A TENSOR FORMULATION FOR INTEGRABLE FRAME FIELDS

Mattéo Couplet[1] Alexandre Chemin[1] Jean-François Remacle[1]

[1] *UCLouvain, Belgium. matteo.couplet@uclouvain.be*

ABSTRACT

In this note we propose a continuous criterion for the local integrability of 2D frame fields. Integrable frame fields form the gradients of a parametrization that is *de facto* valid for quadrilateral meshing. This is a first step towards integrable 3D frame fields which enable the robust generation of hexahedral meshes. We represent frames using orthogonally decomposable (Odeco) tensors, which encode the frame lengths and directions of a frame as the eigenvalues and eigenvectors of a fourth-order symmetric tensor in a way that is invariant to the frame symmetries. The integrability criterion is expressed solely in terms of the tensor coefficients and their spatial derivatives by studying the eigenvalue perturbation problem for Odeco tensors. We propose a simple projected gradient method to enforce this criterion on arbitrary geometries, and show that the algorithm converges to nearly-integrable frame fields.

Keywords: frame field, integrability, quad meshing, hex meshing

1. INTRODUCTION AND RELATED WORK

Meshes composed of quadrilaterals and hexahedra are known to offer superior computational performance, yet generating them on arbitrary geometries remains a challenge. A popular approach to quad/hex meshing is to first generate a *frame field*, i.e., assign to each point of the geometric domain a set of 2 or 3 orthogonal directions (a *frame*) representing the local orientation of the mesh elements. This frame field serves as a guide for computing a $(u, v)/(u, v, w)$-parametrization, which can then be used to build a quad/hex mesh. A desirable property of the frame field is *integrability*, i.e., that locally the 2/3 branches are the gradients of the parametrization's scalar fields.

Among notable works, we can cite [1], who first compute a unitary (non-scaled) frame field, then, despite being non-integrable, compute a parametrization of which the gradient is as closely aligned to the frame field as possible. Representing this parametrization requires transitions across cuts, hence the mixed-integer formulation. In a similar vein, [2] reduce the input frame field to a vector field on a branched covering, and use a Hodge decomposition to make this vector field integrable. Parametrizations obtained from existing frame field-driven methods require a quantization step to extract a quad mesh. Finding a valid quantization is in itself a complex combinatorial problem addressed notably in [3]; this problem even has no solution in some pathological cases. Another type of approach is to work with spatially fixed singularities. When no integrable isotropic frame field matching the singularities exists, it is possible to compute an integrable anisotropic frame field, as in [4].

If we turn to the 3D case, most unitary frame fields cannot be integrated to a parametrization anyhow, as the singularity structure does not match any hexahedral configuration [5]. Computing a topologically valid singularity graph *a priori* is almost impossible and is mostly done by hand. This consideration raises the need for a *one step* integrable frame field solver, which motivates our work.

In this note, we propose a continuous criterion for integrability of a 2D frame field. Frames are represented using the well-known *orthogonally decomposable (Odeco) tensor*, and the criterion is expressed solely in terms of the tensor coefficients and their spatial derivatives. We also propose a robust method for enforcing this criterion on arbitrary geometries, and show empirically that the algorithm converges to integrable 2D frame fields.

2. TENSOR FORMULATION

We lay out the mathematical object proposed by [6] that allows to encode scaled orthogonal frames in a way that is invariant to symmetries. Then we show how we express the 2D integrability condition in terms of the tensor coefficients, which is the main novelty presented in this note.

2.1 Odeco tensors

Let n be the frame dimension (2 or 3) and consider the frame defined by orthogonal unit vectors $\hat{\mathbf{u}}_1, \ldots \hat{\mathbf{u}}_n$ and corresponding lengths $\lambda_1, \ldots, \lambda_n$. We define the fourth-order symmetric tensor

$$\mathbf{T} = \sum_{i=1}^{n} \lambda_i \hat{\mathbf{u}}_i^{\otimes 4}. \tag{1}$$

Such a tensor is said to be *orthogonally decomposable* (Odeco) when the vectors are mutually orthogonal. One can notice that this tensor is indeed invariant to permutations and inversions of the branches. It has n^4 components of which $N = \binom{n+4-1}{4}$ are independent (5 in two dimensions, 15 in three dimensions). A common way to represent the tensor compactly is to first define the tensor polynomial

$$p_T(x_1, \ldots, x_n) = T_{ijkl} x_i x_j x_k x_l$$

(in the sequel, all summations are implicit over repeated indices), then decompose this polynomial in the basis of circular (2D) or spherical (3D) harmonics $Y_1, \ldots, Y_N$:

$$p_T(\mathbf{x}) = \sum_j q_j Y_j(\mathbf{x}). \tag{2}$$

As shown by [6], Odeco tensors form an algebraic variety $\mathcal{F}$ defined by a set of homogeneous quadratic equations,

$$q^{\mathsf{T}} A_i \, q = 0, \tag{3}$$

where $q \in \mathbb{R}^N$ encodes the tensor as in (2), and A_i is a set of symmetric matrices (there are 3 equations in 2D, and 27 in 3D).

An arbitrary tensor can be projected on the Odeco variety using the algebraic projector of [6]. However, for our purposes this projector is computationally prohibitive. We thus propose an approximate "geometric" projection which is much faster to compute. First the tensor is decomposed into its second-order eigentensors $\mathbf{D}_i \in \mathbb{R}^{n \times n}$:

$$\mathbf{T} = \sum_{i=1}^{m} \tilde{\lambda}_i \mathbf{D}_i^{\otimes 2}, \tag{4}$$

with $m = \binom{n+2-1}{2}$ (3 in 2D, 6 in 3D). This decomposition can be reduced to a standard matrix eigendecomposition problem by writing the 2nd- and 4th order tensors in Mandel notation. If $\mathbf{T}$ is near-Odeco, the n greatest eigenvalues $\tilde{\lambda}_i$ are dominant. The frame directions are then found by truncating the $m-n$ smallest eigenvalues and computing the (matrix) eigendecomposition

$$\mathbf{D} \coloneqq \sum_{i=1}^{n} \tilde{\lambda}_i \mathbf{D}_i = \sum_{i=1}^{n} \lambda_i \hat{\mathbf{u}}_i^{\otimes 2}. \tag{5}$$

If we require an *isotropic* frame (branches of equal lengths), we simply take the average of the computed lengths. Note that this approach recovers the frame exactly if $\mathbf{T}$ is exactly Odeco. Otherwise, it serves as a cheap and good enough projection for our algorithms.

2.2 Integrability condition

Let $\mathbf{u}_i = \lambda_i \hat{\mathbf{u}}_i$ denote the scaled frame branches. An orthogonal frame field is *integrable* if all pairs of frame branches have zero *Lie bracket*, i.e.,

$$[\mathbf{u}_i, \mathbf{u}_j] = \boldsymbol{\nabla}_{\mathbf{u}_j} \mathbf{u}_i - \boldsymbol{\nabla}_{\mathbf{u}_i} \mathbf{u}_j = \mathbf{0}, \quad i \neq j.$$

This equation involves the spatial derivatives of the frame branches, which can be expressed with the derivatives of the tensor coefficients. Written compactly,

$$\boldsymbol{\nabla} \mathbf{u} = \frac{\partial \mathbf{u}}{\partial T_{ijkl}} \; \boldsymbol{\nabla} T_{ijkl}. \tag{6}$$

It remains to find $\partial \mathbf{u} / \partial T_{ijkl}$, i.e., how the frame directions vary with the tensor coefficients. Once can notice from the definition (1) that the frame lengths λ_i and directions $\hat{\mathbf{u}}_i$ are in fact *eigenvalues* and *eigenvectors* of the tensor, since

$$\mathbf{T} : \hat{u}_i^{\otimes 3} = \lambda_i \hat{\mathbf{u}}_i, \tag{7}$$

where the left-hand side denotes a triple contraction. From this observation, we follow an approach analoguous to the well-known matrix *eigenvalue perturbation* problem: given an arbitrary tensor perturbation $\delta\mathbf{T}$, we express the variation of the eigendecomposition $\delta\lambda_i$, $\delta\hat{\mathbf{u}}_i$ in terms of $\delta\mathbf{T}$, keeping only first-order terms. This procedure gives the formula for every branch $\mathbf{u}$,

$$\frac{\partial u_i}{\partial T_{j_1 j_2 j_3 j_4}} = \left(\hat{u}_i \hat{u}_{j_1} + M^{-1}_{ij_1} - M^{-1}_{ik} \hat{u}_{j_1} \hat{u}_k\right) \hat{u}_{j_2} \hat{u}_{j_3} \hat{u}_{j_4}, \tag{8}$$

where $M_{ij} = \delta_{ij} - 3\hat{u}_i \hat{u}_j$.

We now turn specifically to the 2D case. Consider a frame with scaled branches $\mathbf{u}$ and $\mathbf{v}$. For more convenient notations we use the polynomial coefficients q_j instead of the tensor coefficients $T_{j_1 j_2 j_3 j_4}$ (these are a linear combination of one another). The Lie bracket can be written compactly as

$$[\mathbf{u}, \mathbf{v}]_i = \left(\frac{\partial u_i}{\partial q_j} v_k - \frac{\partial v_i}{\partial q_j} u_k\right) \frac{\partial q_j}{\partial x_k}. \tag{9}$$

It can be shown, by plugging (8) into the equations, and using symbolic computations, that the expression in brackets *can be expressed entirely as a linear combination of the tensor coefficients* q_j. This observation brings us to our key theoretical result: we can express the Lie bracket of a 2D frame field using only the tensor field and its derivatives:

$$[\mathbf{u}, \mathbf{v}]_i = C_{ijkl}\ q_l\ \frac{\partial q_j}{\partial x_k}. \tag{10}$$

3. INTEGRABLE FRAMES SOLVER

We formulate the search for an integrable frame field over a geometric domain Ω as the minimization of an *integrability energy*:

$$\min_q\ H(q) = \frac{1}{2} \int_\Omega \|[\mathbf{u}, \mathbf{v}]\|^2 \mathrm{d}\Omega \tag{11}$$

Discretization The frame field is discretized over a triangle mesh. Values are stored at nodes and the field is represented using standard P1 finite elements.

Optimization method To minimize the integrability energy $H(q)$, we chose a very simple solver to demonstrate the validity of our formulation. We perform a gradient descent in which, after each gradient step, every frame is projected back onto the Odeco variety. To prevent the gradient step from moving too far away from the variety, the step Δq is projected onto the tangent space of the variety at q. This is easily done since we know that the normal space at q is generated by the vectors $\{A_i q\}$, where A_i are the matrices describing the variety as in (3). Finally, the frame field is uniformly normalized to preserve the total norm; this prevents the solver from shrinking the frames to reduce the integrability energy. As a summary, the following steps are repeated until convergence:

1. Compute gradient step: $\Delta q \coloneqq -\boldsymbol{\nabla} H(q)$,
2. Project step on tangent space: $\Delta q \coloneqq \mathbb{P}_{\mathrm{T}_q \mathcal{F}}[\Delta q]$,
3. Update: $q \coloneqq q + \Delta q$,
4. Project tensors on Odeco variety: $q \coloneqq \mathbb{P}_{\mathcal{F}}[q]$,
5. Normalize uniformly: $q \coloneqq \alpha q$.

Initial condition The solution to (11) is obviously not unique, just like parametrizations are not unique. Among integrable frame fields, we wish to compute one that is smooth. This can be achieved by adding a Dirichlet energy term to the objective, which acts as a regularizer. However, this would prevent the solver from converging to an exactly integrable frame field. We found that simply choosing a smooth frame field (e.g. one obtained with a classical MBO scheme as in [6]) as initial solution provides results that are smooth enough since the solver does not diverge too far from the initial solution.

Boundary condition At the boundary of the geometric domain, a special projector is applied in step 4 such that the orientation of the frame remains fixed (according to the boundary normal) and only the frame scales can be changed.

4. RESULTS

To validate our approach we have run the algorithm on two simple geometric models, shown on Fig. 1.

The *annulus* possesses no singularities but has known analytical integrable solutions, where the frames are aligned with $\hat{\mathbf{e}}_r$ and $\hat{\mathbf{e}}_\theta$, the radial branch increases linearly with the radius, and the tangential branch is any axisymmetric function $f(r)$:

$$(\mathbf{u}, \mathbf{v}) = (ar\ \mathbf{e}_r, f(r)\ \mathbf{e}_\theta). \tag{12}$$

The results show that the algorithm converges to this type of solution, both in the isotropic and non-isotropic cases. Looking at the evolution of the integrability energy on Fig. 2 indicates that the iterations have converged. There is good reason to believe that the residual energy is due only to the discretization, as we have observed that the residual converges to zero as we refined the mesh.

The *piano* model possesses the two most important types of singularities: a $+1/4$ index singularity on the left, and a $-1/4$ index on the right. The isotropic case has converged but as the residual energy is large we suspect a local minimum; this will need further investigation. The non-isotropic case goes in the right direction but has not fully converged despite the high number of iterations performed; this shows that the convergence rate of the solver needs to be improved.

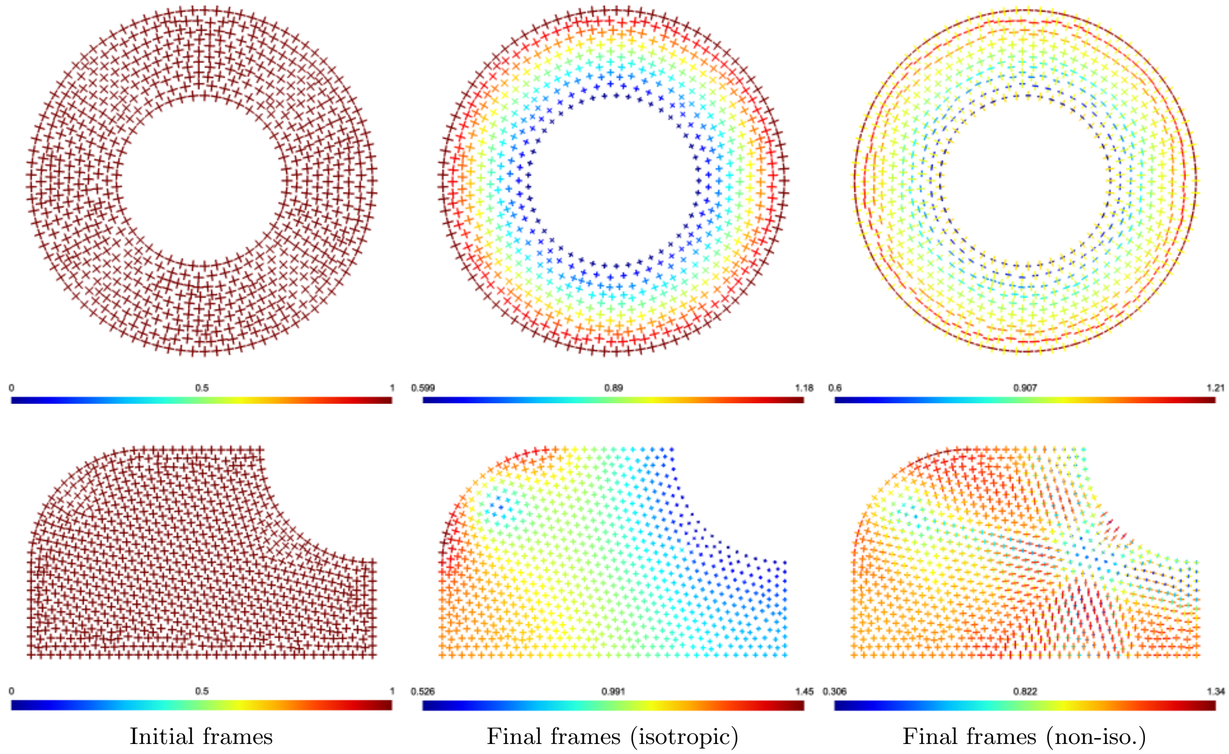

Figure 1: Results of our algorithm on the *annulus* model (top) and a model with two singularities of index $\pm 1/4$, called *piano* (bottom). The algorithm is run with both isotropic and non-isotropic projections.

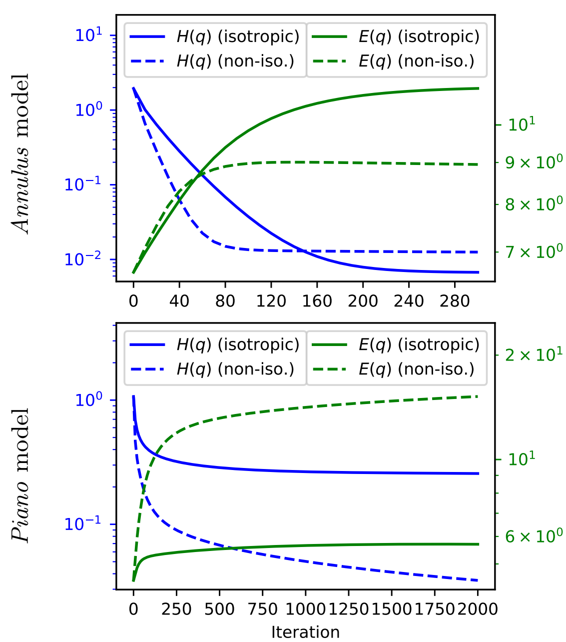

Figure 2: Convergence statistics with evolution of integrability energy $H(q)$ and Dirichlet energy $E(q)$.

5. CONCLUSION AND FUTURE WORK

In this note, we have deduced the equations expressing the Lie bracket of a 2D frame field. These equations depend only on the tensor representation of the frame field and not on any explicit representation. We have implemented a simple solver for imposing integrability on an input frame field. This solver converges towards nearly-integrable solutions, which demonstrates the viability of our approach.

Convergence of our solver leaves much to be desired. We aim to vastly improve the convergence rate by turning to second-order optimization methods, such as Newton or BFGS. We will also investigate whether states such as the isotropic piano model are indeed local minima and if so, how we can escape them.

The next step will be to generalize the approach to 3D frame fields. This can either be done by finding a 3D generalization of the Lie bracket (which we have already found not to be linear), or by designing a method using explicit branches anyway in the expression of (9).

References

[1] Bommes D., Zimmer H., Kobbelt L. "Mixed-Integer Quadrangulation." *ACM Transactions on Graphics*, vol. 28, no. 3, 77:1–77:10, Jul. 2009

[2] Kälberer F., Nieser M., Polthier K. "QuadCover - Surface Parameterization Using Branched Coverings." *Computer Graphics Forum*, vol. 26, no. 3, 375–384, 2007

[3] Campen M., Bommes D., Kobbelt L. "Quantized Global Parametrization." *ACM Transactions on Graphics*, vol. 34, no. 6, 1–12, Nov. 2015

[4] Jezdimirović J., Chemin A., Remacle J.F. "Integrable Cross-Field Generation Based on Imposed Singularity Configuration – the 2D Manifold Case –.", Sep. 2022

[5] Liu H., Zhang P., Chien E., Solomon J., Bommes D. "Singularity-Constrained Octahedral Fields for Hexahedral Meshing." *ACM Transactions on Graphics*, vol. 37, no. 4, 1–17, Jul. 2018

[6] Palmer D., Bommes D., Solomon J. "Algebraic Representations for Volumetric Frame Fields." *ACM Transactions on Graphics*, vol. 39, no. 2, 1–17, Apr. 2020