VI International Conference on Adaptive Modeling and Simulation
ADMOS 2013
J. P. Moitinho de Almeida, P. Díez, C. Tiago and N. Parés (Eds)

MULTI-BLOCK DECOMPOSITION USING CROSS-FIELDS

HAROLD J. FOGG, CECIL G. ARMSTRONG AND TREVOR T. ROBINSON

School of Mechanical and Aerospace Engineering,
Queen's University, Belfast BT9 5AH, N. Ireland.
e-mail: hfogg01@qub.ac.uk

Key words: Quadrilateral Mesh, Cross-Field, Mesh Singularities, Multi-block decomposition

Abstract. A method is presented for automatically generating cross-fields from direction constraints on arbitrary surfaces meshed with triangular elements. A cross-field describes the directionality of a quad mesh and the developed solver produces a cross-field abiding to prescribed target element sizes and directions. Additionally, a simple method of generating a multi-block decomposition suitable for all-quad meshing by tracing critical streamlines of the cross-field is explained and illustrated.

1 INTRODUCTION

1.1 Related work

Computer graphics visual effects methods such as anisotropic shading, morphing, texture synthesis, and global parametrisation of surfaces have been the driving force for the development of many vector field design methods [1, 2, 3, 4]. The vector field purpose is to control a bijective mapping onto the surface and the design objective is to minimise the angular distortion and/or stretching of the mapping. Vector fields with four-way symmetry have been found to be particularly useful for these applications. The concept has been described by many authors with different terminology, such as *cross-field* [5, 6], *4-way symmetry field (4-RoSy field)* [7, 8], and *frame field* [9]. Quad mesh generation is a somewhat equivalent problem where the mapping of an isotropic square mesh onto the surface is sought. In recent years cross-fields have begun to emerge as an important common feature to many new advanced mesh generation methods.

The QuadCover algorithm [9] produces a globally continuous parametrisation whose isoparametric curves describe a global closed quadrilateral mesh from a provided cross field. It was strongly inspired by Ray *et al.* [10] whose Periodic Global Parametrisation method produces comparable results by similar means. The global parametrisation method for generating quad meshes from input cross-fields has been used in other works [5, 11] with modifications. Progress has also been made on the 3-D equivalent problem [12, 13, 14].

Various approaches have been taken for the construction of smooth cross-fields on surfaces. The curvature orientated cross-field is a popular option and used in [15, 16, 9]. It produces appealing cross-fields coupled to inherent surface properties and a piece-wise linear approximation aligned with the principal directions is an optimal representation of the smooth underlying surface [17, 18] (as cited in [16]). However, where curvature tensors are symmetrical the principal directions are ill-defined which results in a discontinuous cross-field. Smoothing or optimisation schemes are needed to put order on the randomly orientated crosses in these regions.

For other approaches, a typical first step is to define a measure of the smoothness of a cross-field based on the angular deviations, variably called the smoothness energy, error or energy functional, and then set about minimising its integral over the surface area. Wei [2] and Turk [3] took a pragmatic approach and developed mixtures of relaxation and interpolation procedures to numerically converge to a vector field result.

Bunin's *continuum theory of unstructured mesh generation* [6] clarifies how singularities are the crucial characteristic features of cross-fields and quad meshes. A rigorous theory is presented that relates the size variation of an infinitesimal quad mesh to the directional variation of a cross-field. The scalar size field, acting as the continuum description of the mesh, is governed by a stationary heat equation where singularities appear as point sources and sinks. A quad mesh singularity corresponds to a node which is not attached to a regular array of quad elements, *e.g.* where three or five quad elements are attached to a given node in the interior of the mesh. Singularities necessarily occur from particular combinations of mesh alignment constraints or significant total Gaussian curvature of the surfaces. Once a valid arrangement of singularities has been identified, the scalar field can be easily solved numerically. In a follow-up paper [19], Bunin describes a method to tackle the *inverse Poisson* type problem of placing mesh singularities on planar surfaces. Ben-Chen *et al.* [20] proposed a method for identifying suitable locations for cone singularities of a conformally related metric to the surface by considering the Gaussian curvatures.

Palacios *et al.* [8], describe a design system in which cross-fields can be created and modified on surfaces from a set of prescribed singularities using interpolation and relaxation algorithms. A vector-based representation of an N-RoSy field is used which is globally continuous over planar surfaces but not for curved surfaces. A *parallel transport* scheme similar to that described by Zhang [4] is proposed to describe the field continuously over local regions.

The same vector-based representation of a cross-field is used in the recent work by Kowalski *et al.* [21] for decomposing planar surfaces for quad meshing. Boundary alignment constraints are applied to the cross-field and it is solved in the interior by a two stage algorithm: The first solves a stationary heat equation for the coordinates of the representation vector. The second normalises the vector solution of the previous step by an optimisation routine. The final cross-field is smooth with a small number of singularities appearing where the neighbourhood boundary alignment constraints are incompatible.

A method for constructing a smooth cross-field from a sparse set of directional constraints is described by Bommes *et al.* [5]. Direction constraints are extracted from the triangulated surface representation along curves where the estimated principal curvatures are significantly anisotropic. An involved optimisation algorithm called a *greedy mixed-integer* solver is designed for minimising the smoothness energy of a cross-field that is locally constant on each tri element. It iteratively solves for a smooth cross-field containing well-positioned singularities between the constrained directions.

With a similar discretised cross-field representation and smoothness energy functional, Liu *et al.* [11] use a non-linear optimisation method to solve for a cross-field from direction constraints along user-specified strokes. A properly initialised cross-field is imperative for the procedure. A simple method is used to propagate the cross-field from the crosses at specified elements to unspecified ones by iterating over an unordered queue of elements. Crosses for elements without specified crosses that neighbour elements with specified crosses are computed by a method mimicking a parallel-transport operation and then they are removed from the queue. This is repeated until the queue is empty. Liu comments that by using a randomly generated initialisation instead of the propagation based initialisation, an optimised cross-field contains many spurious features and singularities are effectively determined by the specifics of the arbitrary initialisation.

The problem of generating smooth *3-D cross-frame fields* for hexahedral mesh generation has been addressed in recent works. As for 2-D, the proposed 3-D methods involve using optimisation schemes to minimise the smoothness energy of a cross-frame field. Huang *et al.* [22, 13] proposed the use of a spherical harmonic functions to describe cross-frames which have convenient properties for measuring field smoothness and for specifying alignments to cross-frames. Before the non-linear optimisation solver is used to minimise the discretised smoothness energy integral, the system is initialised by solving a stationary heat type problem for the spherical harmonic coefficients. The fundamentals of the procedure are like those used by Kowalski [21] to calculate cross-fields in 2-D. Li *et al.* [14] describe an equivalent method to measure the smoothness energy of a cross-frame field. A boundary aligned cross-frame field is solved for by non-linear optimisation similarly. Their initialisation procedure is summarised as propagating the boundary frames into the interior of the tetrahedral mesh of the volume so that for any interior tet, its frame is assigned to be the same as that of its nearest boundary tet.

1.2 Contributions

The work presented in this paper makes the following contributions:

- A methodology is developed for generating smooth cross-fields on triangulated surfaces from an arbitrary number of alignment constraints using an efficient *fast-marching* algorithm to propagate the field into regions between constraints. It improves on the initialisation process described by Liu [11]. A similar fast marching algorithm is used by Lai [23] to parallel-transport a vector over a surface.

- The method is reasoned in the terms of Bunin's *continuum theory* and a smoothness energy functional is developed which is minimised locally in each propagation step. Additional terms for penalising deviations from target element sizes and directions are included in the functional so that the generated cross-fields have improved suitability with respect to prescribed size fields.

- A simple method is described for generating multi-block decompositions of surfaces by tracing critical streamlines of the cross-fields similar to that shown in [21].

2 THEORY

2.1 Continuum theory of unstructured mesh generation

In Bunin's theory [6] a conformal (angle-preserving) transformation from an arbitrary surface of interest, S, to a locally flat or developable surface, $\tilde{S}$, is searched for. A field of equally spaced parallel geodesics and their orthogonal trajectories represent a uniform mesh on $\tilde{S}$ and its image on S describes the quad mesh solution. The elements are ideally shaped squares on $\tilde{S}$ and as the spacing reduces to zero the elements are also squares on S. The key results of Bunin's paper are:

1. The scalar variable ϕ involved in the conformal factor between the metric tensors of S and $\tilde{S}$,
$$\tilde{g}_{ij} = e^{2\phi} g_{ij}\,, \quad \phi \in \mathbb{R}, \tag{1}$$
obeys the Poisson equation,
$$\Delta_{\mathcal{S}}\phi = K + \sum_{i=1}^{N} k_i \frac{\pi}{2} \delta_{P_i}\,, \quad k_i \in \mathbb{Z} \geq -4, \tag{2}$$
where $\Delta_{\mathcal{S}}$ is the Laplace-Beltrami operator, K is the Gaussian curvature of S and the δ_{P_i} terms are weighted delta Dirac functions describing discrete *cone* points of $\tilde{S}$ where a discrete total curvature of multiples of $\pi/2$ occurs at the cone vertex. The cone points correspond to singularities in the quad mesh with their characteristic types given by the integers k_i. From the metric scaling relationship (Eqn. (1)), it follows that
$$\phi = -\ln h, \tag{3}$$
where h is the element size.

2. The geodesic curvature, κ_g, of a mesh edge is directly related to local variation of the ϕ-field by
$$\frac{\partial \phi}{\partial e} = \kappa_g \equiv \frac{\partial \theta}{\partial t}. \tag{4}$$

The tangent vector $\mathbf{t}$, intrinsic normal $\mathbf{e}$, and surface normal $\mathbf{n}$ make a right hand orthonormal basis. The directional derivatives in the directions of $\mathbf{t}$ and $\mathbf{e}$ are represented by $\frac{\partial}{\partial t}$ and $\frac{\partial}{\partial e}$. θ is defined as the angle of the tangent vector from a reference direction on the tangent plane that is parallel transported over the surface (see [6, Section 2]). It follows that

$$\nabla\theta = R(-\tfrac{\pi}{2}\mathbf{n})\nabla\phi, \tag{5}$$

where $R(-\frac{\pi}{2}\mathbf{n})$ represents a $-\pi/2$ rotation about $\mathbf{n}$. Hence, $\nabla\theta$ is also governed by Eqn. (2).

3. For boundary conforming solutions, boundary conditions are applied to curved boundary edges and corners with angles not multiples of $\pi/2$. Neumann boundary conditions are applied to curved boundary edges according to Eqn. (4). Point sources with strengths dependent on the corner angle are applied at boundary corners.

These concepts are illustrated by the example shown in Figure 1.

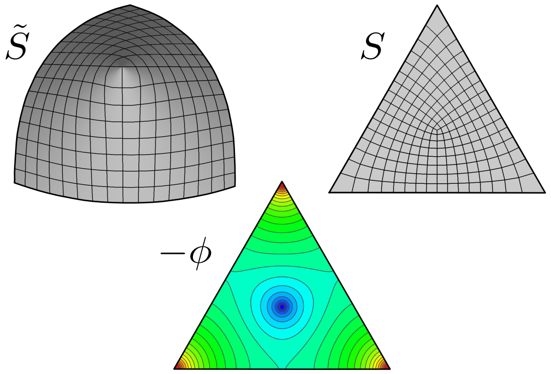

Figure 1: Example of Bunin's *continuum theory* for a flat triangular surface S. The conformally related surface $\tilde{S}$ is conical. The $-\phi$-field describing the logarithm of the element size on S is solved as a temperature field in a FEA heat conduction analysis.

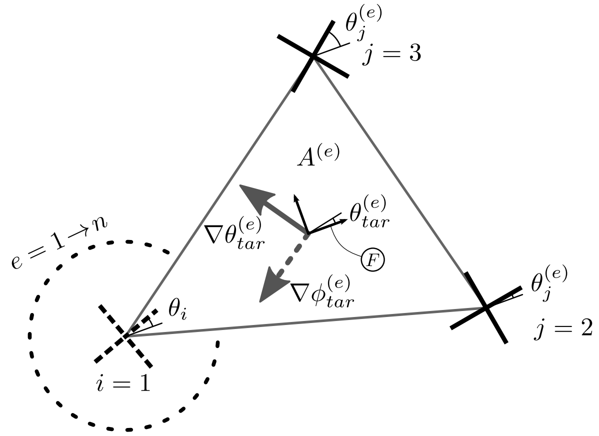

Figure 2: Optimising cross angle at node i adjacent to nodes j considering the target size gradients and orientations in the connected elements 1 to n.

2.2 Energy Functional

For the purpose of designing a cross-field solver, an energy functional is developed for minimisation. Rapid variation in cross-field directions, and hence element size variation, should be penalised. An obvious choice is to use the Dirichlet energy of the ϕ/θ-field,

$$E_{smoo} = \int \|\nabla\phi\|^2 dA = \int \|\nabla\theta\|^2 dA. \tag{6}$$

It is straightforward to prove that its value equals both the sum of the squared local geodesic curvatures of orthogonal mesh edges, and also the square of the norm of the element size gradient normalised by the local size. It is worth noting that that the Laplace equation (Eqn. (2) without source terms) is equivalent to minimising the Dirichlet energy in 2-D.

Given a target element size field for the mesh, a cross-field that accommodates those sizes will change direction according to the gradients of the prescribed size field as by Eqn. (4). Thus another penalty term to account for this is

$$E_{grad} = \int \|\nabla\theta - \nabla\theta_{tar}\|^2 dA, \tag{7}$$

where $\nabla\theta_{tar}$ is the target gradient of the local angle of the cross-field that can be computed from the target size field using Eqn. (3) and Eqn. (5). Similarly, a penalty term for the deviation of the field from prescribed directions is

$$E_{direc} = \int \frac{(\theta - \theta_{tar})^2}{A_{total}} dA. \tag{8}$$

(Note: The total area divisor is included so that the expression yields a dimensionless value)

The overall energy functional is the combination of the individual energy functionals:

$$E = E_{smoo} + w_1 E_{grad} + w_2 E_{direc}\,, \quad w_1, w_2 \in \mathbb{R} > 0, \tag{9}$$

where w_1 and w_2 are arbitrary weighting factors.

3 CROSS-FIELD GENERATION

3.1 Discretisation

A piece-wise linear description of a cross-field on a tri mesh is used in this work. This allows finite-element theory to be utilised to formulate the problem locally as a directly solvable linear system. On a tri element, e, an example of which can be shown in Figure 2, the angle of a cross relative to a local frame is approximated by a bilinear function, or equivalently by blends of the node angle values using linear shape functions:

$$\begin{aligned}\theta^{(e)}(x,y) &= \alpha_0^{(e)} + \alpha_1^{(e)} x + \alpha_2^{(e)} y, \\ &= \sum_{i=1}^{3} L_i^{(e)}(x,y)\theta_i^{(e)}.\end{aligned} \tag{10}$$

A target element size field can be discretised in the same way on the tri mesh.

Considering the problem of minimising E by adjusting the cross at a node i while keeping the crosses at surrounding nodes j fixed, of elements 1 to n, the solution satisfies the equation

$$\frac{\partial E}{\partial \theta_i} = 0$$

$$= \sum_{e=1}^{n} \frac{\partial}{\partial \theta_i} \left\| \nabla \theta^{(e)} \right\|^2 A^{(e)} + w_1 \sum_{e=1}^{n} \frac{\partial}{\partial \theta_i} \left\| \nabla \theta^{(e)} - \nabla \theta_{tar}^{(e)} \right\|^2 A^{(e)} + w_2 \sum_{e=1}^{n} \int_e \frac{(\theta^{(e)} - \theta_{tar}^{(e)})^2}{A_{total}} dA. \quad (11)$$

Using the local linear approximations of $\theta^{(e)}$ and $\theta_{tar}^{(e)}$ in Eqn (10), Eqn. (11) can be reduced and re-expressed with θ_i as a function of $L_i^{(e)}$, $L_j^{(e)}$, $A^{(e)}$, $\theta_j^{(e)}$, and $\theta_{tar}^{(e)}$. In this way a formula is constructed for calculating the optimum cross at a node.[1] Figure 2 shows the set-up for one of the n elements. In a propagation advancement step (Section 3.2) θ_i is optimised from the point of view of a single element. For a smoothing process all n elements would be taken into account.

3.1.1 Measuring angles

The local frame, F, is located at one of the nodes, call it N_f, with its x- and z-axes aligned with a cross direction and the local surface normal, $\mathbf{n}_f$. Angles θ_i and θ_j of crosses *cross*$_i$ and *cross*$_j$ are taken with respect to F, as shown in Figure 2. For an arbitrary node N_x, *cross*$_x$ on tangent plane E_x, is rotated onto tangent plane E_f to give *cross*$_x'$. This is effected by the rotation matrix $R(\mu \mathbf{u})$, where $\mu = \cos^{-1}(\mathbf{n}_x \cdot \mathbf{n}_f)$ and $\mathbf{u} = (\mathbf{n}_x \times \mathbf{n}_f) / |\mathbf{n}_x \times \mathbf{n}_f|$. This corresponds to a discrete parallel transport operation performed on a cross between the nodes (See [6, Section 2]). The rotation about $\mathbf{n}_f$ that moves *cross*$_x'$ onto *cross*$_f$ through the smallest angle (in the range $(-\pi/4, \pi/4]$) is used to describe the change in the cross-field orientation. It corresponds to the integral of the geodesic curvature of the cross-field.

3.2 Cross-field propagation

With the definition of the functional to be minimised and the discretisation of the problem, one route to generating the cross-field is to set-up an optimisation process *à la* Bommes [5] or Liu [11]. However computationally expensive non-linear solvers are used and an effective initialisation phase is required. Such an initialisation algorithm is proposed here. Although, the results show that the stand-alone generated cross-fields are of high quality without optimisation.

To produce boundary conforming meshes the cross-field is set as aligned with the tangent vectors of boundary edges. At a boundary corner, either the bisector direction or an offset direction of $\pi/4$ from the bisector is used. The decision is based on minimising the corresponding point source strength in Bunin's *continuum theory* and depends on the corner angle and the choices are fairly intuitive.

[1] The expression and its derivation are uncomplicated and are omitted here due to space limitations.

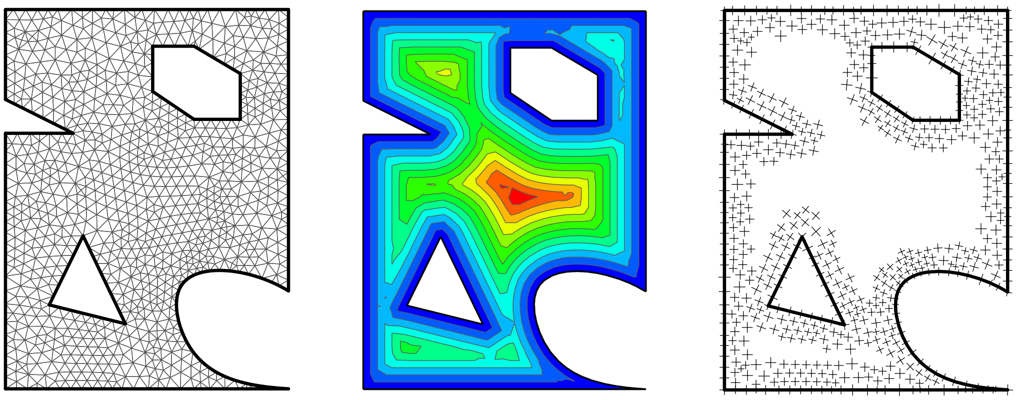

Figure 3: A 2-D test case surface with curved boundaries, sharp indents and hole features. (Left) Tri mesh on surface. (Centre) Distance field of Eikonal equation solved as part of the propagation algorithm. (Right) Propagation of cross-field from boundaries inwards.

The *fast marching alorithm*, introduced by Sethian [24], is used to propagate the cross-field from direction constraints in concordance with the distance field of the Eikonal equation. The algorithm developed in this work is similar to the finite-element based algorithm presented in [25]. A summary is given in Algorithms 1 and 2.

Algorithm 1 Cross-field propagation

```
▷ Initilisation
narrow_band_list $\leftarrow$ {}
alive_list $\leftarrow$ {}
for each Node, N do
    if N.cross $\neq$ None then
        N.d $\leftarrow$ 0
        N.alive $\leftarrow$ True
        alive_list.append(N)
    else
        N.alive $\leftarrow$ False
    end if
end for
for N in alive_list do
    N.method*(narrow_band_list)
end for
narrow_band_list.sort()        ▷ wrt d
▷ Remove first node and advance narrow band to include its neighbouring nodes
while narrow_band_list $\neq$ {} do
    trialN $\leftarrow$ narrow_band_list.pop(0)
    narrow_band_list.remove(0)
    trialN.alive $\leftarrow$ True
    trialN.method*(narrow_band_list)
    narrow_band_list.sort()        ▷ wrt d
end while
```

Algorithm 2 Node method*

```
function METHOD*(narrow_band_list)
    ▷ Description: For all neighbouring nodes, if not alive and not in narrow_band_list compute d and cross members and add to narrow_band_list.
    for N1 in Node.neighbour_Ns do
        dN2N3_list $\leftarrow$ {}
        for Element, E, in N1.neighbour_Es do
            {N2,N3} $\leftarrow$ E.nodes.remove(N1)
            if N2.alive and N3.alive then
                d $\leftarrow$ compute_d(N1.pos,N2.pos,N3.pos, N2.d2,N3.d3)
                dN2N3_list.append({d,N2,N3})
            end if
        end for
        if dN2N3_list={} then
            pass
        else
            {d',N2',N3'} $\leftarrow$ entry in dN2N3_list with smallest d entry
            N1.d $\leftarrow$ d'
            N1.cross $\leftarrow$ compute_cross (N1.pos, N2'.pos, N3'.pos,N2'.cross, N3'.cross)
            narrow_band_list.append(N1)
        end if
    end for
end function
```

The *compute_cross* function is based on the formula suggested in Section 3.1 and the *compute_d* function is described in Appendix A.1. The process is illustrated in Figure 3.

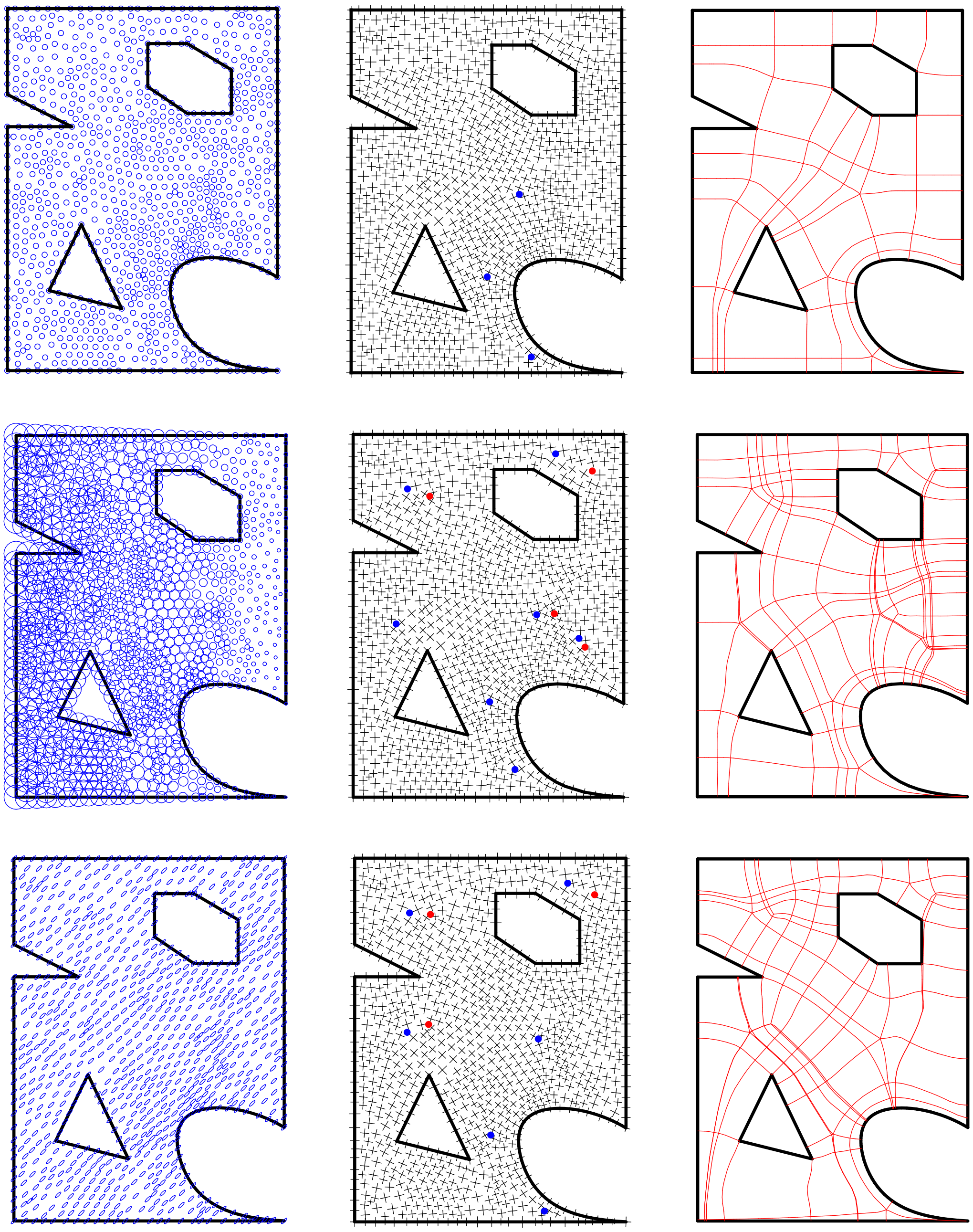

Figure 4: Examples of cross-fields and decompositions generated for three different target size fields. Blue and red dots indicate the occurrence of $k = 1$ and $k = -1$ type singularities respectively. (First row) The default unit target size field is used which results in a simple and aesthetically pleasing solution. (Second row) The target size field is isotropic and increases from right to left. The consequence is that edges tend to curve along paths running vertically which causes the occurrence of numerous extra singularities. (Third row) A constant anisotropic target element size field is used. The edges are to be aligned with the principal axes of the size tensor. Hence the E_{direc} term comes into play with a weighting factor proportional to the ratio of the principal axes lengths. Edges tend to travel diagonally across the surface which causes a different arrangement of singularities.

3.3 Cross-field singularities

Singularities occur in the cross-field in tri elements where the angle of a cross cannot be interpolated continuously between the crosses of its nodes.

From Bunin's *continuum theory*,

$$-k\frac{\pi}{2} = \sum_{i=0}^{n} \measuredangle\big(\mathrm{PT}_{p_i \to p_{i+1}}\mathbf{c}(p_i), \mathbf{c}(p_{i+1})\big) + \iint\limits_{\text{enclosed-area}} K dA, \tag{12}$$

where a closed-path is composed of segments between the points $p_0, p_1, ...p_n$ (p_n=p_0). PT is the parallel transport operator, $\mathbf{c}$ is a cross vector and $k \in \mathbb{Z}$ is the index of the enclosed singularity.

The total Gaussian curvature term can be calculated in the discretised representation of the surface from the three surface normals at the nodes using the standard angular deficit scheme (see *e.g.* [26]). Rotation angles for each edge in the tri mesh are calculated from the propagated cross-field by the method described in Section 3.1.1.

Hence, adding the rotation angles associated with each edge of a tri element with signs according to an anti-clockwise traversal and including the total curvature of the surface over the element (which is mostly negligible) gives a value equal to 0, $-\pi/2$ or $\pi/2$. These values correspond to cases of no singularity ($k = 0$), a positive singularity ($k = 1$) or a negative singularity ($k = -1$). It is certain that higher order singularities cannot occur because all edge rotation angles are in the range $(-\pi/4, \pi/4]$.

By virtue of the propagation method, singularities only occur in tri elements of inner-regions near the medial axis of the surface. For convex geometries with a high degree of symmetry, such as regular hexagons and circles, singularities occur in side-by-side tri elements. Possibly ideally they should be combined into a single higher order singularity. An additional routine would need to be incorporated for this task.

3.4 Tracing decomposition edges

With a cross assigned to every node and elements containing singularities identified, a piece-wise linear C^0 smooth cross-field can be defined. For a tri element not containing a singularity with nodes N_1, N_2 and N_3, a bilinear function describing the change in cross angle over the element with respect to $cross_1$ is determined. Referring to Eqn. (10), θ_1=0, $\theta_2 = \mathit{\Delta}\theta_{12}$ and $\theta_3 = \mathit{\Delta}\theta_{13}$ where $\mathit{\Delta}\theta_{12}$ and $\mathit{\Delta}\theta_{13}$ are the rotations stored for the respective edges, as discussed in Section 3.2. $\theta(x, y)$ is taken to mean the rotation about the element normal. However, $\theta_3 - \theta_2 \neq \mathit{\Delta}\theta_{23}(= -\mathit{\Delta}\theta_{12} + \mathit{\Delta}\theta_{13} + \iint K dA)$ if $\iint K dA \neq 0$. This complication is ignored – a simplification that is not prone to cause problems because the total Gaussian curvature over an element is small.

Singularity elements are divided into three new elements by edges running from a new node at the centroid to the corner nodes. For a new element with nodes N_1, N_2, N_c where N_c is the new node, the singularity is placed at N_c. Thus, θ is set as a function of polar coordinates r, φ with the origin at N_c so that $\theta(r, \vartheta_2) - \theta(r, \vartheta_1) = \mathit{\Delta}\theta_{12}$.

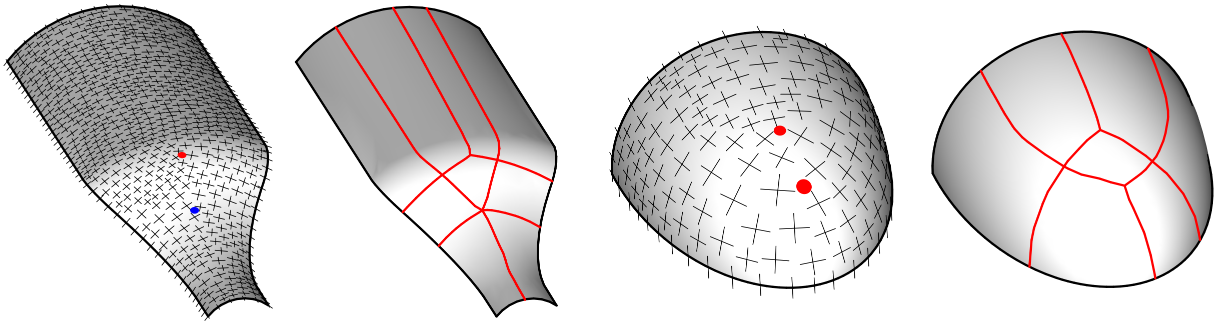

Figure 5: Cross-fields and decompositions generated on two curved surfaces with unit target size fields. In both cases the geodesic curvatures of the boundary edges are zero and the corner angles are $\pi/2$, therefore the singularities emerge as a consequence of the Gaussian curvatures of the surfaces.

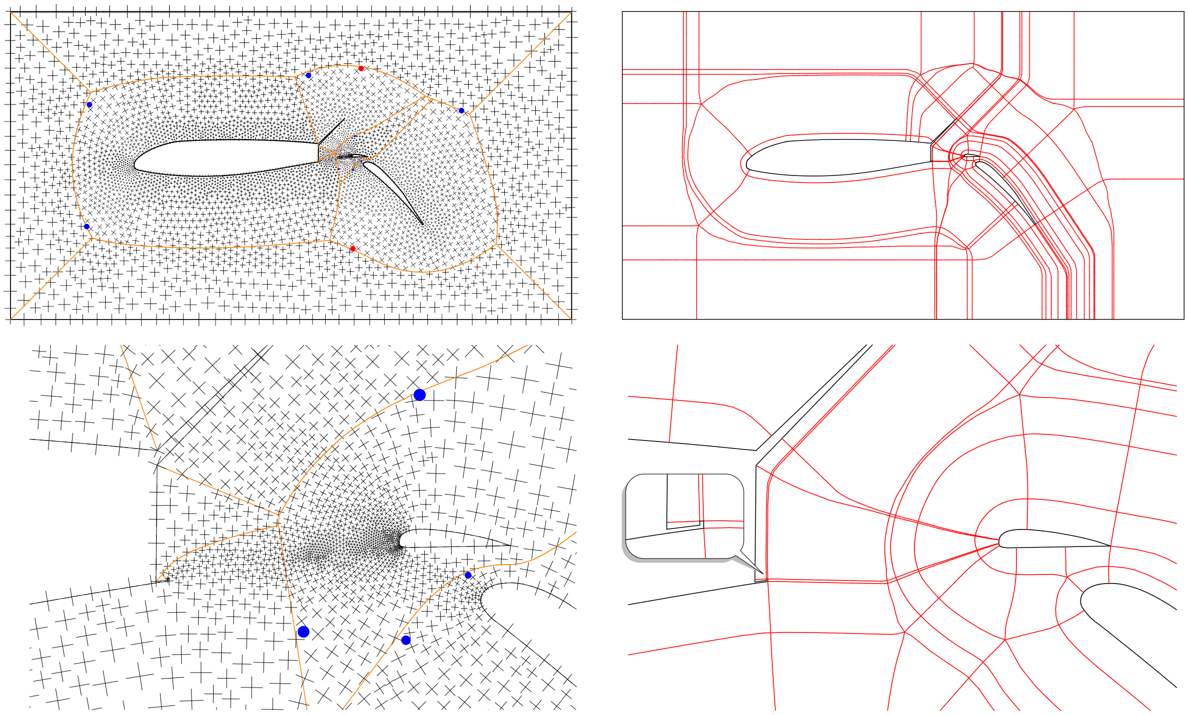

Figure 6: Multi-element aerofoil with flap and spoiler. (Left) Cross-field solution and close-up of the complex cove region. It is evident that the cross-field singularities occur on or near the medial axis represented by the orange lines. (Right) Intricate decomposition created by traced critical streamlines.

A simple numerical procedure is used for completing the task of tracing a cross-field streamline through an element given a curve tangent vector $\mathbf{d}_1$ and a position p_0 on element edge. First $\mathbf{d}_1$ is rotated onto the element plane to give $\mathbf{d}'_1$. Secondly, the reference vector is chosen from one of the four cross vectors at N_1 so that $\theta(p_0)$ best fits the angle that $\mathbf{d}'_1$ makes with the reference vector. Then Heun's method, a basic variation of the Runge-Kutta method, is used to integrate the streamline to another element edge according to

$\theta(x, y)$. The implemented method is an adaption of that described in [21, Section 3.2].

The critical streamlines radiating from singularities and boundary corners partition the surface into quadrilateral blocks, thus forming a multi-block decomposition suitable for all-quad meshing. The cross-field is evenly distorted around singularities and it is a straightforward matter to determine the starting *star geodesic* tangent vectors. The decomposition streamlines are traced until they meet the boundaries or until a pre-decided threshold distance or turn angle is exceeded.

Examples of cross-fields, singularities and multi-block decompositions are shown in Figures 4, 5 and 6.

4 DISCUSSION AND CONCLUSIONS

A novel method has been presented for generating a cross-field on a surface of arbitrary shape and genus with a provided tri mesh. Bunin's *continuum theory* is relied on as a basis for teasing out the best approach and for arguing the rationality of the approach taken. A set of specified direction constraints fix the crosses at certain nodes initially and the cross-field is propagated to the rest of the mesh by a fast marching method. The boundaries are selected automatically as the direction constraints unless alternatives are given, so that the result is boundary conforming. At each advancement step a new cross at a node is calculated by a simple formula derived by locally minimising an energy functional. The energy functional is designed to describe the composition of the cross-field smoothness and deviations from target element size changes and target directions over a region. Singularities of the propagated cross-field occur at isolated locations on the medial axis and are identified in elements of the tri mesh by a simple check.

The cross-field solver is the most important contribution of this work. The fast marching algorithm is efficient and solves the non-linear problem simply and quickly with an asymptotic complexity of $O(N \log N)$ [24]. For the example shown in Figure 6 with the tri mesh containing ∼6k nodes, the time taken for the entire process to finish was under a minute. The produced cross-fields can be tailored to suit a prescribed size field of a quad mesh. A potential application of the presented technology is as an effective cross-field initialisation method for global optimisation based solvers.

A basic streamline tracing algorithm is used to create the decomposition edges starting from singularities and boundary corners. Thus, multi-block decompositions of surfaces can be automatically constructed on which it is possible to generate all-quad meshes using widely-used algebraic mapping algorithms. Bunin's *continuum theory* deals only with the properties of a mesh with infinitesimal elements, hence it lacks guidelines for constructing a discrete mesh. For complex decompositions, such as that shown in Figure 6(right), long thin blocks are created by the simple streamline tracing algorithm. This does not cause an issue if the target element sizes are very small but a difficulty arises when the target sizes are greater than the block height. A post processing block simplification method could be used to overcome this problem by removing thin block rows with heights much smaller than the target element sizes, with care needed to avoid violating the block topology. Depending on requirements, perhaps recently developed global parametrisation based algorithms, such as [5, 11, 9], might be a preferable way to generate the quad mesh rather than by decomposition.

Acknowledgements

The work reported here is sponsored by Aircraft Research Association (ARA). The authors would like to acknowledge Transcendata for their support.

REFERENCES

[1] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 465–470, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[2] Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 355–360, New York, NY, USA, 2001. ACM.

[3] Greg Turk. Texture synthesis on surfaces. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 347–354, New York, NY, USA, 2001. ACM.

[4] Eugene Zhang, Konstantin Mischaikow, and Greg Turk. Vector field design on surfaces. *ACM Trans. Graph.*, 25(4):1294–1326, October 2006.

[5] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3):77:1–77:10, July 2009.

[6] Guy Bunin. A Continuum Theory for Unstructured Mesh Generation in Two Dimensions. *Comput. Aided Geom. Des.*, 25(1):14–40, January 2008.

[7] Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Lévy. Geometry Aware Direction Field Processing. *ACM Transactions on Graphics*, 29(1):Article 1, December 2009.

[8] Jonathan Palacios and Eugene Zhang. Rotational symmetry field design on surfaces. *ACM Trans. Graph.*, 26(3), July 2007.

[9] Felix Kälberer, Matthias Nieser, and Konrad Polthier. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum*, 26(3):375–384, 2007.

[10] Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25(4):1460–1485, October 2006.

[11] Yang Liu, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.*, 30(6):140:1–140:10, December 2011.

[12] M. Nieser, U. Reitebuch, and K. Polthier. Cubecover parameterization of 3d volumes. *Computer Graphics Forum*, 30(5):1397–1406, 2011.

[13] Jin Huang, Tengfei Jiang, Yuanzhen Wang, Yiying Tong, and Hujun Bao. Automatic frame field guided hexahedral mesh generation. Technical Report MSU-CSE-12-9, Department of Computer Science, Michigan State University, East Lansing, Michigan, August 2012.

[14] Yufei Li, Yang Liu, Weiwei Xu, Wenping Wang, and Baining Guo. All-hex meshing using singularity-restricted field. *ACM Trans. Graph.*, 31(6):177:1–177:11, November 2012.

[15] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3):485–493, July 2003.

[16] M. Marinov and L. Kobbelt. Direct anisotropic quad-dominant remeshing. In *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications*, pages 207 – 216, oct. 2004.

[17] R. Bruce Simpson. Anisotropic mesh transformations and optimal error control. *Applied Numer. Math*, 14:183–198, 1992.

[18] E. F. D'Azevedo. Are bilinear quadrilaterals better than linear triangles? *SIAM J. Sci. Comput.*, 22(1):198–217, January 2000.

[19] Guy Bunin. Towards Unstructured Mesh Generation Using the Inverse Poisson Problem. *arXiv/0802.2399*, 2008.

[20] Mirela Ben-Chen, Craig Gotsman, and Guy Bunin. Conformal Flattening by Curvature Prescription and Metric Scaling. *Computer Graphics Forum*, 27(2):449–458, 2008.

[21] Nicolas Kowalski, Franck Ledoux, and Pascal Frey. A PDE Based Approach to Multidomain Partitioning and Quadrilateral Meshing. In Xiangmin Jiao and Jean-Christophe Weill, editors, *Proceedings of the 21st International Meshing Roundtable*, pages 137–154. Springer Berlin Heidelberg, 2013.

[22] Jin Huang, Yiying Tong, Hongyu Wei, and Hujun Bao. Boundary aligned smooth 3d cross-frame field. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 143:1–143:8, New York, NY, USA, 2011. ACM.

[23] Yu-Kun Lai, Miao Jin, Xuexiang Xie, Ying He, Jonathan Palacios, Eugene Zhang, Shi-Min Hu, and Xianfeng Gu. Metric-driven rosy field design and remeshing. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):95–108, January 2010.

[24] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials science.* Cambridge University Press, 2 edition, 6 1999.

[25] Renato N. Elias, Marcos A. D. Martins, and Alvaro L. G. A. Coutinho. Simple finite element-based computation of distance functions in unstructured grids. *International Journal for Numerical Methods in Engineering*, 72(9):1095–1110, 2007.

[26] Zhiqiang Xu and Guoliang Xu. Discrete schemes for gaussian curvature and their convergence. *CoRR*, abs/0804.1046, 2008.

A Appendix

A.1 Calculate distance function in tri element

The Eikonal equation states

$$\|\nabla d\| = 1, \quad d \in \mathbb{R} > 0, \tag{13}$$

where d is the distance function. In a tri element with the distance set at two nodes, N_2 and N_3, the distance at the remaining node, N_1, is calculated as follows.
The distances relative to d_2 are:

$$\begin{aligned} u_1 &= d_1 - d_2, \\ u_2 &= 0, \\ u_3 &= d_3 - d_1. \end{aligned} \tag{14}$$

A local Cartesian coordinate frame, F, on the element plane with its origin at N_2 is used and u is approximated by a bilinear function,

$$u(x, y) = \alpha x + \beta y, \quad \alpha, \beta \in \mathbb{R}, \tag{15}$$

where F is chosen such that its x-axis is along edge-12 and the y-axis points into the element, so that $\beta > 0$. By Eqn. (13),

$$\begin{aligned} \|\nabla u\| &= \alpha^2 + \beta^2 = 1, \\ \Rightarrow \beta &= \sqrt{1 - \alpha^2}. \end{aligned} \tag{16}$$

Substituting known values in Eqn (15) gives

$$\begin{aligned} u_3 &= \alpha x_3 + \sqrt{1 - \alpha^2}\cancelto{0}{y_3,} \\ \Rightarrow \alpha &= \frac{u_3}{x_3} \end{aligned} \tag{17}$$

Therefore,

$$u_1 = \frac{u_3}{x_3} x_1 + \sqrt{1 - \frac{u_3}{x_3}^2}\, y_1, \tag{18}$$

and finally,

$$d_1 = d_2 + \frac{u_3}{x_3} x_1 + \sqrt{1 - \frac{u_3}{x_3}^2}\, y_1. \tag{19}$$