a nine-dimensional vector whose components correspond to the linear combination of nine real spherical harmonics. That vector takes unique value for a given frame. In other words, if a frame has a unique orientation with respect to the cartesian frame, it corresponds to a unit set of nine values corresponding to a linear combination of real spherical harmonics. Besides, the distance between two frames corresponds to the distance between their nine-dimensional vectors owing to the fact that spherical harmonics are mutually orthogonal with the L^2 -norm. [Ray et al. \(2016\)](#) define the orientation of frames in each vertex of a tetrahedron by means of Euler angles. Based on Euler angles, they smooth a frame field through their representation with their nine-dimensional vector.

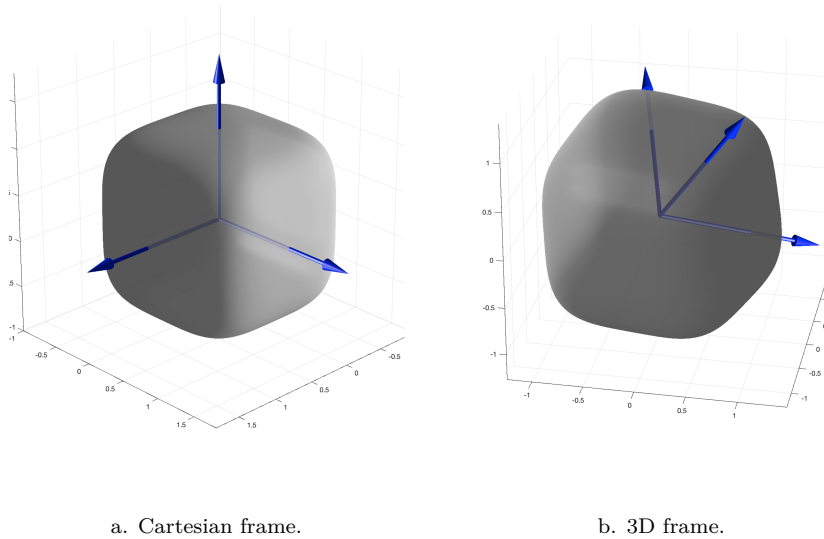


Figure 1.15: 3D frames defined with fourth order tensor. Figures from ([Chemin et al., 2018](#), Fig. 2).

[Chemin et al. \(2018\)](#) describe frame field with a fourth order tensor. They start with the same polynomial, but they use it to define a unit sphere with the L^4 -norm, Fig.1.15. This polynomial may be written as a fourth order

tensor. They then express that tensor with a basis which consists in a rotation of the initial cartesian frame. From that general expression, they show that this tensor is fully symmetric and respects the partial traces. Those algebraic relationships allow to show that this tensor corresponds to a projector. Owing to those properties, they prove that this tensor corresponds to the orientation of a frame. Then, they use the Mandel notation to represent a fourth order tensor with a six-by-six matrix. The algebraic structure enables the parameterization of that matrix with nine real parameters. They propose to smooth the values of those parameters in order to smooth the corresponding frame field. From such a tensor, they get the corresponding frame by computing the three eigentensors associated to the eigenvalues which are equal to one. The eigenvectors of the sum of those eigentensors corresponds to the three orthogonal directions of the underlined frame.

3 | Contributions

The outline of the present manuscript corresponds to the contributions of the thesis in regard to the related works. Achieving the related objectives of the WAVES project is mainly based on three strategies. First, the ability to remesh a discrete surface represented by a triangulation which is unusable for numerical simulation. Once it is possible to remesh such a discrete surface, nodes have to be spawn adequately such that the combination of new triangles into quadrangles is optimal. The icing on the cake would be then to produce a full hexahedral mesh from the surfaces bounding a domain. However, relevant full hexahedral mesh generation is still a challenge. The extension of crossfield into frame field is actually ongoing research.

Those contributions have been developed in three papers which consists of the supported articles. First, remeshing based on triangulation parameterization is presented in Chapter 2 *Automatic surface mesh generation for discrete models: a complete and automatic pipeline based on reparameterization*, which is submitted to Journal of Computational Physics. Then, an innovative way to compute crossfield on two-manifold is derived in Chapter 3 *Cross fields on arbitrary surfaces: a PDE approach based on Ginzburg-Landau theory* in revision with CAD journal, which is a promoted version of *Computing cross fields: a PDE approach based on the Ginzburg-Landau theory* published in Procedia Engineering. Finally, a new parameterization of frames is explored in Chapter 4 *Quaternionic octahedral fields: $SU(2)$ parameterization of 3D frames*, which is intended for submission.

Surface parameterization

The nonnegotiable feature of an automatic process is its robustness. In the case of surface remeshing, the parameterization has to be one-to-one. Otherwise, it is inoperable. In the second chapter, it is shown that the discrete parameterization of a triangulation has to have parametric vertices corresponding to a convex combination of their neighbors, using only geometrical arguments. The discrete parameterization should be smooth, such that the corresponding metric tensor is smooth. Harmonics mappings provide the smoothest parameterizations, but they are not guaranteed to be one-to-one. Mean value coordinates introduced by Floater (2003) are a compromise of smoothness, one-to-oneness and linearity. They are derived explicitly with a finite element approach. It is demonstrated that the corresponding difference scheme does not correspond to a Laplace operator over a structured triangulation. Boundary conditions are discussed, particularly about holes within a triangulation. It is proved that homogeneous Neumann conditions over holes produce convex holes within the parameter space. Since it is a useless deformation of the parameterization, a virtual filling of holes is proposed to lower the deformation. A longest edge bisection of inner edges of triangulations that is performed before parameterization yields smoother parameterizations. The finite scheme corresponding to a Lagrange \mathcal{P}^2 to build piecewise quadratic mean value coordinates is derived. It is shown that this scheme does not correspond to a relevant discretization of the mean value theorem. Finally, it is proved there does not exist a Lagrange \mathcal{P}^2 version of mean value coordinates. Intricate examples of the given pipeline are exhibited.

The parameterization has been implemented as a module within the HXT library. The remeshing process is triggered from Gmsh.

Crossfield computation

Previous works analyze the crossfield quality on basis of the Poincaré-Hopf theorem. This theorem only links the sum of degrees of singularities to the Euler characteristic of the surface. It does not provide any information about the respective degree, nor the mutual distance of singularities. Bethuel et al. (1994) provide precisely that information in the case of a vector field whose the behavior satisfies the Ginzburg-Landau equations in the asymptotic case, with Dirichlet conditions. The analysis of aimed quadrangular meshes about the disposition and type of their irregular vertices correspond to those of such vector field singularities. In the third chapter, the relationship between the index of a singularity and the valence of an irregular vertex is derived. Since a crossfield may be represented by a vector field, results about Ginzburg-Landau equations match. Ginzburg-Landau equations are partial differential equations

corresponding to the Euler-Lagrange equations of the Ginzburg-Landau functional. This functional consists of two terms: the Dirichlet energy and the deviation of unit norm of the vector field. The second term is scaled by a penalty factor which is proportional to the inverse square of the coherence length. This length is a characteristic of the domain. When the coherence length tends to zero, the corresponding vector field yields the aimed properties. Those properties give a minimum number of singularities, and those singularities have all an index whose the absolute value is one. The mutual distance between two singularities is maximized. A discrete scheme is derived by using a FEM with edge shape functions called Crouzeix-Raviart. The discrete scheme is validated in the case of the disk, by comparing the distribution of computed singularities with their distance to the center which is derived analytically. The scheme applied on a sphere gives a surprising result: the singularities of the computed crossfield correspond to the vertices of an anti-cube, instead of those of a cube. It is shown that the former configuration maximizes the mutual distance of singularities. Thanks to the FEM formulation, boundary conditions may be weakened in the case of sharp geometry. In that latter case, the weak boundary conditions hold better results. The use of such a crossfield for quadrangular mesh is demonstrated for both, global parameterization and frontal methods.

The crossfield computation has been completely implemented within the HXT library, in the Ginzburg module.

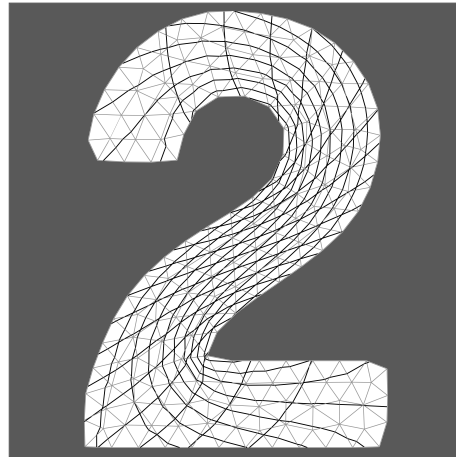
Observe that the Ginzburg-Landau approach has also been used by [Viertel and Osting \(2019\)](#). They also make the observation that the design of a crossfield corresponds to the Ginzburg-Landau theory. Compared to the contributions of this thesis, they give results about the global parameterization. Their result states that from a Ginzburg-Landau crossfield, it is always possible to partition the surface with patches which are four sided. Their numerical method is completely different and does not yield explicitly the coherence length. It is an implicit diffusion which is repeated. The diffusion lasts a characteristic time, which corresponds to the minimum eigenvalue of the discrete laplace operator. Pointwise renormalization is applied between each diffusion. This method is inspired by the one of [Merriman et al. \(1994\)](#), and is called MBO diffusion.

Frame representation

There exist two ways to represent frames without using combinatorics. These are the fourth order real spherical harmonics and the fourth order tensors. In the fourth chapter, a new parameterization of frames is developed based on the work of [Du Val \(1964\)](#). This parameterization is based on a bijective mapping between the unit quaternions and the special unitary group $SU(2)$.

Since quaternions are not familiar for engineers, it is shown how two opposite quaternions correspond to an even rotation. Afterwards, the isomorphism between quaternions of unit norm and $SU(2)$ is defined, showing that transformations within one space are consistent with the corresponding ones of the other one. From three finite groups of quaternions, three invariant forms corresponding to the octahedral group are derived. With a slight modification of the isomorphism between the quaternions and $SU(2)$, those three invariant forms give a parameterization of the frame orientation. It means that all the rotations and thus quaternions giving the same (oriented) frame, return the same value of invariant forms. Those three values correspond to three complex valued coordinates of a surface. This surface is defined by an implicit equation, which is consequently respected by the three invariant forms. From such a triplet, a way to determine all the quaternions returning those invariant values is derived. Numerical experiments demonstrate however that the three values cannot be used as a vector representation of frames. In other words, the Euclidean distance between two triplets does not correspond to the distance between the corresponding frames. The behavior of the complex valued triplet is given in the case of frames sharing a same direction. Two values describe ellipsis in their respective complex plane, and are linked by a linear relationship. Finally, the attempted numerical schemes for frame field smoothing are derived for completeness.

CHAPTER



Triangulation remeshing based on one-to-one parameterization

This chapter is a reproduction of the following paper

Beaufort, P.-A., Geuzaine, C., Remacle, J.-F., submitted, *Automatic surface mesh generation for discrete models – A complete and automatic pipeline based on reparameterization*, Journal of Computational Physics.

Abstract

Triangulations are an ubiquitous input for the finite element community. However, most raw triangulations obtained by imaging techniques are unsuitable as-is for finite element analysis. In this paper, we give a robust pipeline for handling those triangulations, based on the computation of a one-to-one parametrization for automatically selected patches of input triangles, which makes each patch amenable to remeshing by standard finite element meshing algorithms. Using only geometrical arguments, we prove that a discrete parametrization of a patch is one-to-one if (and only if) its image in the parameter space is such that all parametric triangles have a positive area. We then derive a non-standard linear discretization scheme based on mean value coordinates to compute such one-to-one parametrizations, and show that the scheme does *not* discretize a Laplacian on a structured mesh. The proposed pipeline is implemented in the open source mesh generator Gmsh, where the creation of suitable patches is based on triangulation topology and parametrization quality, combined with feature edge detection. Several examples illustrate the robustness of the resulting implementation.

Keywords: Triangulations, Finite Element, Remeshing, Parameterization, Mean Value Coordinates, Gmsh, Feature Edge, Longest Edge Bisection

1 | Introduction

Engineering designs are often encapsulated in Computer Aided Design (CAD) systems. This is usually the case in automotive, shipbuilding or aerospace industries. The finite element method is the prominent technique for performing analysis of these designs and this method requires a finite element mesh, i.e. a subdivision of CAD geometrical entities into a (large) collection of simple geometrical shapes such as triangles, quadrangles, tetrahedra and hexahedra, arranged in such a way that if two of them intersect, they do so along a face, an edge or a node, and never otherwise.

In CAD systems, the geometry of surfaces is described through a parametrization i.e. a mapping

$$\mathbf{x} : A \mapsto \mathbb{R}^3, \quad (u; v) \mapsto \mathbf{x}(u; v) \quad (2.1)$$

where $A \subset \mathbb{R}^2$ is usually a rectangular region $[u_0, u_1] \times [v_0, v_1]$. When finite element mesh generation procedures have access to such parametrizations $\mathbf{x}(u; v)$

of surfaces, it is in general a good idea to generate a planar mesh in the parametric domain A and map it in 3D. This way of generating surface meshes is called *indirect* (Borouchaki et al. (2000)), and is the predominant method for generating high-quality finite element surface meshes in a robust manner. This approach is in particular followed by the open source mesh generator Gmsh (Geuzaine and Remacle (2009)), which directly interacts with CAD systems to get parametrizations $\mathbf{x}(u; v)$ as well as their derivatives (normals, curvatures...). The nature of the mappings $\mathbf{x}(u; v)$ that are provided by CAD systems is such that anisotropic planar meshing capabilities are required in order to be able to generate quality meshes in 3D. Those mappings may be very irregular and even singular, for example at the two poles of a sphere. Gmsh’s surface planar mesh generators have been designed in such a way that they can handle very distorted metrics (Remacle and Geuzaine (2019)) while still providing high quality outputs.

In domains like geophysics or in bio-sciences, however, the geometry of the models is rarely described through CAD models. Most often, those geometries are produced through imaging (segmentation, voxelization, ...) whose geometrical output can be reduced to a triangulation. Several authors have proposed *direct* approaches (such as Frey (2001); Béchet et al. (2002)) that modify this raw “geometrical” mesh to produce a “computational” mesh with elements of controlled shapes and sizes. The aim of this paper is to show that the *indirect* approach is also possible in this case, in which a (global) parametrization $\mathbf{x}(u; v)$ is not readily available. Starting from a triangulation, our aim is to build a *set of parametrizations* that form an atlas of the model, and which can be used as-is by existing finite element mesh generators.

This paper describes the complete pipeline that allows to build the atlas of the model together with the parametrizations of all its maps. It aims at being self-consistent, which makes it quite exhaustive. In §2, some theoretical background on mappings is presented. Then, §3 develops the concept of discrete parametrizations. A complete set of proofs based on purely geometrical arguments is given that assert the injectivity of the discrete maps that are used. The way Gmsh handles the input in order to ease the parametrization and meshing process is described within §4. We point out the drawback of a general processing of coarse discrete surfaces in §5, and discuss two ways to handle such coarse discretizations. Several examples are presented in §6, and conclusions are drawn in §7.

2 | Mappings

A parametrization $\mathbf{x}(u; v)$ as defined in Equation (2.1) is regular if $\partial_u \mathbf{x}$ and $\partial_v \mathbf{x}$ exist and are linearly independent:

$$\partial_u \mathbf{x} \times \partial_v \mathbf{x} \neq \mathbf{0}$$

for any $(u; v) \in A$. In other words, $\mathbf{x}(u; v)$ is regular if and only if the Jacobian matrix

$$J = \frac{\partial \mathbf{x}}{\partial (u; v)} \in \mathbb{R}^{3 \times 2} \quad (2.2)$$

associated to $\mathbf{x}(u; v)$ has rank 2 $\forall (u; v) \in A$. The nature of the mapping $\mathbf{x}(u; v)$ is fully characterized by the *singular value decomposition* (SVD) of its Jacobian (2.2). Its singular values $\sigma_1 \geq \sigma_2 > 0$ allow to characterize \mathbf{x} :

- \mathbf{x} is *isometric* if and only if $\sigma_1 = \sigma_2 = 1$,
- \mathbf{x} is *conformal* if and only if $\frac{\sigma_2}{\sigma_1} = 1$,
- \mathbf{x} is *equiareal* if and only if $\sigma_1 \sigma_2 = 1$.

Isometric parametrizations preserve essentially everything (lengths, areas and angles). With such nice properties, generating well shaped triangles in the planar domain A will lead to a well shaped mesh in 3D. Disappointingly, such length preserving mappings do not exist for surfaces that are not developable (Struik, 1961, Chapter 2, §4) i.e. that have non zero Gaussian curvatures.

Conformal mappings conserve angles. If $\mathbf{x}(u; v)$ is conformal, isotropy is preserved and standard isotropic mesh generators can do the surface meshing job. Again, the odds are against us: although it is possible to build conformal mappings for most surfaces, it is very difficult to ensure global injectivity of such mappings, even though conformal mappings are always locally injective. Thus, ensuring the global one-to-oneness of conformal mappings is still an open question (see Lévy et al. (2002)).

Equiareal mappings have no interest in mesh generation. Thus, in general, mesh generators are faced with general parametrizations that do not preserve anything. This means that anisotropic planar mesh generators are required to generate well shaped meshes in 3D. Anisotropic mesh generators usually take as input a Riemannian metric field defined in each $(u; v)$ of A . If the aim is to produce an isotropic 3D mesh with a mesh size defined by an isotropic mesh size field $h(\mathbf{x}(u; v))$, the metric tensor that is used by the mesh generator will be

$$M(u; v) = \frac{J^T J}{h^2}.$$

Let us assume for example that the surface to be meshed is an ellipsoid. Fig. 2.1 shows a 3D surface mesh that is adapted to the maximal curvature of the surface as well as its counterpart in the parameter plane of the ellipsoid. The particular ellipsoid of Fig. 2.1 is $e = 7$ times wider in the x direction than in

the two other directions y and z . Its parametrization (which is standard to most CAD systems) is

$$\begin{aligned}x(u, v) &= e \sin u \sin v \\y(u, v) &= \sin u \cos v \\z(u, v) &= \cos u\end{aligned}$$

where $u \in [0, \pi]$ is the inclination and $v \in [-\pi, \pi[$ is the azimuth. The metric tensor associated to that mapping is

$$M = \frac{1}{h^2} \begin{pmatrix} \cos^2 v (e^2 \sin^2 v + \cos^2 v) + \sin^2 u & \sin u \sin v \cos u \cos v (e^2 - 1) \\ \sin u \sin v \cos u \cos v (e^2 - 1) & \sin^2 u (e^2 \cos^2 v + \sin^2 v) \end{pmatrix} \quad (2.3)$$

The mapping is obviously not regular when $u = 0$ and when $u = \pi$. This is surprisingly not so much of a problem for mesh generators. [Remacle and Geuzaine \(2019\)](#) propose a way to slightly modify meshing procedures in order to deal with singular mappings such as the one of the ellipsoid. The metric field (2.3) is anisotropic (see Fig. 2.1e) and non-uniform. Yet it is smooth and smoothness of mappings is the most important feature of $\mathbf{x}(u; v)$ in order to allow mesh generators to do a good job. When generating a mesh in an indirect fashion, a planar mesh, possibly anisotropic, is generated in the parameter plane A . Then, one may think that this planar mesh is mapped in 3D through $\mathbf{x}(u; v)$, which is not true: only corners of the triangles are mapped in 3D and those corners are connected together with 3D straight lines that are not the actual mapping of 2D straight lines. In the best case scenario, any 2D straight line connecting points $(u_a; v_a)$ and $(u_b; v_b)$ corresponds the geodesic between those two points. When the metric M is locally constant, geodesics are straight lines and the indirect meshing approach gives good results. When the metric varies rapidly along one given edge, then indirect meshing becomes difficult. In CAD systems, parametrizations are always smooth and indirect mesh generation is always possible.

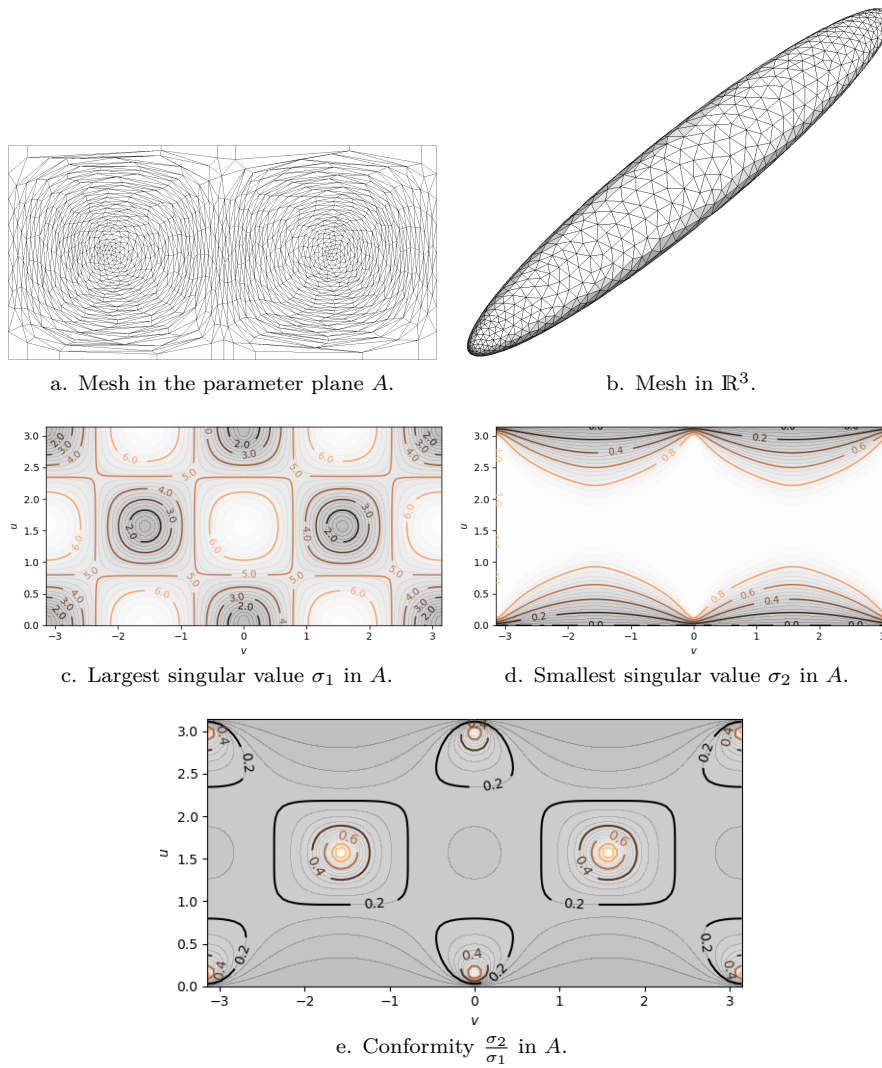


Figure 2.1: The case of an ellipsoid. Fig. a shows the mesh of the ellipsoid in the parameter space while Fig. b shows the same mesh in the 3D space. Fig. c and d show the largest and smallest singular values σ_1 and σ_2 of the jacobian J . Fig. e shows the non conformity parameter σ_2/σ_1 .

3 | Discrete Parametrizations

In Section §2, we have shown that having a smooth parametrization was the condition to allow indirect surface meshing. CAD systems provide smooth parametrizations but CAD models are not the only geometrical representations that are available in engineering analysis. In many domains of engineering interest, geometries of models are described by triangulations. We call such models *discrete models*.

Assume a triangulation T with $\#p$ nodes (vertices), $\#e$ edges and $\#t$ triangles which are correctly oriented to each others. Finding a parametrization of T consists in assigning to every vertex p_i of the triangulation a pair of coordinates $(u_i; v_i)$. If every triangle (p_i, p_j, p_k) , with $p_\bullet \in \mathbb{R}^{3^1}$ of the triangulation has a positive area in the $(u; v)$ plane, then the parametrization is injective.

A parametrization of T onto a subset of $A \subset \mathbb{R}^2$ exists if the triangulation corresponds to the one of a planar mesh. Assume that triangulation T is simply connected with $\#b$ boundaries, $\#h$ vertices on those boundaries and whose the genus is g . Then the surface is parameterizable if and only if

$$\#t = 2(\#p - 1) + 2(\#b - 1) - \#h + 4g$$

In what follows, we present some existing material that is detailed in numerous publications such as Floater and Hormann (2005); Tutte (1963); Remacle et al. (2010). The main interest of this section is that we take here the point of view of the numerical geometer. The main result about the one-to-oneness of mappings is proven without using one single theorem of analysis such as maximum principles of Radó-Kneser-Choquet theorem (see (Dirac, 1953)).

Consider an internal vertex i of T and $J(i)$ the set of indices whose the corresponding nodes are connected to the node i (in other words, edge (i, j) exists $\forall j \in J(i)$). The value of the parametric coordinates (u_i, v_i) at vertex i will be computed as a weighted average of the coordinates (u_j, v_j) of its neighboring vertices:

$$\sum_{j \in J(i)} \lambda_{ij}(u_i - u_j) = 0 \quad , \quad \sum_{j \in J(i)} \lambda_{ij}(v_i - v_j) = 0 \quad (2.4)$$

where λ_{ij} are coefficients. This scheme is called a *difference scheme* that involves only differences $(u_i - u_j)$, with $j \in J(i)$. If every λ_{ij} is positive, values of u_i and v_i are convex combinations of their surrounding values. In a geometrical point of view, it actually means that point (u_i, v_i) lies in the convex hull \mathcal{H}_i of its neighboring vertices.

¹In what follows, a triangle is denoted by the indices of its nodes, i.e. (i, j, k) instead of (p_i, p_j, p_k) .

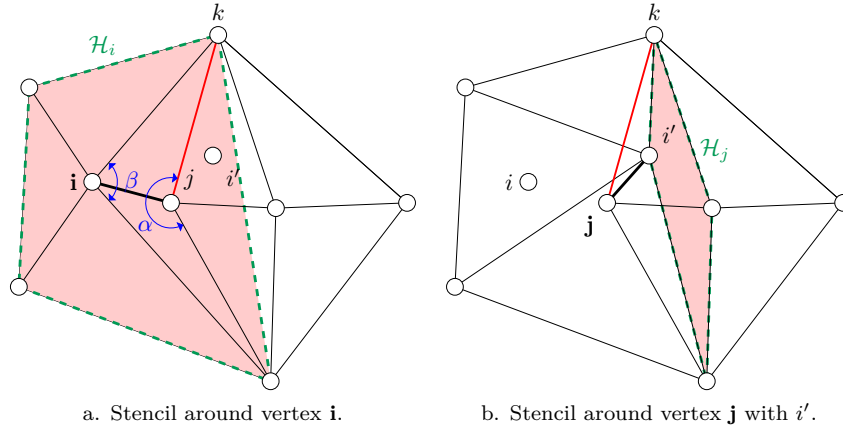


Figure 2.2: Sketch of proof for monotonicity.

With that assumption, it is easy to prove that the mapping provided by any positive scheme of the type (2.4) is one-to-one. Let us consider a triangle (i, j, k) in the parameter plane $(u; v)$, Fig.2.2. If edge (j, k) belongs to \mathcal{H}_i , that triangle (i, j, k) is obviously positive.

On the other hand, if (j, k) is inside \mathcal{H}_i , as it is the case in Fig. 2.2a, then (j, k) does not belong to \mathcal{H}_i and moving i to i' creates an inverted triangle (i', j, k) while keeping $\mathcal{H}_i = \mathcal{H}_{i'}$. In this case, i' is inside \mathcal{H}_i while triangle (i', j, k) is inverted. It is easy to see that moving i to i' implies that j would be outside \mathcal{H}_j which is in contradiction with the hypothesis that each vertex is inside its convex hull, Fig.2.2b. Vertex j being inside \mathcal{H}_i implies that $\alpha > \pi$. The sum of the four angles of a quadrangle is 2π . This implies that $\beta < \pi$ which implies that edge (i, k) belongs to \mathcal{H}_j . So, moving i to i' puts j outside \mathcal{H}_j .

Now see what happens on the outer boundary ∂A of the $(u; v)$ domain A . There, points have no neighboring hull. Yet, assuming that ∂A is convex, then all vertices of ∂A that are connected to internal vertices belong to the convex hull of those latter internal vertices. Besides, no internal vertex cannot be situated outside A . It means that all triangles having (at least) one edge belonging to ∂A are positive. This last part of the proof has some similarities with the one of Floater (2003).

This means that a positive scheme applied to a convex domain implies that the discrete parametrization is one-to-one. In our case, we will always choose ∂A as the unit circle.

Now, the right choice of the λ_{ij} is of outmost importance for ensuring a good parametrization. Our use of parametrization is meshing. The first and non negotiable property of the discrete parametrization is one-to-oneness. We thus choose a positive scheme and a $(u; v)$ domain that is a unit circle. The second priority is smoothness, we will develop that aspect below. The icing on the cake would be conformity (i.e. angle preservation) but, as noted in §2, Gmsh’s mesh generators are comfortable with anisotropic mappings and we will not put any effort on that aspect of the game (our aim is not texture mapping like in computer graphics, so we are OK to map squares on circles).

3.1 Parametrization smoothness

We look here for a smooth function $\mathbf{x}(u, v)$ i.e. a continuous function whose derivatives are smooth as well because we want σ_1 and σ_2 to be smooth and σ_1 and σ_2 are by-products of the metric i.e. a tensor computed using the first derivatives of $\mathbf{x}(u, v)$. Tutte’s barycentric mapping (see [Tutte \(1963\)](#)) consists in choosing $\lambda_{ij} = 1$. This choice leads to very irregular mappings that are useless for mesh generation purposes. The idea that has been advocated by many authors (e.g. [Lévy et al. \(2002\)](#); [Marchandise et al. \(2011\)](#)) is to solve a partial differential equation whose solutions are inherently smooth. For example, the solution of Laplace equations on domains with smooth boundaries and with smooth boundary conditions are C^∞ and it is indeed a good idea to choose the λ_{ij} in such a way the difference operator (2.4) is a discrete version of the Laplace operator.

3.2 Laplace smoothing using \mathcal{P}^1 finite elements

The standard P^1 finite element formulation of the Laplace problem is well known for more than a half of a century. In the early days, some authors ([Duffin \(1959\)](#)) have written coefficients $\lambda_{ij}^{\text{FEM}}$ in a quite geometrical fashion (see [Fig. 2.3](#)):

$$\lambda_{ij}^{\text{FEM}} := \frac{1}{2} \left(\frac{\cos(\theta_k)}{\sin(\theta_k)} + \frac{\cos(\theta_l)}{\sin(\theta_l)} \right). \quad (2.5)$$

For sake of completeness, the so called “cotangent formula” (2.5) is fully derived in [Appendix A.A](#). Coefficients $\lambda_{ij}^{\text{FEM}}$ of (2.5) may be negative for $\theta_\bullet \in (\frac{\pi}{2}; \pi)$, which could lead to scheme that is not provably injective; ([Floater, 1998](#), §5) gives a simple example where the Laplacian smoothing fails to provide an injective mapping. This is the very old result that states that the maximum principle satisfied by solutions of Laplace equations is only guaranteed *a priori* by finite elements computed on acute triangulations, i.e. triangulations without obtuse angles. Acute triangulations are a sufficient condition for injectivity.

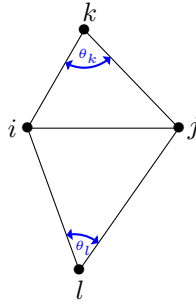


Figure 2.3: Definitions of θ_k and θ_l for the difference scheme corresponding to the linear Galerkin approach.

Yet it is not necessary and it is indeed complicated to find examples where finite elements fail to provide one-to-one parametrizations. Disappointingly, in the world of mesh generation, limit cases that happen once in a thousand have to be avoided. So, we will not use the finite element version of Laplacian smoothing for parametrizing our surfaces.

3.3 Mean value coordinates

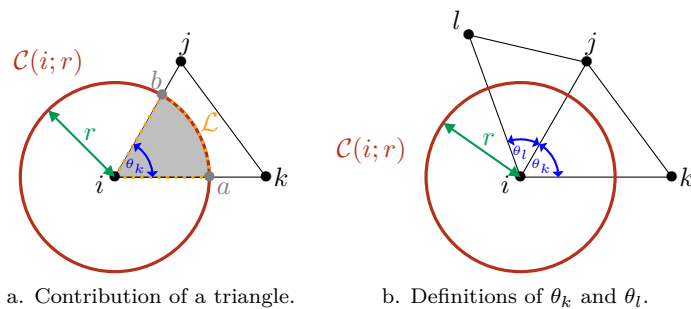


Figure 2.4: Derivation of the difference scheme corresponding to the mean value coordinates.

A continuous function f is a solution of Laplace equation $\nabla^2 f = 0$ on an open set $A \subset \mathbb{R}^2$ if and only if, for every $\mathbf{x} \in A$, $f(\mathbf{x})$ is equal to the average

value of f over every circle of radius r $\mathcal{C}(\mathbf{x}; r)$ that fully belongs to A :

$$f(\mathbf{x}) = \frac{1}{2\pi r} \int_{\mathcal{C}(\mathbf{x}; r)} f(\mathbf{x}') \, d\mathbf{x}'. \quad (2.6)$$

This principle states that the extrema of the mapping are located on the boundary of the domain, and that there is not local extremum inside the domain.

(Floater, 2003) proposes a way to compute λ_{ij} that actually mimics property (2.6): this scheme is called mean value coordinates. In this paper, we re-derive Floater's λ_{ij} corresponding to mean value coordinates using a finite element point of view. According to (2.6), the value f_i is the average of values $f(\mathbf{x})$ along a circle $\mathcal{C}(i; r)$ of radius r centered on i (see Fig. 2.4). A linear interpolation $f(x; y) = \sum_j f_j \phi_j(x; y)$ is assumed over each triangle \mathcal{T}_{ijk} . We are going to compute the contribution of triangle \mathcal{T}_{ijk} for (2.6)

$$\theta_k r f_i = \int_{\widehat{ab}} f_i \phi_i + f_j \phi_j + f_k \phi_k \, ds$$

where θ_k is the angle between edges $[ij]$ and $[ik]$, and \widehat{ab} is the circle arc of $\partial\mathcal{C}(i; r)$ contained in \mathcal{T}_{ijk} , Fig. 2.4a. Since $\phi_i + \phi_j + \phi_k = 1$,

$$\underbrace{\left(\theta_k r - \int_{\widehat{ab}} \phi_i \, ds \right)}_{\int_{\widehat{ab}} \phi_j + \phi_k \, ds} f_i - \int_{\widehat{ab}} \phi_j \, ds f_j - \int_{\widehat{ab}} \phi_k \, ds f_k = 0$$

which gives

$$\underbrace{\int_{\widehat{ab}} \phi_j \, ds}_{\lambda_{ij}} (f_i - f_j) + \underbrace{\int_{\widehat{ab}} \phi_k \, ds}_{\lambda_{ik}} (f_i - f_k) = 0$$

over \mathcal{T}_{ijk} .

Linear shape function ϕ_j associated to node j in \mathcal{T}_{ijk} corresponds to

$$\phi_j(x; y) = \frac{y}{y_j}$$

where y is the vertical coordinate relative to edge $[ik]$ and y_j is the y -coordinate of node j . We compute the integral of y over \widehat{ab} from the contour \mathcal{C} composed of \widehat{ab} , $[bi]$ and $[ia]$

$$\int_{\mathcal{C}} y \, ds$$

From normal vector of $\mathcal{C}(i; r)$ $\hat{n} = \frac{1}{r}(x; y)$, we get

$$\int_{\mathcal{L}} \frac{y}{r} \, ds = \int_{\mathcal{L}} \hat{n} \cdot \mathbf{e}_y \, ds$$

with $\mathbf{e}_y = (0; 1)$. The divergence of \mathbf{e}_y is obviously zero, and owing to the divergence theorem

$$\int_{\mathcal{L}} \frac{y}{r} ds = \int_{\mathcal{R}(\mathcal{L})} \nabla \cdot \mathbf{e}_y dx dy = 0$$

where $\mathcal{R}(\mathcal{L})$ is the region surrounded by \mathcal{L} (gray area, Fig. 2.4a). The integral along the circle arc \widehat{ab} is then equal to the opposite of integrals along edges $[bi]$ and $[ia]$ of triangles \mathcal{T}_{ijk}

$$\begin{aligned} \int_{\widehat{ab}} \underbrace{\hat{n} \cdot \mathbf{e}_y}_{\frac{y}{r}} ds &= - \left(\int_{[ia]} \underbrace{\hat{n} \cdot \mathbf{e}_y}_{-1} ds + \int_{[bi]} \underbrace{\hat{n} \cdot \mathbf{e}_y}_{\cos(\theta_k)} ds \right) \\ &= -(-r + r \cos(\theta_k)) \\ &= r(\cos(\theta_k) - 1) \end{aligned}$$

Since $y_j = l_{ij} \sin(\theta_k)$, with l_{ik} the length of edge $[ij]$

$$\int_{\widehat{ab}} \phi_k ds = r^2 \frac{\tan\left(\frac{\theta_k}{2}\right)}{l_{ij}}$$

Choosing a radius r small enough (i.e. smaller than the smallest edge within the triangulation) allows to simplify the finite scheme (2.5) by r^2 , which means that the scheme does not depend on the circle of integration. The coefficient λ_{ij} is then given by

$$\lambda_{ij} = \frac{\tan\left(\frac{\theta_k}{2}\right) + \tan\left(\frac{\theta_l}{2}\right)}{l_{ij}} \quad (2.7)$$

We notice that $\lambda_{ij} > 0$, $\forall \theta_{\bullet} \in (0; \pi)$. The difference scheme (2.4) with (2.7) builds linear injective mappings. This monotone scheme is not symmetric, except on equilateral triangulations.

At that point, one can raise the question of the actual accuracy of the MVC scheme for discretizing Laplace equation, which is our guarantee of smoothness. A convergence experiment² has been performed on a square $[0; 1] \times [0; 1]$ on various meshes (Fig. 2.5) using the standard technique of manufactured solutions. We choose $f(x; y) = \sin(2\pi x) \cosh(2\pi y)$ whose laplacian $\nabla^2 f$ is zero.

Fig. 2.6 shows that MVC scheme does not exhibit the usual FEM convergence. The absence of symmetry of the MVC scheme implies that only $\mathcal{O}(h)$ convergence is observed for the L^2 norm. Yet, the MVC scheme seems to converge on all meshes except the structured one. This behavior is due to the fact that the MVC scheme does not correspond to a Laplacian over a structured triangulation, Appendix A.B.

²The experiment has been performed with the Gmsh API, given in supplementary material.

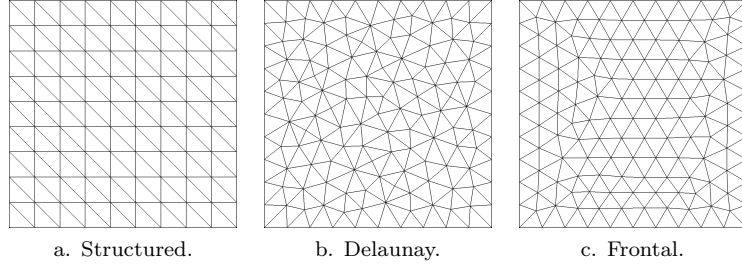
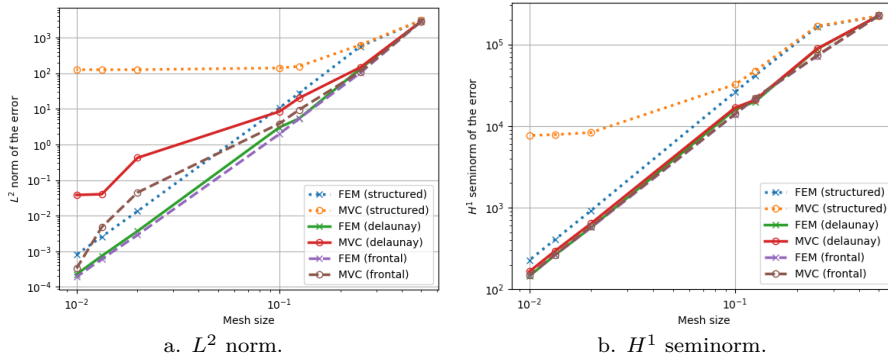


Figure 2.5: Types of meshes on a square.

Figure 2.6: h -convergence of discrete schemes (2.4) with (2.5) VS (2.7) on mesh types of Fig. 2.5.

3.4 Boundary conditions

We consider 3D surfaces that are topologically equivalent to a disk with $\#b - 1$ internal boundaries. The parametric domain that is considered is always a unit disk

$$A = \{(u; v) \in \mathbb{R}^2 : u^2 + v^2 < 1\}.$$

The setup is described in Fig. 2.7. Dirichlet boundary conditions are applied on $\mathbf{x}(\partial A)$ that actually ensure that the u, v coordinates on $\mathbf{x}(\partial A)$ correspond to the unit circle

$$\delta A = \{(u; v) \in \mathbb{R}^2 : u^2 + v^2 = 1\}.$$

We should now decide on what boundary conditions to apply on the other boundaries δB_i . The issue here is that we do not know *a priori* their position in

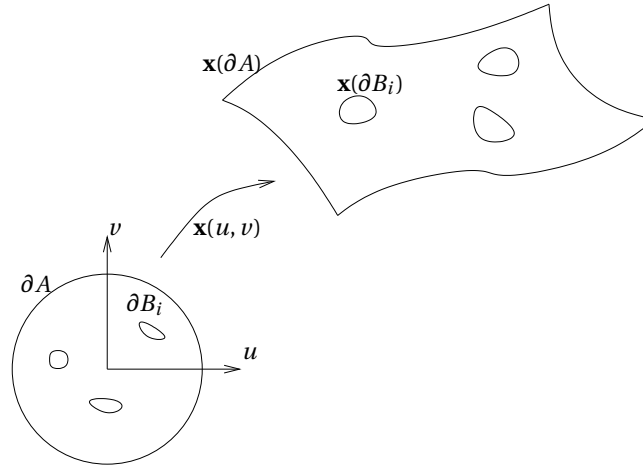


Figure 2.7: A 3D domain that is topologically equivalent to a disk with 3 internal boundaries and its parametric domain A .

the parameter plane. We could decide their position and insert $\#b-1$ small circles inside A . Yet, this would lead to a parametrization that is quite distorted. Another option is to apply the smoother as is to every internal points, including the ones on the internal boundaries. This indeed corresponds to imposing homogeneous Neumann boundary conditions on every internal boundary. It is indeed easy to prove that this choice still leads to a one-to-one parametrization. One first thing to note is that if every ∂B_i is convex and if we use a convex combination map like (2.7), then the mapping is one-to-one.

Assume that points p_1, p_2, \dots, p_k form a closed loop in the parameter plane and that every point lies in the convex hull of its neighbors, such as Fig. 2.8. Then, polygon (p_1, p_2, \dots, p_k) is convex. Indeed, if every three consecutive points i, j, k of such a loop form an angle α_j that is greater or equal to π , then the edges $(i; j)$ and $(j; k)$ lay in the convex hull \mathcal{H}_j . If it is true for every point of the loop corresponding to the hole, then its loop in the parameter plane is convex. From §3, we know that a positive scheme produce a one-to-one parametrization. Hence, if no condition are imposed on the holes - which corresponds to Neumann condition within FEM formulation - the parametric representation of those holes correspond to convex loop, whatever the initial shape of holes (i.e. even if they were concave).

Fig. 2.9a shows a concave domain with a concave hole that is mapped using (2.7) and where homogeneous Neuman boundary conditions were applied to the internal boundary. In this case, $\partial_n u = \partial_n v = 0$ on the internal boundary and

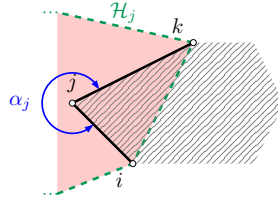


Figure 2.8: Three consecutive points belonging to a loop describing an hole (hatched area) in A .

the parametrization is close to be singular because the two tangent vectors are nearly parallel: both of them are *weakly* orthogonal to the boundary (see Fig. 2.9a)!

Another option consists of *filling the holes*, which leads to better results in practice (see Fig. 2.9b).

A heuristic to fill holes is to link each vertex lying on the hole to a *pseudo center* \mathbf{c} of the hole. This pseudo center corresponds to the center of the circle associated to the hole, Fig. 2.10. The hole is approximated by a circle whose circumference $2\pi r$ corresponds to the perimeter of the hole $\sum_j l_j$. The vertices defining the hole are then assumed to lie on such a circle. New triangles are then defined, by connecting those vertices to the pseudo center of the hole. The angle α_j defined by $\angle \mathbf{v}_i \mathbf{c} \mathbf{v}_{i+1}$ is assumed to be equal to $\frac{l_j}{r}$. Since the triangles filling the hole are assumed to share \mathbf{c} , they are isosceles. All those assumptions enable to average the parametric coordinates of vertices lying on the hole, such that there was no hole. The triangles filling the hole are not explicitly built.

The heuristic performs well, even if the hole is concave and badly shaped, Fig. 2.9b. The improvement compared to the homogeneous Neumann condition is obvious, Fig. 2.9a. Actually, some parametric triangles of Fig. 2.9a are too tight for meshing purposes.

The drawback of filling holes is that it increases the connectivity of the linear system enabling the computation of the underlined parametrization. Indeed, the pseudo centers correspond to extra unknowns which are related to the corresponding unknowns along each hole. Hence, the corresponding rows within the matrix representing the linear system may have a lot of nonzero. The corresponding linear system may become difficult to solve due to those latter rows. In order to conserve a quick process of parameterization, a threshold of the potential connectivity is set: if there are too many vertices on a hole, homogeneous Neumann boundary conditions are set. Otherwise, this hole is filled with the pseudo center.

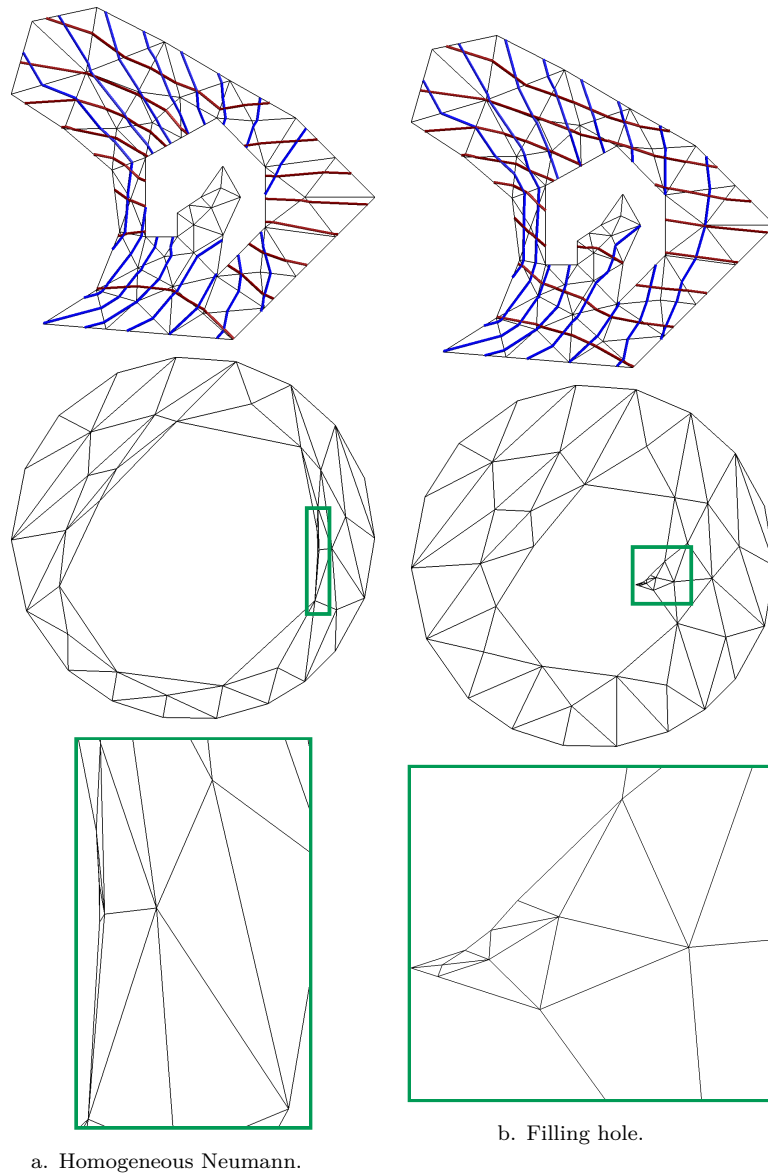


Figure 2.9: Demonstration of filling a concave hole with the circle assumption. Top: parametrization over the (discrete) geometry (u : red isolines, v : blue isolines). Bottom: triangles within the computed parametric space.

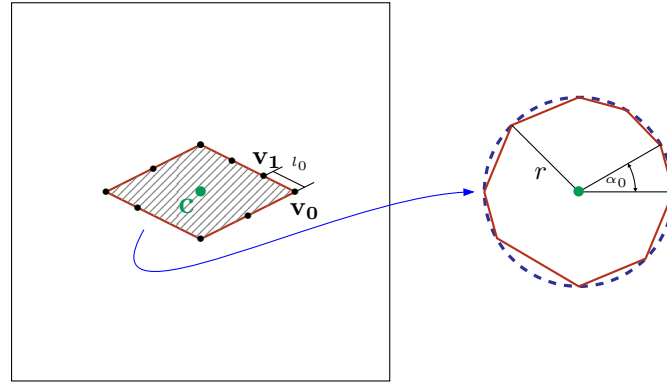


Figure 2.10: Exampled filling hole (hatched area).

4 | Gmsh's Pipeline for Discrete Surface Meshing

The specifications of Gmsh's algorithm for the generation of meshes on discrete surfaces are the following

- A conforming “watertight” geometrical triangulation is given as input.
- A mesh with user specified mesh size parameters is given as output by Gmsh where all mesh vertices lie exactly on the input triangulation.

In Gmsh's new pipeline, the problem of surface meshing is divided in two stages: (i) a pre-processing stage and (ii) a mesh generation stage. In order to explain the usefulness of the two stages of the pipeline, a relatively simple example will serve as a common theme to illustrate the various treatments that have to be undergone by a rough geometric triangulation to become a high quality finite element mesh.

Fig. 2.11 shows the geometric triangulation of a “Batman” object that is connected to a sort of torus.

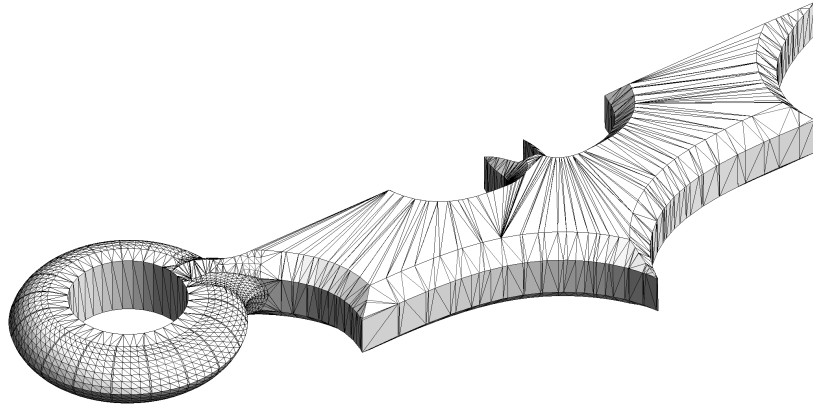


Figure 2.11: The Batman geometry.

In Gmsh's pipeling, a rough geometrical triangulation is taken as input. A triangulation like the one of Fig. 2.11 cannot be processed as is for a number of reasons.

4.1 Detecting feature edges

The geometrical triangulation of Fig. 2.11 is composed of a list of triangles. The first part of our pre-processing is to detect feature edges of the geometry that should be present in the final mesh. We use here a simple angle criterion (typically, user-defined) to detect feature edges. After detecting feature edges, a first version of the final atlas is created. Fig. 2.12 shows the Batman geometry where feature edges have been created for all edges that have two adjacent triangles with normals separated by an angle of more than 40 degrees. A first version of the final topology of the domain is created with model faces that are bounded by the feature edges. After the computation of feature edges, curvature tensors are computed at every vertex of every surface using the method of Rusinkiewicz (2004).

4.2 Creating the atlas

At that point, we are not yet ready to compute the atlas of the model i.e. the final boundary representation of our model together with the parametrization of all its model surfaces. As explained in §3 every model surface of the atlas should have the right topology. In this following step, we ensure that every map of the atlas has this right topology. When a surface has a larger genus,

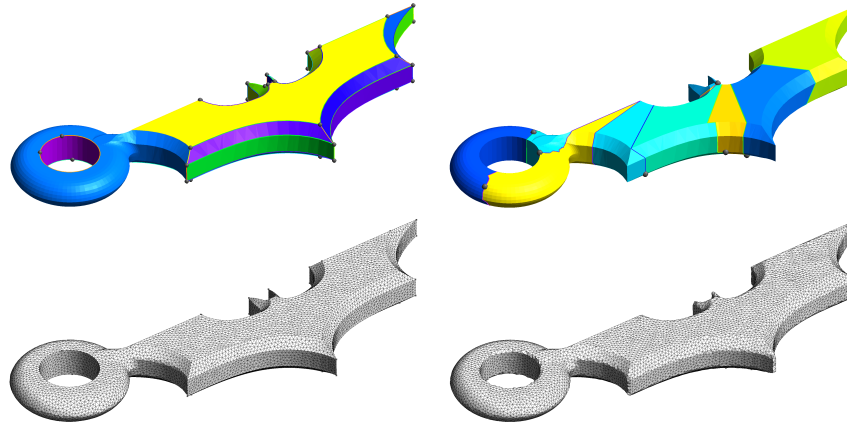


Figure 2.12: Top left Fig. shows the final model with feature edges detection (threshold angle of 40 degrees). Bottom left Fig. shows a uniform mesh on that model. Right Fig. show the final model and mesh without feature edges detection. The domain has been split automatically in such a way that every model face has the right topology.

it is split in two parts using METIS (see [Karypis and Kumar \(2009\)](#)), a graph partitioning software. This operation is applied up to the point when every surface is parametrizable.

It is also known that surfaces with large aspect ratios may lead to parametrizations that have non distinguishable coordinates. When the parametrization is computed, we also ensure that parametric triangles are not too small i.e that their area is not close to machine precision (see [Marchandise et al. \(2011\)](#)). If it is the case, the surface is split in two.

For large models, we also split surfaces that contain a too large number of triangles (typically 100,000). Computing mean value coordinates require to solve a non symmetrical system of equations and one of the design goals of the parametrization process is to be fast.

Fig. 2.12 (top right) shows the decomposition that has been done on the Batman model without pre-computing feature edges.

4.2.1 The final BREP

At that point, the input triangulation has been transformed into a proper boundary representation that has a valid topology and for which each face has been parametrized. All those topological and geometrical informations are now saved in the version 4 of the output mesh format of Gmsh. This “extended”

mesh file can be used as input to Gmsh’s surface mesh generators. Fig. 2.12 (bottom images) show meshes for both models generated using feature edges and automatic splitting.

5 | Improving Parametrization on Coarse Discrete Surfaces

The methodology that has been presented before is general and applies to triangulated surfaces of arbitrary complexity. Yet, geometrical triangulations of CAD surfaces may not be sufficiently dense to allow a smooth parametrization. For example, a good geometrical triangulation of a cylinder may not contain internal vertices as depicted on Fig. 2.13. We have computed the parametrization of this cylinder using mean value coordinates and the result is presented in Fig. 2.13a. Even if this parametrization is said “moderately noised”, it cannot be used for mesh generation purposes. Fig. 2.13b and 2.13c show conformity indicator $\frac{\sigma_2}{\sigma_1}$ both on the real and parameter space of the cylinder.

From this observation, a numerical analyst would suggest two ways to improve the computation: refining the solution (i.e. the input mesh), or increasing the order of the approximation (i.e. second order).

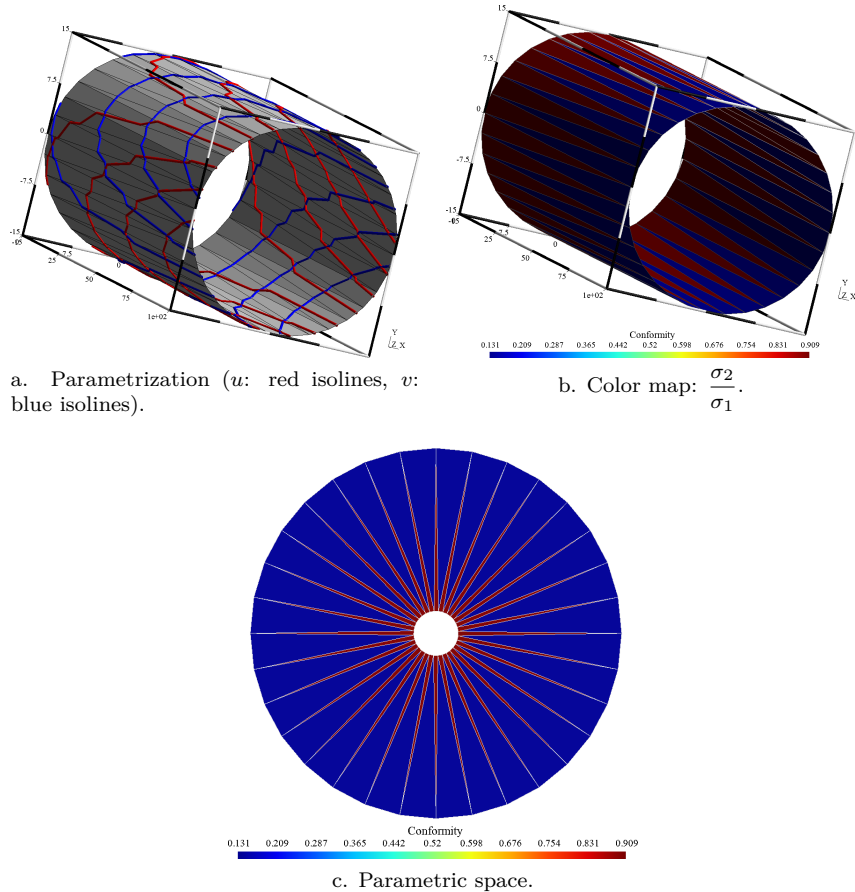


Figure 2.13: Parametrization on coarse stl triangulations: a cylinder.

5.1 Refinement by longest edge bisection

We refine the geometrical triangulation without changing its geometry i.e. only using edge splits. We use here a variant of the well known longest edge bisection process developed by (Rivara, 1997): edges to be split are tagged and the longest edge of the list is split, then the second longest edge is split and the process continues until the shortest edge of the list is split. We repeat the process several times up to the point all inner edges respect a length threshold. Fig. 2.14a shows the new geometrical mesh of the cylinder.

In order to illustrate the effect of this refinement on the parametrization,

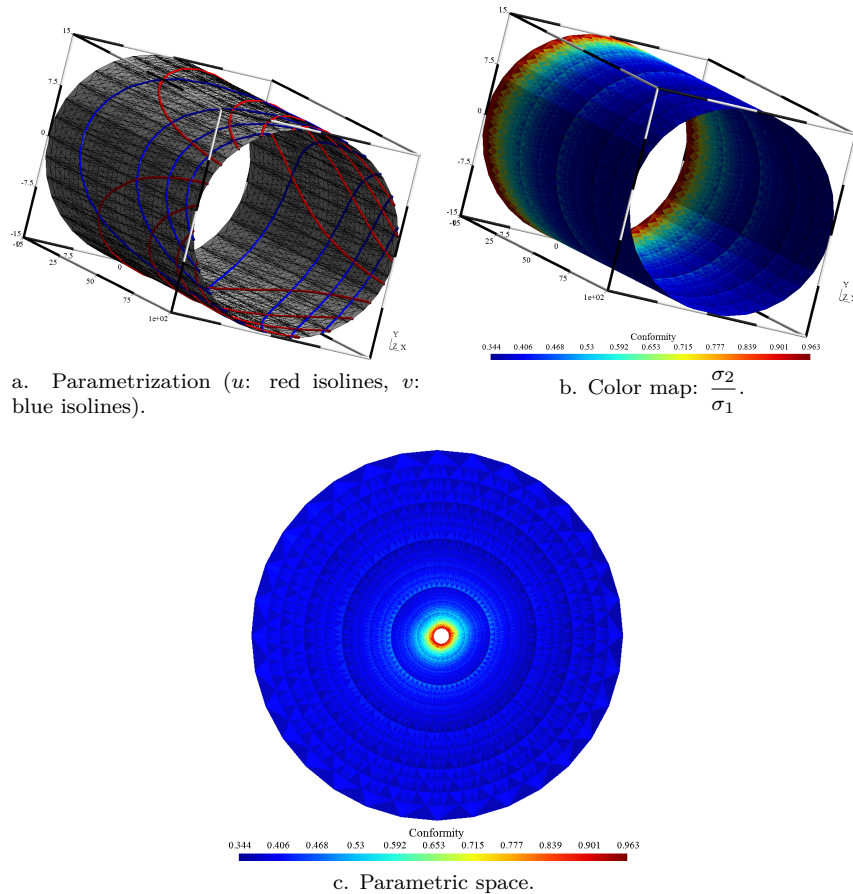
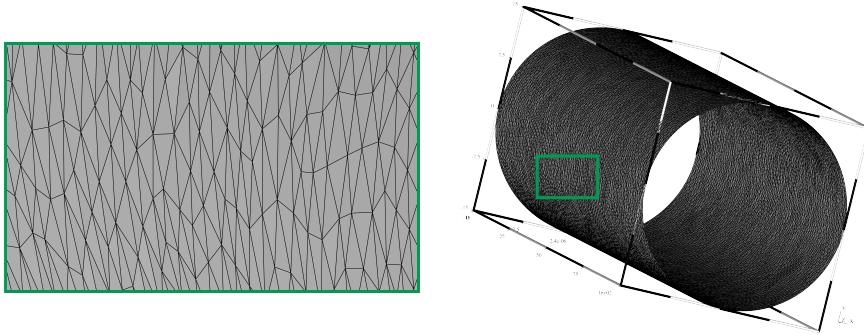
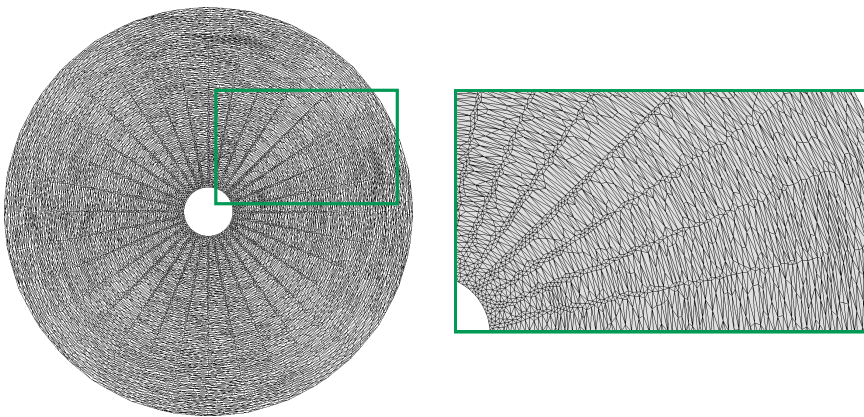


Figure 2.14: Parametrization on refined stl triangulations: the cylinder (5 iterations).

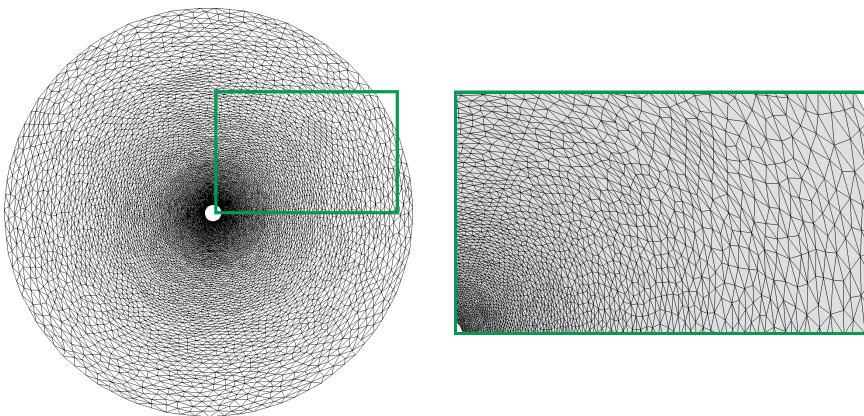
we have pre-computed a “good mesh” of the cylinder in the 3D space (see Fig. 2.15a). This good mesh has been inverse-mapped onto the parameter spaces of the non refined cylinder and on the refined cylinder. While doing that, we can see the meshes that should have been created by Gmsh’s surface meshers in both parameter planes to obtain the same “good mesh”. Fig. 2.15b shows the mesh in the parameter plane of the non refined geometrical cylinder: it contains series of elongated triangles followed by isotropic ones, illustrating the too great variability of the conformity parameter. In Fig. 2.15c, the mesh is anisotropic but element shapes are locally uniform and any good anisotropic mesher is able to generate that kind of mesh.



a. Good mesh on cylinder.



b. Mapping within the parametric space, *without* preprocessing (cf. Fig. 2.13c).



c. Mapping within the parametric space, *with* preprocessing (cf. Fig. 2.14c).

Figure 2.15: Effect on mapping a good mesh on a parametrization with (b) and without (c) edge refinement preprocessing.

5.2 Second order approximation

As in the piecewise linear approximation (see §3.3), we derive λ_{ij} from Lagrange \mathcal{P}^2 function shapes

$$\theta_k r f_i = \int_{\widehat{ab}} f_i \phi_i + f_j \phi_j + f_k \phi_k + f_{ij} \phi_{ij} + f_{jk} \phi_{jk} + f_{ik} \phi_{ik} ds$$

where ϕ_\bullet are the Lagrange \mathcal{P}^2 finite element shape functions, which are defined with the barycentric coordinates $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)$ (Ern and Guermond, 2013, Chapter 1, §1.2.4)

$$\begin{cases} \phi_a &= \mathbf{v}_a(2\mathbf{v}_a - 1), \quad a \in \{i, j, k\} \\ \phi_{ab} &= 4\mathbf{v}_a\mathbf{v}_b, \quad a, b \in \{i, j, k\} : a \neq b \end{cases}$$

Assigning coordinates relative to v_i , Fig. 2.16

$$\begin{aligned} \mathbf{v}_i &= (0; 0) \\ \mathbf{v}_j &= (l_{ij} \cos(\theta_k); l_{ij} \sin(\theta_k)) \\ \mathbf{v}_k &= (l_{ik}; 0) \end{aligned}$$

Again, $\phi_i + \phi_j + \phi_k + \phi_{ij} + \phi_{jk} + \phi_{ik} = 1$ enables us to write

$$\begin{aligned} &\underbrace{\left(\theta_k r - \int_{\widehat{ab}} \phi_i ds \right)}_{\int_{\widehat{ab}} \phi_j + \phi_k + \phi_{ij} + \phi_{jk} + \phi_{ik}} f_i - \int_{\widehat{ab}} \phi_j ds f_j - \int_{\widehat{ab}} \phi_k ds f_k \\ &- \int_{\widehat{ab}} \phi_{ij} ds f_{ij} - \int_{\widehat{ab}} \phi_{jk} ds f_{jk} - \int_{\widehat{ab}} \phi_{ik} ds f_{ik} = 0 \end{aligned}$$

which gives

$$\begin{aligned} &\underbrace{\int_{\widehat{ab}} \phi_j ds (f_i - f_j)}_{\lambda_{ij}} + \underbrace{\int_{\widehat{ab}} \phi_k ds (f_i - f_k)}_{\lambda_{ik}} \\ &+ \underbrace{\int_{\widehat{ab}} \phi_{ij} ds (f_i - f_{ij})}_{\lambda_{i(ij)}} + \underbrace{\int_{\widehat{ab}} \phi_{jk} ds (f_i - f_{jk})}_{\lambda_{i(jk)}} + \underbrace{\int_{\widehat{ab}} \phi_{ik} ds (f_i - f_{ik})}_{\lambda_{i(ik)}} = 0 \end{aligned}$$

We use SymPy (see (Meurer et al., 2017)) to compute $\lambda_{ij}^{\mathcal{P}^2}$ (code in supplementary material)

$$\lambda_{ij}^{\mathcal{P}^2} = \frac{r^2}{l_{ij}^2 \sin^2(\theta_k)} ((l_{ij} - r) \cos(\theta_k) \sin(\theta_k) + r\theta_k - l_{ij} \sin(\theta_k)) \quad (2.8)$$

We should derive the other coefficients $\lambda_{\bullet}^{\mathcal{P}^2}$, but something is wrong with (2.8). We cannot get rid of r within the expression. It means that the coefficients give the average for a certain circle of radius r . Yet, it has to be for *any* circle, whatever the radius. It is then not possible to derive $\lambda^{\mathcal{P}^2}$ for a monotone scheme.

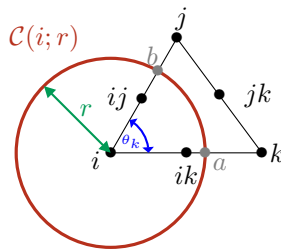


Figure 2.16: Sketch for quadratic approximation of $\lambda_{ij}^{\mathcal{P}^2}$.

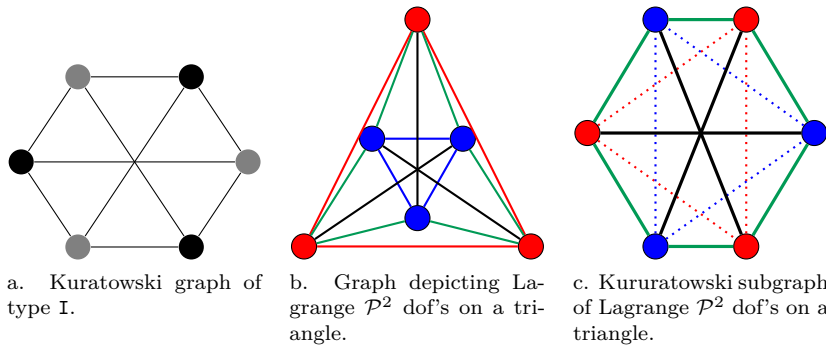


Figure 2.17: Graph corresponding to Lagrange \mathcal{P}^2 dof's on a triangle has no planar representation.

Actually, graph theory states such a result. Lagrange \mathcal{P}^2 degrees of freedom on a triangle may be depicted by a 3-connected graph, Fig. 2.17b. Tutte (Tutte, 1963, §4) claims that any graph having a Kuratowski subgraph is nonplanar. Fig. 2.17a corresponds to a Kuratowski graph. A graph is planar if it can be drawn on a plane, in such a way that its edges intersect only on vertices of the graph. It means that each vertex of the graph may correspond to a convex combination of its neighbors, which we aim. However, Fig. 2.17b has such a Kuratowski subgraph, Fig. 2.17c. The graph of Fig. 2.17b has no planar

representation. Hence, it means it is not possible to write Lagrange \mathcal{P}^2 scheme which is monotone.

6 | Examples

In this section, several complex examples are presented that show the level of robustness that has been attained by our methodology. The examples that have been chosen in order to challenge our algorithm and push it to the limit.

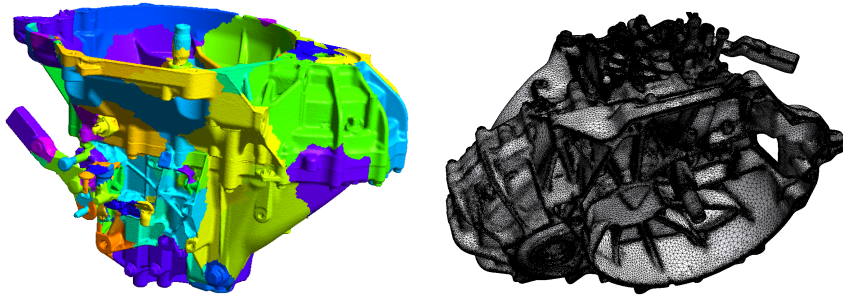


Figure 2.18: Complex scanned mechanical part. The initial triangulation (left) that contains 797,666 triangles has been split into 194 surfaces that are parametrizable. The mesh on the right that contains 1,762,388 triangles and has been adapted to the curvature of the original discrete surface. It has been generated by Gmsh in 640 seconds, including IO's.

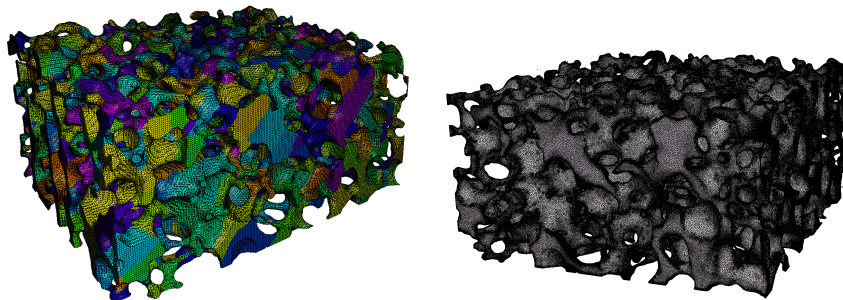


Figure 2.19: X-ray tomography image of a silicon carbide foam (from P. Duru, F. Muller and L. Selle, IMFT, ERC Advanced Grant SCIROCCO). The initial triangulation (left) that contains 1,288,116 triangles has been split into 1,802 surfaces that are parametrizable. The mesh on the right contains 4,922,322 triangles and has been adapted to the curvature of the original discrete surface. It has been generated by Gmsh in 1,187 seconds, including IO's.

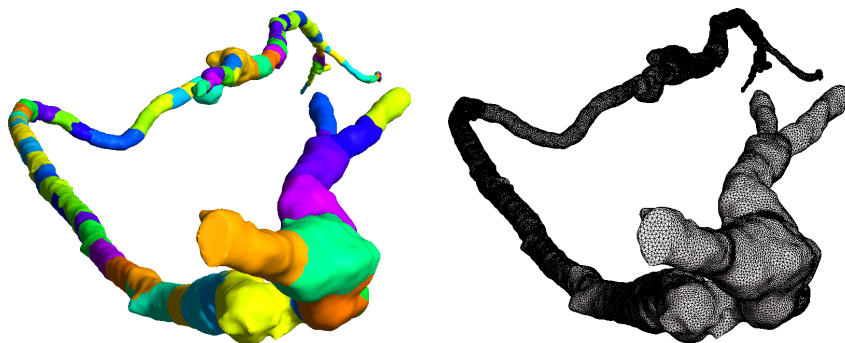


Figure 2.20: CT scan of an artery. The initial triangulation (left) that contains 63,468 triangles has been split into 101 surfaces that are parametrizable. Most of the cuts were done because of the large aspect ratio of the tubular domains. The uniform mesh on the right that contains 170,692 triangles has been generated by Gmsh in 22 seconds, including IO's.

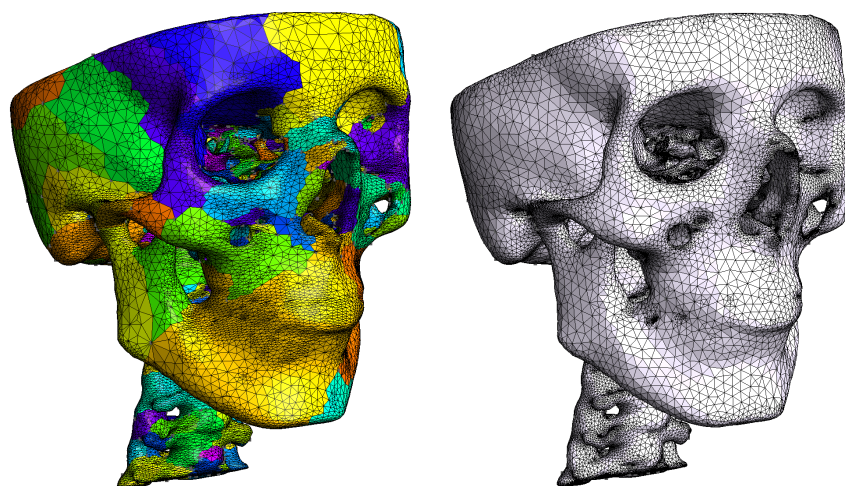


Figure 2.21: Remeshing of a skull. The initial triangulation (left) that contains 142,742 triangles has been split into 715 surfaces that are parametrizable. The mesh on the right is adapted to the surface curvature and contains 323,988 triangles and has been generated by Gmsh in 58 seconds, including IO's.

7 | Conclusion

This paper has demonstrated the Gmsh's ability to remesh robustly poor quality triangulations, for the purpose to run finite element analysis. Gmsh's

pipeline essentially relies on the one-to-oneness of parametrization, where conformity is not mandatory since a mesher has to deal with anisotropic meshes. We have shown that such a discrete parametrization is possible only if the corresponding mapping orients all parametric triangles in the same way.

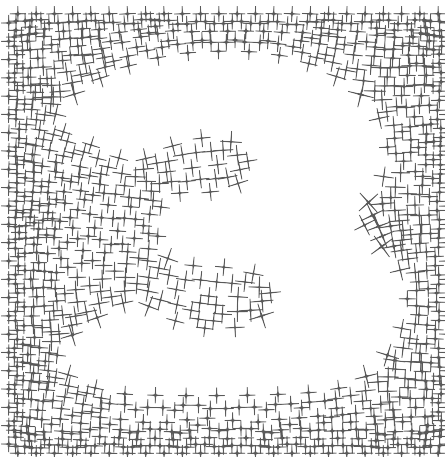
Based on the mean value theorem and assuming a linear approximation, we have derived the well-known mean value coordinates. We performed a convergence test of the corresponding scheme: it does not discretize properly a Laplacian on a structured mesh; otherwise, it has the expected convergence for a scheme that is not symmetric. We have proved that if homogeneous Neumann conditions are set along the boundary of holes (within a triangulation), the mean value coordinates give parametric holes whose boundary is convex. Since it unnecessarily deforms the parametrization, we gave an heuristic that fills the holes as they were circular in order to produce better parametrizations.

With one simple but graphic example, we shown the effect of feature edge detection on the atlas creation. We have discussed how to improve the parameterization of a coarse triangulation: the only way is to perform a longest edge bisection before parametrization. We have shown there is no Lagrange \mathcal{P}^2 version of the mean value coordinates. Finally, several difficult examples were exhibited as a demonstration of the robustness of Gmsh's pipeline.

Acknowledgements

The present study was carried out in the framework of the project “Large Scale Simulation of Waves in Complex Media”, which is funded by the Communauté Française de Belgique under contract ARC WAVES 15/19-03.

CHAPTER



Crossfield computation based on Ginzburg-Landau theory

This chapter is a reproduction of the following paper

Beaufort, P.-A., Georgiadis, C., Lambrechts, J., Henrotte, F., Geuzaine, C., Remacle, J.-F., revision, *Cross fields on arbitrary surfaces: A PDE approach based on Ginzburg-Landau theory*, Computer-Aided Design.

which is a promoted¹ version of

Beaufort, P.A., Lambrechts, J., Henrotte, F., Geuzaine, C. and Remacle, J.F., 2017. *Computing cross fields: A PDE approach based on the Ginzburg-Landau theory*. *Procedia engineering*, 203, pp.219-231.

¹It has been selected by the IMR26 committee.

Abstract

This paper proposes a method to compute crossfields based on the Ginzburg-Landau theory in Magnetism. According to this theory, the magnetic moment distribution in a ferromagnetic material can be regarded as a vector field with fixed norm, i.e., a directional field. The energy is the integral over the sample of the squared norm of the distribution gradient, and the sought distribution is a minimizer of this energy under the fixed norm constraint. The Ginzburg-Landau functional, which describes mathematically this situation, has two terms: the Dirichlet energy of the distribution and a term penalizing the mismatch between the fixed and actual norm of the distribution. Directional fields on surfaces are known to have a number of critical points, which are properly identified with the Ginzburg-Landau approach: the asymptotic behavior of Ginzburg-Landau problem provides well-distributed critical points over the 2-manifold, whose indices are as low as possible. The central idea in this paper is to exploit this theoretical background for crossfield computation on arbitrary surfaces. Such crossfields are instrumental in the generation of meshes with quadrangular elements. The relation between the topological properties of quadrangular meshes and crossfields are hence first recalled. It is then shown that a crossfield on a surface can be represented by a complex function of unit norm with a number of critical points, i.e., a nearly everywhere smooth function taking its values in the unit circle of the complex plane. As maximal smoothness of the crossfield is equivalent with minimal energy, the crossfield problem is equivalent to an optimization problem based on Ginzburg-Landau functional. A discretization scheme with Crouzeix-Raviart elements is applied and the correctness of the resulting finite element formulation is validated on the unit disk by comparison with an analytical solution. The method is also applied to the 2-sphere where, surprisingly but rightly, the computed critical points are not located at the vertices of a cube, but at those of an anticube.

Keywords: Ginzburg-Landau theory, Crossfields, Poincaré-Hopf, critical points (singularities), Crouzeix-Raviart, Asterisk Fields

1 | Introduction

The Finite element method (FEM) provides a powerful and versatile framework for numerical simulation, which however heavily relies on *mesh generation*, the decomposition of a geometrical region into simple shaped finite elements. In

two-dimensional geometries, two kinds of elements exist: triangles and *quadrangles*. Quadrangular meshes are deemed better than triangular ones because (i) there are half as many quadrangles than triangles for the same number of vertices; (ii) it is possible to define tensorial operations on quadrangles; and (iii) quadrangular meshes ease the tracking of preferred directions in mesh refinement.

However, the generation of quadrangular meshes remains a challenging task, for which many strategies have been explored. Some of them, based on surface parameterization, are suitable for the generation of structured quadrangular meshes, close to regular (square) grids. A crossfield may be used to determine the appropriate parameterization, either on a patch (Bommes et al., 2009) or globally (Kälberer et al., 2007). A crossfield can also be used for partitioning the surface into a set of curvilinear quadrangular regions (a polyquad), then trivially quadrangulable (Kowalski et al., 2013). The parameterization can also be deduced from a singularity graph (Cohen and Desbrun, 2006). In this paper, the primary concern is however to use crossfields as part of another meshing strategy: a frontal approach firstly proposed by (Lee and Lo, 1994) that consists in recombining triangles into quadrangles. This can be done efficiently (Remacle et al., 2012) but the quality of the quadrangles strongly depends on the node location. A heuristic to obtain well distributed nodes is to spawn them following consistent directions, such as those suggested by a smooth crossfield. Such a frontal approach allows building unstructured quadrangular meshes with varying element size. Other advantages of quadrilateral meshes exist for specific finite element models: for examples, triangular plate bending elements are stiffer than quadrilateral ones with the same number of vertices

Although there exist various ways to represent discrete crossfields (Vaxman et al., 2016, §5), their computation generally relies on some smoothing process, possibly under constraints. For an angle-based representation, a crossfield is pictured as four orthogonal or opposite vectors. From this representation, it is possible to formulate the quadrangulation as a mixed-integer problem (Bommes et al., 2009). More advanced mathematical notions such as holonomy (Lai et al., 2009) may be used as well to design crossfields. This approach requires to build a metric on the 2-manifold.

In this paper, the so-called Cartesian (complex) representation (Palacios and Zhang, 2007) is adopted. This representation naturally takes the symmetries of the cross into account, and the crossfield is identified with a complex-valued function. Complex analysis gives then a large and useful background, especially about the theoretical analysis of critical points. The second term of the Ginzburg-Landau functional is controlled by a parameter depending on the local mesh size. When this parameter is small enough, the minimization of the functional results in a smooth crossfield whose critical points are optimally located and whose critical points have indices with minimal absolute

values, according to the theory. The previous approach closest to ours is that in (Kowalski et al., 2013). It has only the energy term, but the vector field is constrained to have a norm close to the unity. Critical points are identified in this approach by computing an argument (angle) from the vector field, whereas we only need to compute the vector field norm, critical points being in our approach points where the crossfield norm locally vanishes.

Our main contribution is to express the crossfield problem with Ginzburg-Landau equations. Those equations rely on an interesting mathematical and physical backgrounds. In order to grasp the great understanding that Ginzburg-Landau functional provides to the crossfield problem, we first recall the topological constraints of full quadrangular (and triangular) mesh in section 2 and the link with cross (and asterisk, respectively) field in section 3. In section 4, we develop the intuition of using the Ginzburg-Landau functional for the crossfield problem and we give the related Ginzburg-Landau theory. We derive in section 5 a simple FEM scheme from the Ginzburg-Landau equations. Our numerical scheme is validated on the unit disk in regards with Ginzburg-Landau theory, section 6. On the 2-sphere section 7 we get a surprising but correct result. In section 8, the Ginzburg-Landau equations are modified to get better results on NACA profiles. Finally, we apply our simple finite scheme on the torus in section 9.

2 | Topology of Triangular and Quadrilateral Meshes

Assume an orientable surface \mathcal{S} embedded in \mathbb{R}^3 . Let g be the number of handles of the surface. The topological characteristic g , which is also called the genus of the surface, is the maximum number of cuttings along non-intersecting closed curves that won't make the surface disconnected. Let also b be the number of connected components of the boundary $\partial\mathcal{S}$ of the surface. The Euler characteristic of \mathcal{S} is then the integer

$$\chi = 2 - 2g - b.$$

One has $\chi = 2$ for a sphere, whereas $\chi = 1$ for a disk ($b = 1$), and $\chi = 0$ for a torus ($g = 1$) or a cylinder ($b = 2$).

Consider now a mesh on \mathcal{S} with n nodes (also called vertices), n_e edges and n_f facets. The Euler formula

$$\chi = n - n_e + n_f \tag{3.1}$$

provides a general relationship between the numbers of nodes, edges and facets in the mesh (details in Eppstein (2009)). If n_b nodes (and hence n_b edges) are on the boundary $\partial\mathcal{S}$, and if the number of edges (or nodes) per facet is noted

n_{evf} ($n_{evf} = 3$ for triangulations and $n_{evf} = 4$ for quadrangulations, meshes mixing triangles with quadrangles being excluded), the following identity holds: all facets have n_{evf} edges, $n_e - n_b$ edges have two adjacent facets and n_b edges have one adjacent facet. Hence the relationship

$$n_{evf}n_f = 2(n_e - n_b) + n_b. \quad (3.2)$$

Elimination of n_e between (3.2) and (3.1) yields

$$2n - n_b + (2 - n_{evf})n_f = 2\chi, \quad (3.3)$$

which is true for any triangulation or quadrangulation.

A regular mesh has only regular vertices. An internal vertex is regular if it has exactly 6 adjacent triangles or 4 adjacent quadrangles, whereas a boundary vertex is regular if it has exactly 3 adjacent triangles or 2 adjacent quadrangles. One has then

$$6(n - n_b) + 3n_b = 3n_f \quad \Rightarrow \quad n_f = 2n - n_b \quad (3.4)$$

and

$$4(n - n_b) + 2n_b = 4n_f \quad \Rightarrow \quad n_f = n - \frac{n_b}{2} \quad (3.5)$$

respectively for a regular triangulation and a regular quadrangulation, from a topological point of view. Substitution of (3.4) and (3.5) into (3.3) shows that only surfaces with a zero Euler characteristic can be paved with a regular mesh. If $\chi \neq 0$, irregular vertices will necessarily be present in the mesh.

The number and the index of the irregular vertices is tightly linked to the Euler characteristic χ , which is a topological invariant of the surface. We call valence of a vertex the number of facets adjacent to the vertex in the mesh. In a regular mesh, all vertices have the same valence v_{reg} . In a non regular mesh, on the other hand, a number of irregular vertices have a valence $v \neq v_{reg}$, and one notes the integer $k = v_{reg} - v$ the valence mismatch of a vertex.

Assume a quadrangulation with n_k irregular internal vertices of valence $v = 4 - k$, and n_{bk} irregular boundary vertices of valence $2 - k$, k given. All other vertices are regular. There are then $n - n_b - n_k$ regular internal vertices of valence 4, and $n_b - n_{bk}$ regular boundary vertices of valence 2, so that one can write

$$4n_f = 4(n - n_b - n_k) + 2(n_b - n_{bk}) + (4 - k)n_k + (2 - k)n_{bk}, \quad (3.6)$$

and the subtraction of (3.3) with $n_{evf} = 4$ yields

$$\chi = \frac{k}{4}(n_k + n_{bk}),$$

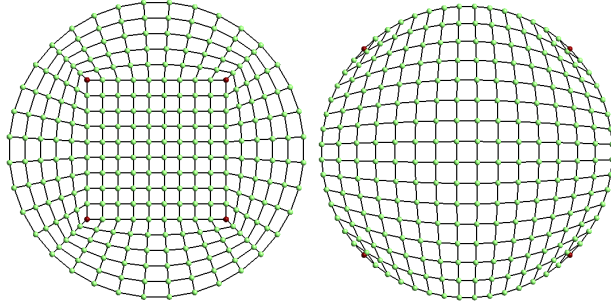


Figure 3.1: A quadrilateral mesh of a circle. Four irregular vertices of index $1/4$ (in red) are required to obtain such a mesh. The irregular vertices may be inside the disk (left) or on its boundary (right)

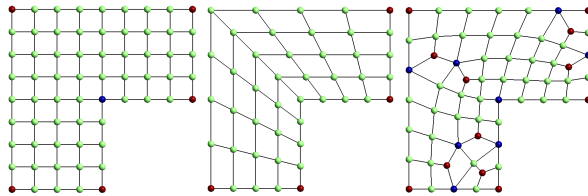


Figure 3.2: Different quadrangulations of a L-shaped domain. Irregular vertices of index $1/4$ are displayed in red, whereas ones of index $-1/4$ are displayed in blue. The sum of the indices of the irregular vertices is equal to $\chi = 1$ in all cases.

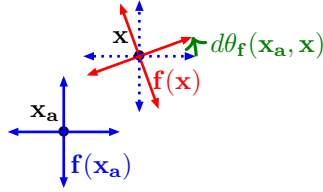
showing that, in a quadrangulation, each irregular vertex counts for $\text{index}(\mathbf{x}_i) = k/4$ in the Euler characteristic, a quantity called the indice of the irregular vertex \mathbf{x}_i .

Summing up now on different possible values for k , one can establish that a quadrangulation of a surface with Euler characteristic χ verifies

$$\chi = \sum_k \frac{k}{4} (n_k + n_{bk}) = \sum_{i=1}^N \text{index}(\mathbf{x}_i). \quad (3.7)$$

Consider, for instance, the quadrangulation of a disk, which is a surface with $\chi = 1$. A minimum of $n_1 = 4$ irregular vertices of index $1/4$ must be present. They can be located either on the boundary (vertices of valence 1) or inside the disk (vertices of valence 3), Fig. 3.1.

Fig. 3.2 shows three different quadrangulations of a L-shaped domain ($\chi = 1$). Regular boundary nodes should all have a valence of 2. The mesh on the

Figure 3.3: Differential function $d\theta_f$.

left has 6 irregular vertices located at the corners of the domain: five with index $1/4$, and one with index $-1/4$. The central mesh, on the other hand, has the minimum amount of irregular vertices, i.e. four ones of index $1/4$. The right mesh generated by recombination of a standard Delaunay triangular mesh (see [Remacle et al. \(2012\)](#)) has twelve vertices of index $1/4$, and eight vertices of index $-1/4$, both on the boundary and inside the domain. Quality meshes should have as few irregular vertices as possible. In what follows, a general approach allowing to compute the position of such irregular vertices before meshing the surface is presented.

3 | Why Crossfields?

Crossfields are auxiliary in the generation of quadrangular meshes. We shall show that nonregular vertices defined in the previous section are precisely the critical points of a crossfield, and that these critical points of the crossfield can also be related to the Euler characteristic of the meshed surface. This result represents an important theoretical limit on the regularity of quadrangular meshes.

3.1 Continuity

A crossfield f is a field defined on a surface \mathcal{S} with values in the quotient space S^1/Q , where S^1 is the circle group and Q is the group of quadrilateral symmetry. Pictorially, it associates to each point of the surface \mathcal{S} , which has to be meshed, a cross made of four unit vectors that are orthogonal with each others in the tangent plane $T\mathcal{S}$ of the surface.

A surface \mathcal{S} can be identified with its tangent space in any neighborhood $\sigma \subset \mathcal{S}$ that is sufficiently small to have curvature effects negligible. This local identification of the surface with a vector space endows it with a natural parallel transport rule, so that the angular differential $d\theta_f(\mathbf{x}_a, \mathbf{x}_b)$ can be defined as the minimal angle, with its sign, between the branches of $f(\mathbf{x}_a)$ and any of the

branches of $f(\mathbf{x}_b)$ for any pair of points $\mathbf{x}_a, \mathbf{x}_b \in \sigma$ where f is defined, Fig. 3.3. Taking now as reference the cross $f(\mathbf{x}_a)$, an angular coordinate

$$\theta_f(\mathbf{x}) = d\theta_f(\mathbf{x}_a, \mathbf{x}) \quad (3.8)$$

can be defined for crosses in σ . The crossfield f is deemed continuous (regular) at \mathbf{x}_b if the limit

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}_b} \theta_f(\mathbf{x}) = \theta_f(\mathbf{x}_b) \quad (3.9)$$

exists (i.e. is unique). It is then equal to $\theta_f(\mathbf{x}_b)$. Isolated points \mathbf{x}_i , $i = 1 \dots N$, of \mathcal{S} where the limit (3.9) does not exist are called critical points or zeros of the crossfield.

3.2 Index and degree

Although defined locally, the notion of continuity gives unexpectedly valuable information about the topology of \mathcal{S} , which is a nonlocal concept. To see this, consider a crossfield f defined on a quadrangular element delimited by four (possibly curvilinear) edges. Assume the crossfield is parallel to the four edges (i.e. one of the four branches of the cross is parallel to the tangent vector of the edge at each point of the edge, except the extremities) and prolongates smoothly inside the quadrangle. This field is discontinuous at corners where edges do not meet at right angle, but it is continuous everywhere else. Making the same construction for all elements of a quadrangular mesh, one obtains a crossfield f topologically identified with the quadrangular mesh, and that is continuous everywhere except at the vertices of the mesh. This field has thus got isolated critical points at mesh vertices, but not all critical points have the same significance. Some critical points have a specific topological value, associated with the notion of index.

To introduce the notion of index, an angular coordinate needs to be defined for points in a neighborhood σ_i of a critical point \mathbf{x}_i . Picking up an arbitrary regular point $\mathbf{x}_a \in \sigma_i$, $\mathbf{x}_a \neq \mathbf{x}_i$, the local unit vector basis

$$\mathbf{e}_1 = \frac{\mathbf{x}_a - \mathbf{x}_i}{|\mathbf{x}_a - \mathbf{x}_i|}, \quad \mathbf{e}_2 = \mathbf{n} \times \mathbf{e}_1,$$

with \mathbf{n} the normal to \mathcal{S} , is constructed, and hence a local polar coordinate system

$$r(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_i|, \quad \theta(\mathbf{x}) = \text{atan2}\left((\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{e}_2, (\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{e}_1\right) \quad (3.10)$$

can be defined for points in σ_i .

A circular curve \mathcal{C}_i of infinitesimal radius centered around the vertex \mathbf{x}_i is now considered. As the angles $\theta(\mathbf{x})$ (3.10) and $\theta_f(\mathbf{x})$ (3.8) are precisely the

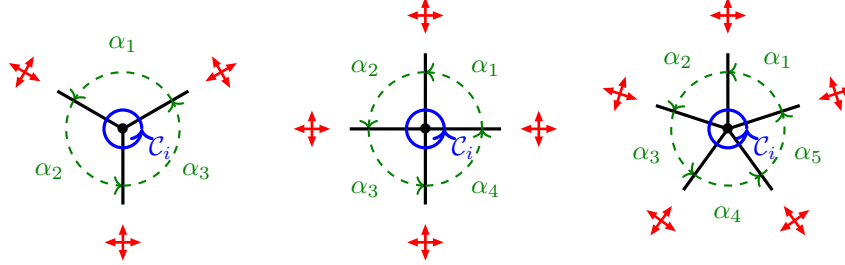


Figure 3.4: Illustration of vertices where the indices of the crossfield (in red) are respectively $1/4, 0$ and $-1/4$, from left to right. The index only depends on the number of quadrangles that are adjacent to the vertex, independently of the values of the angles α_p , which don't need to be identical as they are in the figure.

elements of the groups S^1 and S^1/Q , respectively, the crossfield on C_i can be regarded as a mapping

$$f : S^1 \mapsto S^1/Q. \quad (3.11)$$

The mapping is continuous, since C_i circles around the critical point \mathbf{x}_i , but it does not cross it. The index of f at \mathbf{x}_i is the degree of the mapping (3.11), i.e. the number of times the codomain wraps around the domain under the mapping. Its algebraic expression is easily expressed in terms of the angles θ and θ_f as

$$\text{index}(\mathbf{x}_i) = \frac{1}{2\pi} \oint_{C_i} d\theta_f$$

where 2π is $\oint_{C_i} d\theta$. In case of a vertex \mathbf{x}_i of valence v_i , i.e. a vertex adjacent to v_i quadrangular elements, the integral evaluates as

$$\text{index}(\mathbf{x}_i) = \frac{1}{2\pi} \sum_{p=1}^{v_i} \left(\alpha_p - \frac{\pi}{2} \right) = \frac{1}{2\pi} (2\pi - v_i \frac{\pi}{2}) = \frac{4 - v_i}{4}, \quad (3.12)$$

where the α_p 's are the angles of the v_i quadrangular elements adjacent to the considered vertex \mathbf{x}_i , and where the obvious relationship $\sum_{p=1}^{v_i} \alpha_p = 2\pi$ has been used. The crossfield f has index 0 at vertices adjacent to four quadrangular elements, whereas it has index $1/4$ ($-1/4$) at vertices adjacent to 3 (5, respectively) quadrangular elements meet, Fig. 3.4. As one sees, the index is a topological characteristic of the crossfield f at the critical point \mathbf{x}_i . It does not depend on the choice of the curve C_i , nor on the choice of an angular reference for the angles $\theta(\mathbf{x})$ and $\theta_f(\mathbf{x})$.

3.3 Poincaré-Hopf theorem

Equation (3.12) relates the index of the crossfield at a critical point \mathbf{x}_i with one fourth of valence $k_i = 4 - v_i$ of the corresponding mesh vertex. This result can be combined with the algebraic topology result of previous section (3.7) that each internal irregular vertex of valence k_i counts for $k_i/4$ in the Euler characteristic of the underlying surface. This yields the relationship

$$\sum_{i=1}^N \text{index}(\mathbf{x}_i) = \chi \quad (3.13)$$

for the critical points of a crossfield f defined on a surface \mathcal{S} .

This is a generalization Poincaré-Hopf theorem, which states that the sum of the indices of the critical points of a *vector* field \mathbf{v} defined on a surface \mathcal{S} without boundary is equal to the Euler characteristic of the surface. This famous theorem draws an unexpected and profound link between two apparently distinct areas of mathematics, topology and analysis. Whereas vector fields have integer indices at critical points, crossfields have indices that are multiples of $1/4$. Still the topological relationship (3.13) of Poincaré-Hopf holds in both cases. Actually, our developments reach same inferences as Ray et al. (2008).

4 | Crossfield Computation: the Planar Case

We introduce the representation of a crossfield by means of a vector field. From this representation, we derive the problem to solve that corresponds to minimize Ginzburg-Landau functional. Its asymptotic behavior provides suitable critical points, if any.

4.1 Vector representation of crossfields

Only scalar quantities can be compared at different points of a manifold. For the comparison or, more generally, for differential calculus with nonscalar quantities like crossfields, a parallel transport rule needs to be defined on the manifold. On a surface (two-manifold), this rule can take the form of a regular vector field which gives at each point the direction of the reference angle 0. Poincaré-Hopf theorem says that such a field does not exist in general, and in particular on manifolds whose Euler characteristic is not zero. The situation is however easier in the planar case. A global Cartesian coordinate frame can always be defined over the plane, and be used to evaluate the orientation of the crossfield. We shall therefore expose the crossfield computation method in

the planar case, and then generalize to nonplanar surfaces, where we will have to deal with local reference frames, in a subsequent section.

A cross $f(\mathbf{x})$ is an element of the group S^1/Q , which can be represented by the angle $\theta_f(\mathbf{x})$ it forms with the local reference frame. Yet, due to the quadrilateral symmetry, four different angles in $[0, 2\pi[$ represent the same crossfield $f(\mathbf{x})$. Let for instance the angles $\theta_1 = 0$ and $\theta_2 = \pi/2$ represent the same cross. The average $(\theta_1 + \theta_2)/2 = \pi/4$ represents another cross, whereas the difference $\theta_2 - \theta_1 = \pi/2$ is not zero. So, we have $\frac{1}{2}(x + x) \neq x$ and $x - x \neq 0$, which clearly indicates that the values of the crossfield f do not live in a linear (affine) space. This makes the representation by θ_f improper for finite element interpolation. The solution is two-fold. First, the angle θ_f is multiplied by four, so that the group S^1/Q is mapped on the unit circle S^1 , and the cross f is therefore represented by a unit norm vector \mathbf{f} . Then, the vector is represented in components in the reference frame as

$$\mathbf{f} = (\cos 4\theta_f, \sin 4\theta_f) \equiv (f_1, f_2).$$

This vector may be represented by a complex-valued function

$$f = f_1 + i f_2$$

This representation corresponds to a vector field that is described by a complex exponential whose argument is 4θ . A crossfield is thus depicted by the fourth roots of a (unit) complex number. This observation may be generalized for directional fields with n symmetries (Vaxman et al., 2016, §5.2).

4.2 Laplacian smoothing

Computing the crossfield f consists thus now of computing the vector field representation \mathbf{f} , which obviously lives in a linear space (a 2D plane). The components of \mathbf{f} are fixed on the boundaries of $\Gamma = \partial\mathcal{S}$ so that the crosses are parallel with the exterior normal vector $\mathbf{n} = (\cos \theta_n, \sin \theta_n)$ i.e.

$$\mathbf{f} = (\cos 4\theta_n, \sin 4\theta_n) \quad \text{on } \Gamma.$$

Propagating \mathbf{f} inside \mathcal{S} is here done by solving a Laplacian problem. Even though the vector representation \mathbf{f} is unitary on Γ , it tends to drift away from S^1 inside the domain. The computed finite element solution \mathbf{f} lies therefore outside the unit circle and must be projected back on S^1 to recover the angle

$$\theta_f = \frac{\text{atan2}(f_2, f_1)}{4}.$$

Due to the multiplication by 4, the indices of the critical points of the vector field \mathbf{f} verify

$$\sum_{i=1}^N \text{index}(\mathbf{x}_i) = 4\chi. \quad (3.14)$$

4.3 The Ginzburg-Landau model

Numerical experiments show that the norm of the vector field \mathbf{f} computed by Laplacian smoothing (see previous section) decreases quite rapidly as one moves away from the boundary ∂S , leaving in practice large zones in the bulk of the computational domain where the solution is small, and the computed crossfield inaccurate, Fig. 3.5a. A more satisfactory formulation consists of ensuring that the norm of \mathbf{f} remains unitary over the whole computational domain, Fig. 3.5b. This problem can be formulated in variational form in terms of the Ginzburg-Landau functional

$$E(f_1, f_2) = \underbrace{\frac{1}{2} \int_S (|\nabla f_1|^2 + |\nabla f_2|^2) dS}_{\text{smoothing}} + \underbrace{\frac{1}{4\epsilon^2} \int_S (f_1^2 + f_2^2 - 1)^2 dS}_{\text{penalty}}. \quad (3.15)$$

The first term minimizes the gradient of the crossfield and is therefore responsible for the laplacian smoothing introduced in the previous section. The second term is a penalty term that vanishes when $\mathbf{f} \in S^1$. The penalty parameter ϵ , called *coherence length*, has the dimension of a length. The Euler-Lagrange equations of the functional (3.15) are the quasi-linear PDE's

$$\nabla^2 f_i - \frac{1}{\epsilon^2} (f_1^2 + f_2^2 - 1) f_i = 0 \quad i = 1, 2. \quad (3.16)$$

called Ginzburg-Landau equations. If ϵ is small (enough) with respect to the dimension of S , then \mathbf{f} is of norm 1 everywhere but in the vicinity of the isolated critical points \mathbf{x}_i^c .

The asymptotic behavior of Ginzburg-Landau energy can be written as

$$E = \pi \left(\sum_{i=1}^N \text{index}(\mathbf{x}_i^c)^2 \right) \log(1/\epsilon) + W + \mathcal{O}(1/|\log \epsilon|). \quad (3.17)$$

with

$$W = -\pi \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \text{index}(\mathbf{x}_i^c) \text{index}(\mathbf{x}_j^c) \log |\mathbf{x}_i^c - \mathbf{x}_j^c| + \mathcal{R} \quad (3.18)$$

as $\epsilon \rightarrow 0$ (see Bethuel et al. (1994), Introduction, Formulae 11 and 12).

In asymptotic regime, the energy is thus composed of three terms. The first term of (3.17) blows up as $\epsilon \rightarrow 0$, i.e. energy becomes unbounded if critical points are present. When ϵ is small, this first term dominates, and one is essentially minimizing $\sum_{i=1}^N \text{index}(\mathbf{x}_i^c)^2$ with the constraint (3.14). This indicates that a critical point of index 2 has a cost of 4 in terms of energy, whereas 2 critical points of index 1 have a cost of 2. All critical points should

therefore be of index ± 1 , and their number should be $N = 4 |\chi|$. This is indeed good news for our purpose: good crossfields should have few critical points of lower indices.

The second term of (3.17) is the *renormalized energy* W (3.18). It remains bounded when ϵ tends to 0. The double sum in W reveals the existence of a logarithmic force between critical points. The force is attractive between critical points with indices of opposite signs, and repulsive between critical points with indices of the same signs. The second term in (3.18) is more complicated and is detailed in Bethuel et al. (1994). Basically, \mathcal{R} represents a repulsing force that forbids critical points to approach the boundaries.

Finally, the third term in (3.17) vanishes as $\epsilon \rightarrow 0$. At the limit, all energy is thus carried by the critical points of the field. All this together allows to believe that Ginzburg-Landau model is a good choice for computing crossfields. It produces few critical points, which are moreover well-distributed over the domain.

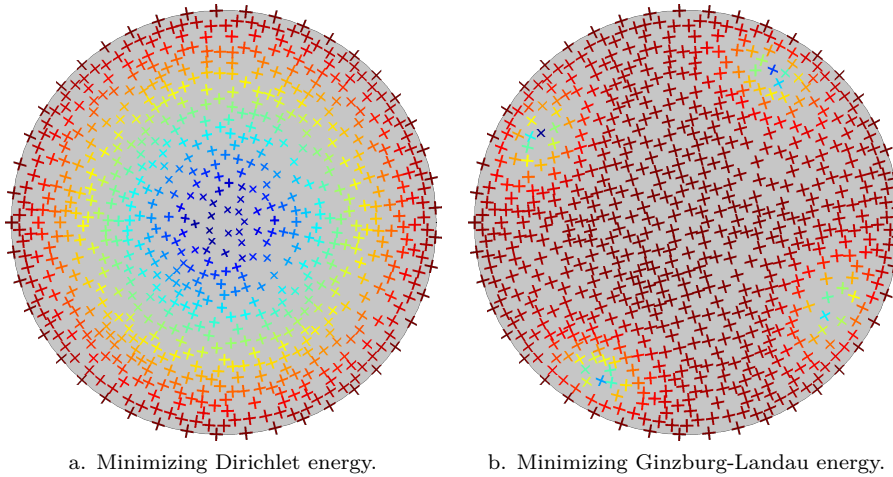


Figure 3.5: Crossfield over a disk. The color describes the field norm: blue is close to zero, red close to unity.

5 | Computation of Crossfields: Nonplanar Generalization

The finite element computation method for crossfields is now generalized to the case of nonplanar surfaces. Consider the conformal triangulation $\mathcal{S} = \cup_{ijk} \Omega_{ijk}$

of a nonplanar surface manifold \mathcal{S} , each triangle Ω_{ijk} being defined by the vertices \mathbf{p}_i , \mathbf{p}_j and \mathbf{p}_k . Since no global reference frame exists on a nonplanar surface, a local reference frame is associated to each edge of the triangulation. Let e^p be the p^{th} edge of the mesh, joining nodes \mathbf{p}_i and \mathbf{p}_j , and \mathbf{n}^p be the average of the normals vectors of the two triangles adjacent to e^p . The vectors

$$\{\mathbf{e}^p = \mathbf{p}_j - \mathbf{p}_i, \mathbf{t}^p = \mathbf{n}^p \times \mathbf{e}^p\}$$

form a local frame $\{\hat{e}^p, \hat{t}^p\}$ which enables the representation of the connector values of the discretized crossfield \mathbf{f} ,

$$f_1^p = \cos 4\theta_f^p \quad , \quad f_2^p = \sin 4\theta_f^p,$$

which are attached to the center of the edges of the triangulation. Actually, θ_f^p is assumed to be the same along e^p within both planes of triangles sharing e^p . This assumption eases computations and gives a planar-like representation, Fig. 3.6a.

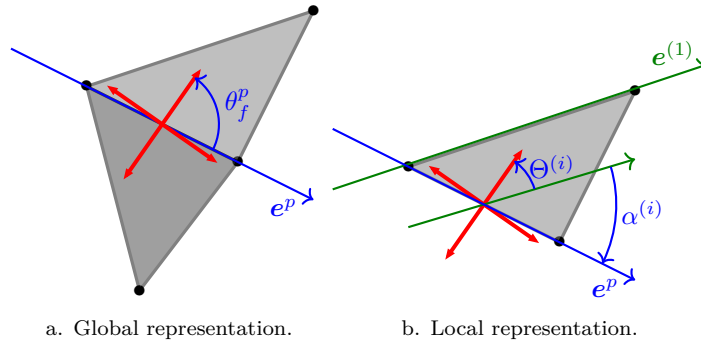


Figure 3.6: Crossfield over the p^{th} edge of a mesh.

As the connector values are attached to the edges of the mesh, and not to the nodes, Crouzeix-Raviart interpolation functions are used instead of conventional Lagrange shape functions, [Crouzeix and Raviart \(1973\)](#). The Crouzeix-Raviart shape functions ω^p equal 1 on corresponding edge e^p , and -1 on the opposite vertices (Fig. 3.7) in the two adjacent triangular elements. They are polynomial and their analytic expression in the reference triangle $\{\xi \in [0, 1], \eta \in [0, 1 - \xi]\}$ reads

$$\omega^{(1)}(\xi, \eta) = 1 - 2\eta \quad , \quad \omega^{(2)}(\xi, \eta) = 2(\xi + \eta) - 1 \quad , \quad \omega^{(3)}(\xi, \eta) = 1 - 2\xi,$$

where indices (1), (2) and (3) enclosed in parentheses denote the local edge numbering in the considered triangular element.

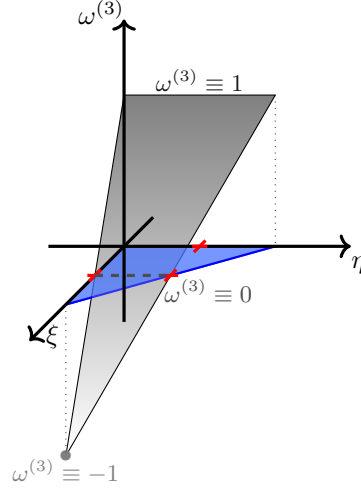


Figure 3.7: Third Crouzeix-Raviart function shape (shaded in grey) over reference triangle (in blue).

Each of the three edges of a triangle Ω_{ijk} has its own local reference frame. If one is to interpolate expressions involving the vector field \mathbf{f} over this element, the three edge-based reference frames have to be appropriately related with each other (see [Ray et al. \(2016\)](#)). We arbitrarily take the reference frame of the first edge of the element as reference, and express the angular coordinate of the two other edges in function of this one with the relationships (Fig. 3.6b)

$$\Theta^{(1)} = \theta_f^{(1)} \quad , \quad \Theta^{(2)} = \theta_f^{(2)} + \alpha^{(2)} \quad , \quad \Theta^{(3)} = \theta_f^{(3)} + \alpha^{(3)} .$$

Thus, the 6 local unknowns of triangle Ω_{ijk} can be expressed as a function of the 6 edge unknowns by

$$\begin{pmatrix} \cos 4\Theta^{(1)} \\ \cos 4\Theta^{(2)} \\ \cos 4\Theta^{(3)} \\ \sin 4\Theta^{(1)} \\ \sin 4\Theta^{(2)} \\ \sin 4\Theta^{(3)} \end{pmatrix} \equiv \underbrace{\begin{pmatrix} F_1^{(1)} \\ F_1^{(2)} \\ F_1^{(3)} \\ F_2^{(1)} \\ F_2^{(2)} \\ F_2^{(3)} \end{pmatrix}}_{\mathbf{F}^{ijk}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos 4\alpha^{(2)} & 0 & 0 & \sin 4\alpha^{(2)} & 0 \\ 0 & 0 & \cos 4\alpha^{(3)} & 0 & 0 & \sin 4\alpha^{(3)} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -\sin 4\alpha^{(2)} & 0 & 0 & \cos 4\alpha^{(2)} & 0 \\ 0 & 0 & -\sin 4\alpha^{(3)} & 0 & 0 & \cos 4\alpha^{(3)} \end{bmatrix}}_{\mathbf{R}^{ijk}} \underbrace{\begin{pmatrix} f_1^{(1)} \\ f_1^{(2)} \\ f_1^{(3)} \\ f_2^{(1)} \\ f_2^{(2)} \\ f_2^{(3)} \end{pmatrix}}_{\mathbf{f}^{ijk}}$$

and we have the interpolation

$$F_1^{ijk}(\xi, \eta) = \sum_{i=1}^3 \omega^{(i)}(\xi, \eta) F_1^{(i)} \quad , \quad F_2^{ijk}(\xi, \eta) = \sum_{i=1}^3 \omega^{(i)}(\xi, \eta) F_2^{(i)}$$

for the vector field \mathbf{f} in the triangle Ω_{ijk} .

A Newton scheme is proposed to converge to the solution. The Newton iteration at stage n for solving (3.16) consists of solving:

$$\nabla^2 \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}_n - \frac{1}{\epsilon^2} \begin{pmatrix} 3f_1^2 + f_2^2 - 1 & 2f_1 f_2 \\ 2f_1 f_2 & f_1^2 + 3f_2^2 - 1 \end{pmatrix}_{n-1} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}_n = -\frac{2}{\epsilon^2} \begin{pmatrix} f_1^3 + f_1 f_2^2 \\ f_1^2 f_2 + f_2^3 \end{pmatrix}_{n-1} \quad (3.19)$$

The 6×6 elementary matrix \mathbf{K}^{ijk} and the 6×1 elementary vector \mathbf{B}^{ijk} of element Ω_{ijk} are then given by \mathbf{K}^{ijk}

$$\left[\begin{array}{cc} \left(\int_{\Omega_{ijk}} \nabla \omega_m \cdot \nabla \omega_n + \frac{1}{\epsilon^2} (3F_1^2 + F_2^2 - 1) \omega_m \omega_n \, d\Omega \right) & \left(\int_{\Omega_{ijk}} \frac{\partial}{\partial x} F_1 F_2 \omega_m \omega_n \, d\Omega \right) \\ \left(\int_{\Omega_{ijk}} \frac{\partial}{\partial x} F_1 F_2 \omega_m \omega_n \, d\Omega \right) & \left(\int_{\Omega_{ijk}} \nabla \omega_m \cdot \nabla \omega_n + \frac{1}{\epsilon^2} (F_1^2 + 3F_2^2 - 1) \omega_m \omega_n \, d\Omega \right) \end{array} \right] \quad (3.20)$$

and

$$\mathbf{B}^{ijk} = \left(\begin{array}{c} \left(\int_{\Omega_{ijk}} \nabla F_1 \cdot \nabla \omega_n + \frac{1}{\epsilon^2} (F_1^3 + F_2^2 - F_1) \omega_n \, d\Omega \right) \\ \left(\int_{\Omega_{ijk}} \nabla F_2 \cdot \nabla \omega_n + \frac{1}{\epsilon^2} (F_1^2 + F_2^3 - F_2) \omega_n \, d\Omega \right) \end{array} \right). \quad (3.21)$$

with $m, n = 1 \dots 3$.

It is then necessary to transform those elementary matrix and vector in the reference frames of the edges as

$$\mathbf{k}^{ijk} = (\mathbf{R}^{ijk})^T \mathbf{K}^{ijk} \mathbf{R}^{ijk} \quad \text{and} \quad \mathbf{b}^{ijk} = (\mathbf{R}^{ijk})^T \mathbf{B}^{ijk}.$$

Then, standard finite element assembly can be performed. Boundary conditions are simply

$$f_1^p = 1 \quad , \quad f_2^p = 0$$

on every edge e^p of $\partial\mathcal{S}$. This nice simplification is due to the fact that unknowns are defined on the reference frame of the edges.

6 | Numerical Validation: the Unit Disk

We compute the analytical location of critical points of a directional field defined on the unit disk. The calculations are based on the Ginzburg-Landau results, described in section 4.3. The numerical location obtained by our FEM is compared to the analytical one.

Let \mathcal{S} be the open unit disk in \mathbb{R}^2 , i.e.

$$\mathcal{S} := \{(x_1; x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 < 1\}$$

For a star-shaped planar domain such as \mathcal{S} with a smooth boundary $\partial\mathcal{S}$ of exterior normal ν and tangent τ , whose vector field has d critical points of index $+1$ at $X^c = \{\mathbf{x}_1^c, \dots, \mathbf{x}_d^c\} \in \mathcal{S}$, the asymptotic energy E_ϵ (in complex form) becomes

$$E_\epsilon \xrightarrow{\epsilon \rightarrow 0} \pi d |\log(\epsilon)| + W(X^c) + \mathcal{O}(\epsilon) \quad (3.22)$$

where $W(X^c)$ is the renormalized energy

$$W(X^c) = -\pi \sum_{i \neq j} \log |\mathbf{x}_i^c - \mathbf{x}_j^c| + \frac{1}{2} \int_{\partial\mathcal{S}} \Phi f \times \nabla f \cdot \tau ds - \pi \sum_i R(\mathbf{x}_i^c) \quad (3.23)$$

where Φ is given by the following Neumann problem

$$\left. \begin{aligned} \nabla^2 \Phi(\mathbf{x}) &= 2\pi \sum_{i=1}^d \delta(\mathbf{x} - \mathbf{x}_i^c) & \text{in } \mathcal{S} \\ \nabla \Phi \cdot \nu &= f \times \nabla f \cdot \tau & \text{on } \partial\mathcal{S} \end{aligned} \right\} \quad (3.24)$$

and R is the regular part of Φ :

$$R(\mathbf{x}) = \Phi(\mathbf{x}) - \sum_{i=1}^d \log |\mathbf{x} - \mathbf{x}_i^c| \quad (3.25)$$

E_ϵ is minimum when the critical points are located appropriately, i.e. when (3.23) is minimum. The renormalized energy W corresponds to the Ginzburg-Landau energy (3.22) when the singular core energy $\pi d |\log(\epsilon)|$ has been removed. Since W depends only on the location of the critical points, it is possible to compute their location in the case of the unit disk, in order to get an optimal directional field.

The minimum of W is obtained by sampling points within the unit disk. It is assumed that the d critical points exhibit the d symmetries of their group (the quadrilateral group in the case $d = 4$). In other words, it means that they are at the same distance r^c from the center of the disk (i.e. the origin $(0; 0)$), and separated two-by-two with an angle of $2\pi/d$ radians.

The Neumann problem (3.24) is solved by decomposing $\Phi = \Phi^0 + \Phi^1$. The first term Φ^0 is the Green function of a two-dimensional Laplacian operator, while the second one Φ^1 is obtained by separation of variables (r, θ) . The solution is then

$$\Phi(r, \theta) = \sum_{i=1}^d \left[\underbrace{\frac{1}{2} \log(r^2 - 2r r^c \cos(\theta) + r^{c2})}_{\Phi_i^0} + \underbrace{\sum_{n=1}^{\infty} A_{i,n} r^n \cos(n \theta)}_{\Phi_i^1} \right] \quad (3.26)$$

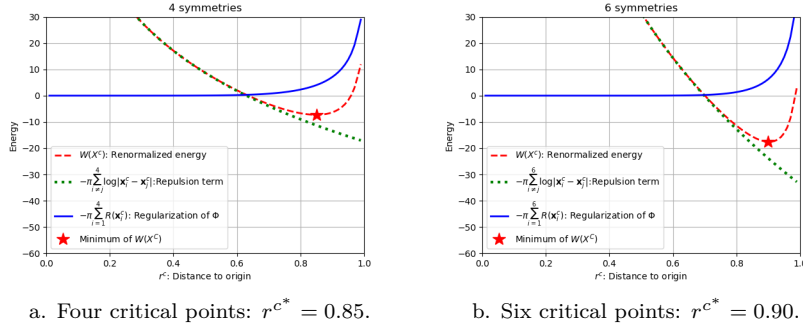


Figure 3.8: Python evaluations of renormalized energy W for different X^c on a unit disk S .

where $A_{i,n}$ depends on the location of the i -th critical point, which is parameterized by r^c . It is possible to show that the second term of (3.23) is zero, Appendix B.B. The analytical solution of Neumann problem is derived into the Appendix B.A.

The evaluation of W consists of computing the first and last terms, by sampling the disk. The sampling is done by selecting d critical points spaced by $2\pi/d$ radians. The distance r^c is sampled between zero and one. The distance r^{c^*} which gives the lowest value of W defines the location of the critical points. A Python script performs the evaluations and returns the optimal distance r^{c^*} , Fig. 3.8.

The corresponding directional fields are computed, and their critical point locations are compared with circles which radii correspond to r^{c^*} , Fig. 3.9. The location of critical points are really close to the estimation based on the analytical solution of $W(X^c)$ in the case of the unit circle. They tend to draw the corners of the polygon of symmetry: a square in the case of the crossfield, Fig. 3.9a and a regular hexagon for the asterisk field, Fig. 3.9b. The critical points are quite close to the unit circle. The more critical points, the closer to the unit circle they are. We understand that the repulsion term is stronger than the regularization term within the domain. The regularization term is only able to forbid critical points to be on the boundary, i.e. the unit circle.

7 | A Surprising Result: the Sphere

Let us compute the crossfield on a unit sphere. The sphere has no boundary so we choose randomly one edge of the mesh and fix the crossfield for this

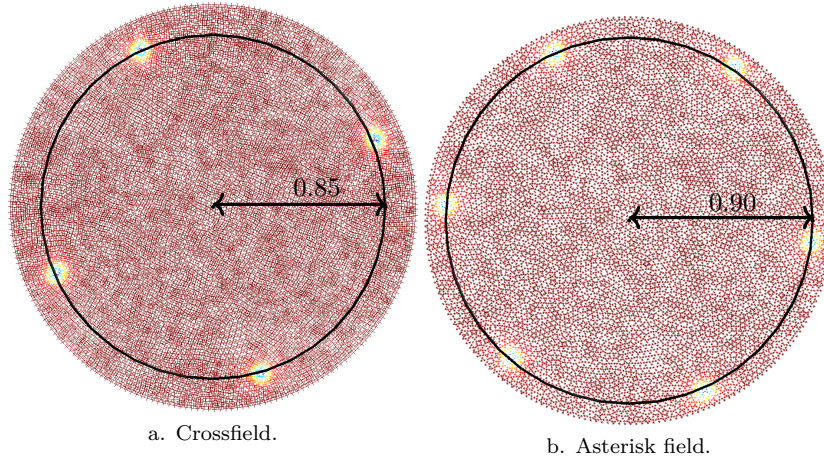


Figure 3.9: FEM computations of direction fields on a unit disk \mathcal{S} : the critical points are in blue areas.

specific edge. The mesh of the sphere is made of 2960 triangles (see Fig. 3.10). A value of $\epsilon = 0.1$ was chosen for the computation. A total of 29 Newton iterations were necessary to converge, by reducing the residual norm to 10^{-12} . The location of the 8 critical points is indeed not what we expected: our initial intuition was that critical points would be located at the corners of an inscribed cube of side $1/\sqrt{3}$. In all our computations i.e. while changing the mesh and ϵ , critical points are located on two squares of side $1/\sqrt{3}$, those two squares being tilted by 45 degrees around their common axis (see Fig. 3.10). Equilateral triangle patterns are formed between critical points that belong to both squares. In reality, our solution is the right solution. In the asymptotic regime, the location \mathbf{x}_i^c of the 8 critical points tends to minimize $-\sum_i \sum_{i \neq j} \log |\mathbf{x}_i^c - \mathbf{x}_j^c|$ (see Equations (3.17) and (3.18)). We have thus computed $-\sum_i \sum_{i \neq j} \log |\mathbf{x}_i^c - \mathbf{x}_j^c|$ for tilting angles ranging from 0 to $\pi/2$. Fig. 3.11 shows clearly that the minimum of the energy corresponds to an angle of $\pi/4$, which is exactly what is found by the finite element formulation. Fig. 3.12 shows the crossfield as well as the separatrices. The separatrices were computed “by hand”. The solution that has been found is related to what is called the Whyte’s problem (cf. Saff and Kuijlaars (1997); Dragnev et al. (2002)) that consists of finding N points on the sphere which positions maximize the product of their distances. The critical points are called *logarithmic extreme points* or *elliptic Fekete points* (see Fekete (1923)).

The specific configuration that corresponds to $N = 8$ is called an anticube

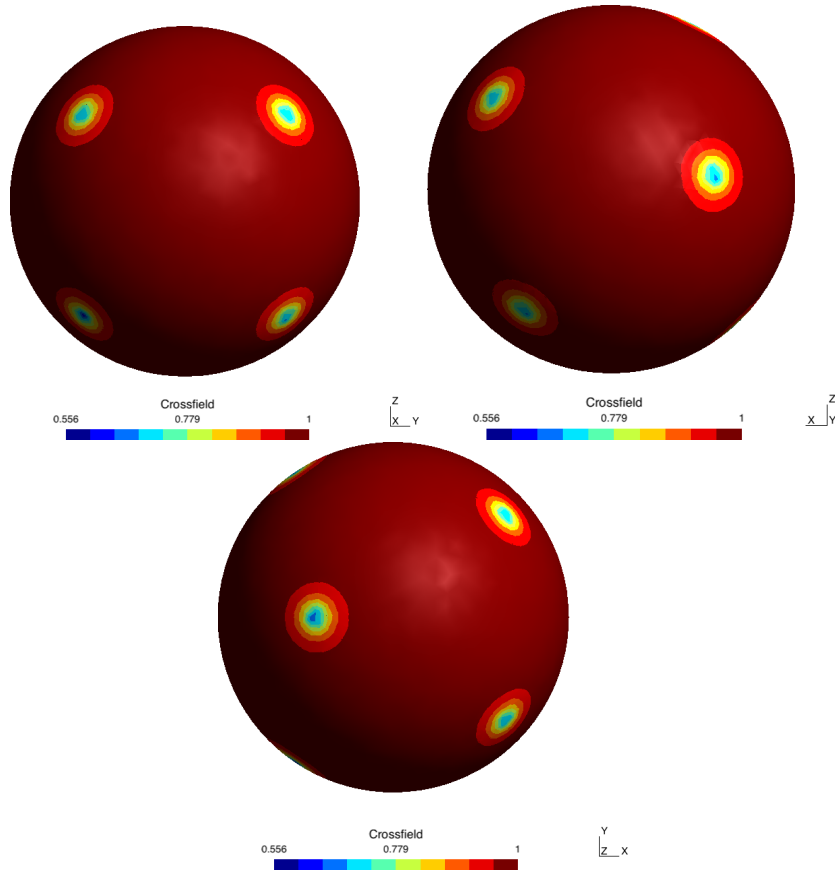


Figure 3.10: Mesh of the sphere. Colors correspond to the 2-norm $\|\mathbf{f}\|$ of the crossfield. The 8 critical points are located on two squares of side $1/\sqrt{3}$, which corresponds to the size of the inscribed cube. The two squares are tilted by 45 degrees.

(or square antiprism) and is exactly the one that was found numerically.

In the case of an asterisk field, the critical points are the summits of an icosahedron, which is the solution of Whyte's problem for $N = 12$. This superb result shows that it is indeed possible to use crossfields not only for building quadrangles but also to build equilateral triangles.

Actually, it is possible to show that the critical points computed over the sphere by Ginzburg-Landau correspond to the solution of Whyte's problem for any even value of N (see [Jezdimirovic et al. \(2017\)](#)).

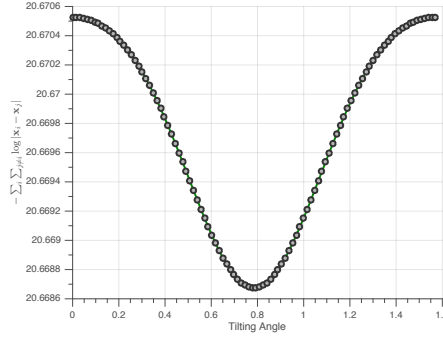


Figure 3.11: Energy vs. tilting angle for the sphere. The minimum corresponds to a tilting angle of $\pi/4$.

8 | Weak Boundary Conditions

In this section, we have computed the graph of singularities of a standard CFD test case: a three component wing domain with $\chi = -2$. This example is very similar to the one presented by (Kowalski et al., 2013). The solution has been computed on a non uniform triangular mesh of about 15,000 triangles. The graph of singularities has been depicted on Fig. 3.14. Weak boundary conditions have been applied to the different components of the wing where a penalization replaces the strong imposition of f on boundaries. A new term is thus added to Energy (3.15) for taking into account boundary conditions:

$$\frac{1}{2} \int_{\mathcal{S}} (|\nabla f_1|^2 + |\nabla f_2|^2) d\mathcal{S} + \frac{1}{4\epsilon^2} \int_{\mathcal{S}} (f_1^2 + f_2^2 - 1)^2 d\mathcal{S} + \frac{L}{2\epsilon^2} \int_{\partial\mathcal{S}} [(f_1 - \bar{f}_1)^2 + (f_2 - \bar{f}_2)^2] d\partial\mathcal{S} \quad (3.27)$$

where \bar{f}_1 and \bar{f}_2 are values of the crosses that are weakly imposed on the boundary and L the characteristic size of the problem. This new treatment allows singularities to migrate on the boundary, making their repulsive action finite. Figure 3.14 clearly shows that effect: a singularity of index $1/4$ sits on the leading edge of the slat, allowing a clean decomposition of the domain. The same migration is also observed on the leading edge of the profile. A strong imposition of boundary conditions naturally leads to singularities that are very close to regions of the boundary with high curvature, usually at a distance from the boundary that is one mesh size. Artificial boundary layers are thus added to the decomposition (see (Kowalski et al., 2013, Fig. 12 and 14)).

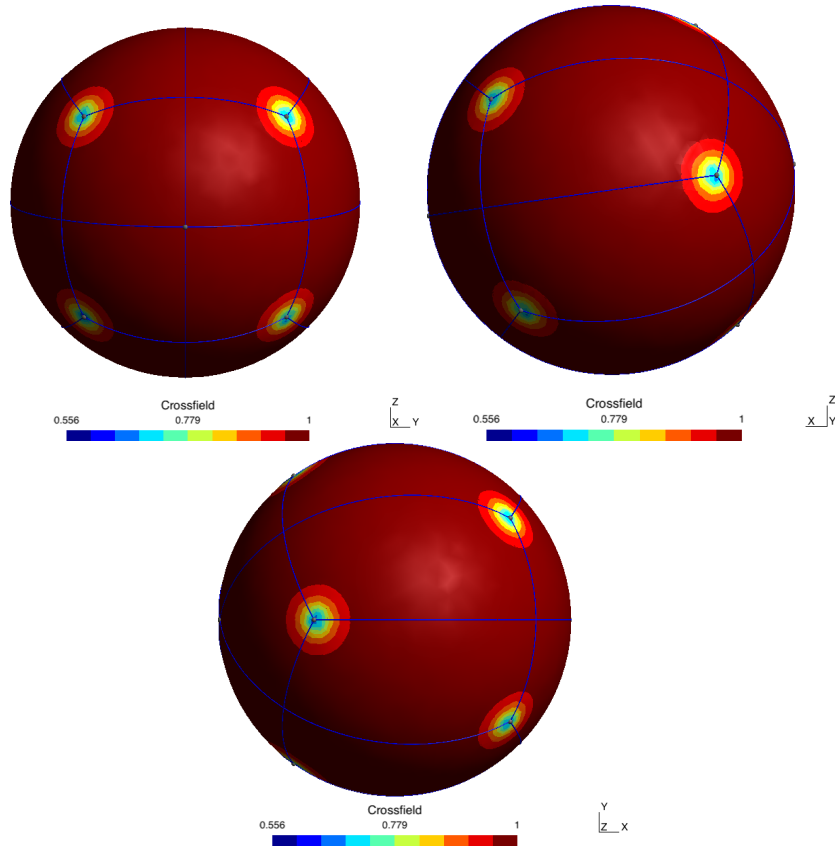


Figure 3.12: Separatrices from crossfield.

9 | Application of our FEM Scheme to the Torus

The Euler-Poincaré characteristic of the torus is $\chi = 0$. Theoretically, we should obtain a crossfield without critical points. But our FEM scheme gives crossfield with twelve critical points, located where the Gaussian curvature is maximal (exterior) or minimal (interior), Fig. 3.15. Fig. 3.15a shows that the six critical points located on the maximal Gaussian curvature line are facing the six corresponding critical points located on the minimal Gaussian curvature line. Moreover, as the former have an index $+1/4$, and the latter an index $-1/4$, Fig. 3.15b, the index sum of the surface is zero, as predicted by the Poincaré-Hopf theorem.

Our FEM scheme does not reach however the asymptotic behavior ($\epsilon \rightarrow 0$)

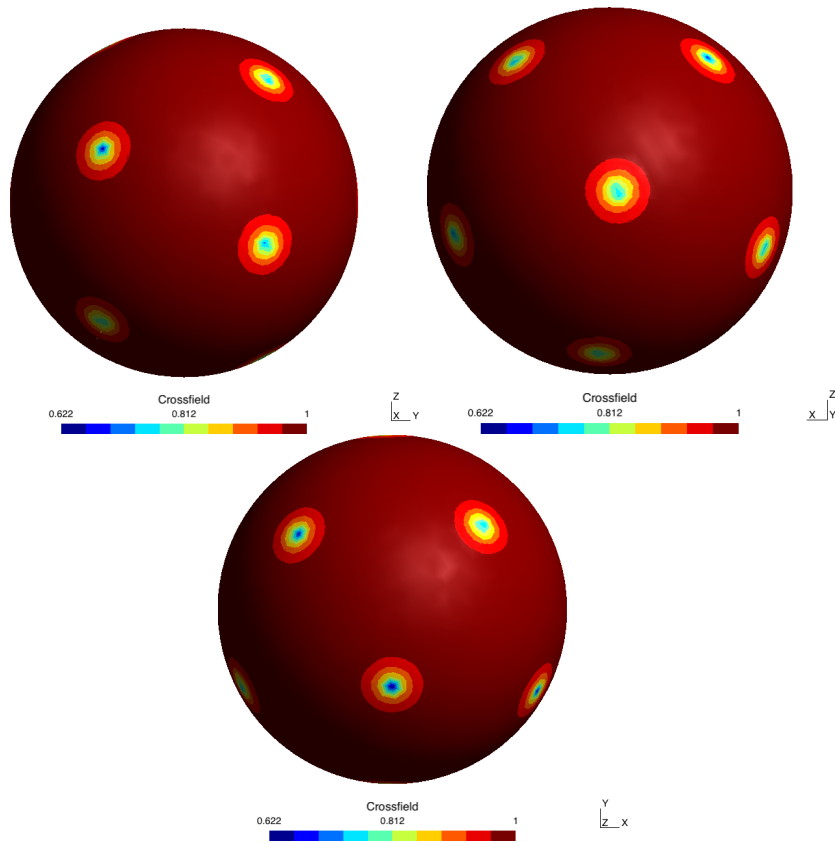


Figure 3.13: Asterisk field (6 symmetries) which the critical points correspond to the corners of an icosahedron.

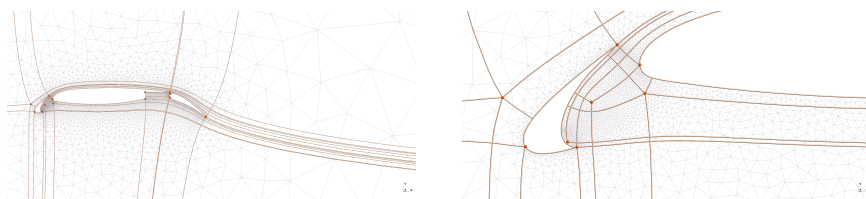


Figure 3.14: Graph of singularities for the three component wing. Right figure is a zoom on the leading edge slat.

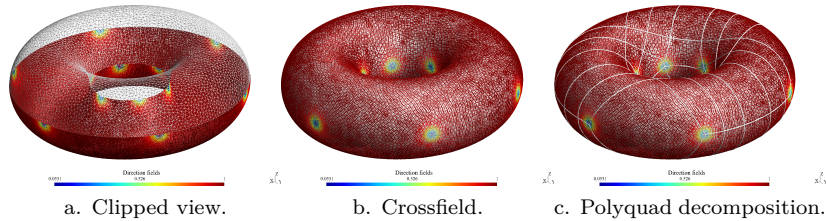


Figure 3.15: FEM ($\epsilon = 0.06$) crossfield on a torus discretized by 20612 triangles. Twelve critical points of opposite indices ($\pm 1/4$).

of the Ginzburg-Landau functional. It means that our penalty factor ϵ is not low enough. Otherwise, the computed crossfield should not have any critical points owing to (3.17). Actually, the computed crossfield has a lower energy (72.10) than the crossfield with no critical point that could be drawn by aligning crosses with the main curvatures of the surface (84.58). The tentative polyquad decomposition shown in Fig. 3.15c indicates that the field computed with the Ginzburg-Landau approach tends to be more uniform, in order to reduce the Dirichlet energy. It confirms that the Dirichlet term is stronger than the penalty term.

10 | Conclusion

This article has demonstrated the consistency of the Ginzburg-Landau theory to compute directional fields on arbitrary surfaces. The proposed approach relies on a physical and mathematical backgrounds. This provides proofs, analytical solutions and helps delineating fundamental mathematical properties that can be exploited in algorithms.

In particular, the Ginzburg-Landau theory states that when the coherence length ϵ is small enough, the asymptotic behavior is reached, i.e., the number of critical points of the crossfield is minimal, their index is also minimal and they are optimally distributed. A simple FEM scheme has been implemented to validate numerically this assertion. Crossfields have been computed on the unit disk and solutions conform with the Ginzburg-Landau theory have been found. The location of computed critical points on the 2-sphere corresponds to the solution of Whyte's problem: for a crossfield they are at the summits of an anticube whereas for an asterisk field they are at the summits of a regular dodecahedron.

By weakening the boundary conditions of the Ginzburg-Landau problem, critical points are no longer repelled in the interior of the domain and can be

located on the boundary, which improves the polyquad decomposition in the case of the NACA profiles.

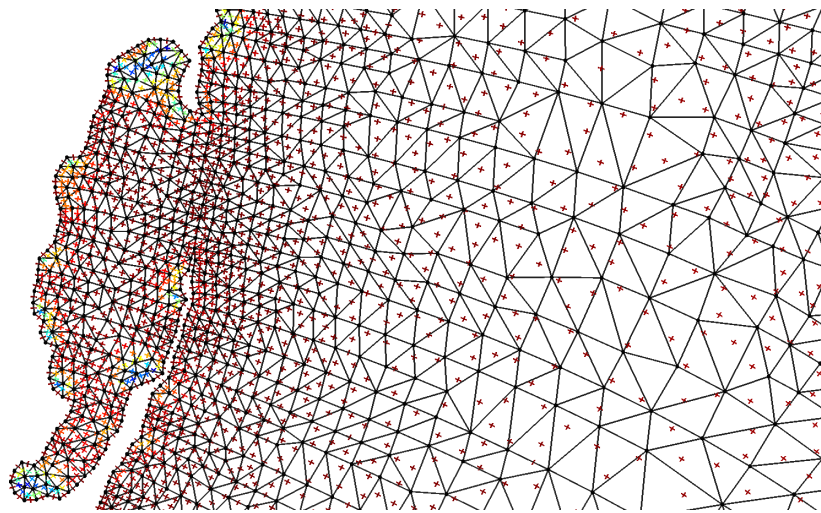
Finally, the process is applied to the quadrangular meshing of the coastal domain around Florida peninsula, Fig. 3.16. Quadrangles are merged from right-angled triangles whose vertices have been spawned along the integral lines of a crossfield, Fig. 3.17a. One sees on Fig. 3.17b how the edges of the recombined quadrangular elements tend to follow the crossfield, and the final mesh is of satisfying quality, Fig. 3.18.

The input triangular mesh can be improved by using an asterisk field. This field is used to spawn vertices which are consistent with an equilateral triangular grid, Fig. 3.19a. The vertices tend to have the correct valence, except in some regions where the size field changes, Fig. 3.19b. The final triangular mesh exhibits a smoother distribution of equilateral triangles through the domain, while the mean quality $\bar{\gamma}$ has been improved to 0.9559 (from $\bar{\gamma}=0.9505$ for the initial mesh), Fig. 3.20.

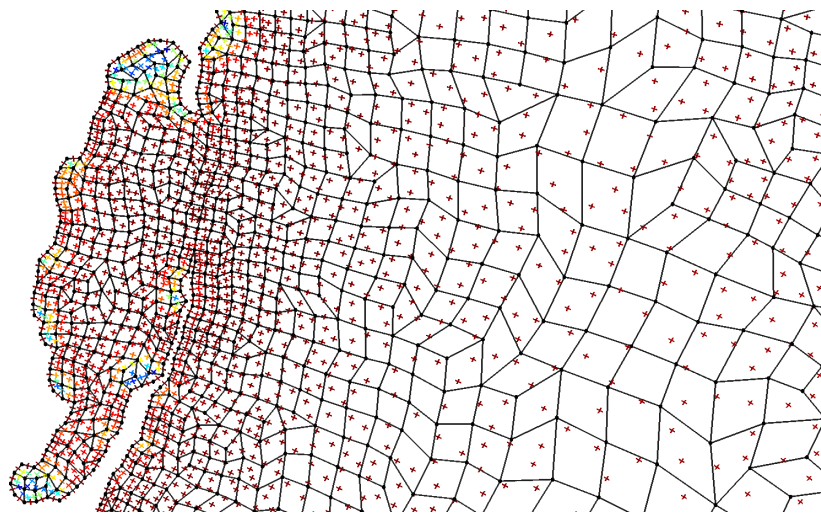
Further work will focus on highly improving the numerical scheme that solves Ginzburg-Landau equations, in order to make it competitive.



Figure 3.16: Florida keys: input triangular mesh ($\bar{\gamma} = 0.9504$). The rectangle will be enlarged.



a. Right-angled triangles from crossfield.



b. Final quadrangular mesh after optimization.

Figure 3.17: Zoom on the Florida keys, the color map is 0 (blue) to 1 (red) and describes the norm of directions.

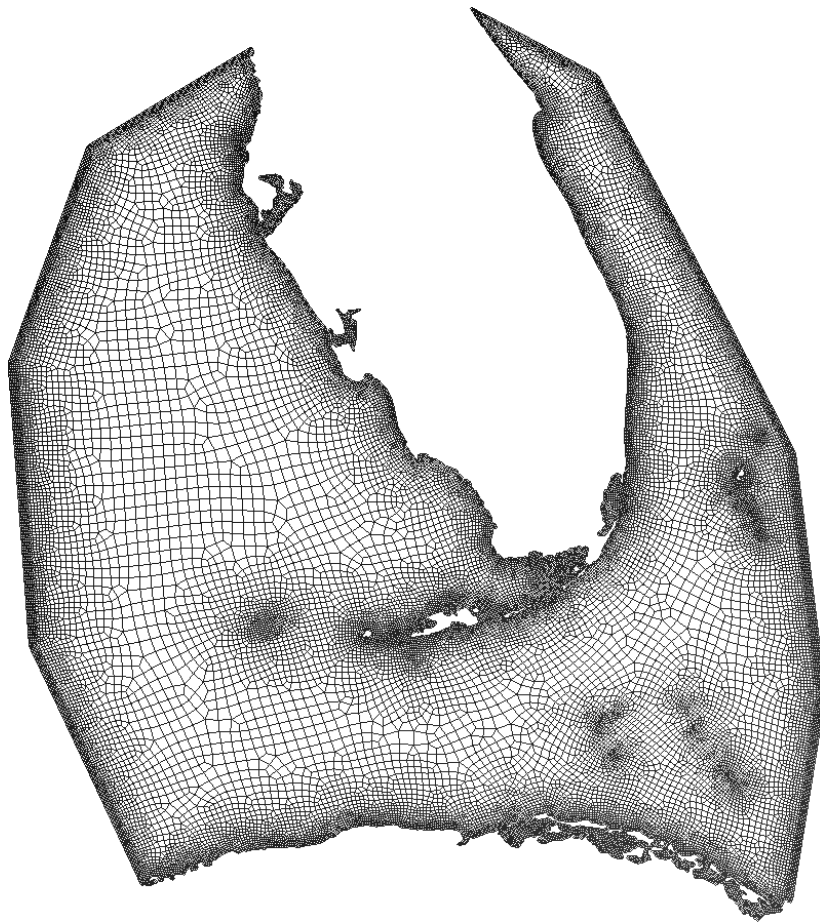
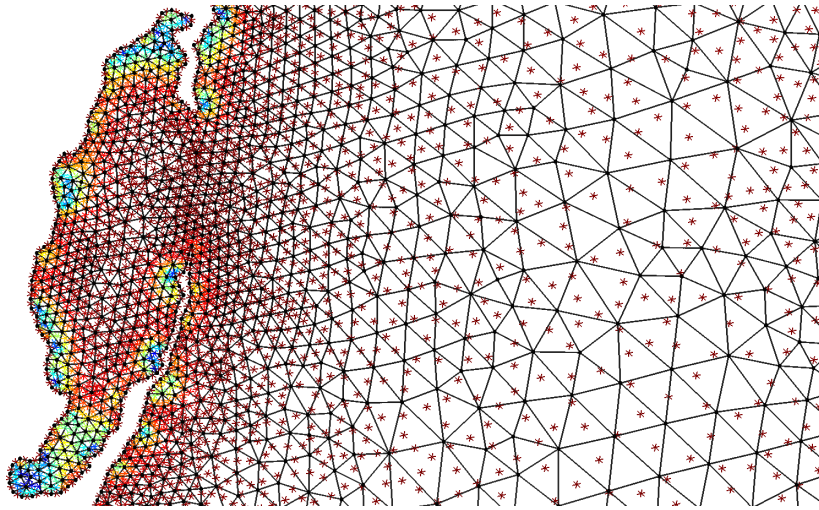
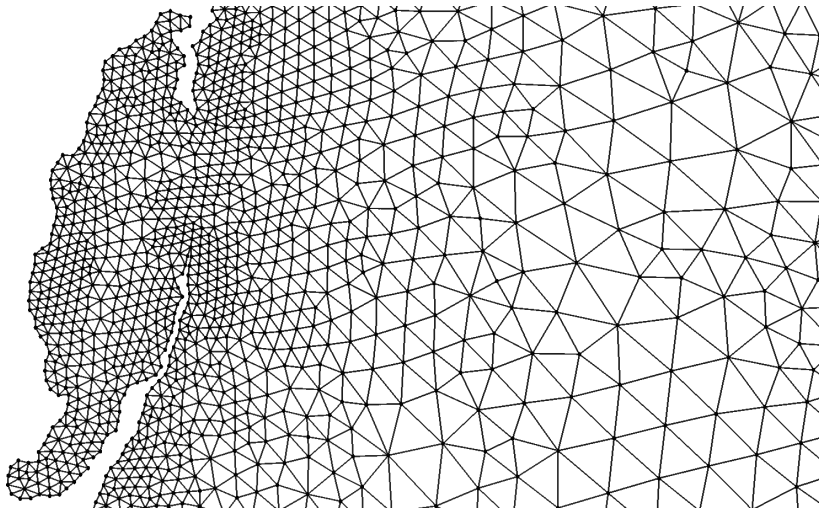


Figure 3.18: Final quadrangular mesh over the Florida keys.

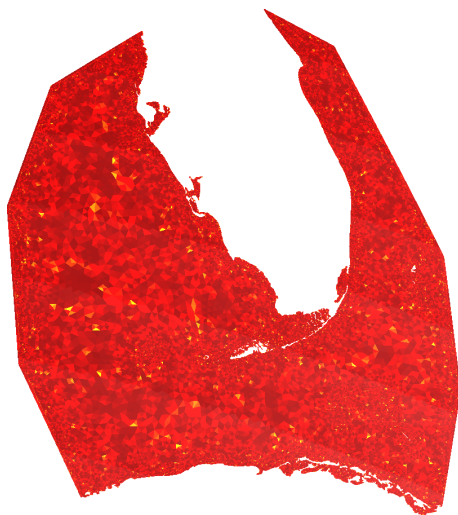


a. Asterisk field over the new triangles.

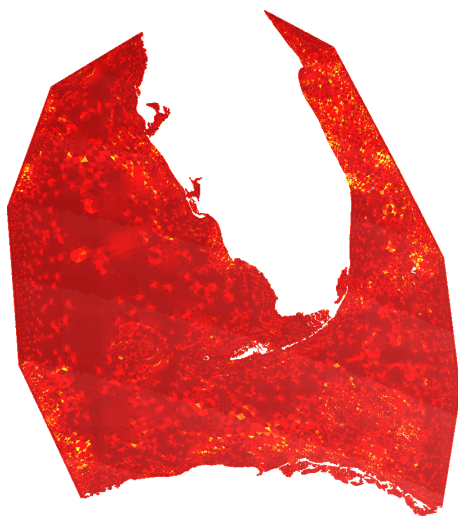


b. New triangular mesh.

Figure 3.19: Zoom on the Florida keys, the color map is 0 (blue) to 1 (red) and describes the norm of directions.



a. Quality value γ of initial triangles.



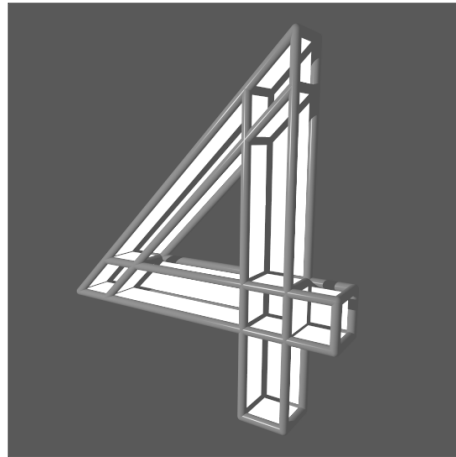
b. Quality value γ of final triangles.

Figure 3.20: Zoom on the Florida keys, the color map is 0 (blue) to 1 (red) and describes the quality of triangles.

Acknowledgements

The present study was carried out in the framework of the project "Large Scale Simulation of Waves in Complex Media", which is funded by the Communauté Française de Belgique under contract ARC WAVES 15/19-03 with the aim of developing and using Gmsh ([Geuzaine and Remacle \(2009\)](#)).

CHAPTER



Quaternionic parameterization of 3D frames

This chapter is a reproduction of the following paper

Beaufort, P.-A., Lambrechts, J., Geuzaine, C., Remacle, J.-F., to submit,
Quaternionic octahedral fields: $SU(2)$ parameterization of 3D frames.

Abstract

Full hexahedral meshes are required to be as regular as possible, which means that the local topology has to be constant almost everywhere. This constraint is usually modelled by 3D frames. A 3D frame consists of three mutual orthogonal (unit) vectors, defining a local basis. 3D frame fields are auxiliary for hexahedral mesh generation. Computation of 3D frame fields is an active research field. There mainly exist three ways to represent 3D frames: combination of rotations, spherical harmonics and fourth order tensor. We propose here a representation carried out by the special unitary group. The article strongly relies on [Du Val \(1964\)](#). We first describe the rotations with quaternions, ([Du Val, 1964, §13-15](#)). We define and show the isomorphism between unit quaternions and the special unitary group, ([Du Val, 1964, §16](#)). The frame field space is identified as the quotient group of rotations by the octahedral group, ([Du Val, 1964, §20](#)). The invariant forms of the vierer, tetrahedral and octahedral groups are successively built, without using homographies ([Du Val, 1964, §39](#)). Modifying the definition of the isomorphism between unit quaternions and the special unitary group allows to use the invariant forms of the octahedral group as a unique parameterization of the orientation of 3D frames. The parameterization consists in three complex values, corresponding to a coordinate of a variety which is embedded in a three complex valued dimensional space. The underlined variety is the model surface of the octahedral group, which can be expressed with an implicit equation. We prove that from a coordinate of the surface, we may identify all the quaternions giving the corresponding 3D frames. We show that the euclidean distance between two coordinates does not correspond to the actual distance of the corresponding 3D frames. We derive the expression of three components of a coordinate in the case of frames sharing an even direction. We then derive a way to ensure that a coordinate corresponds to the special unitary group. Finally, the attempted numerical schemes to compute frame fields are given.

Keywords: Hexahedral Mesh, 3D Frames, Quaternions, Special Unitary Group, Invariant Forms, Model Surface, Variety

1 | Introduction

Full hexahedral mesh is still an open question (see ([Shepherd and Johnson, 2008](#))). Yet, it seems that there is an easy way to produce a full hexahedral

mesh: first produce a tetrahedral mesh, then split each of them into 4 hexahedra. But this way is not convenient: the hexahedra are not regular, they tend to have bad quality and do not form a structured mesh. Finite element community aims to get full hexahedral meshes, possibly structured, which are as regular as possible.

The regularity of an hexahedral mesh is related to the topology of a given domain $R \subset \mathfrak{R}^3$. Let us consider a mesh on $R \subset \mathfrak{R}^3$ with N nodes (i.e. vertices), N_E edges, N_F facets and N_C cells (i.e. element-wise volumes, here being hexahedra) is such that

$$\chi(R) = N - N_E + N_F - N_C \quad (4.1)$$

with $\chi(R)$ the Euler characteristic of the region R . The Euler characteristic of a region is half the one of its boundary (Gross and Kotiuga, 2004, §4C, (4-15))

$$\chi(R) = \frac{\chi(\partial R)}{2} \quad (4.2)$$

We assume there are n nodes, n_e edges and n_f facets making the mesh of ∂R .

$$\chi(R) = \frac{1}{2} (n - n_e + n_f) \quad (4.3)$$

From a topological point of view, an hexahedral mesh is assumed to be regular if each inner (boundary) vertex is shared by

- 8 (4) hexahedra
- 6 (5) edges
- 12 (8) facets

whose corresponding equations are

$$\left. \begin{aligned} 8N_V &= 8(N - n) + 4n \\ 2N_E &= 6(N - n) + 5n \\ 4N_F &= 12(N - n) + 8n \end{aligned} \right\} \quad (4.4)$$

Using (4.4) into (4.1), we get

$$8\chi(R) = 0$$

It means that if a region R may be meshed by regular hexahedra, its Euler characteristic is zero. But the opposite is not true: a region whose characteristic is zero does not mean that it may be meshed by regular hexahedra. For example, let us consider a region that is meshed such that there are k inner loops, each made of L edges. Those kL edges are shared by 3 hexahedra; those

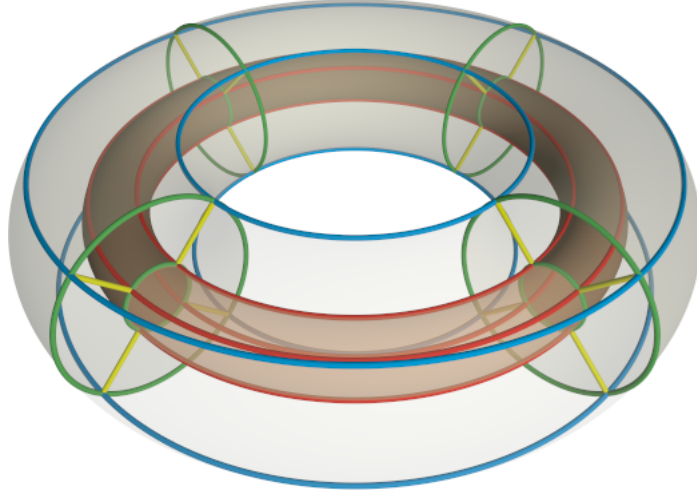


Figure 4.1: Block structure decomposition of a torus and cut torus. The red lines are singular if the inner torus is not an hole, but 4 block structures (front, back, left & right parts). The front part of the tori is more transparent than the other parts.

edges are then singular. The $N - kL$ remaining edges are regular. Equations (4.4) become

$$\left. \begin{aligned} 8N_V &= 8(N - n - kL) + 4n + 6kL \\ 2N_E &= 6(N - n - kL) + 5n + 5kL \\ 4N_F &= 12(N - n - kL) + 8n + 9kL \end{aligned} \right\} \quad (4.5)$$

Again, using (4.5) into (4.1) gives an Euler characteristic that is zero. While the Euler characteristic defines completely the topology of an oriented 2-manifold (surface), it is not the case for an oriented 3-manifold (region). Indeed, from (4.2) a full torus and a torus cut by a smaller one (i.e. the larger one contains the smaller one, Fig. 4.1) have the same Euler characteristic, which is zero. Obviously, the cut torus may be meshed by regular hexahedra: you produce a regular quadrangulation of the outer boundary that is mapped onto the inner one, then you link the corresponding vertices. A full torus cannot be meshed with regular hexahedra; its block structure decomposition corresponds to four singular inner loops. Both situations are represented by Fig. 4.1. Unfortunately, topological constraints for hexahedrizations are not as nicely summarized as the ones for quadrangular meshes, (Beaufort et al., 2017, §1, (7)).

In order to build full hexahedral mesh that are as regular as possible, we use a three-dimensional frame field designing the desired connectivity of a regular

hexahedral mesh, Fig. 4.2a. A 3D frame field gives in each point a 3D frame, picturing the local orientation (and thus the vertex connectivity) of a hexahedron. Since an orientation is relative, it is measured from the cartesian frame, which is the reference 3D frame Fig. 4.2b. Observe that the corresponding vector field is symmetric, since a frame shares the symmetries of an octahedron, Fig. 4.2c.

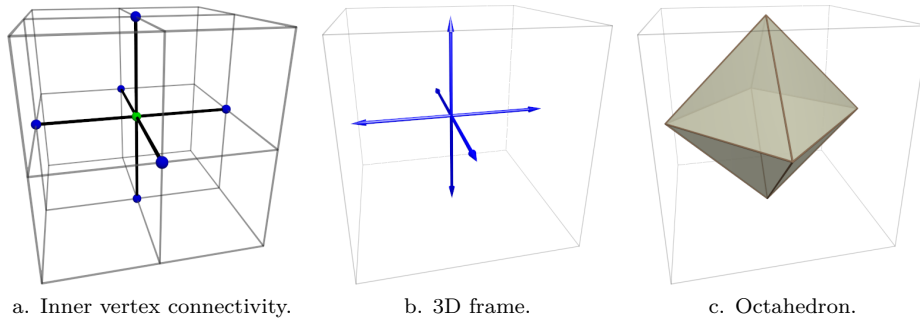


Figure 4.2: Hexahedral features.

2 | Rotation Representations

There exist various ways to represent a frame field. But at the end of the day, they all essentially consist in rotations of a vector field representing an object that exhibits the 24 symmetries of the octahedral group. Such objects may be for example fourth order tensors (Chemin et al., 2018), or spherical harmonics (Ray et al., 2016). In those latter cases, they are represented by a nine-dimensional vector. Actually, they are both based on the representation of

$$\hat{f}(x; y; z) = x^4 + y^4 + z^4 \quad (4.6)$$

which is the polynomial exhibiting the 24 octahedral symmetries corresponding to the cartesian frame.

In the case of *spherical harmonics*, it is seen as a polynomial taking values on the sphere S_2 , Fig. 4.3. This polynomial may be decomposed with the real spherical harmonics of fourth degree.

$$\hat{f}(x; y; z) = \frac{4\sqrt{\pi}}{15} \left(Y_{4,0}(x; y; z) + \sqrt{\frac{5}{7}} Y_{4,4}(x; y; z) \right) + \frac{3}{5}, \forall (x; y; z) \in S_2$$

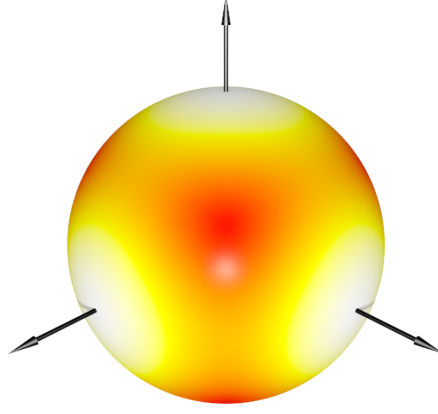


Figure 4.3: $\hat{f}(x; y; z) = x^4 + y^4 + z^4$ on the sphere S_2 .

with

$$\begin{aligned} Y_{4,0}|_{S_2} &= \frac{3}{16} \sqrt{\frac{1}{\pi}} (3x^4 + 3y^4 + 8z^4 + 6x^2y^2 - 24x^2z^2 - 24y^2z^2) \\ Y_{4,4}|_{S_2} &= \frac{3}{16} \sqrt{\frac{35}{\pi}} (x^4 + y^4 - 6x^2y^2 + 12) \end{aligned}$$

If we rotate the cartesian frame \hat{f} by means of a matrix $R \in \text{SO}(3)$, we get f^1

$$f(x_0; x_1; x_2) := \hat{f}(R_{0i}x_i; R_{1j}x_j; R_{2k}x_k)$$

with $x_m = R_{mn}x_n$. The function f may still be expressed with real spherical harmonics of fourth degree.

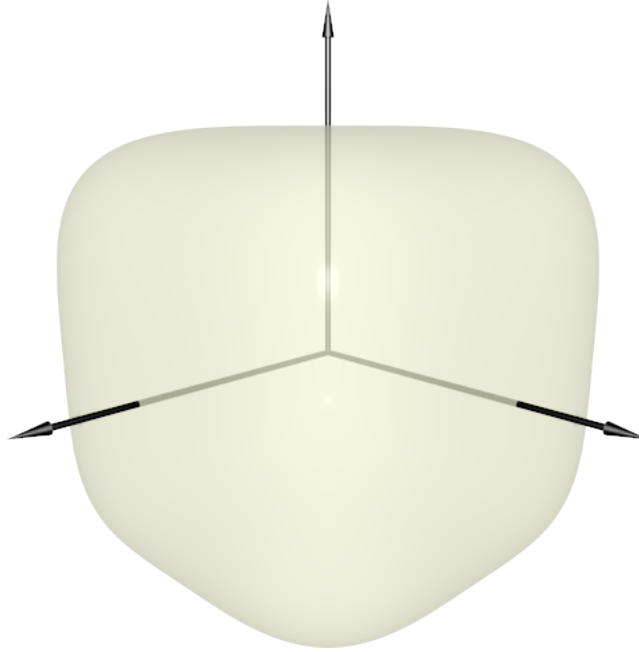
If we consider the isosurface described by the points where polynomial (4.6) is equal to one, it corresponds to a unit sphere in 4-norm which may be written as a *fourth order tensor*

$$\begin{aligned} \hat{f}(x_0; x_1; x_2) &= \hat{A}_{ijkl} x_i x_j x_k x_l \\ \text{with } \hat{A}_{ijkl} &= \sum_{q=1}^3 \delta_{iq} \delta_{jq} \delta_{kq} \delta_{lq} \end{aligned}$$

Again, if we rotate the frame \hat{f} with a matrix $R \in \text{SO}(3)$, we have

$$f(x_0, x_1, x_2) = \hat{f}(R_{1i}x_i, R_{2i}x_i, R_{3i}x_i)$$

¹by denoting $x = x_0$, $y = x_1$, $z = x_2$

Figure 4.4: $\hat{f}(x; y; z) = 1$.

which generalizes \hat{A} as fourth order tensor

$$A_{ijkl} = R_{im}R_{jn}R_{ko}R_{lp}\hat{A}_{mnop}$$

giving

$$f(x_0, x_1, x_2) = A_{ijkl}x_ix_jx_kx_l$$

As illustrated by those two representations, we understand that 3D frame fields consist in rotations depicted by the quotient group

$$\text{SO}(3)/\text{O} \subset \text{SO}(3)$$

where O is the octahedral group, i.e. the 24 rotations leaving invariant the orientation of an octahedron whose vertices are at the units of each axis, Fig. 4.2c.

Those two representations work with an object sharing the octahedral symmetries, which enables to identify this quotient group. We here propose to work directly with the corresponding rotational group. To do so, we need to describe three rotational groups by means of quaternions. We later build the corresponding invariant forms by avoiding their symmetries.

2.1 Quaternions

A real quaternion q consists of four real numbers $(q_0; q_1; q_2; q_3) \in \mathfrak{R}^4$. Using three imaginary units $\mathbf{i}, \mathbf{j}, \mathbf{k}$ such that

- $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$,
- $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$,
- $\mathbf{jk} = -\mathbf{kj} = \mathbf{i}$,
- $\mathbf{ki} = -\mathbf{ik} = \mathbf{j}$

the quaternion q may be written as

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

Hence, addition of quaternions is common

$$p + q = (q_0 + q_0) + (p_1 + q_1)\mathbf{i} + (p_2 + q_2)\mathbf{j} + (p_3 + q_3)\mathbf{k}$$

while their product is *not* commutative

$$p q = \begin{aligned} &(q_0 q_0 - (p_1 q_1 + p_2 q_2 + p_3 q_3)) + (p_0 q_1 + q_0 p_1 + (p_2 q_3 - q_2 p_3))\mathbf{i} \\ &+ (p_0 q_2 + q_0 p_2 + (p_3 q_1 - q_3 p_1))\mathbf{j} + (p_0 q_3 + q_0 p_3 + (p_1 q_2 - q_1 p_2))\mathbf{k} \end{aligned} \quad (4.7)$$

The norm of a quaternion q is defined by means of its conjugate q^*

$$\begin{aligned} q^* &= q_0 - (q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}) \\ |q|^2 &= q q^* = q^* q \end{aligned}$$

Then, it follows that the inverse of q is

$$q^{-1} = \frac{q^*}{|q|^2}$$

The imaginary part $q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ of a quaternion q may be identified as a vector $\mathbf{q} \in \mathfrak{R}^3$

$$q = q_0 + \mathbf{q}$$

Using dot (\cdot) and cross (\times) products of vectors $\mathbf{p}, \mathbf{q} \in \mathfrak{R}^3$, we may write (4.7) as

$$p q = (p_0 q_0 - \mathbf{p} \cdot \mathbf{q}) + \underline{(p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q})}$$

where the underlined part is the 3D vector representing the imaginary part of the product. Scalar and vector products as well as vector addition are covariant with rotations of $\text{SO}(3)$, which implies that the imaginary part of a quaternion is also covariant. It means that every rotation of the imaginary

part \mathbf{q} corresponds to an *automorphism* of quaternions, i.e. a bijective mapping from quaternions to quaternions which preserves the structure of quaternions (i.e. their product). We are going to identify this automorphism, which is related to quaternion product.

Assuming that the four components defining a quaternion correspond to coordinates of an euclidean space of fourth dimension, we consider unit quaternions \hat{q} such that

$$\hat{q}_0^2 + \hat{q}_1^2 + \hat{q}_2^2 + \hat{q}_3^2 = 1 = \hat{q}_0^2 + |\hat{\mathbf{q}}|^2$$

A unit quaternion \hat{q} may define right (\mathcal{R}) and left (\mathcal{L}) screws on a quaternion p (whose norm may be different from 1),

$$\mathcal{R}_{\hat{q}}(p) : p \mapsto \hat{q} p, \quad \mathcal{L}_{\hat{q}}(p) : p \mapsto p \hat{q} \quad (4.8)$$

which are automorphisms of quaternions².

The screws do not alter the norm of p

$$\begin{aligned} |p \hat{q}|^2 &= (p_0^2 \hat{q}_0^2 + (\mathbf{p} \cdot \hat{\mathbf{q}})^2 - 2p_0 \hat{q}_0 \mathbf{p} \cdot \hat{\mathbf{q}}) + \\ &\quad (p_0^2 |\hat{\mathbf{q}}|^2 + \hat{q}_0^2 |\mathbf{p}|^2 + |\mathbf{p}|^2 |\hat{\mathbf{q}}|^2 - (\mathbf{p} \cdot \hat{\mathbf{q}})^2 + 2p_0 \hat{q}_0 \hat{\mathbf{q}} \cdot \mathbf{p}) \\ &= (p_0^2 + |\mathbf{p}|^2)(\hat{q}_0^2 + |\hat{\mathbf{q}}|^2) \\ &= |\hat{q} p|^2 \\ &= |p|^2 \end{aligned}$$

where we used the relationship $|\mathbf{a} \times \mathbf{b}|^2 = |\mathbf{a}|^2 |\mathbf{b}|^2 - (\mathbf{a} \cdot \mathbf{b})^2$, with $\mathbf{a}, \mathbf{b} \in \mathfrak{R}^3$.

Besides, if we consider the angle γ between the fourth dimensional vectors corresponding to quaternions p and q , it is the same between their image produced by any screws \hat{r}

$$\begin{aligned} (\hat{r} p) \odot_{\mathfrak{R}^4} (\hat{r} q) &= (\hat{r}_0 p_0 - \hat{\mathbf{r}} \cdot \mathbf{p} + \hat{r}_0 \mathbf{p} + p_0 \hat{\mathbf{r}} + \hat{\mathbf{r}} \times \mathbf{p}) \\ &\quad \odot_{\mathfrak{R}^4} (\hat{r}_0 q_0 - \hat{\mathbf{r}} \cdot \mathbf{q} + \hat{r}_0 \mathbf{q} + q_0 \hat{\mathbf{r}} + \hat{\mathbf{r}} \times \mathbf{q}) \\ &= (\hat{r}_0^2 + |\hat{\mathbf{r}}|^2)(p_0 q_0 + \mathbf{p} \cdot \mathbf{q}) \\ &= (p \hat{r}) \odot_{\mathfrak{R}^4} (q \hat{r}) \\ &= p \odot_{\mathfrak{R}^4} q \\ &= \cos(\gamma) |p| |q| \end{aligned}$$

where we used the following relationships

- $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$
- $(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c})$

²Actually, the definition of \mathcal{L} will be modified to be an automorphism.

We understand that screws (4.8) correspond to some kind of rotations. We are going to describe their properties starting with a well chosen unit quaternion $\hat{q} = \cos(\alpha) + \sin(\alpha)\mathbf{k}$.

The corresponding screws give

$$(p_0 \cos(\alpha) - p_3 \sin(\alpha)) + (p_1 \cos(\beta) - p_2 \sin(\beta))\mathbf{i} \\ + (p_2 \cos(\beta) - p_1 \sin(\beta))\mathbf{j} + (p_3 \cos(\alpha) + p_0 \sin(\alpha))\mathbf{k}$$

with $\beta = \pm\alpha$ for respectively $\mathcal{L}_{\hat{q}}(p)$ and $\mathcal{R}_{\hat{q}}(p)$. For general values of β , the transformation may be seen as acting on a fourth dimensional vector, i.e. a matrix-vector product whose matrix is

$$\begin{bmatrix} \cos(\alpha) & 0 & 0 & -\sin(\alpha) \\ 0 & \cos(\beta) & -\sin(\beta) & 0 \\ 0 & \sin(\beta) & \cos(\beta) & 0 \\ \sin(\alpha) & 0 & 0 & \cos(\alpha) \end{bmatrix}$$

It is a *compound* rotation of angles α in the $\langle 1; \mathbf{k} \rangle$ -plane and β in the $\langle \mathbf{i}; \mathbf{j} \rangle$ -plane. Those two planes are *absolutely* orthogonal, which means that they have *no* nonzero vector in common. Then, the compound rotations act only on vectors in their corresponding plane. Those planes actually define the invariant planes of the corresponding rotations. As we said that every rotation of the imaginary part \mathbf{q} of a quaternion q corresponds to an automorphism of quaternions, our arbitrary choice $\hat{q} = \cos(\alpha) + \sin(\alpha)\mathbf{k}$ may be generalized to $\hat{r} = \cos(\alpha) + \sin(\alpha)\mathbf{v}$ with $\mathbf{v} \in \mathfrak{R}^3$ s.t. $|\mathbf{v}|^2 = 1$. The two invariant planes of $\mathcal{R}_{\hat{r}}(p), \mathcal{L}_{\hat{r}}(p)$ correspond to the one joining the real axis to \mathbf{v} and the one that is normal to \mathbf{v} contained within the region $\langle \mathbf{i}; \mathbf{j}; \mathbf{k} \rangle = \mathfrak{R}^3 \subset \mathfrak{R}^4$ (called *imaginary prime*), since we have rotated the imaginary part of q only. It means that the invariant planes are still absolutely orthogonal.

$\mathcal{R}_{\hat{r}}(p) \forall \hat{r}$ is isomorphic to the product of unit quaternions denoted by \hat{Q} . It means that the combination of $\mathcal{R}_{\hat{a}}, \mathcal{R}_{\hat{b}}$ corresponds to $\mathcal{R}_{(\hat{b}\hat{a})}$, i.e. the right screw parameterized by the product of units quaternions $\hat{r}_b \hat{r}_a$. However, $\mathcal{L}_{\hat{r}}(p)$ is not isomorphic to the product of unit quaternions; it is not *yet* an automorphism. Indeed, the combination of $\mathcal{L}_{\hat{a}}, \mathcal{L}_{\hat{b}}$ (in this order) has to act from right in the quaternion product. To ensure a consistent combination, we have to modify the left screw definition, which is *now* an automorphism

$$\mathcal{L}'_{\hat{r}}(p) : p \mapsto p\hat{r}^{-1}$$

The combination

$$\mathcal{L}'_{\hat{b}\hat{a}}(p) : p \mapsto p\hat{a}^{-1} \hat{b}^{-1}$$

is then consistent, since the left screws act on the left, in the correct order.

Owing to the associative multiplication of quaternions, combination of $\mathcal{R}_{\hat{r}}$ and $\mathcal{L}'_{\hat{r}}$ is simply

$$\mathcal{A}_{\hat{r}}(p) : p \mapsto \hat{r} p \hat{r}^{-1} \quad (4.9)$$

Automorphisms of form (4.9) are *inner* automorphisms.

Taking again the particular case $q = \cos(\alpha) + \sin(\alpha)\mathbf{k}$, $\mathcal{A}_{\hat{q}}$ may be expressed with products of the two matrices

$$\begin{aligned} & \underbrace{\begin{bmatrix} \cos(\alpha) & 0 & 0 & \sin(\alpha) \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ -\sin(\alpha) & 0 & 0 & \cos(\alpha) \end{bmatrix}}_{\mathcal{L}'_{\hat{q}}} \underbrace{\begin{bmatrix} \cos(\alpha) & 0 & 0 & -\sin(\alpha) \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ \sin(\alpha) & 0 & 0 & \cos(\alpha) \end{bmatrix}}_{\mathcal{R}_{\hat{q}}} \\ &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\alpha) & -\sin(2\alpha) & 0 \\ 0 & \sin(2\alpha) & \cos(2\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathcal{A}_{\hat{q}}} \end{aligned}$$

First, observe that the matrices may commute since the compound rotations act in two same invariant planes. We notice that the two rotations in the $\langle 1; \mathbf{k} \rangle$ -plane avoid each other, while in the $\langle \mathbf{i}; \mathbf{j} \rangle$ -plane they add to each other. Reminding that \hat{q} is arbitrary and may be generalized to $\hat{r} = \cos(\alpha) + \mathbf{v} \sin(\alpha)$, we understand that $\mathcal{R}_{\hat{r}}$ is a compound rotation of angle α in the planes joining the real axis to \mathbf{v} and the one which is normal to \mathbf{v} contained in the imaginary prime, while $\mathcal{L}'_{\hat{r}}$ is respectively a rotation of $-\alpha$ and α in those latter planes. Hence, $\mathcal{A}_{\hat{r}}$ is a simple rotation of angle 2α around \mathbf{v} in the imaginary prime, which coincides with \mathfrak{R}^3 . We have identified the automorphism of quaternions which corresponds to a rotation of the imaginary part of a quaternion, leaving unchanged its real part: it is the inner automorphism (4.9). Those inner automorphisms provides us a *homomorphic* mapping between \hat{Q} and $\text{SO}(3)$, since two (opposite) unit quaternions correspond to a single rotation of $\text{SO}(3)$: $\pm \hat{r}$.

$$\hat{Q} \xrightarrow{2:1} \text{SO}(3) \quad (4.10)$$

Obviously, the opposite is also true: any inner automorphisms of nonzero quaternions $q \in \mathfrak{R}^4$ represented by

$$\mathcal{A}_q(p) : p \mapsto q p q^{-1} \quad (4.11)$$

is a rotation of the imaginary prime, since any nonzero $q \in \mathfrak{R}^4$ may be written as $q := |q|\hat{q}$.

2.2 Special unitary group $\text{SU}(2)$

Let a quaternion $q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$. We define $u, v \in \mathbb{C}$

$$u := q_0 + q_3 i, \quad v := -(q_2 + q_1 i) \quad (4.12)$$

such that they parameterize the complex matrix of the following form

$$\begin{bmatrix} u & -v^* \\ v & u^* \end{bmatrix} \quad (4.13)$$

The above relationship (4.12) defines an isomorphism between the quaternions and the two-by-two skew-Hermitian matrices³ of form (4.13), which are called *quaternionic* matrices. Let us consider two quaternions $p, q \in \mathfrak{H}^4$ and their corresponding quaternionic matrices $P, Q \in \mathbb{C}^{2 \times 2}$. It is straightforward that the addition $P + Q$ corresponds $p + q$, and conversely. The products PQ and $p q$ correspond to each other

$$\begin{bmatrix} u_p & -v_p^* \\ v_p & u_p^* \end{bmatrix} \begin{bmatrix} u_q & -v_q^* \\ v_q & u_q^* \end{bmatrix} = \begin{bmatrix} u_p u_q - v_p^* v_q & -(u_p v_q^* + u_p^* u_q^*) \\ v_p u_q + u_p^* v_q & -v_p v_q^* + u_p^* u_q^* \end{bmatrix}$$

where

- $u_p u_q - v_p^* v_q = (q_0 q_0 - (p_1 q_1 + p_2 q_2 + p_3 q_3)) + (p_0 q_3 + q_0 p_3 + (p_1 q_2 - q_1 p_2)) i$
- $v_p u_q + u_p^* v_q = -(p_0 q_2 + q_0 p_2 + (p_3 q_1 - q_3 p_1)) - (p_0 q_1 + q_0 p_1 + (p_2 q_3 - q_2 p_3)) i$

whose real and imaginary parts correspond to the quadruplet defining the product $p q$.

If we apply (4.12) to a unit quaternion $\hat{q} \in \hat{Q}$, the quaternionic (4.13) matrix has a determinant $\hat{u}\hat{u}^* + \hat{v}\hat{v}^* = 1$. This matrix form corresponds to the SU(2) group. Owing to (4.12) for unit quaternions, we have

$$\text{SU}(2) \xleftrightarrow{1:1} \hat{Q} \xleftrightarrow{2:1} \text{SO}(3) \quad (4.14)$$

At this point, we could wonder why using (4.12) to define (4.13). If we use the common definition $u, v \in \mathbb{C}$

$$u = q_0 + q_1 i, \quad v = q_2 + q_3 i$$

the product $p q$ would correspond to the product of quaternionic matrices QP (instead of PQ).

Let us come back to screws \mathcal{L} and \mathcal{R} . We consider $\hat{q}, \hat{p} \in \hat{Q}$ and $r \in \mathfrak{H}^4$, with their respective complex coordinates $(\hat{u}_q; \hat{v}_q), (\hat{u}_p; \hat{v}_p) \in \text{SU}(2)$ and $(u_r; v_r) \in \mathbb{C}^2$ defined by (4.12). We know those screws correspond to rotations of the fourth dimensional vector corresponding to \mathfrak{H}^4 , which may be represented by a matrix-vector product. The corresponding quaternionic representation of $\mathcal{R}_{\hat{r}}(p)$ is

$$\begin{bmatrix} u_p \\ v_p \end{bmatrix} \mapsto \begin{bmatrix} \hat{u}_{\hat{r}} & -\hat{v}_{\hat{r}}^* \\ \hat{v}_{\hat{r}} & \hat{u}_{\hat{r}}^* \end{bmatrix} \begin{bmatrix} u_p \\ v_p \end{bmatrix} = \begin{bmatrix} \hat{u}_{\hat{r}} u_p - \hat{v}_{\hat{r}}^* v_p \\ \hat{v}_{\hat{r}} u_p + \hat{u}_{\hat{r}}^* v_p \end{bmatrix} \quad (4.15)$$

³ $A \in \mathbb{C}^{2 \times 2}$ s.t. $A_{ij} = -A_{ji}^*$

whose real and imaginary parts correspond to the four components of $\hat{p} r \in \mathfrak{R}^4$. The transformation (4.15) is a *complex affine transformation*. However, there is no such affine transformation (acting on $(u_p; v_p)$) corresponding to the left screw $\mathcal{L}'_{\hat{r}}$. The corresponding quaternionic representation is

$$[u_p \quad -v_p^*] \mapsto [u_p \quad -v_p^*] \begin{bmatrix} \hat{u}_{\hat{r}} & -\hat{v}_{\hat{r}}^* \\ \hat{v}_{\hat{r}} & \hat{u}_{\hat{r}}^* \end{bmatrix}^{-1} = [\hat{u}_{\hat{r}}^* u_p + \hat{v}_{\hat{r}} v_p^* \quad \hat{v}_{\hat{r}}^* u_p - \hat{u}_{\hat{r}} v_p^*]$$

which is a complex affine transformation on $(u_p; -v_p^*)$.

3 | Frame Field Space

In the last section, we have seen that a frame can be represented by a rotation of the cartesian frame. However, there are 24 distinct rotations (i.e. 48 unit quaternions) giving a unique frame. We need functions which give a unique set of values for each frame, in other words a same set of values for the 48 unit quaternions generating a given frame.

We are going to analyze how some rotational groups act on $(u; v) \in \mathbb{C}^2$ as affine transformations. This analysis will give three invariant forms per group, giving a set of values corresponding to coordinates of a surface embedded in \mathbb{C}^3 . We eventually need to redefine $(u; v)$ from a quaternion, in order to properly parameterize the frames from those invariant forms.

3.1 Finite groups of quaternions

We are mainly interested in the octahedral group, which shares the symmetries of a frame. Nevertheless, we need to define it from two smaller finite groups of quaternions, which are the vierer and binary tetrahedral groups.

The *vierer group* $\mathcal{V} \subset \hat{Q}$, only consists of 4 rotations that are of angle π around the axes of the cartesian frame including the identity

$$\mathcal{V} = \{\pm 1; \pm \mathbf{i}; \pm \mathbf{j}; \pm \mathbf{k}\}$$

The *binary tetrahedral group* $\mathcal{T} \subset \hat{Q}$ is composed of the 12 rotations that leave unchanged the orientation of a regular tetrahedron whose 4 vertices are located at $(1; 1; 1), (1; -1; -1), (-1; 1; -1), (-1; -1; 1)$ and of its dual whose vertices have respectively opposite components of the primal.

$$\mathcal{T} = \mathcal{V} \oplus \left\{ \frac{1}{2} (\pm 1 \pm \mathbf{i} \pm \mathbf{j} \pm \mathbf{k}) \right\}$$

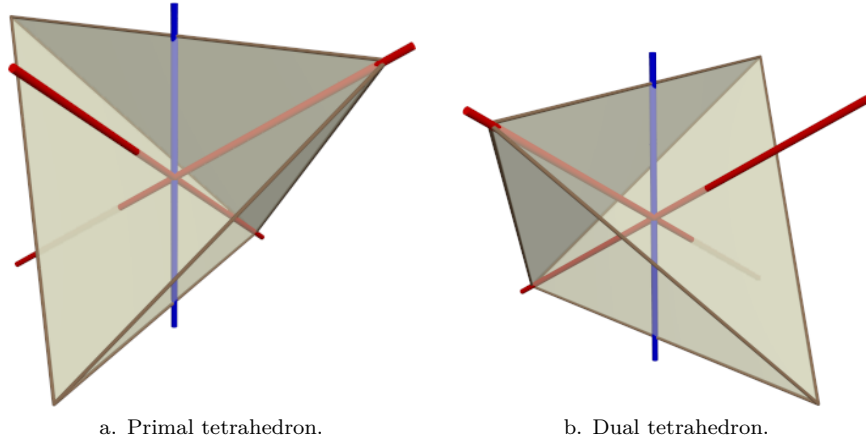


Figure 4.5: Three axes of rotations generating the binary tetrahedral group. Blue and red axes correspond respectively to rotations of π and $\frac{2\pi}{3}$.

The *binary octahedral group* $\mathcal{O} \subset \hat{Q}$ has 24 rotations that preserve the orientation of an octahedron whose vertices are $(\pm 1; 0; 0)$, $(0; \pm 1; 0)$, $(0; 0; \pm 1)$ and of the dual cube whose the centers of each face correspond to the vertices of the primal octahedron.

$$\mathcal{O} = \mathcal{T} \oplus \left\{ \frac{1}{\sqrt{2}} (\pm 1 \pm \mathbf{i}); \frac{1}{\sqrt{2}} (\pm 1 \pm \mathbf{j}); \frac{1}{\sqrt{2}} (\pm 1 \pm \mathbf{k}); \right. \\ \left. \frac{1}{\sqrt{2}} (\pm \mathbf{i} \pm \mathbf{j}); \frac{1}{\sqrt{2}} (\pm \mathbf{j} \pm \mathbf{k}); \frac{1}{\sqrt{2}} (\pm \mathbf{i} \pm \mathbf{k}) \right\}$$

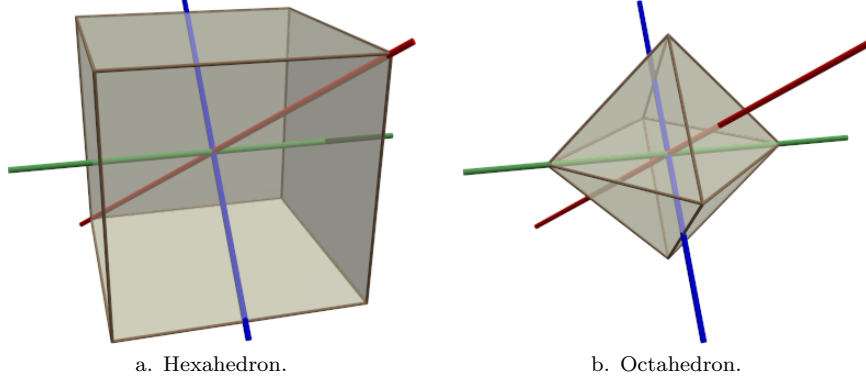


Figure 4.6: Three axes of rotations generating the binary octahedral group. Blue, red and green axes correspond respectively to rotations of π , $\frac{2\pi}{3}$ and $\frac{\pi}{2}$.

Each finite group may be generated by three unit quaternions

$$\hat{a} = \cos\left(\frac{\pi}{p}\right) + \mathbf{a} \sin\left(\frac{\pi}{p}\right), \quad \hat{b} = \cos\left(\frac{\pi}{q}\right) + \mathbf{b} \sin\left(\frac{\pi}{q}\right), \quad \hat{c} = \cos\left(\frac{\pi}{2}\right) + \mathbf{c} \sin\left(\frac{\pi}{2}\right) \quad (4.16)$$

with $\mathbf{a}, \mathbf{b}, \mathbf{c}$ the corresponding axes of rotation. Those unit quaternions are such that they satisfy the relationships

$$\hat{a}^p = \hat{b}^q = \hat{c}^2 = \hat{a}\hat{b}\hat{c} = -1 \quad (4.17)$$

which is depicted by the triplet $(p; q; 2)$; it corresponds to the powers of (4.17).

The triplet $(2; 2; 2)$ corresponds to the vierer group \mathcal{V} , with (e.g.) $\mathbf{a} = \mathbf{i}, \mathbf{b} = \mathbf{j}, \mathbf{c} = \mathbf{k}$. The triplet $(3; 3; 2)$ corresponds to the binary tetrahedral group \mathcal{T} , with (e.g.) $\mathbf{a} = \frac{1}{\sqrt{3}}(\mathbf{i} + \mathbf{j} + \mathbf{k}), \mathbf{b} = \frac{1}{\sqrt{3}}(-\mathbf{i} + \mathbf{j} + \mathbf{k}), \mathbf{c} = \mathbf{k}$. The triplet $(4; 3; 2)$ corresponds to the binary octahedral group \mathcal{O} , with (e.g.) $\mathbf{a} = \mathbf{i}, \mathbf{b} = \frac{1}{\sqrt{3}}(\mathbf{i} + \mathbf{j} + \mathbf{k}), \mathbf{c} = \frac{1}{\sqrt{2}}(\mathbf{j} - \mathbf{k})$.

3.2 Model surface

We look for a set of functions $f_k : \mathbb{C}^2 \mapsto \mathbb{C}$ such that

$$f_k \left(A_i \begin{bmatrix} u \\ v \end{bmatrix} \right) = f \left(\begin{bmatrix} u \\ v \end{bmatrix} \right), \quad \forall A_i \in \mathcal{V}, \mathcal{T}, \text{ xor } \mathcal{O} \quad (4.18)$$

where A_i is a quaternionic matrix corresponding to an element of the finite group $(p; q; 2)$, i.e.

$$A_i^p = -I, A_i^q = -I, \text{ xor } A_i^2 = -I$$

with I the two-by-two identity matrix.

We know that a finite group $(p; q; 2)$ may be generated by three unit quaternions written as (4.16). Hence, if (4.18) is satisfied for such three unit quaternions, it is satisfied by all quaternions of the finite group. It means we have three unit vectors, i.e. 6 real degrees of freedom. It implies that the set of functions is composed of three functions $(f_0; f_1; f_2) \in \mathbb{C}^3$. However, the last equality of (4.17) is not satisfied for all quaternions written as (4.16). It is a constraint, that will appear as a polynomial relationship between f_0, f_1 and f_2 .

$$f_2^2 = \mathcal{P}(f_0; f_1) \quad (4.19)$$

where \mathcal{P} is a bivariate polynomial. This relationship defines a surface embedded in \mathbb{C}^3 . It is parameterized by $(u; v) \in \mathbb{C}^2$. Besides, we observe that there exists an affine transformation on $(u; v)$ which does not alter $f_0(u; v)$ and $f_1(u; v)$, but which gives an opposite sign to $f_2(u; v)$. This latter transformation does not belong to the current finite group $(p; q; 2)$.

Such a surface (4.19) is a *model surface*, which defines the rotations of a quotient group: $\hat{Q}/\mathcal{V}, \hat{Q}/\mathcal{T}$ or \hat{Q}/\mathcal{O} (abusing of the isomorphism between $SU(2)$ and \hat{Q}). The functions f_k parameterizing it are homogeneous polynomial in $(u; v) \in \mathbb{C}^2$.

3.3 Invariant forms

We are going to derive the invariant forms relative to each finite group.

The vierer group

The vierer group \mathcal{V} has the identity rotation, which may be represented by the following affine transformation

$$\begin{bmatrix} u \\ v \end{bmatrix} \mapsto \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -u \\ -v \end{bmatrix}$$

We find that uv, u^2 and v^2 are invariant under such a transformation. However, they do change for half turns around axes $\mathbf{i}, \mathbf{j}, \mathbf{k}$: in the case of \mathbf{i}

$$\begin{bmatrix} u \\ v \end{bmatrix} \mapsto \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} i v \\ i u \end{bmatrix}$$

which respectively gives $-uv, -v^2$ and $-u^2$. We observe that we have to square every expression and then to combine the two last ones and thus to make a third expression from the three initial ones: $(uv)^2, u^4 + v^4$ and $uv(u^2 + v^2)(u^2 - v^2)$.

Those latter expressions are also invariant for half turns around \mathbf{j} and \mathbf{k} . They are thus invariant forms of \mathcal{V}

$$\begin{cases} f_0(u; v) &= (uv)^2 \\ f_1(u; v) &= u^4 + v^4 \\ f_2(u; v) &= uv(u^4 - v^4) \\ \text{s.t. } f_2^2(u; v) &= f_0(u; v) (f_1^2(u; v) - 4f_0^2(u; v)) \end{cases} \quad (4.20)$$

As claimed, there is an affine transformation that does not belong to \mathcal{V} that does not alter $f_0(u; v), f_1(u; v)$ but which changes the sign of $f_2(u; v)$

$$\begin{bmatrix} u \\ v \end{bmatrix} \mapsto \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u \\ -v \end{bmatrix}$$

The binary tetrahedral group

The binary tetrahedral group \mathcal{T} contains \mathcal{V} . Since a quaternion group is generated by three unit quaternions, if we build invariant forms for \mathcal{T} from those of \mathcal{V} , we just need to have them invariant for two unit quaternions of \mathcal{T} (since it will be invariant to unit quaternions of \mathcal{V}) such that $(3; 3; 2)$. We consider first the rotation $\frac{1}{2}(1 + \mathbf{i} + \mathbf{j} + \mathbf{k})$ described by the following transformation

$$\begin{bmatrix} u \\ v \end{bmatrix} \mapsto \frac{1}{2} \begin{bmatrix} 1+i & i-1 \\ 1+i & 1-i \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{2}(1+i) \begin{bmatrix} u+iv \\ u-iv \end{bmatrix}$$

We first notice that $\tilde{f}_0 = f_1 + 2\sqrt{3}i f_0 = u^4 + v^4 + 2\sqrt{3}i(uv)^2$, $\tilde{f}_1 = f_1 - 2\sqrt{3}i f_0 = u^4 + v^4 - 2\sqrt{3}i(uv)^2$, $\tilde{f}_2 = f_2$ are also invariant forms of \mathcal{V} s.t. $\tilde{f}_2^2 = \frac{\sqrt{3}i}{36}(\tilde{f}_1^3 - \tilde{f}_0^3)$. We may see that

$$\begin{aligned} \tilde{f}_0(u+iv; u-iv) &= (u+iv)^4 + (u-iv)^4 + 2\sqrt{3}i(u^2+v^2)^2 \\ &= 2(1+\sqrt{3}i)(u^4+v^4) + 4(\sqrt{3}i-3)(uv)^2 \\ &= 2(1+\sqrt{3}i)(u^4+v^4 + 2\sqrt{3}i(uv)^2) \\ &= 4 \exp\left(\frac{\pi i}{3}\right) \tilde{f}_0(u; v) \end{aligned}$$

and that

$$\begin{aligned} \tilde{f}_1(u+iv; u-iv) &= (u+iv)^4 + (u-iv)^4 - 2\sqrt{3}i(u^2+v^2)^2 \\ &= 2(1-\sqrt{3}i)(u^4+v^4) - 4(3+\sqrt{3}i)(uv)^2 \\ &= 2(1-\sqrt{3}i)(u^4+v^4 - 2\sqrt{3}i(uv)^2) \\ &= 4 \exp\left(-\frac{\pi i}{3}\right) \tilde{f}_1(u; v) \end{aligned}$$

Knowing that $\left(\frac{1+i}{2}\right)^4 = -\frac{1}{4}$, we notice that

$$\bullet \tilde{f}_0\left(\frac{1}{2}(1+i)(u+iv); \frac{1}{2}(1+i)(u-iv)\right) \tilde{f}_1\left(\frac{1}{2}(1+i)(u+iv); \frac{1}{2}(1+i)(u-iv)\right) = (\tilde{f}_0 \tilde{f}_1)(u; v)$$

- $\tilde{f}_0^3 \left(\frac{1}{2}(1+i)(u+iv); \frac{1}{2}(1+i)(u-iv) \right) = \tilde{f}_0^3(u; v)$
- $\tilde{f}_1^3 \left(\frac{1}{2}(1+i)(u+iv); \frac{1}{2}(1+i)(u-iv) \right) = \tilde{f}_1^3(u; v)$
- $\tilde{f}_2 \left(\frac{1}{2}(1+i)(u+iv); \frac{1}{2}(1+i)(u-iv) \right) = \tilde{f}_2(u; v)$

Since \tilde{f}_i are homogeneous polynomials in $(u; v)$, we get the same results for $\frac{1}{2}(1-i+j-k)$. It is then possible to build invariant forms from the above relationships. Indeed, the product $\tilde{f}_0\tilde{f}_1$ and \tilde{f}_2 are invariant for elements of \mathcal{V} , $\frac{1}{2}(1+i+j+k)$ and $\frac{1}{2}(1-i+j-k)$ which satisfy (3; 3; 2). Those two expressions are thus invariant to all elements of \mathcal{T} . Remembering that $\tilde{f}_2^2 = \frac{\sqrt{3}i}{36}(\tilde{f}_1^3 - \tilde{f}_0^3) = (uv(u^4 - v^4))^2$ is a linear combination of \tilde{f}_0^3 and \tilde{f}_1^3 , which are invariant to \mathcal{T} , we end up with the following invariant forms for \mathcal{T}

$$\begin{cases} g_0(u; v) &= uv(u^4 - v^4) = \tilde{f}_2(u; v) \\ g_1(u; v) &= u^8 + v^8 + 14(uv)^4 = \tilde{f}_0(u; v)\tilde{f}_1(u; v) \\ g_2(u; v) &= u^{12} + v^{12} - 33(uv)^4(u^4 + v^4) = \tilde{f}_0^3(u; v) + \tilde{f}_1^3(u; v) \\ \text{s.t. } g_2^2(u; v) &= g_1^3(u; v) - 108g_0^4(u; v) \end{cases} \quad (4.21)$$

The affine transformation

$$\begin{bmatrix} u \\ v \end{bmatrix} \mapsto \begin{bmatrix} \exp\left(\frac{\pi i}{4}\right) & 0 \\ 0 & -\exp\left(-\frac{\pi i}{4}\right) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \exp\left(\frac{\pi i}{4}\right)u \\ -\exp\left(-\frac{\pi i}{4}\right)v \end{bmatrix} \quad (4.22)$$

leaves g_0 and g_1 unchanged while g_2 has its sign changed.

The binary octahedral group

The binary octahedral group \mathcal{O} containing \mathcal{T} , which contains \mathcal{V} . Since \mathcal{O} may be generated by three unit quaternions such that (4; 3; 2), we just have to build invariant forms from \mathcal{T} that are invariant to a rotation consisting in a quarter turn around one of the axes $\mathbf{i}, \mathbf{j}, \mathbf{k}$ (or any combinations of two of them). In the case of a quarter turn around \mathbf{i} , it corresponds to the following affine transformation

$$\begin{bmatrix} u \\ v \end{bmatrix} \mapsto \begin{bmatrix} \exp\left(\frac{\pi i}{4}\right) & 0 \\ 0 & \exp\left(-\frac{\pi i}{4}\right) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \exp\left(\frac{\pi i}{4}\right)u \\ \exp\left(-\frac{\pi i}{4}\right)v \end{bmatrix}$$

which obviously leaves unchanged $g_1(u; v)$, but it changes the sign of $g_0(u; v)$ and $g_2(u; v)$. It means that g_0^2 and the product g_0g_1 are invariant under the above affine transformation. Therefore, we get the following invariant forms

for \mathcal{O}

$$\begin{cases} h_0(u; v) &= u^8 + v^8 + 14(uv)^4 = g_1(u; v) \\ h_1(u; v) &= (uv(u^4 - v^4))^2 = g_0^2(u; v) \\ h_2(u; v) &= uv(u^4 - v^4)(u^{12} - 33u^4v^4(u^4 + v^4) + v^{12}) = (g_0g_2)(u; v) \\ \text{s.t. } h_2^2(u; v) &= h_1(u; v) (h_0^3(u; v) - 108h_1^2(u; v)) \end{cases} \quad (4.23)$$

$h_0(u; v)$ and $h_1(u; v)$ are invariant under the affine transformation (4.22), while $h_2(u; v)$ takes the opposite sign.

3.4 Going back to frames

We have identified invariant forms for *right* screws taking their value in either \mathcal{V}, \mathcal{T} or \mathcal{O} . In the case of \mathcal{O} , it is expressed as

$$\hat{o}_i \hat{r} = \hat{q}_i$$

with $\hat{o}_i \in \mathcal{O}, 0 \leq i < 48$ and $\hat{r} \in \hat{Q}$. Hence, the 48 rotations of the cartesian frame $\hat{\mathbf{f}} = \{\pm\mathbf{i}; \pm\mathbf{j}; \pm\mathbf{k}\}$ giving another 3D frame are described by \hat{q}_i

$$(\hat{o}_i \hat{r}) \hat{\mathbf{f}} (\hat{o}_i \hat{r})^{-1}$$

However, these rotations do not describe a 3D frame. The rotations \hat{o}_i, \hat{r} are performed in the wrong order. The 48 rotations \hat{o}_i have to be applied first on $\hat{\mathbf{f}}$, then only the rotation \hat{r} giving the new 3D frame has to be applied.

To reverse the order, we have to conjugate the quaternions \hat{q}_i corresponding to $(h_0; h_1; h_2)$. Indeed,

$$(\hat{o}_i \hat{r})^* \hat{\mathbf{f}} ((\hat{o}_i \hat{r})^*)^{-1} = \hat{r}^* \hat{o}_i^* \hat{\mathbf{f}} \hat{o}_i \hat{r} \quad (4.24)$$

Since $\hat{o}_i^* \in \mathcal{O} \forall i$, (4.24) applies in the correct order the rotations, which produces a new 3D frame corresponding to a rotation \hat{r} of $\hat{\mathbf{f}}$ up to a symmetry. Therefore, we have to update (4.12) in order to take account of the conjugation of \hat{q}

$$\left. \begin{aligned} u &= q_0 - q_1 i \\ v &= q_2 + q_3 i \end{aligned} \right\} \quad (4.25)$$

4 | Numerical Insights

A set of 48 $(\hat{u}; \hat{v}) \in \text{SU}(2)$ which preserves the orientation of a given cube is a *groupset*. We have shown that all elements of a groupset are mapped onto the same complex valued coordinates $(h_0; h_1; h_2) \in \mathbb{C}^3$ of the model surface (4.23). It is possible to do the reverse way: from any coordinate of the model surface,

the corresponding groupset can be identified. The groupset may be composed of nonunit quaternions; however, all the 48 quaternions have the same norm owing to the fact that the corresponding affine transformations do not alter the norm.

We are going to rewind the derivations, starting with h_0, h_1, h_2 and going through g_0, g_1, g_2 and \tilde{f}_0, \tilde{f}_1 in order to get u, v . Owing to (4.23), we get

$$\begin{aligned} g_1 &= h_0 \\ g_0 &= (-1)^{k_0} \sqrt{h_1} \\ g_2 &= \begin{cases} (-1)^{k_0} \frac{h_2}{\sqrt{h_1}} & , \text{ if } |h_1| \neq 0 \\ (-1)^{k_0} \sqrt{h_1^3} & , \text{ otherwise} \end{cases} \end{aligned}$$

with $k_0 = \{0, 1\}$. There are thus two possibilities. Then, we obtain with (4.21)

$$\begin{aligned} \tilde{f}_0 &= \exp\left(i \frac{2\pi}{3} k_1\right) \sqrt[3]{\frac{2g_2 + 12\sqrt{3}ig_0^2}{2}} \\ \tilde{f}_1 &= \begin{cases} \frac{g_1}{\tilde{f}_0} & , \text{ if } |\tilde{f}_0| \neq 0 \\ \exp\left(i \frac{2\pi}{3} k_1\right) \sqrt[3]{\frac{2g_2 - 12\sqrt{3}ig_0^2}{2}} & , \text{ otherwise} \end{cases} \end{aligned}$$

with $k_1 = \{0, 1, 2\}$. There are thus three possibilities. Knowing that $\tilde{f}_{0,1}(u; v) = u^4 \pm 4\sqrt{3}i(uv)^2 + v^4$ (respectively), we may identify

$$(uv)^2 = \frac{\tilde{f}_0 - \tilde{f}_1}{4\sqrt{3}i}$$

- $\tilde{f}_0 \neq \tilde{f}_1$

We can compute v^4 from the following quadratic polynomial

$$48(v^4)^2 - 24(\tilde{f}_0 + \tilde{f}_1)v^4 - (\tilde{f}_0 - \tilde{f}_1)^2 = 0$$

whose roots are

$$v = i^{k_3} \sqrt[4]{\frac{\tilde{f}_0}{4} + \frac{\tilde{f}_1}{4} + (-1)^{k_2} \frac{\sqrt{3}\sqrt{\tilde{f}_0^2 + \tilde{f}_0\tilde{f}_1 + \tilde{f}_1^2}}{6}}$$

with $k_2 = \{0, 1\}, k_3 = \{0, 1, 2, 3\}$. In that case, there are eight possibilities for v , which each gives

$$u = (-1)^{k_4} \sqrt{v}$$

with $k_4 = \{0, 1\}$. We end up with 96 possible $(u; v)$, which is twice what we expected. It is due to the choice of the sign of the square root in the

expression $g_0 = (-1)^{k_0} \sqrt{h_1}$. Indeed, in one case we choose the wrong sign for g_0 , which gives the wrong sign to g_2 . A wrong choice gives the groupset defined by $(h_0; h_1; -h_2)$, instead of $(h_0; h_1; h_2)$. A wrong choice may be easily mapped onto an element of $(h_0; h_1; h_2)$, by using the affine transformation (4.22). That latter transformation is performed in practice, since we only need one quaternion to rotate the cartesian frame onto the underlined 3D frame.

- $\tilde{f}_0 = \tilde{f}_1$

Otherwise if uv is zero, it means that $u = 0$ or $v = 0$, giving

$$\begin{cases} u &= k_2 i^{k_3} \sqrt[4]{\frac{\tilde{f}_0 + \tilde{f}_1}{2}} \\ v &= (1 - k_2) i^{k_3} \sqrt[4]{\frac{\tilde{f}_0 + \tilde{f}_1}{2}} \end{cases}$$

with $k_2 = \{0, 1\}, k_3 = \{0, 1, 2, 3\}$. There are then eight possibilities of u and v . We end up with 48 possible $(u; v)$. We do not encounter the latter issue, because if u or v is zero, $h_2 = 0$.

4.1 Euclidean distance \mathbb{C}^3

We analyze here the distance between 3D frames $\tilde{\mathbf{f}}$ and the cartesian frame $\hat{\mathbf{f}}$. The aim is to check if $(h_0; h_1; h_2) \in \mathbb{C}^3$ such that (4.23) may be used to define an euclidean distance between 3D frames.

First, we produce frames rotated around a single axis,

$$\tilde{\mathbf{f}} = \left[\cos\left(\frac{\alpha}{2}\right) + \mathbf{v} \sin\left(\frac{\alpha}{2}\right) \right] \hat{\mathbf{f}} \left[\cos\left(\frac{\alpha}{2}\right) - \mathbf{v} \sin\left(\frac{\alpha}{2}\right) \right]$$

with $\mathbf{v} \in \{\mathbf{i}; \mathbf{j}; \mathbf{k}\}$ and $\alpha \in [0; \frac{\pi}{2}]$. We then compute their distance to the cartesian frame

$$d_{\mathbb{C}^3}(\tilde{\mathbf{f}}; \hat{\mathbf{f}}) = \sqrt{dh_0 dh_0^* + dh_1 dh_1^* + dh_2 dh_2^*} \quad (4.26)$$

with $dh_i = \tilde{h}_i - \hat{h}_i$ the difference of the i -th components of the triplet (4.23). Fig. 4.7a shows that frames rotated around \mathbf{k} appear to be further than the ones rotated around \mathbf{i}, \mathbf{j} . Obviously, the former ones should be as far than the latter ones. (4.26) is not isotropic.

Let us compare (4.26) with the euclidean distance in \mathfrak{R}^4 , i.e. the shortest distance between any of 48 unit quaternions giving $\tilde{\mathbf{f}}$ from $\hat{\mathbf{f}}$ and the unit quaternion 1

$$d_{\mathfrak{R}^4}(\tilde{\mathbf{f}}; \hat{\mathbf{f}}) = \min_i \sqrt{(\hat{q}_{0(i)} - 1)^2 + \hat{q}_{1(i)}^2 + \hat{q}_{2(i)}^2 + \hat{q}_{3(i)}^2 + \hat{q}_{4(i)}^2}$$

This actually defines a consistent distance between frames. The frames $\tilde{\mathbf{f}}$ are produced from random unit quaternions \hat{q} . Fig. 4.7b shows that the distances do not correspond at all. Frames having triplet (4.23) close to the one of $\hat{\mathbf{f}}$ may be either close or far of the cartesian frame.

We may conclude (4.26) is not suitable to define a distance between frames. It means that averaging sets of values $(h_0; h_1; h_2)$ in \mathbb{C}^3 is inconvenient. Indeed, two sets that are close according to (4.26) could give a frame which is not the average of the two frames corresponding to the two sets.

4.2 Around an axis

We analyze the behavior of h_0, h_1, h_2 of 3D frames having one of their axes in common. We define this axis with $\mathbf{n} = (n_x; n_y; n_z) \in \mathfrak{R}^3$ such that $n_x^2 + n_y^2 + n_z^2 = 1$. Tanks to the automorphism of quaternions, we consider

$$\hat{q}_z = \cos\left(\frac{\alpha}{2}\right) + \sin\left(\frac{\alpha}{2}\right) \mathbf{k}$$

parameterizing rotations around \mathbf{k} of angle α . Afterwards, we rotate \mathbf{k} onto $\mathbf{n} = (n_x; n_y; n_z) \in \mathfrak{R}^3$ by means of $\hat{q}_{z \rightarrow n}$. We thus aim

$$\hat{q}_n = \hat{q}_{z \rightarrow n} \hat{q}_z$$

The unit quaternion $\hat{q}_{z \rightarrow n}$ may be described by a rotation of γ around a unit vector \mathbf{v} . It gives

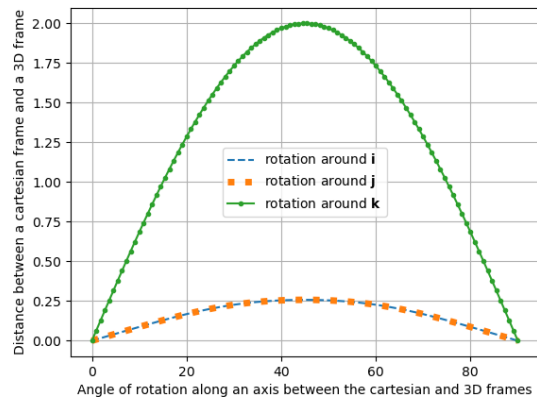
$$\begin{aligned} \cos(\gamma) &= \mathbf{k} \cdot \mathbf{n} = n_z \\ \sin(\gamma) &= |\mathbf{k} \times \mathbf{n}| = \sqrt{n_x^2 + n_y^2} = \sqrt{1 - n_z} \sqrt{1 + n_z} \\ \mathbf{v} &= \frac{\mathbf{k} \times \mathbf{n}}{|\mathbf{k} \times \mathbf{n}|} = \frac{(-n_y; n_x; 0)}{\sqrt{1 - n_z} \sqrt{1 + n_z}} \end{aligned}$$

Using trigonometric identities for \cos and \sin of $\frac{\gamma}{2}$, we get

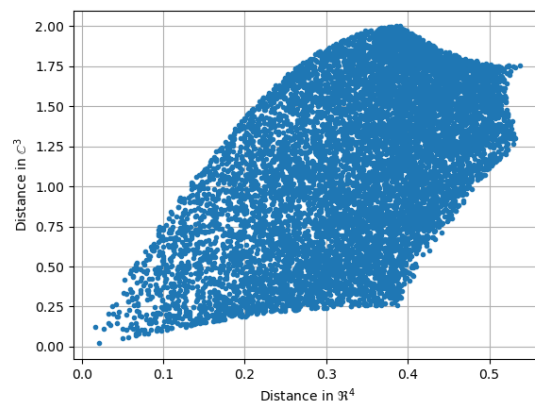
$$\hat{q}_{z \rightarrow n} = \frac{1}{\sqrt{2}} \left(\sqrt{1 + n_z} + \frac{1}{\sqrt{1 + n_z}} (-n_y \mathbf{i} + n_x \mathbf{j}) \right)$$

We then get

$$\begin{aligned} \hat{q}_n &= \frac{1}{\sqrt{2}} \left(\sqrt{1 + n_z} \cos\left(\frac{\alpha}{2}\right) + \frac{1}{\sqrt{1 + n_z}} [n_x \sin\left(\frac{\alpha}{2}\right) - n_y \cos\left(\frac{\alpha}{2}\right)] \mathbf{i} \right. \\ &\quad \left. + \frac{1}{\sqrt{1 + n_z}} [n_x \cos\left(\frac{\alpha}{2}\right) + n_y \sin\left(\frac{\alpha}{2}\right)] \mathbf{j} + \sqrt{1 + n_z} \sin\left(\frac{\alpha}{2}\right) \mathbf{k} \right) \end{aligned}$$



a. 3D frames are frame rotated around one axis.



b. Distances between 10^4 random 3D frames and the cartesian frame.

Figure 4.7: Distance between 3D frames and the cartesian frame. The distances are the euclidean ones of \mathbb{C}^3 (to compare $(h_0; h_1; h_2)$) and of \mathbb{R}^4 (to compare $(q_0; q_1; q_2; q_3)$.)

Owing to (4.25), we have the following SU(2) representation

$$\begin{aligned}\hat{u}_n &= \sqrt{\frac{1+n_z}{2}} \exp\left(-i\frac{\alpha}{2}\right) \\ \hat{v}_n &= \frac{n_x - in_y}{\sqrt{2(1+n_z)}} \exp\left(i\frac{\alpha}{2}\right)\end{aligned}$$

It then gives the invariant forms

$$\begin{aligned}h_{0[n]}(\alpha) &= \left(\frac{n_x - in_y}{2}\right)^4 \left(\left[\frac{n_x - in_y}{n_z + 1} \exp(i\alpha) \right]^4 + 14 + \left[\frac{n_z + 1}{n_x - in_y} \exp(-i\alpha) \right]^4 \right) \\ h_{1[n]}(\alpha) &= \left(\frac{n_x - in_y}{2}\right)^6 \left(\left[\frac{n_x - in_y}{n_z + 1} \exp(i\alpha) \right]^2 - \left[\frac{n_z + 1}{n_x - in_y} \exp(-i\alpha) \right]^2 \right)^2 \\ h_{2[n]}(\alpha) &= -\left(\frac{n_x - in_y}{2}\right)^9 \left(\left[\frac{n_x - in_y}{n_z + 1} \exp(i\alpha) \right]^2 - \left[\frac{n_z + 1}{n_x - in_y} \exp(-i\alpha) \right]^2 \right) \\ &\quad \left(\left[\frac{n_x - in_y}{n_z + 1} \exp(i\alpha) \right]^6 + \left[\frac{n_z + 1}{n_x - in_y} \exp(-i\alpha) \right]^6 \right) \\ &\quad - 33 \left\{ \left[\frac{n_x - in_y}{n_z + 1} \exp(i\alpha) \right]^3 + \left[\frac{n_z + 1}{n_x - in_y} \exp(-i\alpha) \right]^3 \right\}\end{aligned}\tag{4.27}$$

Let

$$\left. \begin{aligned}w(\alpha) &:= \frac{n_x - in_y}{n_z + 1} \exp(i\alpha) \\ a &:= \frac{n_x - in_y}{2}\end{aligned} \right\}$$

which simplifies the writing of (4.27)

$h_0(w)$	$= a^4(w^4 + w^{-4} + 14)$	(4.28)
$h_1(w)$	$= a^6(w^2 - w^{-2})^2$	
	$= a^6(w^4 + w^{-4} - 2)$	
$h_2(w)$	$= -a^9(w^2 - w^{-2})(w^6 + w^{-6} - 33(w^2 + w^{-2}))$	
	$= -a^9(w^8 - w^{-8} - 34(w^4 - w^{-4}))$	
	$= -a^9(w^4 - w^{-4})(w^4 + w^{-4} - 34)$	

We notice that

$$\begin{aligned}a^2 h_0(w) - h_1(w) &= 16a^6 \\ -a^3(w^4 - w^{-4})(h_1(w) - 32) &= h_2(w)\end{aligned}$$

giving a linear relationship between h_0 and h_1 parameterized by two first components of the axis \mathbf{n} , and updating the model surface equation.

Let us define

$$H_0 := \frac{h_0}{a^4}, \quad H_1 := \frac{h_1}{a^6}$$

and write

$$\frac{n_x - in_y}{n_z + 1} = \frac{1 - n_z}{1 + n_z} \exp(i\theta)$$

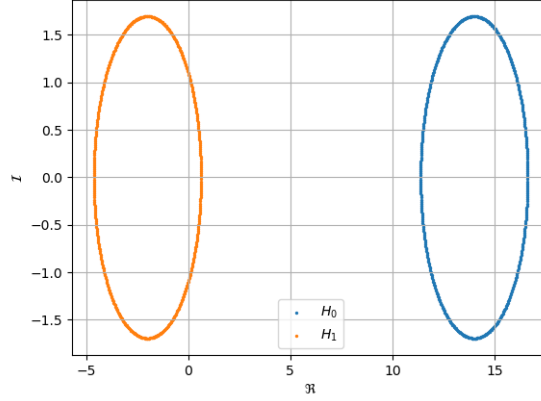


Figure 4.8: Scattering of H_0, H_1 in the complex plane \mathbb{C} for 500 three-dimensional frames rotated around an axis \mathbf{n} which corresponds to one of the axes of those frames.

in polar coordinates with $\theta = \arctan\left(\frac{-n_y}{n_x}\right)$. We then let $t = 4(\alpha + \theta)$ to obtain

$$\begin{aligned} H_0(t) &= p \exp(it) + q \exp(-it) + 14 \\ H_1(t) &= p \exp(it) + q \exp(-it) - 2 \end{aligned}$$

with $p = \frac{1}{q} = \left(\frac{1 - n_z}{1 + n_z}\right)^4$. We notice H_0, H_1 have expressions corresponding to ellipses in the complex plane, Fig. 4.8. It follows that h_0, h_1 are ellipses which are scaled and rotated. The corresponding relationships could be used to set boundary conditions on frames in order to align one of their axes with the normal of the boundary of a volume.

4.3 Ensuring $SU(2)$ from model surface

We are interested in unit quaternions. We know that a coordinate of the model surface provides a groupset whose quaternions have the same norm. We could compute a quaternion corresponding to a coordinate by using the above procedure. However, it is possible to derive a simpler way to check if a given coordinate $(h_0; h_1; h_2)$ laying on the model surface corresponds to $(\hat{u}; \hat{v}) \in SU(2)$.

We do not need to compute explicitly a quaternion. We just need to know the value of some powers of u and v . We remind that

$$|q|_{\mathbb{R}^4}^2 = |u|_{\mathbb{C}}^2 + |v|_{\mathbb{C}}^2$$

We are going to derive a way to compute those values u^n and v^m .

First, we notice that applying affine transformations (as right screws) from the octahedral group to $f_0(u; v) = (uv)^{24}$ produces⁵ the following subset

$$(uv)^2, -\frac{1}{4}(u^2 + v^2), \frac{1}{4}(u^2 - v^2)^2$$

The three elements of this subset may be written as the roots of polynomial of third order $(x - (uv)^2)(x + (u^2 + v^2)/4)(x - (u^2 - v^2)/4) = 0$. Scaling and expanding the latter expression allows to identify h_0 and h_1 as coefficients of the polynomial expression

$$16x^3 - h_0x + h_1 = 0 \quad (4.29)$$

Let $x^* := (uv)^2$ for an arbitrary root value of the above polynomial; choosing an other root value accounts of choosing an other quaternion of the same groupset.

We then compute either u^8 and v^8 as the two roots of the quadratic polynomial $(y - u^8)(y - v^8) = 0$. Expanding that latter polynomial in terms of h_0 and x^* , we get

$$y^2 - (h_0 - 14x^{*2})y + x^{*4} = 0 \quad (4.30)$$

Then the norm of the corresponding $(u; v)$ is given by

$$|(u; v)|^2 = |u^8|^{\frac{1}{4}} + |v^8|^{\frac{1}{4}}$$

Knowing that the discriminant of (4.30) is

$$\Delta^2 = h_0^2 - 28h_0x^{*2} + 192x^{*4}$$

we have the following norm

$$|(u; v)|^2 = \left| \frac{h_0 - 14x^{*2} + \Delta}{2} \right|^{\frac{1}{4}} + \left| \frac{h_0 - 14x^{*2} - \Delta}{2} \right|^{\frac{1}{4}}$$

which is independent of the choice of root for x^* and Δ , since modifying a choice corresponds to choosing an other quaternion of the same groupset.

⁴We remind that f_0 is an invariant form of the vierer group.

⁵ $f_0 \left(A_i \cdot \begin{bmatrix} u \\ v \end{bmatrix} \right) = \dots$

4.4 Attempted numerical schemes

As mentioned in §4.1, the use of the euclidean distance related to \mathbb{C}^3 for measuring the distance between frames from their triplet $(h_0; h_1; h_2)$ is inconsistent. Yet, we have tried different numerical schemes based on a finite element approach. Even if those have been unsuccessful due to this latter statement, we describe them.

The schemes are based on a tetrahedral mesh that discretizes the region R of interest. A Lagrange \mathcal{P}^1 approximation is built from that mesh.

4.4.1 Linear formulation

We consider that the frames laying on the boundary ∂R are given. In practice we compute the corresponding crossfield (Beaufort et al., 2017), and then identify a rotation sending the cartesian frame to the frame. From that rotation, we have a corresponding complex pair $(u; v) \in \mathbb{C}^2$. We eventually have $(h_0(u; v); h_1(u; v); h_2(u; v))$ all over the boundary ∂R .

We assume that h_i is function of $(x; y; z) \in R \subset \mathfrak{R}^3$, $\forall i$. We aim to get smooth values of h_i within R , which corresponds to minimize their Dirichlet energy

$$\min_{(h_0; h_1; h_2)} \int_R |\nabla h_0(x; y; z)|_{\mathbb{C}}^2 + |\nabla h_1(x; y; z)|_{\mathbb{C}}^2 + |\nabla h_2(x; y; z)|_{\mathbb{C}}^2 dx dy dz$$

with $|\bullet|_{\mathbb{C}}$ the usual complex norm. The weak finite formulation is then given by

$$\sum_{i=0}^2 \sum_j \int_R \nabla \phi_j \cdot \nabla \phi_k dx dy dz h_{i,j} = 0, \forall k$$

with $h_{i,j}$ the nodal value of h_i in node $(x_j; y_j; z_j)$.

We get a linear system with three complex unknowns by vertex (node). The solution is not projected onto the model surface.

4.4.2 Collocation method

We parameterize the rotations using the Euler angles (α, β, γ) respectively around $\mathbf{k}, \mathbf{j}, \mathbf{k}$, i.e. the following matrix belonging to $\text{SO}(3)$

$$\begin{bmatrix} -\sin(\alpha) \sin(\gamma) + \cos(\alpha) \cos(\beta) \cos(\gamma) & -\sin(\alpha) \cos(\gamma) - \sin(\gamma) \cos(\alpha) \cos(\beta) & \sin(\beta) \cos(\alpha) \\ \sin(\alpha) \cos(\beta) \cos(\gamma) + \sin(\gamma) \cos(\alpha) & -\sin(\alpha) \sin(\gamma) \cos(\beta) + \cos(\alpha) \cos(\gamma) & \sin(\alpha) \sin(\beta) \\ -\sin(\beta) \cos(\gamma) & \sin(\beta) \sin(\gamma) & \cos(\beta) \end{bmatrix} \quad (4.31)$$

whose the columns correspond to the 3 directions of a frame. We know that (4.31) is equivalent to two opposite quaternions,

$$\pm \left(\cos \left(\frac{\beta}{2} \right) \cos \left(\frac{\alpha + \gamma}{2} \right); \sin \left(\frac{\beta}{2} \right) \sin \left(\frac{\gamma - \alpha}{2} \right); \sin \left(\frac{\beta}{2} \right) \cos \left(\frac{\gamma - \alpha}{2} \right); \cos \left(\frac{\beta}{2} \right) \sin \left(\frac{\alpha + \gamma}{2} \right) \right)$$

From this quaternion, we use the relationships (4.25) and (4.23) in order to get the octahedral forms parameterized with $(\alpha; \beta; \gamma)$. Thanks to this parameterization, we may express the minimization of the Dirichlet energy based on the Euler angles

$$\min_{(\alpha_j; \beta_j; \gamma_j)} \sum_{i=0}^2 \sum_j \int_R |h_i(\alpha_j; \beta_j; \gamma_j) \nabla \phi_j(x; y; z)|_{\mathbb{C}}^2 dx dy dz \quad (4.32)$$

which is a nonlinear optimization problem.

We solve (4.32) using a Newton's method. The required derivatives are computed by means of the chain rule based on the Wirtinger calculus (Wirtinger, 1927). Owing to $|f|_{\mathbb{C}}^2 = f \cdot f^*$, we rewrite (4.32)

$$\min_{(\alpha_i; \beta_i; \gamma_i)} \sum_{i=0}^2 \int_R \left(\sum_j h_{i,j} \nabla \phi_j \right) \cdot \left(\sum_k h_{i,k}^* \nabla \phi_k \right) dx dy dz$$

with $h_{i,l} = h_i(\alpha_l; \beta_l; \gamma_l)$. The corresponding Euler-Lagrange equations are thus

$$\sum_{i=0}^2 \sum_j \int_R \left(\frac{\partial h_{i,l}}{\partial \bullet_l} h_{i,j}^* + h_{i,j} \frac{\partial h_{i,l}^*}{\partial \bullet_l} \right) \nabla \phi_l \cdot \nabla \phi_j dx dy dz = 0, \forall l$$

for $\bullet_l \in \{\alpha_l; \beta_l; \gamma_l\}$, with

$$\frac{\partial h_{i,l}}{\partial \bullet_l} = \frac{\partial h_{i,l}}{\partial u} \frac{\partial u}{\partial \bullet_l} + \frac{\partial h_{i,l}}{\partial v} \frac{\partial v}{\partial \bullet_l}$$

We eventually need the corresponding hessian⁶

$$\sum_{i=0}^2 \int_R \left(\frac{\partial h_{i,l}}{\partial \bullet_l} \frac{\partial h_{i,k}^*}{\partial \bullet_k} + \frac{\partial h_{i,k}}{\partial \bullet_k} \frac{\partial h_{i,l}^*}{\partial \bullet_l} + \sum_j \delta_{kl} \left(\frac{\partial^2 h_{i,l}}{\partial \bullet_l \partial \star_l} h_{i,j}^* + h_{i,j} \frac{\partial^2 h_{i,l}^*}{\partial \bullet_l \partial \star_l} \right) \right) \nabla \phi_l \cdot \nabla \phi_k dx dy dz$$

for $\bullet_l, \star_l \in \{\alpha_l; \beta_l; \gamma_l\}$ with

$$\frac{\partial h_{i,l}}{\partial \bullet_l \partial \star_l} = \frac{\partial^2 h_{i,l}}{\partial u^2} \frac{\partial u}{\partial \bullet_l} \frac{\partial u}{\partial \star_l} + \frac{\partial^2 h_{i,l}}{\partial v^2} \frac{\partial v}{\partial \bullet_l} \frac{\partial v}{\partial \star_l} + \frac{\partial h_{i,l}}{\partial u} \frac{\partial^2 u}{\partial \bullet \partial \star} + \frac{\partial h_{i,l}}{\partial v} \frac{\partial^2 v}{\partial \bullet \partial \star} + \frac{\partial^2 h_{i,l}}{\partial u \partial v} \left(\frac{\partial u}{\partial \bullet_l} \frac{\partial v}{\partial \star_l} + \frac{\partial v}{\partial \bullet_l} \frac{\partial u}{\partial \star_l} \right)$$

⁶for the entry (k, l)

The boundary conditions are set by imposing that each frame on ∂R shares a direction with the outward normal $\mathbf{n} = (n_x; n_y; n_z)$. We consider that the last column of (4.31) corresponds to \mathbf{n} , which implies

$$\alpha = \begin{cases} \arctan\left(\frac{n_y}{n_z}\right), & \text{if } n_x \neq 0 \neq n_y \\ \frac{\pi}{2}, & \text{if } n_x = 0, n_y \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\beta = \pm \arccos(n_z), \text{ the sign is determined with } n_x \text{ or } n_y$$

4.4.3 Metric

Fig. 4.7b shows that the euclidean distance between the invariant forms is irrelevant, compared to the euclidean distance between the quaternion (which is relevant). We aim to measure the variation among the quaternions δq from the variation of their invariant form δh . A linear approximation is to build a metric M . We start with the jacobian

$$J = \begin{bmatrix} \frac{\partial h_0}{\partial u} & \frac{\partial h_0}{\partial v} \\ \frac{\partial h_1}{\partial u} & \frac{\partial h_1}{\partial v} \\ \frac{\partial h_2}{\partial u} & \frac{\partial h_2}{\partial v} \end{bmatrix}$$

which measures the variation δh from δq . We thus need to compute a pseudo inverse J^+ of J . Then, we can build the aimed metric M

$$M = J^+ \cdot (J^+)^{\dagger}$$

where A^{\dagger} is the transposed conjugate (i.e. hermitian) of A .

Therefore, the Dirichlet energy corresponds to

$$\int_R \underline{\nabla h} \cdot M \cdot \underline{\nabla h}^{\dagger} \, dx dy dz \quad (4.33)$$

where $\underline{\nabla h} = [\nabla h_0 \quad \nabla h_1 \quad \nabla h_2]$.

We can evaluate (4.33) by means of a collocation method, as we have done in §4.4.2. The minimization is a bit trickier, since the metric M is implicit, i.e. we cannot express it analytically from $(u; v)$, neither from $(\alpha; \beta; \gamma)$. The derivatives have to be computed numerically.

However, (4.33) remains an approximation which is only valid for points $(h_0; h_1; h_2)$ close to each other (laying in the neighborhood of the same tangent space). The use of the metric is not sufficient to get a consistent distance of frames from their invariant forms.

5 | Conclusion

We essentially gave a new parameterization of 3D frame fields, involving only 3 complex values related by an implicit equation describing a variety. The $SU(2)$ parameterization is based on (Du Val, 1964), with a slight modification about the isomorphism between the special unitary group and the unit quaternions. We derived the invariant forms without involving homographies; we used the fact that a finite group of quaternions can be defined by three unit quaternions. We showed how to get the quaternions from a coordinate of the variety. Through numerical experiments, we noticed that the euclidean distance between 2 coordinates of the variety is not consistent with the distance between the corresponding 3D frames. We derived the relationship of components for 3D frames sharing an even direction: the two first ones describe an ellipsis in their respective complex plane. We showed how to ensure that a coordinate gives a unit quaternion. Finally, three attempted numerical schemes have been described; they do not compute properly a frame field because of the inconsistent use of the euclidean norm.

Acknowledgements

The present study was carried out in the framework of the project “Large Scale Simulation of Waves in Complex Media”, which is funded by the Communauté Française de Belgique under contract ARC WAVES 15/19-03. The authors thank Professor Jean-Pierre Tignol who has enlightened them about invariant theory. §4.3 is based on a Professor Tignol’s note.

CHAPTER



Conclusion

Mesh generation is an enthralling topic. Meshes are essential for computational methods and computer graphics, whose ends are quite different. The present thesis has been concerned to enable auxiliary tools for mesh generation in order to perform efficient and consistent numerical computations. Even if computational methods needed by the ARC WAVES project have still to be implemented, most of work related to mesh generation has been achieved. The initial methodology fulfilling the requirements of mesh generation for the WAVES project is almost completed.

The surface parameterization allows to use standard mesh generation for stereolithography triangulations. The mean value coordinates provide a robust and bijective parameterization of the underlined triangulations. The computation of the parameterization has been optimized about the boundary conditions by lowering the deformation due to holes within the the triangulation. The input triangulation may be refined in order to improve the quality of the parameterization.

The crossfield computation leverages the generation of quadrangular mesh over surfaces. The Ginzburg-Landau theory gives a complete understanding about the crossfield design, specially about singularities. A discrete scheme solving the Ginzburg-Landau equations has been validated on the disk, by comparing the theoretical/analytical position of singularities with the computed ones. The numerical method performs well, even if it creates singularities that are not required by the topology of the surface – like the torus. Anyway, the singularities are suitably located in that latter case and the crossfield provides a convenient polyquad decomposition. The finite element formulation allows to set weak boundary conditions, which is convenient for sharp geometries such as NACA airfoils.

Full hexahedral mesh generation remains a challenge among mesh community. The privileged technique relies on 3D frame field, either for frontal approaches or block structure decomposition. Frame field computation is ongoing research. The present thesis proposes a new way to parameterize frames by means of the special unitary group. The orientation of frames is described by quaternions, whose the corpus is one-to-one with the special unitary group. This results in three complex numbers corresponding to coordinates of a surface embedded in a three complex-valued dimensional space. Unfortunately, a consistent numerical method producing smooth frame field from this parameterization has not been achieved.

The parameterization pipeline based on mean value coordinates always returns a valid atlas to remesh poor quality stl (conforming) triangulation. The only drawback is the *a posteriori* assessment about the aspect ratio of the parameterization. Solving the linear equations related to the mean value coordinates is time-consuming. Avoiding unnecessary computations by checking *a priori* the aspect ratio of the parameterization would be time-saving. However, such an *a priori* criterion seems utopian for every kind of geometry. Those lost computations cost a small price for the guarantee that it works with every conforming triangulation.

Ginzburg-Landau theory predicts optimal crossfields in regards of the topology of the underlined surface. Complete information is provided about the singularities, i.e. their index and their location. Performances of the proposed numerical scheme are limited by the nonlinearity of the corresponding equations to solve. Yet, implementation of the scheme is quite basic. Investigations for a better scheme would be relevant, in order to quicken the computations.

Frame parameterization based on special unitary group yields the smallest¹ unique characterization of the attitude of a frame. Real spherical harmonics and fourth order tensor characterize the orientation of a frame with 9 real parameters, while the invariant theory provides only three complex parameters.

¹to the best knowledge of the author

Unfortunately, the related variety is equipped with a metric far from being euclidean. Common numerical methods computing frame field cannot then be used.

The three next sections give some perspectives. The remeshing process may be improved by computing better geometrical mesh from the mean value coordinates. The crossfield has to be scaled in order to produce a regular quadlayout. The generation of high quality three-dimensional meshes relies on variety optimization.

1 | Remeshing with Winslow Smoothing

A common drawback about remeshing from an atlas is the formation of seams along two maps. Besides, the one-to-oneness of the parameterization is not sufficient to draw valid triangles from a map. To overcome those downsides, the use of *Winslow smoothing* appears to be suitable for the optimization of the parametric mesh in order to produce the best geometrical mesh.

Winslow (1966) proposes to build numerically a structured mesh with triangles. To do so, he considers that a structured mesh is defined by two equipotentials $\chi(x; y)$ and $\psi(x; y)$ such that

$$\nabla^2 \chi = 0, \quad \nabla^2 \psi = 0 \quad (5.1)$$

where $\chi \equiv \text{constant}$ and $\psi \equiv \text{constant}$ define edges of a regular triangulation, Fig. 5.1a. He then computes (5.1) from parameter space $(\chi; \psi)$, which gives

$$\left. \begin{aligned} \alpha \frac{\partial^2 x}{\partial \chi^2} - 2\beta \frac{\partial^2 x}{\partial \chi \partial \psi} + \gamma \frac{\partial^2 x}{\partial \psi^2} &= 0 \\ \alpha \frac{\partial^2 y}{\partial \chi^2} - 2\beta \frac{\partial^2 y}{\partial \chi \partial \psi} + \gamma \frac{\partial^2 y}{\partial \psi^2} &= 0 \end{aligned} \right\} \quad (5.2)$$

where

$$\left\{ \begin{aligned} \alpha &= \left(\frac{\partial x}{\partial \psi} \right)^2 + \left(\frac{\partial y}{\partial \psi} \right)^2 \\ \beta &= \frac{\partial x}{\partial \chi} \frac{\partial x}{\partial \psi} + \frac{\partial y}{\partial \chi} \frac{\partial y}{\partial \psi} \\ \gamma &= \left(\frac{\partial x}{\partial \chi} \right)^2 + \left(\frac{\partial y}{\partial \chi} \right)^2 \end{aligned} \right.$$

Those equations (5.2) are better known as the Laplace-Beltrami equations.

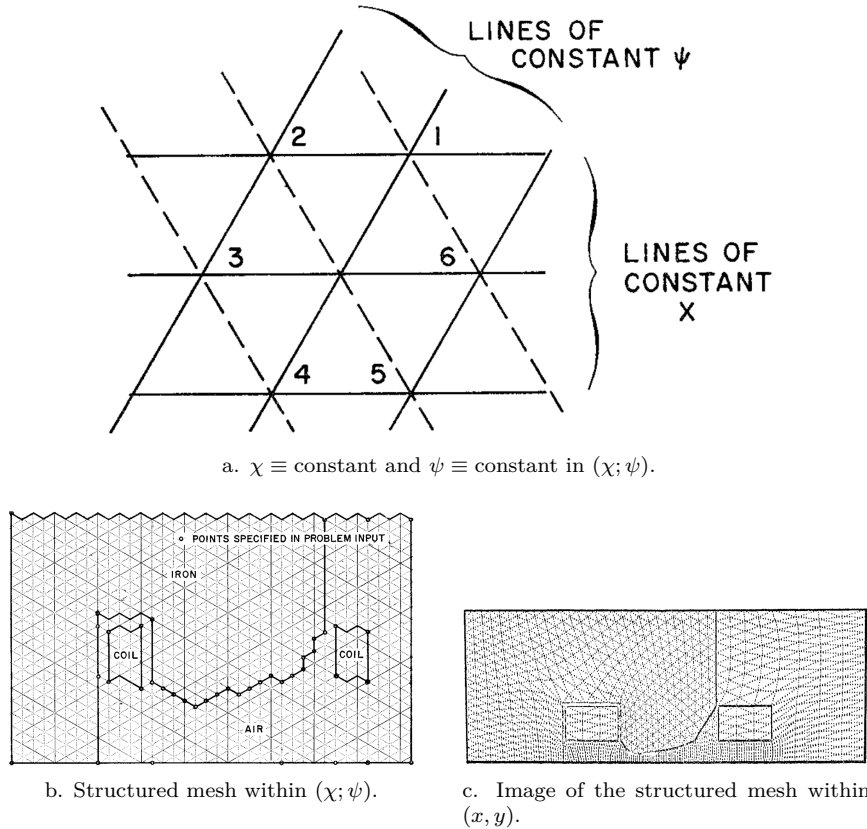


Figure 5.1: Mapping a parametric structured mesh onto a geometry. Figures from (Winslow, 1966, Fig. 12, 4 & 5).

Solving (5.2) allows to map a structured mesh from the parameter space (Fig.5.1b) onto the physical space (Fig.5.1c). The constant local topology of this mesh enables finite difference methods, which are used by Winslow to solve the quasilinear Poisson equation.

The Winslow smoothing has been applied to unstructured mesh by Fortunato and Persson (2016), in order to produce curvilinear triangles, Fig.5.2. They start with a straight sided mesh whose the parameterization fits exactly along a boundary. They then compute the curved mesh by solving the Laplace-Beltrami equations based on a harmonic parameterization

$$g^{ij} \partial_i \partial_j x_k = 0, \quad \text{for } k = 0, 1, 2 \quad (5.3)$$

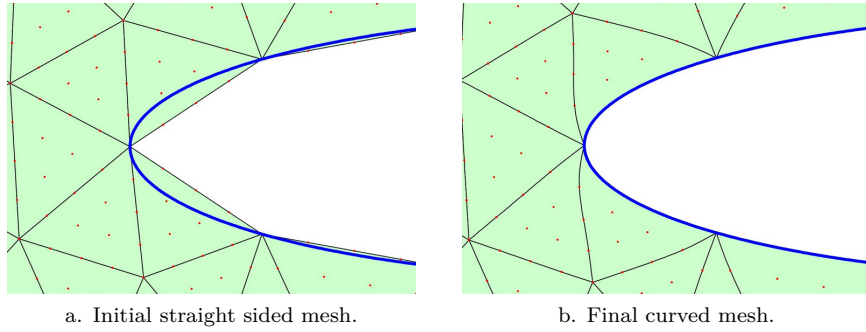


Figure 5.2: Curving adequately a straight sided mesh in order to fit to a geometry. Figures from (Fortunato and Persson, 2016, Fig. 2).

where Einstein convention is applied, g^{ij} is the inverse metric tensor and ∂_a is the partial derivative with respect to the a -th variable in the parameter space. Due to the nonlinearity of (5.3), they express this equation as a conservative second-order term with a first-order term

$$\begin{cases} \partial_i(g^{ij}) + \alpha^j &= 0 \\ \partial_i(g^{ij}\partial_j x_k) + \alpha^j \partial_j x_k &= 0 \end{cases}$$

for $j, k = 0, 1, 2$. They solve those latter equations by using a finite element approach.

This latter approach could be used to improve a parametric mesh, producing a geometrical mesh of high quality. Besides, the seams along maps could be smoothed out.

2 | Size Matters for Directions

Parameterization methods based on crossfields require the integrability of the crossfield. The integrability of the crossfield means that following one direction and then symmetrical one may be done in any order. In other words, the four vector fields corresponding to the different branches of a cross(field) can commute².

The commutativity between two vector fields u, v can be measured by the *Lie bracket*

$$[u; v] = u(v) - v(u) \tag{5.4}$$

²As long they do not surround any singularities.

where

$$v(u) = u^i \partial_i (v^j \partial_j)$$

measures the change of vector field u along the vector field v . Actually, the Lie bracket measures how much vector field flow curves fail to close. Those curves correspond to the isovalues of a parameterization. If the Lie Bracket is nonzero, the corresponding vector fields do not commute.

Lately, [Ježdmirovic et al. \(2019\)](#) have pointed this out. They claim that a unit crossfield can be hardly integrable, since

$$[u; v] = \nabla \theta$$

with

$$u = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}, \quad v = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix}$$

One way to get integrability is to scale the vector flow curves,

$$U = hu, \quad V = hv$$

giving

$$[U; V] = h^2 \nabla \theta - h(\nabla h \times \hat{n})$$

with \hat{n} the unit normal to the surface. Choosing appropriately h allows to get integrability by avoiding the Lie bracket

$$h \nabla \theta = \nabla h \times \hat{n} \tag{5.5}$$

In the case of a given unit crossfield (almost everywhere), the gradient of the angle may be easily computed from the complex expression of a vector field. Indeed, such a vector field may be expressed as

$$f = \exp(i 4\theta)$$

giving

$$\nabla f = i 4 \exp(i 4\theta) \nabla \theta = i 4f \nabla \theta$$

meaning that

$$\nabla \theta = -i \frac{\nabla f}{4f} \tag{5.6}$$

Combining (5.6) with (5.5), we may compute a scaling from a given unit crossfield, Fig.5.3. This is convenient for parameterization. The scaling is related to a sizefield.

However, the primary concern is here to provide high quality meshes for numerical simulation. It means that a given sizefield should constrain the

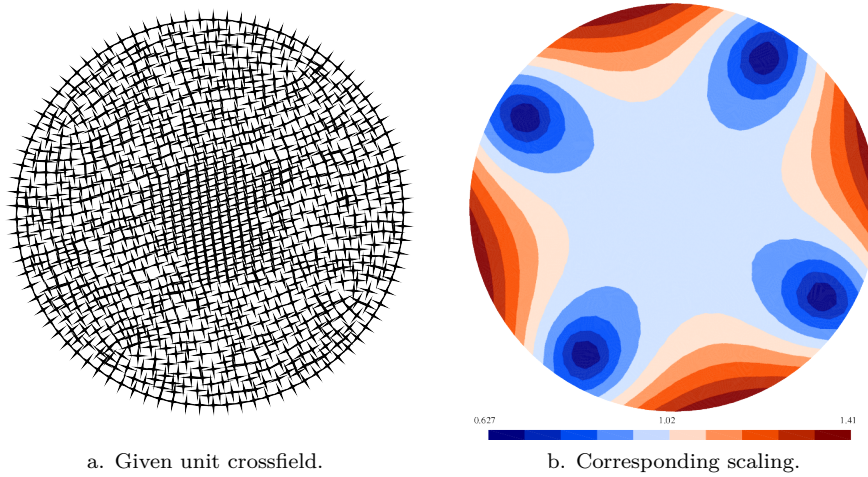


Figure 5.3: Scaling a given crossfield to get integrability.

crossfield, and not the opposite. Indeed, the sizefield is usually based on the needs of the numerical simulation, e.g. the required accuracy of the solution.

A given sizefield can hardly drive to a consistent crossfield from (5.5) and (5.6). It has to be projected on the closest sizefield corresponding to a crossfield producing the best structured mesh.

(Bethuel et al., 1994, §X.2, Eqn (13) & (14)) give Ginzburg-Landau equations by expressing the vector field with its norm r and angle θ . Owing to the fact that a vector may be expressed as a complex, one can rewrite (5.5)

$$H = \log(h) \implies \nabla\theta = -i\nabla H \quad (5.7)$$

which translates (Bethuel et al., 1994, §X.2, Eqn (13) & (14)) into

$$\left. \begin{aligned} \nabla \cdot (r^2 \nabla H) &= 0 \\ -\nabla^2 r + r |\nabla H|^2 &= \frac{1}{\epsilon^2} r (1 - r^2) \end{aligned} \right\} \quad (5.8)$$

Obviously, a singularity correspond to $r = 0$.

Those equations (5.8) could be used to compute the nearest H corresponding to a given sizefield, which yields a structured mesh. For example, a discrete scheme could use the first equation of (5.8) to compute H from r , and the second equation to compute r from H . As an insight of concept, r has been computed from a given H (Fig.5.3) by using the second equation of (5.8), Fig. 5.4. This could also drive crossfield computation for anisotropic structured meshes.

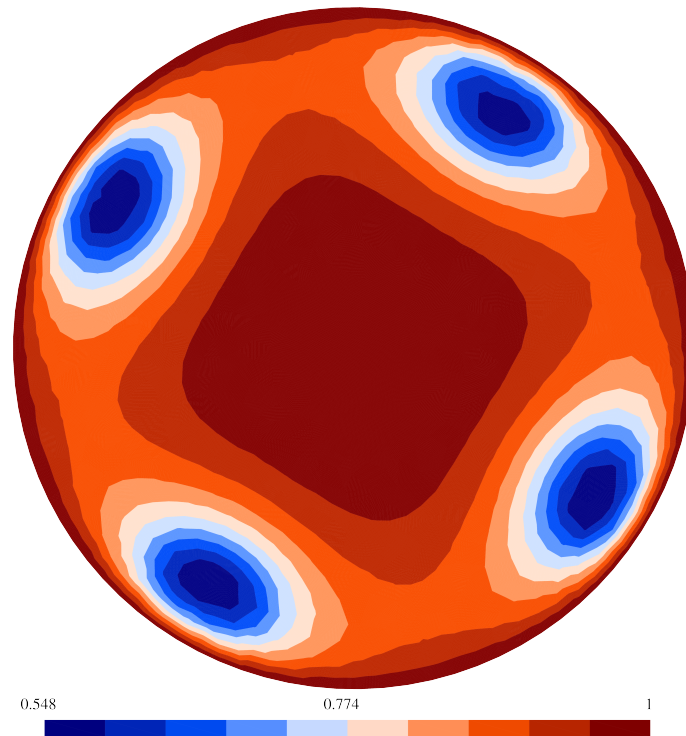


Figure 5.4: Norm of the vector field representing a crossfield, which is computed by using the second equation of (5.8) by using the H corresponding to Fig.5.3.

3 | Frames are about Variety Optimization

Future work should definitely aim yielding a discrete scheme based on the $SU(2)$ parameterization of 3D frames. This work has to focus on variety optimization, in order to efficiently compute geodesics between two coordinates of the model surface representing uniquely the orientation of frames.

Recently, [Palmer et al. \(2019\)](#) have presented a larger representation of frames, by allowing the scaling of three directions of frames. It appears to ease the computations, by quickening the convergence of the algorithm, Fig.5.5. Their variety is defined by 15 components, which are not yet given in their note. Their schemes consist in variety optimization. The most efficient seems to be a modified MBO (mMBO, MBO from [Merriman et al. \(1994\)](#)), which is a heat equation that is projected on the variety at some times. Concretely, it diffuses for a large time at first and then reduce the time between each projection. An

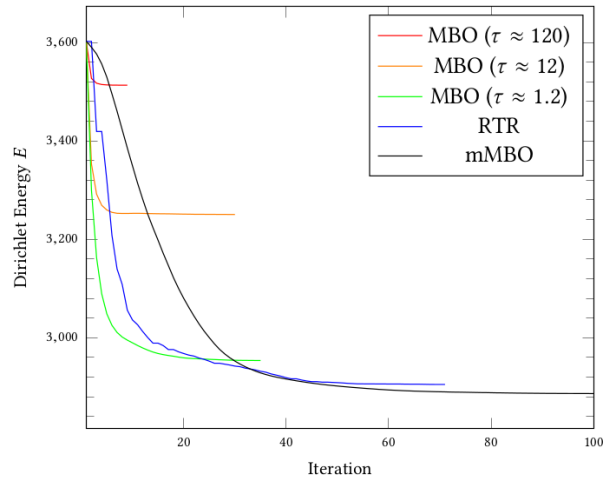


Figure 5.5: Convergence of the finite scheme based on the so-called odeco frames, which is compared to other approaches. Figure from (Palmer et al., 2019, Fig. 8)

analogous scheme should be derived for the $SU(2)$ parameterization of 3D frames.

Like asterisk fields for structured equilateral triangular meshes, there may be a way to provide a tool for structured tetrahedral mesh generation. The dodecahedral group seems consistent; Du Val (1964) gives the corresponding invariant forms and model surface. Yet, the corresponding directions should be scaled for two reasons. There is obviously the matter about integrability, but there is a larger issue. It is the fact it is not possible to fill space with regular (conforming) tetrahedra. But there may be a way to fill space with structured tetrahedra whose edges are scaled. It would be interesting to study the connections between the odeco frames and the $SU(2)$ parameterization, in order to derive odeco dodecahedra.

Bibliography

- Beaufort, P.A., Lambrechts, J., Henrotte, F., Geuzaine, C., Remacle, J.F., 2017. Computing cross fields A PDE approach based on the Ginzburg-Landau theory. *Procedia Eng.* 203, 219–231. doi:[10.1016/j.proeng.2017.09.799](https://doi.org/10.1016/j.proeng.2017.09.799).
- Béchet, E., Cuilliere, J.C., Trochu, F., 2002. Generation of a finite element mesh from stereolithography (stl) files. *Computer-Aided Design* 34, 1–17.
- Bennis, C., Vézien, J.M., Iglésias, G., 1991. Piecewise surface flattening for non-distorted texture mapping, in: *ACM SIGGRAPH computer graphics*, ACM. pp. 237–246.
- Bernard, P.E., Remacle, J.F., Kowalski, N., Geuzaine, C., 2016. Frame field smoothness-based approach for hex-dominant meshing. *Computer-Aided Design* 72, 78–86.
- Bethuel, F., Brezis, H., Hélein, F., et al., 1994. *Ginzburg-Landau Vortices*. volume 13. Springer.
- Bommes, D., Zimmer, H., Kobbelt, L., 2009. Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG)* 28, 77.
- Borouchaki, H., Laug, P., George, P.L., 2000. Parametric surface meshing using a combined advancing-front generalized delaunay approach. *International Journal for Numerical Methods in Engineering* 49, 233–259.
- Boubendir, Y., Antoine, X., Geuzaine, C., 2012. A quasi-optimal non-overlapping domain decomposition algorithm for the helmholtz equation. *Journal of Computational Physics* 231, 262–280.
- Campen, M., Bommes, D., Kobbelt, L., 2012. Dual loops meshing: quality quad layouts on manifolds. *ACM Transactions on Graphics (TOG)* 31, 110.
- Chemin, A., Henrotte, F., Remacle, J.F., Van Schaftingen, J., 2018. Representing three-dimensional cross fields using fourth order tensors, in: *International Meshing Roundtable*, Springer. pp. 89–108.

- Cohen, Y.T.P.A.D., Desbrun, S.M., 2006. Designing quadrangulations with discrete harmonic forms, in: Eurographics symposium on geometry processing, pp. 1–10.
- Constant, M., Dubois, F., Lambrechts, J., Legat, V., 2019. Implementation of an unresolved stabilised fem–dem model to solve immersed granular flows. *Computational Particle Mechanics* 6, 213–226.
- Crouzeix, M., Raviart, P.A., 1973. Conforming and nonconforming finite element methods for solving the stationary stokes equations i. *Revue française d’automatique informatique recherche opérationnelle. Mathématique* 7, 33–75.
- Delandmeter, P., Lambrechts, J., Legat, V., Vallaey, V., Naithani, J., Thiery, W., Remacle, J.F., Deleersnijder, E., 2018. A fully consistent and conservative vertically adaptive coordinate system for slim 3d v0. 4 with an application to the thermocline oscillations of lake tanganyika. *Geoscientific Model Development* 11, 1161–1179.
- Dirac, P., 1953. The lorentz transformation and absolute time. *Physica* 19, 888–896. doi:[10.1016/S0031-8914\(53\)80099-6](https://doi.org/10.1016/S0031-8914(53)80099-6).
- Dragnev, P., Legg, D., Townsend, D., 2002. On the separation of logarithmic points on the sphere, in: *Approximation Theory X: Abstract and Classical Analysis*, Vanderbilt University Press, Nashville, TN. pp. 137–144.
- Du Val, P., 1964. Homographies, quaternions and rotations .
- Duffin, R.J., 1959. Distributed and lumped networks. *Journal of Mathematics and Mechanics* , 793–826.
- Edelsbrunner, H., 2001. *Geometry and topology for mesh generation*. Cambridge University Press.
- Eppstein, D., 2009. Nineteen proofs of euler’s formula: $V - e + f = 2$. *Information and Computer Sciences*, University of California, Irvine .
- Ern, A., Guermond, J.L., 2013. *Theory and practice of finite elements*. volume 159. Springer Science & Business Media.
- Fekete, M., 1923. Über die verteilung der wurzeln bei gewissen algebraischen gleichungen mit ganzzahligen koeffizienten. *Mathematische Zeitschrift* 17, 228–249.
- Floater, M.S., 1998. Parametric tilings and scattered data approximation. *International Journal of Shape Modeling* 4, 165–182.

- Floater, M.S., 2003. Mean value coordinates. *Computer aided geometric design* 20, 19–27.
- Floater, M.S., Hormann, K., 2005. Surface parameterization: a tutorial and survey, in: *Advances in multiresolution for geometric modelling*. Springer, pp. 157–186.
- Fortunato, M., Persson, P.O., 2016. High-order unstructured curved mesh generation using the winslow equations. *Journal of Computational Physics* 307, 1–14.
- Freitag, L.A., Knupp, P.M., 2002. Tetrahedral mesh improvement via optimization of the element condition number. *International Journal for Numerical Methods in Engineering* 53, 1377–1391.
- Frey, P., 2001. Yams a fully automatic adaptive isotropic surface remeshing procedure .
- Georgiadis, C., Beaufort, P.A., Lambrechts, J., Remacle, J.F., 2017. High quality mesh generation using cross and asterisk fields: Application on coastal domain, *Research Notes, 26th International Meshing Roundtable, Sandia National Laboratories, September 18-21 2017*.
- Geuzaine, C., Remacle, J.F., 2009. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering* 79, 1309–1331.
- Gross, P.W., Kotiuga, P.R., 2004. *Electromagnetic Theory and Computation: A Topological Approach (Mathematical Sciences Research Institute Publications)*. Cambridge University Press.
- Hamann, B., 1993. Curvature approximation for triangulated surfaces, in: *Geometric modelling*. Springer, pp. 139–153.
- Huang, J., Tong, Y., Wei, H., Bao, H., 2011. Boundary aligned smooth 3d cross-frame field, in: *ACM transactions on graphics (TOG), ACM*. p. 143.
- Hughes, T., Cottrell, J., Bazilevs, Y., 2004. "consider a spherical cow"—conservation of geometry in analysis: Implications for computational methods in engineering, in: *IMA Hot Topics Workshop: Compatible Spatial Discretizations for Partial Differential Equations*.
- Ježdmirovic, J., Chemin, A., Beaufort, P., Remacle, J., 2017. Elliptic feket points obtained by ginzburg-landau pde, *Research Notes, 26th International Meshing Roundtable, Sandia National Laboratories, September 18-21 2017*.

- Jezdimirovic, J., Chemin, A., Remacle, J., 2019. Multi-block decomposition and meshing of 2d domain using ginzburg-landau pde, 28th International Meshing Roundtable, Sandia National Laboratories, October 15-17 2019.
- Johnen, A., 2016. Indirect quadrangular mesh generation and validation of curved finite elements. Ph.D. thesis. Université de Liège (Belgium), Montefiore Institute.
- Kälberer, F., Nieser, M., Polthier, K., 2007. Quadcover-surface parameterization using branched coverings, in: Computer graphics forum, Wiley Online Library. pp. 375–384.
- Karypis, G., Kumar, V., 2009. MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0. <http://www.cs.umn.edu/~metis>.
- Knöppel, F., Crane, K., Pinkall, U., Schröder, P., 2013. Globally optimal direction fields. *ACM Transactions on Graphics (ToG)* 32, 59.
- Kowalski, N., Ledoux, F., Frey, P., 2013. A pde based approach to multidomain partitioning and quadrilateral meshing, in: Proceedings of the 21st international meshing roundtable. Springer, pp. 137–154.
- Lai, Y.K., Jin, M., Xie, X., He, Y., Palacios, J., Zhang, E., Hu, S.M., Gu, X., 2009. Metric-driven rosy field design and remeshing. *IEEE Transactions on Visualization and Computer Graphics* 16, 95–108.
- Lawson, C.L., 1972. Generation of a triangular grid with applications to contour plotting. *Jet Propul. Lab. Techn. Memo* 299, 2.
- Lee, C., Lo, S., 1994. A new scheme for the generation of a graded quadrilateral mesh. *Computers & structures* 52, 847–857.
- Lévy, B., Petitjean, S., Ray, N., Maillot, J., 2002. Least squares conformal maps for automatic texture atlas generation, in: *ACM transactions on graphics (TOG)*, ACM. pp. 362–371.
- Li, Y., Liu, Y., Xu, W., Wang, W., Guo, B., 2012. All-hex meshing using singularity-restricted field. *ACM Transactions on Graphics (TOG)* 31, 177.
- Marchandise, E., de Wiart, C.C., Vos, W., Geuzaine, C., Remacle, J.F., 2011. High-quality surface remeshing using harmonic maps—part ii: Surfaces with high genus and of large aspect ratio. *International Journal for Numerical Methods in Engineering* 86, 1303–1321.
- Merriman, B., Bence, J.K., Osher, S.J., 1994. Motion of multiple junctions: A level set approach. *Journal of Computational Physics* 112, 334–363.

- Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J.K., Singh, S., et al., 2017. Sympy: symbolic computing in python. *PeerJ Computer Science* 3, e103.
- Nieser, M., Reitebuch, U., Polthier, K., 2011. Cubecover–parameterization of 3d volumes, in: *Computer graphics forum*, Wiley Online Library. pp. 1397–1406.
- Palacios, J., Zhang, E., 2007. Rotational symmetry field design on surfaces, in: *ACM Transactions on Graphics (TOG)*, ACM. p. 55.
- Palmer, D., Bommers, D., Solomon, J., 2019. Algebraic representations for volumetric frame fields. *arXiv preprint arXiv:1908.05411* .
- Pellerin, J., Johnen, A., Verhetsel, K., Remacle, J.F., 2018. Identifying combinations of tetrahedra into hexahedra: A vertex based strategy. *Computer-Aided Design* 105, 1–10.
- Ray, N., Li, W.C., Lévy, B., Sheffer, A., Alliez, P., 2006. Periodic global parameterization. *ACM Transactions on Graphics (TOG)* 25, 1460–1485.
- Ray, N., Sokolov, D., Lévy, B., 2016. Practical 3d frame field generation. *ACM Transactions on Graphics (TOG)* 35, 233.
- Ray, N., Vallet, B., Li, W.C., Lévy, B., 2008. N-symmetry direction field design. *ACM Transactions on Graphics (TOG)* 27, 10.
- Remacle, J., Geuzaine, C., 2019. Gmsh’s approach to robust mesh generation of surfaces with irregular parameterizations, 28th International Meshing Roundtable, Sandia National Laboratories, October 15-17 2019.
- Remacle, J.F., Geuzaine, C., Compere, G., Marchandise, E., 2010. High-quality surface remeshing using harmonic maps. *International Journal for Numerical Methods in Engineering* 83, 403–425.
- Remacle, J.F., Henrotte, F., Carrier-Baudouin, T., Béchet, E., Marchandise, E., Geuzaine, C., Mouton, T., 2013. A frontal delaunay quad mesh generator using the ∞ norm. *International Journal for Numerical Methods in Engineering* 94, 494–512.
- Remacle, J.F., Lambrechts, J., Seny, B., Marchandise, E., Johnen, A., Geuzaine, C., 2012. Blossom-quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. *International journal for numerical methods in engineering* 89, 1102–1119.
- Rivara, M.C., 1997. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *International journal for numerical methods in Engineering* 40, 3313–3324.

- Rusinkiewicz, S., 2004. Estimating curvatures and their derivatives on triangle meshes, in: Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004., IEEE. pp. 486–493.
- Saff, E.B., Kuijlaars, A.B., 1997. Distributing many points on a sphere. *The mathematical intelligencer* 19, 5–11.
- Sarrate Ramos, J., Ruiz-Gironés, E., Roca Navarro, F.J., 2014. Unstructured and semi-structured hexahedral mesh generation methods. *Computational Technology Reviews* 10, 35–64.
- Sawhney, R., Crane, K., 2018. Boundary first flattening. *ACM Transactions on Graphics (ToG)* 37, 5.
- Shepherd, J.F., Johnson, C.R., 2008. Hexahedral mesh generation constraints. *Engineering with Computers* 24, 195–213.
- Shewchuk, J.R., 2002. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley* 73, 137.
- Sitnik, R., Karaszewski, M., 2008. Optimized point cloud triangulation for 3d scanning systems. *Machine Graphics & Vision International Journal* 17, 349–371.
- Struik, D.J., 1961. *Lectures on classical differential geometry*. Courier Corporation.
- Turner, E., 2012. Watertight planar surface reconstruction of voxel data .
- Tutte, W.T., 1963. How to draw a graph. *Proceedings of the London Mathematical Society* 3, 743–767.
- Vaxman, A., Campen, M., Diamanti, O., Panozzo, D., Bommers, D., Hildebrandt, K., Ben-Chen, M., 2016. Directional field synthesis, design, and processing, in: *Computer Graphics Forum*, Wiley Online Library. pp. 545–572.
- Viertel, R., Osting, B., 2019. An approach to quad meshing based on harmonic cross-valued maps and the ginzburg–landau theory. *SIAM Journal on Scientific Computing* 41, A452–A479.
- Vincent, D., Karatekin, Ö., Vallaey, V., Hayes, A.G., Mastrogiuseppe, M., Notarnicola, C., Dehant, V., Deleersnijder, E., 2016. Numerical study of tides in ontario lacus, a hydrocarbon lake on the surface of the saturnian moon titan. *Ocean Dynamics* 66, 461–482.

-
- Wang, D., Hassan, O., Morgan, K., Weatherill, N., 2007. Enhanced remeshing from stl files with applications to surface grid generation. *Communications in Numerical Methods in Engineering* 23, 227–239.
- Winslow, A.M., 1966. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *Journal of computational physics* 1, 149–172.
- Wirtinger, W., 1927. Zur formalen theorie der funktionen von mehr komplexen veränderlichen. *Mathematische Annalen* 97, 357–375.
- Yamakawa, S., Shimada, K., 2003. Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells. *International journal for numerical methods in engineering* 57, 2099–2129.
- Zheng, Z., Wang, R., Gao, S., Liao, Y., Ding, M., 2018. Dual surface based approach to block decomposition of solid models. *Procedia engineering* 127, 149–167.



Discretizations of mappings

A.A | Derivation of the FEM scheme for harmonic mapping

Continuous harmonic maps minimize the *Dirichlet energy*

$$\int_{P_i} |\nabla \phi_i|^2 d\mathbf{x}$$

of the parametrization ϕ_i on the patch P_i . In other words, it minimizes the distortion between the patches and their planar representation.

It is possible to write a Laplacian as a finite difference scheme

$$\nabla^2 f|_i \approx \sum_{j \in J(i)} \lambda_{ij} (f_i - f_j)$$

It is a linear approximation of a Laplace operator at a vertex i . Indeed, the Laplace operator corresponds to the Euler-Lagrange equations derived from the Dirichlet energy

$$\int |\nabla f|^2 d\mathbf{x} \approx \int \left\| \sum_j f_j \nabla \phi_j \right\|^2 d\mathbf{x}$$

$$\frac{d}{df} \int |\nabla f|^2 d\mathbf{x} \Big|_i \approx 2 \int \sum_j f_j \nabla \phi_j \cdot \nabla \phi_i d\mathbf{x}$$

with ϕ_\bullet denoting the linear function shape associated to a node. On a triangle \mathcal{T}_{ijk} , knowing that $\phi_i + \phi_j + \phi_k = 1$ over \mathcal{T}_{ijk}

$$\int_{\mathcal{T}_{ijk}} f_i \nabla \phi_i \cdot \nabla \underbrace{(1 - \phi_j - \phi_k)}_{\phi_i} + f_j \nabla \phi_i \cdot \nabla \phi_j + f_k \nabla \phi_i \cdot \nabla \phi_k d\mathbf{x}$$

Rewriting last relation with terms $(f_i - f_j)$ and $(f_i - f_k)$, we obtain

$$\lambda_{ij} = - \int_{\mathcal{T}_{ijk}} \nabla \phi_i \cdot \nabla \phi_j d\mathbf{x}$$

which corresponds to the standard Galerkin finite element.

On triangle \mathcal{T}_{ijk} (Fig. 2.3)

$$\nabla \phi_i \cdot \nabla \phi_j = |\nabla \phi_i| |\nabla \phi_j| \cos(\pi - \theta_k)$$

where

$$|\nabla \phi_i| = \frac{1}{l_{ik} \sin(\theta_k)} \quad |\nabla \phi_j| = \frac{1}{l_{jk} \sin(\theta_k)}$$

with $l_{\bullet k}$ the length of edge $[\bullet k]$. Knowing that $|\mathcal{T}_{ijk}| = \frac{1}{2} l_{ik} l_{jk} \sin(\theta_k)$

$$- \int_{\mathcal{T}_{ijk}} \nabla \phi_i \cdot \nabla \phi_j d\mathbf{x} = \frac{1}{2} \frac{\cos(\theta_k)}{\sin(\theta_k)}$$

Adding the contribution of \mathcal{T}_{ilj}

$$\lambda_{ij}^{\text{FEM}} := \frac{1}{2} \left(\frac{\cos(\theta_k)}{\sin(\theta_k)} + \frac{\cos(\theta_l)}{\sin(\theta_l)} \right).$$

A.B | MVC difference scheme on a structured mesh is not a Laplacian

The MVC difference scheme relative to f_i corresponds to

$$\frac{\sqrt{2}}{h} (4f_i - (f_{i1} + f_{i2} + f_{i4} + f_{i5})) + \frac{2 - \sqrt{2}}{h} (2f_i - (f_{i3} + f_{i6})) = 0 \quad (\text{A.1})$$

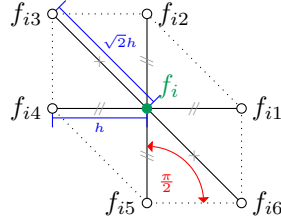


Figure A.1: Stencil within a structured mesh.

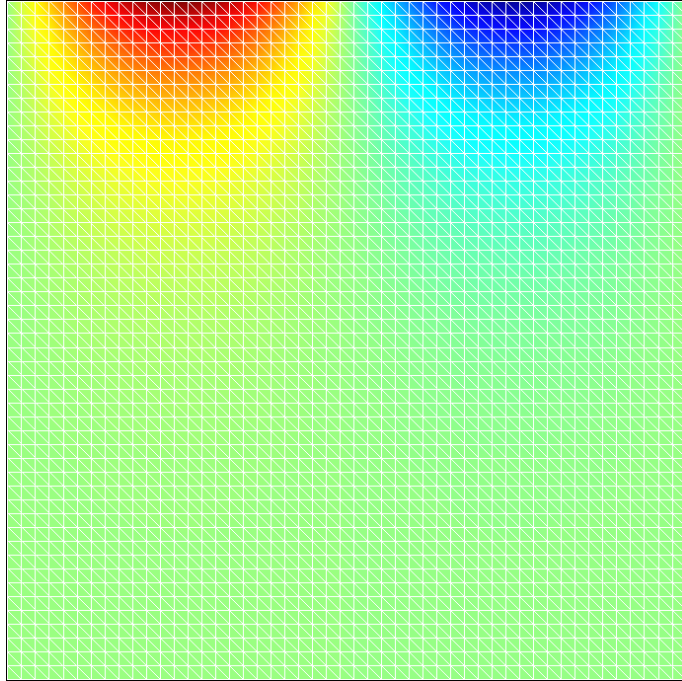
The first term of (A.1) (without the coefficient) corresponds to the well known linear combination of a centered finite difference to approximate a Laplacian. However, the second term does not approximate a continuous Laplacian. Indeed, the Taylor expansion of f_{i3} and f_{i6} is

$$\begin{aligned}
 f_{i3} &= f_i - \frac{\partial f}{\partial x} \Big|_i h + \frac{\partial f}{\partial y} \Big|_i h + \frac{\partial^2 f}{\partial x^2} \Big|_i h^2 + \frac{\partial^2 f}{\partial y^2} \Big|_i h^2 - \frac{\partial^2 f}{\partial x \partial y} \Big|_i h^2 + \text{hot} \\
 f_{i6} &= f_i + \frac{\partial f}{\partial x} \Big|_i h - \frac{\partial f}{\partial y} \Big|_i h + \frac{\partial^2 f}{\partial x^2} \Big|_i h^2 + \frac{\partial^2 f}{\partial y^2} \Big|_i h^2 - \frac{\partial^2 f}{\partial x \partial y} \Big|_i h^2 + \text{hot}
 \end{aligned}$$

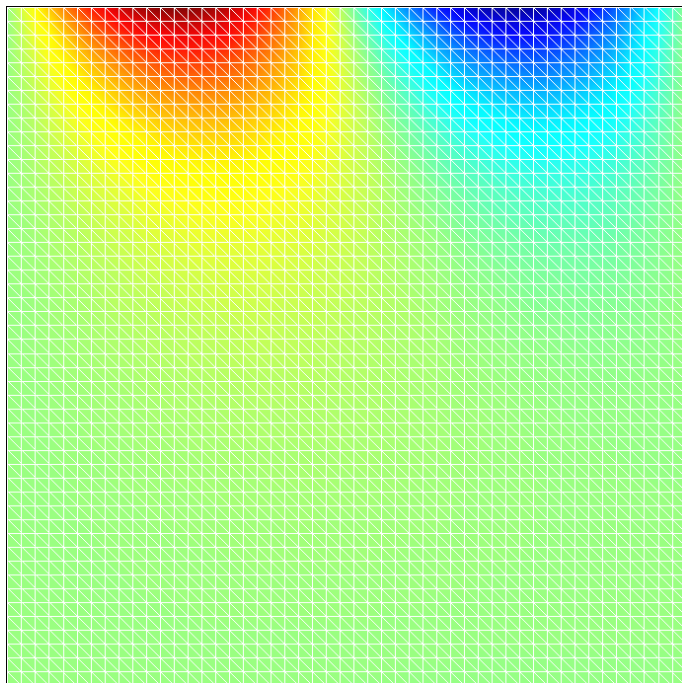
Hence,

$$f_{i3} + f_{i6} - 2f_i = \overbrace{2 \frac{\partial^2 f}{\partial x^2} \Big|_i h^2 + 2 \frac{\partial^2 f}{\partial y^2} \Big|_i h^2}^{2h^2 \nabla^2 f \Big|_i} - 2 \frac{\partial^2 f}{\partial x \partial y} \Big|_i h^2$$

Because of that last term, the MVC scheme on a structured mesh such as Fig. A.1 is not the approximation of a continuous Laplacian.



a. FEM computation.



b. MVC computation.

Figure A.2: Approximations of $\nabla^2 f = 0$ on a structured mesh over a square $[0; 1] \times [0; 1]$.

APPENDIX B

Computing renormalized energy on a unit disk

This appendix provides an analytical form of the renormalized energy $W(X^c)$ of equation (3.24) for a unit disk . We first compute Φ and then prove that second term of (3.23) is equal to zero in the case of a unit circle.

B.A | Solving the Neumann Problem i.e. Computing $\Phi(x)$ of Equation (3.24)

Assume a unit circle S . The analytical value of f on the boundary ∂S of S is

$$f = \exp(i d\theta) \text{ on } \partial S$$

as one direction has to be aligned with τ along the circle. The Neumann boundary condition is thus

$$\frac{\partial \Phi}{\partial \nu} = d \text{ on } \partial \mathcal{S} \quad (\text{B.1})$$

since $f \times \nabla f \cdot \tau = d$ on $\partial \mathcal{S}$. Indeed, from

$$a\bar{b} = a \cdot b - i a \times b, \forall a, b \in \mathbb{C}$$

and

$$\nabla f \cdot \tau = d \quad i f \nabla \theta \cdot \tau = i df$$

the condition (B.1) corresponds to the imaginary part of the corresponding complex product.

The Green function of the two-dimensional Laplacian operator is part of Φ

$$\Phi^0(\mathbf{x}) = \sum_{i=1}^d \log |\mathbf{x} - \mathbf{x}_i^c| \quad (\text{B.2})$$

Even if $\nabla^2 \Phi^0 = 2\pi \sum_{i=1}^d \delta(\mathbf{x} - \mathbf{x}_i^c)$, the flux (per unit of length) $\nabla \Phi^0 \cdot \nu$ does not correspond to (B.1). The solution Φ contains another term Φ^1 . It may also be written as a sum of the contributions coming from the d critical points. Therefore,

$$\Phi = \Phi^0 + \Phi^1 = \sum_{i=1}^d (\Phi_i^0 + \Phi_i^1)$$

such that

$$\begin{aligned} \Phi_i^0(\mathbf{x}) &= \log |\mathbf{x} - \mathbf{x}_i^c| \\ \nabla^2 \Phi_i^1 &= 0 \text{ in } \mathcal{S} \\ (\nabla \Phi_i^0 + \nabla \Phi_i^1) \cdot \nu &= 1 \text{ on } \partial \mathcal{S} \end{aligned} \quad (\text{B.3})$$

Function Φ_i^1 can be written as series of circular harmonics

$$\Phi_i^1(r; \theta) = A_{i,0} + \sum_{n=1}^{\infty} r^n [A_{i,n} \cos(n \theta) + B_{i,n} \sin(n \theta)]$$

where $(r; \theta)$ are polar coordinates. We search for the solution of a Neumann problem which is defined to a constant. Setting set $A_{i,0} = 0$ assigns to zero the average of Φ_i^1 . The idea is simple. We use Φ^1 which is harmonic to remove all oscillatory parts of $\nabla \Phi \cdot \nu$ along the boundary $\partial \mathcal{S}$.

Let us assume the i -th critical point is located on the x axis (the real axis), i.e. $\mathbf{x}_i^c = (r^c; 0)$ with cartesian coordinates (see Figure B.1). One has

$$|\mathbf{x} - \mathbf{x}_i^c|^2 = r^2 + r^{c2} - 2rr^c \cos(\theta)$$

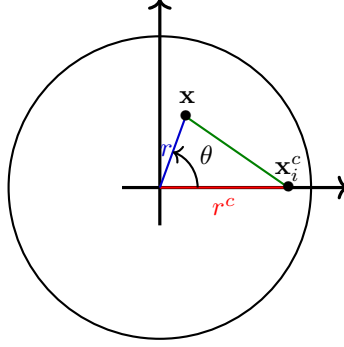


Figure B.1: Unit disk where the i -th critical point is depicted.

and

$$\frac{\partial \Phi_i^0}{\partial r} \Big|_{r=1} v = \frac{1}{|\mathbf{x} - \mathbf{x}_i^c|} \frac{\partial |\mathbf{x} - \mathbf{x}_i^c|}{\partial r} \Big|_{r=1} = \frac{1 - r^c \cos \theta}{1 + r^{c2} - 2r^c \cos \theta}.$$

The last expression can be reformulated as

$$\frac{(1 + r^{c2} - 2r^c \cos \theta) + 1 - r^{c2}}{2(1 + r^{c2} - 2r^c \cos \theta)} = \frac{1 - s \cos \theta + \frac{1-r^{c2}}{1+r^{c2}}}{2(1 - s \cos \theta)}$$

with $s = 2r^c / (1 + r^{c2})$. Taking into account the identity

$$\frac{1}{1 - x} = \sum_{n=0}^{\infty} x^n, \quad |x| < 1,$$

we have

$$\frac{\partial \Phi_i^0}{\partial r} \Big|_{r=1} = \frac{1}{1 + r^{c2}} \left(1 + \left[\sum_{n=1}^{\infty} s^n (\cos \theta)^n \right] \frac{1 - r^{c2}}{2} \right) \quad (\text{B.4})$$

and

$$\frac{\partial \Phi_i^1}{\partial r} \Big|_{r=1} = \sum_{n=1}^{\infty} n(A_{i,n} \cos(n\theta) + B_{i,n} \sin(n\theta)). \quad (\text{B.5})$$

Powers of $\cos(\theta)$ appear in (B.4). In order to replace such powers by $\cos(n\theta)$'s like in Equation (B.5), we use a well known property of Chebyshev polynomials: $P_n(\cos(\theta)) = \cos(n\theta)$. We thus have

$$\cos(m\theta) = \sum_{n=0}^m \mathcal{P}_{mn}(\cos \theta)^n \quad (\text{B.6})$$

where the \mathcal{P}_{mn} 's are the entries of the Chebyshev coefficient matrix \mathcal{P} . Equation (B.6) can thus be regarded as a system of equations

$$X_i = \mathcal{P}_{in} Y_n \quad , \quad X_i = \cos(i\theta) \quad , \quad Y_n = (\cos\theta)^n.$$

The system matrix \mathcal{P} is lower triangular, so the system can be inverted easily

$$Y_n = \mathcal{P}_{ni}^{-1} X_i,$$

or equivalently, back with the initial notation,

$$(\cos(\theta))^n = \sum_{i=0}^n \mathcal{P}_{ni}^{-1} (\cos(i\theta)).$$

Thus,

$$\sum_{n=1}^{\infty} s^n (\cos\theta)^n = \sum_{n=1}^{\infty} s^n \sum_{i=0}^n \mathcal{P}_{ni}^{-1} (\cos i\theta) = \sum_{n=0}^{\infty} w_n \cos n\theta$$

with

$$w_n = \sum_{i=n}^{\infty} s^i \mathcal{P}_{in}^{-1}.$$

Finally, we get the following series for the normal derivative of Φ_i^0 :

$$\left. \frac{\partial \Phi_i^0}{\partial r} \right|_{r=1} = \underbrace{\frac{1}{1+rc^2} \left(1 + w_0 \frac{1-rc^2}{2} \right)}_{W_0=1} + \sum_{n=1}^{\infty} \underbrace{\frac{1-rc^2}{2(1+rc^2)} w_n (\cos n\theta)}_{W_n}$$

We get the final condition

$$\left. \frac{\partial(\Phi_i^0 + \Phi_i^1)}{\partial r} \right|_{r=1} = W_0 + \sum_{n=1}^{\infty} [(W_n + nA_{i,n}) \cos(n\theta) + nB_{i,n} \sin(n\theta)]. \quad (\text{B.7})$$

The boundary condition should be non oscillatory: So,

$$B_{i,n} = 0 \quad \text{and} \quad A_{i,n} = -\frac{W_n}{n}.$$

Finally

$$\Phi(\mathbf{x}) = \sum_{i=1}^d \left[\log |\mathbf{x} - \mathbf{x}_i^c| + \sum_{n=1}^{\infty} r^n A_{i,n} \cos(n\theta) \right].$$

B.B | $\int_{\partial S} \Phi f \times f_\tau ds$ is Zero along a Circle

We want to show that

$$\int_{\partial S} \Phi f \times \nabla f \cdot \tau ds = 0$$

when ∂S is a unit circle.

In the previous section, we have shown that

$$f \times \nabla f \cdot \tau = d$$

Besides, Φ^1 has been derived such that it is non oscillatory along the unit circle. Hence, it remains to show

$$\sum_{k=1}^d \int_{\partial S} \log |\mathbf{x} - \mathbf{x}_k^c| ds = 0$$

We can express that integral with complex variables

$$\Re \left\{ \sum_{k=1}^d \oint_{|z|=1} \log(z - z_k^c) dz \right\} = 0$$

with the complex logarithm, which has two features:

- the complex logarithm is a multivalued function
- the complex logarithm has a peculiar singularity in zero

Those features are due to the fact that zero is a branch point. In our case, the branch points are the critical points z_k^c . A branch cut has to be drawn for each critical point. If $z_k^c = r^c \exp(i \gamma_k)$, the branch cut is such that $\theta \in [\gamma_k; \gamma_k + 2\pi)$, $z = r \exp(i \theta) \in \mathbb{C}$ (red line on Fig. B.2). The logarithm is analytical thanks to the branch cut, and its contour integral may be written as

$$\begin{aligned} \oint_{|z|=1} \log(z - z_k^c) dz &= \int_0^{\gamma_k^-} \log(\exp(i t) - r^c \exp(i \gamma_k)) i \exp(i t) dt \\ &+ \int_{\gamma_k^+}^{2\pi} \log(\exp(i t) - r^c \exp(i \gamma_k)) i \exp(i t) dt \end{aligned}$$

The antiderivative of $\log(w)$ is $w(\log(w) - 1)$, yielding the following

$$\begin{aligned} \oint_{|z|=1} \log(z - z_k^c) dz &= \left[(\exp(i t) - r^c \exp(i \gamma_k)) (\log(\exp(i t) - r^c \exp(i \gamma_k)) - 1) \right]_0^{\gamma_k^-} \\ &+ \left[(\exp(i t) - r^c \exp(i \gamma_k)) (\log(\exp(i t) - r^c \exp(i \gamma_k)) - 1) \right]_{\gamma_k^+}^{2\pi} \\ &= \exp(i \gamma_k)(1 - r^c) (\log(\exp(i \gamma_k^-)) - \log(\exp(i \gamma_k^+))) \\ &= 2\pi i \exp(i \gamma_k)(1 - r^c) \end{aligned}$$

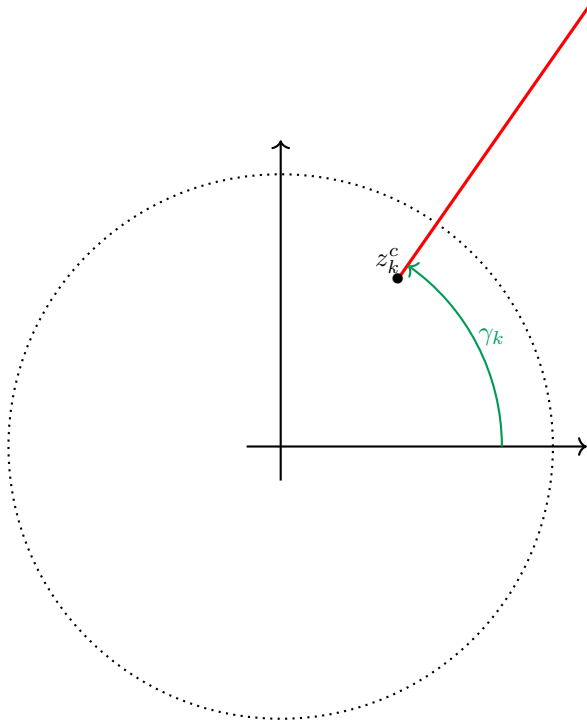


Figure B.2: Illustration of the k -th critical point defining a branch cut (in red) for the corresponding complex logarithm.

The last step is due to the fact that $\gamma_k^- - \gamma_k^+ = 2\pi$ (the branch cut, in other words).

$$\oint_{\mathcal{C}} \log(w) dw = 2\pi i \exp(i \gamma_k)(1 - r^c)$$

which depends on the branch cut.

The d critical points z_k^c being at the same distance r^c from the origin and evenly spaced by an angles $2\pi/d$, we get

$$\sum_{k=1}^d \oint_{|z|=1} \log(z - z_k^c) dz = 2\pi i(1 - r^c) \sum_{k=1}^d \exp(i [\gamma_1 + (k-1)2\pi/d])$$

which is zero since the sum of $d > 1$ complex numbers corresponding to points evenly distributed along on a circle centered at the origin is zero.