

Geometric Optimisation Via Spectral Shifting

ROMAN POYA, Siemens Digital Industries Software, United Kingdom

ROGELIO ORTIGOSA, Technical University of Cartagena, Spain

THEODORE KIM, Yale University, United States

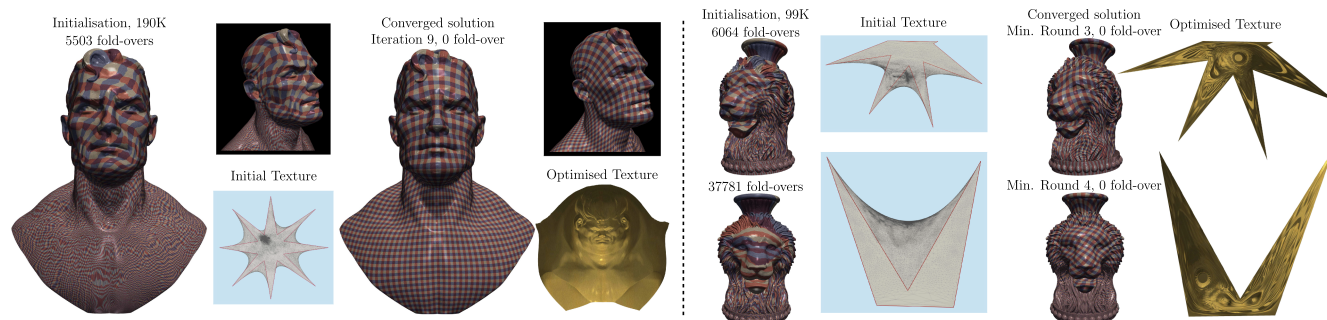


Fig. 1. Our framework turns infinite energies into finite ones, even in the presence of fold-overs, by re-expressing distortion in terms of shifted singular values. We can minimise popular flip-preventing distortion energies over non-injective domains at speeds competitive with locally injective methods. We apply our algorithm to the Symmetric Dirichlet energy on folded initial states, on both free (left) and fixed (right) boundary surface parametrisation problems.

We present a geometric optimisation framework that can recover fold-over free maps from non-injective initial states using popular flip-preventing distortion energies. Since flip-preventing energies are infinite for folded configurations, we propose a new regularisation scheme that shifts the singular values of the deformation gradient. This allows us to re-use many existing algorithms, especially locally injective methods for initially folded maps. Our regularisation is suitable for both singular value- and invariant-based formulations, and systematically contributes multiple stabilisers to the Hessian. In contrast to proxy-based techniques, we maintain second-order convergence. Compact expressions for the energy eigensystems can be obtained for our extended stretch invariants, enabling the use of fast projected Newton solvers. Although spectral shifting in general has no theoretical guarantees that the global minimum is an injection, extensive experiments show that our framework is fast and extremely robust in practice, and capable of generating high-quality maps from severely distorted, degenerate and folded initialisations.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; **Redundancy**; **Robotics** • **Networks** → **Network reliability**;

Additional Key Words and Phrases: Fold-over free mappings, projected Newton solver, mesh, parametrisation, numerical optimisation

Authors' addresses: R. Poya (corresponding author), Siemens Digital Industries Software, Francis House, Cambridge, United Kingdom, CB2 1PH; email: roman.poya@siemens.com; R. Ortigosa, Technical University of Cartagena, Cartagena, Campus Muralla del Mar, 30202, Cartagena, Spain; email: rogelio.ortigosa@upct.es; T. Kim, Yale University, New Haven, Connecticut, CT 06520, United States; email: theodore.kim@yale.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0730-0301/2023/04-ART29 \$15.00

<https://doi.org/10.1145/3585003>

ACM Reference format:

Roman Poya, Rogelio Ortigosa, and Theodore Kim. 2023. Geometric Optimisation Via Spectral Shifting. *ACM Trans. Graph.* 42, 3, Article 29 (April 2023), 15 pages.

<https://doi.org/10.1145/3585003>

1 INTRODUCTION

Generating fold-over free maps is a common goal in geometry processing and physical simulation. Nonlinear techniques for generating locally injective maps often require a valid starting point (such as a Tutte [1963] embedding) as many popular distortion energies become infinite if the initial domain is not locally injective. Such energies have been studied in continuum mechanics since the foundational work of Ball [1976, 1981, 1983] on invertibility theorems.

Coercivity (i.e., barriers) is a key concept in flip-preventing energies which dictates that the energy should go to infinity when the map inverts [Hartmann and Neff 2003]. A series of recent works in computer graphics have proposed such barrier energies [Fu et al. 2015; Rabinovich et al. 2017; Schüller et al. 2013; Smith and Schaefer 2015], while in computational physics, the topic of *recovering* local injectivity has been studied since the work of Winslow [1966] and its variational formulation proposed later in Brackbill and Saltzman [1982]. We refer the reader to Garanzha et al. [2021a] for a historical overview and Fu et al. [2021] for a detailed survey.

In this paper, we propose an optimisation framework for generating fold-over free mappings using popular rotation invariant flip-preventing energies, even when the initial domain is non-injective or degenerate. Our work uses a continuum mechanics approach to introduce the following contributions:

- A **regularisation technique** based on shifted singular values that yields a simple, consistent and general formulation.

- **Compact eigensystems** for regularised energy Hessians which can be embedded into fast second-order solvers, using extended, regularised stretch invariants [Smith et al. 2019] with a tailor-made **optimisation scheme**.
- Between $1.43\times$ to $12.9\times$ **mean speedups** over existing methods on a large-scale benchmark.

More specifically, we solve for the regularisation parameter ahead of minimisation (in an outer loop; see Algorithm 2) and run a series of **Projected Newton (PN)** steps that make use of energy eigensystems. Once the map is repaired, our workflow reduces to well-known locally injective methods. This property then enables the re-use of established techniques for tackling global injectivity, while still starting from folded configurations (see Figure 4). While there are no theoretical guarantees that the global minimum of the energy corresponds to an inversion-free map, our framework passes large-scale datasets with up to 100% success rate. To aid reproducibility, we provide kernel computation code in the supplement.

2 RELATED WORK

2.1 Problem Statement

To facilitate our discussion in the next section, we will first articulate the problem from a mechanics perspective [Bonet et al. 2016b]. Given a mesh of nodes \mathcal{V} and elements \mathcal{T} , potentially containing fold-overs, our goal is to minimise energies with barriers such that¹

$$\Pi(\mathbf{x}^*) = \min_{\mathbf{x}} \int_{\Omega_0} \mathcal{D}(\mathbf{F}) \, d\Omega \quad (1a)$$

$$J = \det \mathbf{F} > 0 \quad (1b)$$

where Π is the total potential energy of the system, \mathbf{x} is the vertex positions in the mesh, \mathbf{x}^* the optimal solution, \mathbf{F} the deformation gradient tensor, $\mathcal{D}(\mathbf{F})$ the distortion energy density function, and J the determinant (also known as the Jacobian). Second-order methods require both first and second derivatives of the integral in (1a):

$$D\Pi[\delta\mathbf{u}] = \int_{\Omega_0} \frac{\partial \mathcal{D}(\mathbf{F})}{\partial \mathbf{F}} : \nabla_{\mathbf{w}} \delta\mathbf{u} \, d\Omega \quad (2a)$$

$$D^2\Pi[\delta\mathbf{u}, \Delta\mathbf{u}] = \int_{\Omega_0} \nabla_{\mathbf{w}} \Delta\mathbf{u} : \frac{\partial^2 \mathcal{D}(\mathbf{F})}{\partial \mathbf{F} \partial \mathbf{F}} : \nabla_{\mathbf{w}} \delta\mathbf{u} \, d\Omega \quad (2b)$$

where $:$ denotes the inner product of matrices ($\mathbf{A} : \mathbf{B} = \text{tr}(\mathbf{A}^T \mathbf{B})$), $\nabla_{\mathbf{w}}$ is the gradient operator with respect to a desired domain (such as the initial mesh) and $\delta\mathbf{u}$ and $\Delta\mathbf{u}$ are finite element trial and test functions. The first term in integrand (2a) is the gradient of energy, also called the **first Piola-Kirchhoff stress tensor (PK1)**, which we express as

$$\mathbf{P} = \frac{\partial \mathcal{D}(\mathbf{F})}{\partial \mathbf{F}}. \quad (3)$$

The middle term in integrand (2b) is the Hessian of energy, which we denote as

$$\mathbb{H} = \frac{\partial^2 \mathcal{D}(\mathbf{F})}{\partial \mathbf{F} \partial \mathbf{F}}. \quad (4)$$

¹Unless otherwise stated, lower case bold letters \mathbf{a} denote vectors, upper case bold letters \mathbf{A} are second order tensors (matrices), and blackboard bold letters \mathbb{A} are fourth order tensors (matrix of matrices).

2.2 Inversion-free Mappings

In graphics, methods for generating inversion-free mappings include Tutte [1963], **Bounded Distortion (BD and LBD)** [Aigerman and Lipman 2013; Kovalsky et al. 2015], extremal quasi-conformal mapping [Weber et al. 2012], **Simplex Assembly (SA)** [Fu and Liu 2016], **Adaptive Block Coordinate Descent (ABCD)** [Naitat et al. 2020], **Total Lifted Content (TLC)** [Du et al. 2020a] and **Fold-over Free Maps (FFM)** [Garanzha et al. 2021a; Su et al. 2019]. Progressive embeddings and intrinsic triangulations have also been successfully applied in 2D (parametrisation) contexts [Campen et al. 2021; Gillespie et al. 2021; Liu et al. 2018; Shen et al. 2019], which relax the requirements on the input triangulation to guarantee finding a discrete, locally injective map. To our knowledge, TLC and FFM are the only frameworks that pass the whole local injectivity benchmark of Du et al. [2020b] in 2D and 3D.

To guarantee inversion-free maps, Equation (1b) must be satisfied, so one common approach is to perform an unconstrained minimisation of barrier energies that implicitly encode this constraint. However, the barrier prevents these energies from being applied when the initialisation contains fold-overs ($J < 0$). Consequently, state-of-the-art algorithms either lift the space of simplices to a logical space by minimising some simple field [Du et al. 2020a], or consider a metric that bounds the amount of distortion [Aigerman and Lipman 2013; Fu and Liu 2016; Kovalsky et al. 2015]. Most of these approaches change the metric (transformation matrix) by employing an auxiliary strictly non-degenerate map.

Minimising barrier energies with fold-overs without changing the triangulation/tetrahedralisation relies on either filtered folded elements [Naitat et al. 2020] or regularised Jacobians [Garanzha et al. 2021a]. Filtering then requires adding energies for inverted elements that can unnecessarily compete with the primary distortion energy. Alternatively, Jacobian regularisation allows the same energy to be used, but leads to non-convex energies and inconsistent treatment of the map's metric, which implies that the regularised Jacobian cannot be retrieved from the product of singular values of the deformation gradient. This complicates the use of state-of-the-art second-order approaches [Shtengel et al. 2017; Smith et al. 2019], which exploit the structure of the deformation gradient.

Invertible distortion energies have also been studied in simulation [Irving et al. 2004; Smith et al. 2018; Stomakhin et al. 2012], but these formulations tackle a different problem. They are neither designed to solve inversion-free mappings nor are they a substitute for locally injective methods (further discussion in Section 5.8).

2.3 Energy Minimisation

Minimising barrier energies typically require stabilisation of the Hessian to make them semi-definite positive. We use the term stabilisation here as opposed to regularisation because Hessian stabilisation is distinct from Jacobian regularisation. The former is necessary to obtain a descent direction while the latter makes a barrier energy finite under fold-overs. To overcome the problem of Hessian indefiniteness, one can clamp the negative eigenvalues of the Hessian [Fu and Liu 2016]. Alternatively, a proxy Hessian such as a Laplacian [Gargallo-Peiró et al. 2015], or a

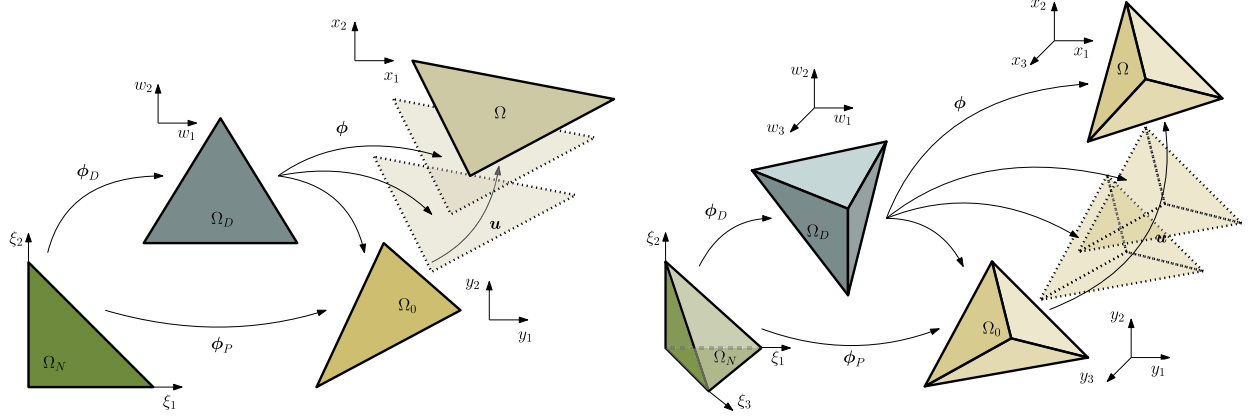


Fig. 2. $\mathbb{R}^d \rightarrow \mathbb{R}^d$ mapping (left) $d = 2$ (right) $d = 3$, between a parametric element Ω_N in the *natural* coordinates, auxiliary element Ω_D , and the physical element Ω_0 . The mapping of interest here is from the desired shape to the final physical element whose diffeomorphism can be established as a composition $\phi = \phi_P \circ \phi_D^{-1}$. Subsequent deformation of the domain is computed in the moving frame Ω in an Arbitrary Lagrangian Eulerian (ALE) fashion. In case the desired element is the rest configuration, we recover the standard Lagrangian formulation.

linearised version of the same Hessian [Poya et al. 2016] can be used/added, or a semidefinite-indefinite decomposition can be performed [Garanzha et al. 2021a].

Projected Newton (PN) [Stomakhin et al. 2012; Teran et al. 2005] is not a good candidate for certain cases of untangling, as it can lead to damped convergence, and coupled with a non-filtered line search, can even stall. Over 10,000 rounds of L-BFGS or Projected Newton are commonly needed [Du et al. 2020a; Garanzha et al. 2021a]. Alternatively, **Quasi-Newton (QN)** methods can smooth away barrier terms and cause the optimisation to converge to the incorrect (still folded) minima. As reported in Du et al. [2020a], the QN method in TLC can fail to recover local injectivity in moderately complex problems, even though the energy is guaranteed to achieve its global minimum at a locally injective state.

3 OUR REGULARISATION APPROACH

We will present our overall approach by starting with continuum mechanics preliminaries (Section 3.1), and present our core regularisation scheme (Section 3.2). From this, we will show how to build a set of regularised stretch invariants (Section 3.3), which allow us to derive compact expressions for their Hessians. After describing a strategy for setting a key regularisation parameter (β) in Section 3.4, we will apply our regularisation scheme to several common energies (Section 3.5), and end with a brief discussion (Section 3.6).

3.1 Continuum Mechanics Preliminaries

In the context of mechanics, the mapping between a mesh with inverted elements and an inversion-free mesh [Aigerman and Lipman 2013; Du et al. 2020a; Fu and Liu 2016; Garanzha et al. 2021a; Kovalsky et al. 2015; Su et al. 2019] can be formalised as the following **Arbitrary-Lagrangian-Eulerian (ALE)** scheme.

Consider an element in the auxiliary (“ideal”) configuration Ω_D , with position vectors \mathbf{w} and a desired shape and size. A corresponding element Ω_0 in the physical configuration has position vectors \mathbf{y} . This element undergoes motion to arrive at the final

physical configuration Ω , denoted with position vectors \mathbf{x} [Bonet et al. 2016b].

To measure distortion we consider the affine mapping ϕ , from Ω_D to Ω (see Figure 2). This can be decomposed into $\phi_D : \Omega_N \rightarrow \Omega_D$ from the natural element Ω_N to the auxiliary element Ω_D , and $\phi_P : \Omega_N \rightarrow \Omega_0$, from the natural element Ω_N to the physical element Ω_0 . Thus, ϕ can be determined by the composition $\phi = \phi_P \circ \phi_D^{-1}$.

Following nonlinear mechanics [Bonet et al. 2016b], the *fi-bre map* transforms an edge between an auxiliary and physical element:

$$\mathbf{F} = \nabla_{\mathbf{w}} \mathbf{x} = (\nabla_{\xi} \mathbf{x})(\nabla_{\xi} \mathbf{w})^{-T}. \quad (5)$$

The domain Ω_D can be a single element, a polygon soup, or a fully conformal mesh. In the case of a conformal mesh, \mathbf{F} is the deformation gradient tensor $\mathbf{F} = \mathbf{F}_{\mathbf{y}}$, where $\mathbf{w} = \mathbf{y}$ is the position vector of the initial configuration.

Following Bonet et al. [2015, 2016a] the three fundamental distortion measures are \mathbf{F} (edge map), its cofactor \mathbf{H} (area map) and its determinant (volume map) such that (see Figure 3)

$$\mathbf{F} = \nabla_{\mathbf{w}} \mathbf{x}, \quad \mathbf{H} = \text{cof} \mathbf{F} = \mathbf{J} \mathbf{F}^{-T} = \frac{1}{2} \mathbf{F} \times \mathbf{F}, \quad \mathbf{J} = \det \mathbf{F}, \quad (6)$$

where \times is the *tensor cross product* between two second order tensors expressed as $[\mathbf{A} \times \mathbf{A}]_{il} = \mathcal{E}_{ijk} \mathcal{E}_{ljk} A_{jj} B_{kk}$ where \mathcal{E} is the third order Levi-Civita tensor [de Boer 1982]. Furthermore, \mathbf{F} admits a left polar decomposition

$$\mathbf{F} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \mathbf{R} \mathbf{S}, \quad (7)$$

into a rotation tensor $\mathbf{R} = \mathbf{U} \mathbf{V}^T$ and a stretch tensor $\mathbf{S} = \mathbf{V} \mathbf{\Sigma} \mathbf{V}^T$. Here, \mathbf{U} , $\mathbf{\Sigma}$ and \mathbf{V} are the tensors from the **signed singular value decomposition (SSVD)** of \mathbf{F} [Irving et al. 2004; Kovalsky et al. 2015].

3.2 Regularisation of \mathbf{F}

We will now regularise the deformation gradient tensor \mathbf{F} , which in turn will allow us to formulate regularised invariants which we can then use to regularise arbitrary barrier energies. Recall that

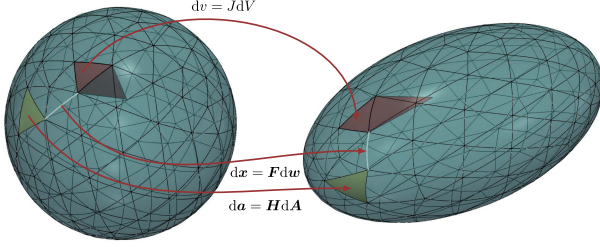


Fig. 3. Mapping a target to a source: the deformation gradient F maps edges (white) from the target to source mesh, its cofactor H maps areas (green) and its determinant J maps volumes (red).

a tensor invariant (e.g. the determinant) is a scalar function of a tensor that remains unchanged under a basis change or a rotation of the target frame. We use the subscript r to denote a regularised quantity, e.g. F_r or σ_{r_i} . We regularise F by spectrally shifting its signed singular values:

$$F_r = U(\Sigma + \Sigma_s)V^T = U\Sigma_r V^T \quad (8a)$$

$$\text{with } \sigma_{r_i} = \sigma_i + \beta \quad i = 1, 2, 3 \quad (8b)$$

$$\text{s.t. } \det F_r > 0. \quad (8c)$$

Above, $\Sigma_s = \beta I$ is a strictly positive definite regularisation tensor. The tensor Σ_r then has a smallest signed singular value of at least $\sigma_i + \beta \geq \epsilon$ where ϵ is a small positive number that we iteratively decrease over the course of optimisation (see ϵ_k in Algorithm 2). We set $\epsilon = 10^{-6}$ as the minimum value.

We consider β a constant during minimisation; this assumption is discussed extensively in Section 3.4. We could alternatively regularise only one signed singular value using the convention ($\text{sign}(\sigma_3) < 0$) [Irving et al. 2004], but such an approach prohibits the derivation of generic regularised energies. The complexity of the expressions becomes unwieldy [Stomakhin et al. 2012], as regularisation in one principal direction introduces anisotropy into isotropic energies. Geometrically, regularising F scales-up the deformation metric i.e., increases the Jacobian (volume ratio), edge length, and area ratios (see Figure 3). However, as soon as fold-overs are repaired, we fall back to the original metric.

3.3 Regularisation of Stretch Invariants

We can now use the regularised deformation gradient F_r to formulate regularised stretch invariants. We first express an arbitrary distortion energy \mathcal{D} in terms of an extended set of stretch tensor invariants:

$$\mathcal{D}(F) = \hat{\mathcal{D}}(I_F, II_F, I_H, II_H, J) \quad (9)$$

with the following definitions for invariants in 3D and 2D

Invariants in 3D	Invariants in 2D
$\begin{aligned} I_F &= \sigma_1 + \sigma_2 + \sigma_3 \\ II_F &= \sigma_1^2 + \sigma_2^2 + \sigma_3^2 \\ I_H &= \sigma_1\sigma_2 + \sigma_1\sigma_3 + \sigma_2\sigma_3 \\ II_H &= \sigma_1^2\sigma_2^2 + \sigma_1^2\sigma_3^2 + \sigma_2^2\sigma_3^2 \\ J &= \sigma_1\sigma_2\sigma_3 \end{aligned}$	$\begin{aligned} I_F &= \sigma_1 + \sigma_2 \\ II_F &= \sigma_1^2 + \sigma_2^2 \\ J &= \sigma_1\sigma_2 \end{aligned} \quad (10)$

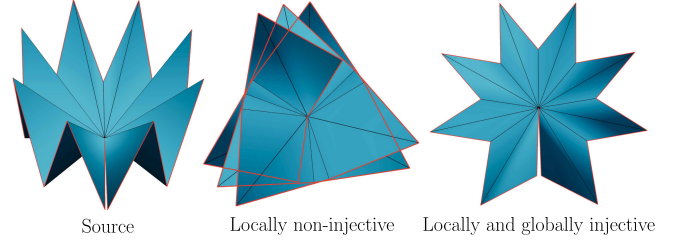


Fig. 4. An inversion-free but locally non-injective mapping. The map in the middle is self-intersecting (not globally injective) and overwound around the mid-vertex (sum of angles $> 2\pi$) (not locally injective) [Lipman 2014; Weber and Zorin 2014]. Our framework allows re-use of existing locally-injective methods on folded maps. Result on the right is obtained with our regularised Symmetric Dirichlet energy (RSD) and contact penalties on the boundary. More results in Figure 14.

Two additional cofactor invariants, I_H and II_H , are introduced on top of those from Smith et al. [2019], and are “cross” terms that model area distortion in 3D (see Figure 3). They are redundant in 2D, as $I_H = I_F = \sigma_1 + \sigma_2$ and $II_H = II_F = \sigma_1^2 + \sigma_2^2$, but in 3D they can be expressed as

$$I_H = \frac{1}{2}(I_F^2 - II_F) \quad \text{area-rotation in 3D} \quad (11a)$$

$$II_H = I_H^2 - 2I_F J \quad \text{area-preservation in 3D} \quad (11b)$$

We discuss these two cofactor invariants further in Section 3.6.

Using Equation (8), we can now formulate a set of regularised invariants,

Regularised stretch tensor invariants

$$\begin{aligned} I_{F_r} &= I_F + d\beta \\ II_{F_r} &= II_F + 2\beta I_F + d\beta^2 \\ I_{H_r} &= I_H + 2\beta I_F + 3\beta^2 \\ II_{H_r} &= II_H + 2\beta(I_F II_F - III_F) + 2\beta^2 I_F^2 + 4\beta^3 I_F + 3\beta^4 \\ J_r &= \begin{cases} J + \beta I_F + \beta^2 & 2D \\ J + \beta I_H + \beta^2 I_F + \beta^3 & 3D \end{cases} \end{aligned} \quad (12)$$

where $III_F = \sigma_1^3 + \sigma_2^3 + \sigma_3^3$ is an auxiliary invariant.

These invariants can be plugged in to any existing distortion energy to yield a regularised counterpart. The gradients and Hessian of these regularised invariants, as well as the exact stabilisers that they contribute to the Hessian, are given in the supplement. Critically, our choice of invariants allows us to derive compact eigensystems for the regularised energy Hessians.

3.4 Finding the Regulariser, β

The decision to regularise using shifted singular values can seem empirical. However, we will show that *all* state-of-the-art approaches, including TLC [Du et al. 2020a], FFM [Garanzha et al. 2021a] and ABCD [Naitsat et al. 2020] can be understood as similar regularisers. Our scheme will then combine the merits of these approaches.

3.4.1 State-of-the-art Approaches. We start with TLC [Du et al. 2020a], where the value of regulariser emerges from a strictly non-degenerate (lifted) transformation. The regulariser is part of the TLC energy itself and cannot be applied to other energies.

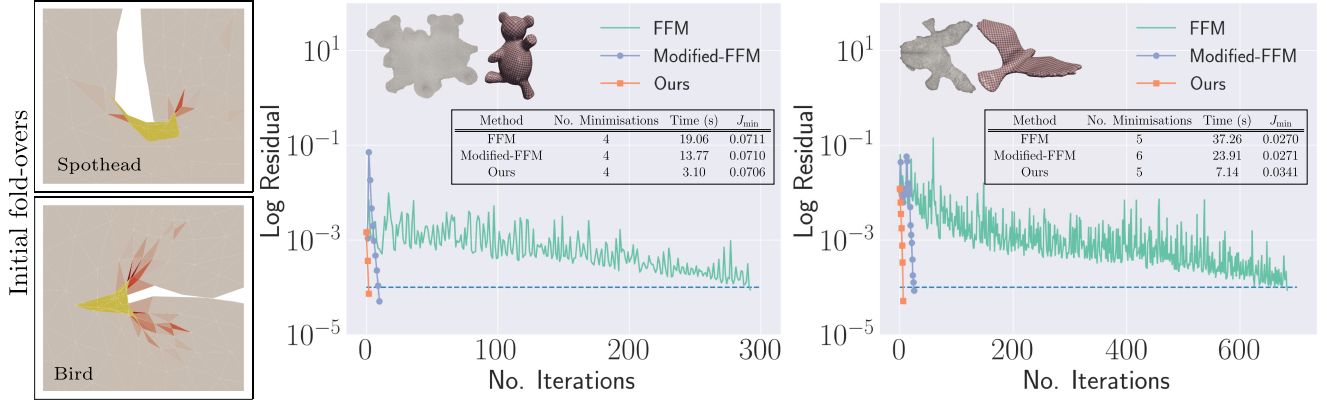


Fig. 5. Comparison of our framework with FFM [Garanzha et al. 2021a] for two seamless surface parametrisation problems. The left contains 87,984 triangles 87,824 and 27 invalid elements and the right has 87,824 and 31 invalid elements. We also implemented the modified element-wise SPD Hessian from Garanzha et al. [2021a]. Our regularisation of MIPS energy achieved the same quality as FFM but was almost **5X faster**. Modified-FFM almost degraded to a first order in subsequent rounds of minimisation but overall was faster than FFM (L-BFGS). Convergence plots are only for the first round of minimisation.

In contrast, FFM [Garanzha et al. 2021a] performs a penalty-style regularisation of J such that

$$J_r(J) = \frac{1}{2} \left(J + \sqrt{J^2 + \delta^2} \right) \quad (13)$$

for every element for which $J < 0$. The δ is a positive constant that typically encodes the minimum Jacobian value over the entire mesh. This approach allows the same energy to be used for folded elements, but introduces metric inconsistency.² This inconsistency implies that: (a) J_r does not correspond to the product of singular values of any metric (F or F_r) and (b) results in non-convex energies, even when applied to convex energies. Moreover, it becomes impossible to formulate the problem in terms of singular values and obtain Hessian eigensystems. Our regularised invariants do not suffer from these problems.

Finally, ABCD [Naitat et al. 2020] considers a separate flip term to modify non-positive Jacobians, which is then used in conjunction with a desired distortion energy:

$$J_r(J) = \prod_i^{d-1} \sigma_i \delta, \quad (14)$$

where negative Jacobians are replaced with a positive constant δ .

3.4.2 Our Approach. We can combine the strengths of the aforementioned approaches. In general, given a regularised Jacobian of, e.g. Equations (13) or (14), we can solve the last line in Equation (12) for β . This allows for both Jacobian and singular-value regularisations to be embedded into a unified approach (see Algorithm 2).

In 2D, this amounts to solving a quadratic equation ($(\sigma_1 + \beta)(\sigma_2 + \beta) = J_r$), and a cubic in 3D ($(\sigma_1 + \beta)(\sigma_2 + \beta)(\sigma_3 + \beta) = J_r$). This is achieved by polynomial root finding on Line 12 of Algorithm 1.

Alternatively, β can be computed for every element, since the last line of Equation (12) can be solved for β analytically. In 2D,

²Metric consistency demands that F and any subsequent derived quantity should also be regularised, not just J .

ALGORITHM 1: Compute β using root finding in nD

```

1: procedure ComputeBeta( $\mathcal{V}, \mathcal{T}, J_{min}, \delta$ )  $\triangleright$  Used within Algorithm 2
2:    $\beta \leftarrow -\inf$ 
3:   for  $T \in \mathcal{T}$  do
4:     Compute invariants  $I_F, I_H, J$   $\triangleright$  (10)
5:     Compute  $J_r$   $\triangleright$  (13) or (14)
6:     Compute flip-penalty  $\kappa \leftarrow J_r - J_{min}$ 
7:     if  $d = 2$  then
8:       polyCoeffs  $\leftarrow [1, I_F, -\kappa]$ 
9:     else if  $d = 3$  then
10:      polyCoeffs  $\leftarrow [1, I_F, I_H, -\kappa]$ 
11:     end if
12:      $\beta \leftarrow \max(\beta, \max(\text{FindRoots}(\text{polyCoeffs})))$ 
13:   end for
14:   return  $\beta$ 
15: end procedure

```

we obtain $\beta = \frac{1}{2}(-I_F + \sqrt{I_F^2 + 4(J_r - J)})$, which is always positive and can be back-substituted to carry out the energy derivatives.

We have now fully parametrised over β , where β can originate from any kind of constraint, not just those described here. In contrast to Garanzha et al. [2021a], a constant β in 2D amounts to a linear transform of a standard energy into a regularised energy (note that, I_F is linear in F and has SPD Hessian), and linear transforms preserve convexity. With this set of regularised invariants, we have established a consistent way of expressing infinite energies over injective, non-injective and degenerate mappings.

3.5 Regularised Energies

We can now regularise several well-known distortion energies and give explicit expressions for their eigensystems. As mentioned, we can either pre-compute β ahead of minimisation or re-substitute its analytical form and compute energy derivatives. The latter is particularly useful in 2D energies such as MIPS or **Symmetric Dirichlet (SD)** where β does not explicitly appear in the energy (c.f. Table 1), and only the Jacobian regulariser δ from Equations (13) or (14) are needed for a consistent

Table 1. Regularised Energy ($\hat{\mathcal{D}}$), Gradient (\mathbf{P}), and Hessian Eigensystem (Four λ_i s) for Regularised MIPS (RMIPS), Regularised Symmetric Dirichlet (RSD), and Regularised Symmetric ARAP (RSARAP) in 2D

Regularised MIPS (RMIPS)	Regularised Symmetric Dirichlet (RSD)	Regularised Symmetric ARAP (RSARAP)
$\mathcal{D}_{\text{RMIPS}}(\mathbf{F}_r) = \frac{\ \mathbf{F}_r\ ^2}{2J_r}$ $\hat{\mathcal{D}}_{\text{RMIPS}}(I_F, I_F, J) = \frac{II_F + 2(\beta I_F + \beta^2)}{2(J + \beta I_F + \beta^2)} = \frac{II_F}{2J_r} + \left(1 - \frac{J}{J_r}\right)$ $\mathbf{P}_{\text{RMIPS}} = \frac{1}{J_r}(\mathbf{F} - \mathbf{H}) - \frac{1}{J_r^2}(J - \frac{II_F}{2})(\mathbf{H} + \beta \mathbf{R})$ $\lambda_{1,2} = \text{eig}(\mathbf{A})$ $\lambda_3 = \frac{2}{J_r} + \frac{II_F - 2J}{2J_r^2}$ $\lambda_4 = -\frac{I_F(II_F - 2J)}{2I_F J_r^2}$ <p>where the entries of the symmetric matrix \mathbf{A} are</p> $\mathbf{A}_{11} = \frac{1}{J_r} + 2\frac{\beta(\sigma_2 - \sigma_1) + \sigma_2^2 - J}{J_r^2} + \frac{(\sigma_2 + \beta)^2(II_F - 2J)}{J_r^2}$ $\mathbf{A}_{22} = \frac{1}{J_r} + 2\frac{\beta(\sigma_1 - \sigma_2) + \sigma_1^2 - J}{J_r^2} + \frac{(\sigma_1 + \beta)^2(II_F - 2J)}{J_r^2}$ $\mathbf{A}_{12} = -\frac{1}{J_r} - \frac{II_F - 2J}{2J_r^2}$ <p>where τ (appearing in eigenmatrices) can now be computed as $\tau = (\lambda_1 - \mathbf{A}_{22})/\mathbf{A}_{12}$.</p>	$\mathcal{D}_{\text{RSD}}(\mathbf{F}_r) = \frac{1}{2}(\ \mathbf{F}_r\ ^2 + \ \mathbf{F}_r^{-1}\ ^2) = \frac{1}{2}(\ \mathbf{F}_r\ ^2 + J_r^{-2}\ \mathbf{H}_r\ ^2)$ $\hat{\mathcal{D}}_{\text{RSD}}(II_F, J) = \frac{1}{2}(II_{F_r} + J_r^{-2}II_{H_r})$ $= \frac{1}{2}II_{F_r}(1 + J_r^{-2})$ $= \frac{1}{2}II_F(1 + J_r^{-2}) + (J_r + J_r^{-1} - J - JJ_r^{-2})$ $\mathbf{P}_{\text{RSD}} = (1 + J_r^{-2})(\mathbf{F} - \mathbf{H})$ $+ (1 - J_r^{-2} - (II_F - 2J)J_r^{-3})(\mathbf{H} + \beta \mathbf{R})$ $\lambda_1 = 1 + \frac{J_r^2 - 4J_r(J + \beta\sigma_1) + \beta^2(II_{F_r} + 2II_F - 4J)}{J_r^4}$ $+ \frac{2\beta\sigma_2(2II_{F_r} + II_F - 2J) + 3\sigma_2^2 II_{F_r}}{J_r^4}$ $\lambda_2 = 1 + \frac{J_r^2 - 4J_r(J + \beta\sigma_2) + \beta^2(II_{F_r} + 2II_F - 4J)}{J_r^4}$ $+ \frac{2\beta\sigma_1(2II_{F_r} + II_F - 2J) + 3\sigma_1^2 II_{F_r}}{J_r^4}$ $\lambda_3 = 1 + \frac{II_F - 2J + 3J_r}{J_r^3}$ $\lambda_4 = 1 + \frac{2\beta}{I_F} - \frac{I_{F_r}(II_F - 2J + J_r)}{I_F J_r^3}$	$\mathcal{D}_{\text{RSARAP}}(\mathbf{F}_r) = \frac{1}{2}(\ \mathbf{F}_r - \mathbf{R}_r\ ^2 + \ \mathbf{F}_r^{-1} - \mathbf{R}_r^{-1}\ ^2)$ $\hat{\mathcal{D}}_{\text{RSARAP}}(I_F, II_F, J) = \frac{1}{2}(II_{F_r} + J_r^{-2}II_{H_r}) - (I_{F_r} + J_r^{-1}I_{H_r})$ $= \frac{1}{2}II_{F_r}(1 + J_r^{-2}) - I_{F_r}(1 + J_r^{-1})$ $= \frac{1}{2}II_F(1 + J_r^{-2}) - I_F(1 + J_r^{-1})$ $+ (J_r + J_r^{-1} - J - JJ_r^{-2}) - 2\beta(1 + J_r^{-1})$ $\mathbf{P}_{\text{RSARAP}} = (1 + J_r^{-2})(\mathbf{F} - \mathbf{H}) - \mathbf{R}(1 + J_r^{-1})$ $+ (1 + J_r^{-2}(I_{F_r} - 1) - (II_F - 2J)J_r^{-3})(\mathbf{H} + \beta \mathbf{R})$ $\lambda_1 = 1 + \frac{1}{J_r^2} + \frac{3(\sigma_2 + \beta)^2(II_F - 2J)}{J_r^4}$ $+ 2\frac{(\sigma_2 + \beta)(3\sigma_2 + \beta + J_r) - (\sigma_2 - \beta)^2 I_{F_r} - 2(\sigma_1 \beta + J)}{J_r^4}$ $\lambda_2 = 1 + \frac{1}{J_r^2} + \frac{3(\sigma_1 + \beta)^2(II_F - 2J)}{J_r^4}$ $+ 2\frac{(\sigma_1 + \beta)(3\sigma_1 + \beta + J_r) - (\sigma_1 - \beta)^2 I_{F_r} - 2(\sigma_2 \beta + J)}{J_r^4}$ $\lambda_3 = 1 - \frac{I_{F_r}\beta - 3}{J_r^2} + \frac{II_F - 2J}{J_r^3}$ $\lambda_4 = 1 + \frac{2(\beta - 1)}{I_F} - \frac{2}{I_F J_r} + \frac{I_{F_r}(I_{F_r} - 1)}{I_F J_r^2} - \frac{I_{F_r}(II_F - 2J)}{I_F J_r^2}$

The Hessians are computed according to Algorithm 2. C++ implementations are provided in the supplement.

formulation. In such cases, our approach becomes a generalisation of FFM. In the supplement we work out the MIPS energy with such a δ regulariser. We also follow this setup whenever possible (back-substitution/per-element β for 2D MIPS and SD and a global β for the rest) for numerical experimentation.

For the case when β is a constant, the 2D energies, gradients and analytic Hessian eigensystems are given in Table 1 for **Regularised MIPS (RMIPS)**, **Regularised Symmetric Dirichlet (RSD)** and **Regularised Symmetric ARAP (RSARAP)**. The Hessian expressions for the 3D cases are given in the supplement, as well as a script to generate them symbolically. We follow Smith et al. [2019] notation.

3.6 Discussion of Invariant-based Formulation

Distortion energies are typically expressed in terms of singular values [Xu et al. 2015], Cauchy-Green invariants [Bonet et al. 2016b], polyconvex invariants [Bonet et al. 2015, 2016a],³ or stretch tensor invariants [Smith et al. 2019].

Stretch tensor invariants and singular value formulations can both reveal the full eigensystem of the Hessian [Chen and Weber 2017; Smith et al. 2019; Zhu 2021], but invariant formulations generally have the advantage that they encode specific distortion metric such as length, area, volume. While these metrics can also be re-expressed in terms of singular-values, invariant-based formulations additionally allow convexity to be studied in closed-form, see Equation (15). By adopting an invariant-based formulation, we can re-use nearly a century of well-established literature [Ball 1976; Bonet et al. 2016b]. Without loss of generality, we chose to work with stretch tensor invariants because it allowed us to derive analytic eigensystems for our *regularised* energy Hessians.

³A function $\mathcal{D}(\mathbf{F})$ is said to be polyconvex if there exists a function $\hat{\mathcal{D}}(\mathbf{F}, \mathbf{H}, J) = \mathcal{D}(\mathbf{F})$ such that $\hat{\mathcal{D}}$ is convex with respect to \mathbf{F} , \mathbf{H} , and J [Ball 1976]. Many non-convex problems are solvable, but if they are also non-polyconvex they exhibit numerical artefacts such as thinning and unnatural deformations.

4 OPTIMISATION APPROACH

We use a numerical continuation technique, coupled with a backtracking scheme [Nocedal and Wright 2006] inspired by FFM, that runs a series of minimisation by decreasing an auxiliary penalty parameter ϵ_k starting from $\epsilon_0 = 1$ (see line 9 of 2nd procedure in Algorithm 2). This parameter is also called the finite untangling sequence parameter. We compute β (Algorithm 1) in an outer loop so it remains constant for all elements and minimisation iterations. As shown in Algorithm 2, β can be computed using a wide range of techniques such as singular-value regularisation [Liao et al. 2021; Naitsat et al. 2020], Jacobian regularisation using a well-known empirical formula (line 11 of 2nd procedure) [Gargallo-Peiró et al. 2015], or using the finite untangling sequence [Garanzha et al. 2021a]. We have found the latter two result in faster convergence. In general, we perform the following (in case of per-element/analytical β , step 2 is not necessary and computing δ in Algorithm 2 is sufficient; see supplement for C++ code):

- (1) At every round (outer loop in Algorithm 2) compute the worst Jacobian, and then compute δ by setting $\epsilon_0 = 1$.
- (2) Compute the maximum regulariser β over the entire mesh.
- (3) Perform a full PN minimisation (inner loop in Algorithm 2):
 - (a) For elements with $J \leq 0$ compute the regularised invariants, energy, PK1 and filtered Hessian.
 - (b) For elements with $J > 0$ use the standard (non-regularised) invariants and compute the energy, PK1 and filtered Hessian.
- (4) At the end of every minimisation, if $J_{\min} \leq 0$, recompute β using ϵ_k based on a sufficient energy decrease, i.e., $\epsilon_k = \epsilon_{k-1}(1 - \mathcal{D}^k/\mathcal{D}^{k-1})$ and δ , and repeat.

4.0.1 Non-injective States. In a non-injective state, it is not necessary to drive the minimisation to gradient- or energy-based convergence. We show a simple example in Figure 6. Two opposite boundary vertices are moved inwards and optimised for two

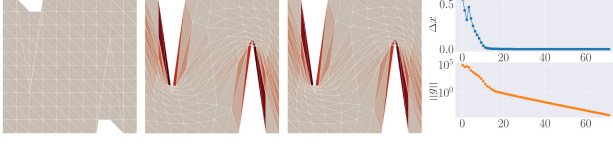


Fig. 6. Every minimisation round with a constant regulariser value can move the vertices towards local injectivity (but may not recover it). It is not necessary to drive these intermediate rounds to convergence. Starting from a non-injective map (left), the vertex displacements almost vanish (middle, iteration 13, $\|g\| = 14.56$) until convergence is reached (right, iteration 71, $\|g\| = 9.05 \times 10^{-5}$).

minimisation rounds. We show convergence of the first minimisation round, where displacements drop quickly despite a high residual. To detect such cases, we first determine if the number of fold-overs have remained unchanged in three consecutive iterations, and whether the displacements have dropped below minimum characteristic length of the mesh. The characteristic length is computed by scaling minimum edge lengths with the ratio of energy decrease. Similar and better alternatives can be used here [Nait-sat et al. 2020; Zhu et al. 2018]. Furthermore, when β is assumed constant, the positivity of J_r may not be guaranteed since new negative Jacobian may appear during iteration that exceed the value of precomputed β . In such cases, we assign the absolute minimum energy over the entire mesh to the corresponding flipped elements that exceed β , and filter their contributions from the line search.

5 NUMERICAL EXPERIMENTS

In this section, we provide results obtained with our framework and compare against other algorithms. For most studies, the code for other frameworks were obtained either from the authors or official project page. For fairness, we also implemented a per-element, numerically projected Newton, similar to the PN solvers used in Simplex Assembly [Fu and Liu 2016] and TLC-PN [Du et al. 2020a].

For completeness, we also implemented the Hessian modified version of FFM which performs a semidefinite-indefinite decomposition as presented in Garanzha et al. [2021a] (see Appendix A.2 in that paper and Appendix B here). We call this version *Modified-FFM*. The paper presented but did not implement the approach, and their final version of FFM used an L-BFGS solver. A direct comparison with other code bases give us approximate performance comparisons, but a clearer picture emerges by comparing our framework's results to our own implementation of other techniques such as Modified-FFM and numerical PN. Unless otherwise stated, all analyses were performed single-threaded on an Intel Xeon(R) E5-2650. We use MKL PARDISO, Eigen [Guennebaud and Jacob 2010] and Fastor [Poya et al. 2017] for tensor manipulations.

5.1 Surface Parametrisation

5.1.1 Comparing with FFM. We begin by comparing against FFM. We include our *Modified-FFM*. We chose two boundary surface parametrisation problems [Liu et al. 2018] with a small number of folded elements. For comparison with FFM, we used the same termination tolerance for both our analytic PN solver and FFM's L-BFGS solver (10^{-4}) with an outer loop tolerance of 10^{-5} . For fairness, we used the same ϵ_k -decay formula (finite untangling sequence [Garanzha et al. 2021a]) and back-substituted the analyt-

ALGORITHM 2: Optimise locally non-injective map

```

1: procedure OptimiseMap( $\mathcal{V}, \mathcal{T}$ )
2:    $k \leftarrow 0, \epsilon_0 \leftarrow 1, \rho \leftarrow 0.9, c \leftarrow 0.5, \text{threshold} \leftarrow 10^{-4}$ 
3:   repeat                                     ▷ Outer minimisation loop
4:      $\beta \leftarrow \text{ComputeRegulariser}(\mathcal{V}, \mathcal{T}, \epsilon_k)$ 
5:     while  $\|g\| > \text{threshold}$  do               ▷ Inner minimisation loop
6:        $\hat{\mathcal{D}}, g, \mathbb{H} \leftarrow \text{GetFuncGradProjectedHessian}(F, \beta)$  ▷ Use
        $\beta = 0$  for elements with  $J > 0$ 
7:        $\mathbb{H}\Delta x = -g$ 
8:        $\alpha \leftarrow \arg\min_{\alpha} \mathcal{D}(x + \alpha\Delta x)$            ▷ Line search
9:        $x \leftarrow x + \alpha\Delta x$ 
10:    end while
11:     $k \leftarrow k + 1$ 
12:     $\epsilon_k \leftarrow \max(10^{-6}, \epsilon_{k-1} \max(c, 1 - \hat{\mathcal{D}}^k / \hat{\mathcal{D}}^{k-1}))$ 
13:  until  $\hat{\mathcal{D}}^{k+1} > \rho \hat{\mathcal{D}}^k$  and  $J_{\min} > 0$ 
14: end procedure

1: procedure ComputeRegulariser( $\mathcal{V}, \mathcal{T}, \epsilon_k$ )
2:    $\sigma_{\min} \leftarrow \inf, J_{\min} \leftarrow \inf$ 
3:   for  $T \in \mathcal{T}$  do
4:     Compute SSVD( $F$ )                                     ▷ (7)
5:      $\sigma_{\min} \leftarrow \min(\min(\sigma_1, \dots, \sigma_i), \sigma_{\min})$ 
6:      $J_{\min} \leftarrow \min(\prod_i \sigma_i, J_{\min})$ 
7:   end for
8:   if SingularValueRegulariser then
9:      $\delta = \epsilon_k - \sigma_{\min}$                                      ▷ ABCD formula
10:  else if JacobianRegulariser then
11:     $\delta = \sqrt{\epsilon_k^2 + 0.04 \min(0, J_{\min})^2}$                  ▷ FFM empirical formula
12:  end if
13:  return ComputeBeta( $\mathcal{V}, \mathcal{T}, J_{\min}, \delta$ )                 ▷ Algorithm 1
14: end procedure

1: procedure GetFuncGradProjectedHessian( $F, \beta$ )
2:   Compute regularised energy  $\hat{\mathcal{D}}$                                ▷ (1a)
3:   Compute regularised gradient  $g$  using PK1                   ▷ (2a)
4:    $[\lambda_i, e_i] = \text{GetRegularisedEigenSystem}(F, \beta)$ 
5:    $\mathbb{H} \leftarrow \sum_i \max(0, \lambda_i) e_i \otimes e_i$  ▷ Project Hessian to SPD, (16), (18)
6:   return  $\hat{\mathcal{D}}, g, \mathbb{H}$ 
7: end procedure

```

ical β into energy derivatives to ensure the differences were only solver related. The volumetric term in FFM was chosen as $1/128$ which corresponded well with our F -based MIPS. We confirmed that the initially computed Jacobians matched.

Figure 5 shows comparison of our framework with FFM and Modified-FFM. All three repaired fold-overs successfully, but our regularisation achieved the same quality almost **5x faster**. Modified-FFM almost degraded to a first order in the subsequent iteration of outer loop but performed better than FFM.

5.1.2 Comparison with TLC. To compare against TLC, we used the David-to-H local injectivity benchmark from Du et al. [2020a] (Figure 7). We chose the regularised MIPS energy and symbolically generated a regularised **Advanced MIPS (AMIPS)** energy for quasi-isometric mapping. This adds a volumetric term to MIPS which can be written in terms of regularised stretch invariants as

$$\hat{\mathcal{D}}_{\text{RAMIPS}}(II_F, J) = \mu \frac{II_F}{2J_r} + \mu \left(1 - \frac{J}{J_r}\right) + \lambda (J_r + J_r^{-1}) - (2\lambda + \mu)$$

where μ and λ are, respectively, shape and area/volumetric coefficients.

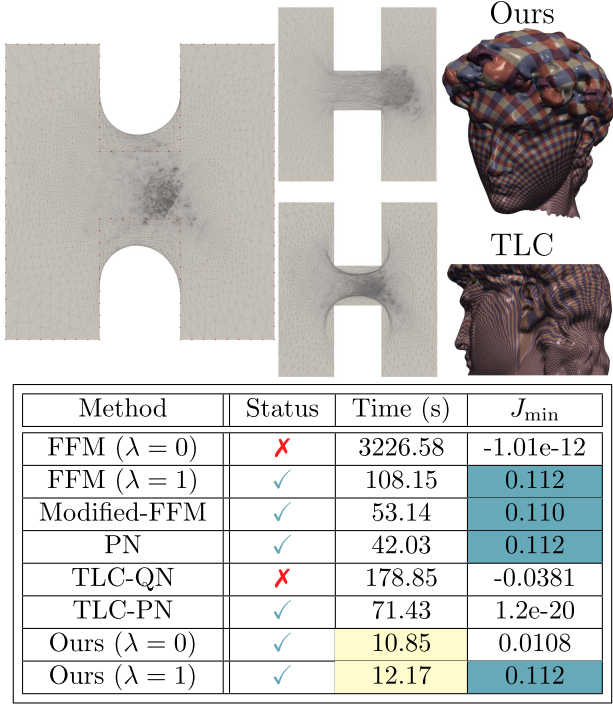


Fig. 7. David-to-H challenge from Du et al. [2020b], using our conformal (RMIPS) and quasi-isometric (RAMIPS, $\lambda = 1$) energies. For compatibility with TLC, all solvers were stopped when local injectivity was recovered. TLC’s QN solver and FFM conformal energy (MIPS only part) failed to recover local injectivity. Projected Newton and Modified-FFM converged quicker than QN. Our framework was up to **10X** faster than FFM and TLC.

We ran experiments for $\lambda = 0$ (MIPS) and $\lambda = 1$, and set $\mu = 1$ in both cases. All solvers were stopped as soon as local injectivity was recovered. Both FFM (with $\lambda = 0$) and TLC’s QN solvers failed to produce a valid “all” injective parametrisation for this model. With FFM, regardless of how we tuned ϵ_k , it was unable to solve the original MIPS energy. FFM with a volumetric term succeeded.

We further ran the Modified-FFM and a numerical PN solver for the case of quasi-isometric mappings ($\lambda = 1$). In this example, Modified-FFM was faster than FFM, and while PN converged quickly, the cost of the per-element eigen-decomposition dominated. On the other hand, our framework recovered injectivity up to **10X faster** (Figure 7). While this decomposition is embarrassingly parallel, we ran our benchmarks serially for fairness of comparison.

TLC generated huge distortion with $J_{\min} = 1.2 \times 10^{-20}$, which is 10^{19} times worse than the ones obtained by our RAMIPS energy. Best running times and qualities are highlighted in Figure 7.

5.1.3 Comparison with SA. Simplex Assembly (SA) [Fu and Liu 2016] can respect user-specified distortion bounds, but loses the robustness of TLC, FFM, or our framework, with respect to repairing fold-overs. SA minimises the same AMIPS energy as FFM, so it is natural to compare it against our regularised MIPS energy.

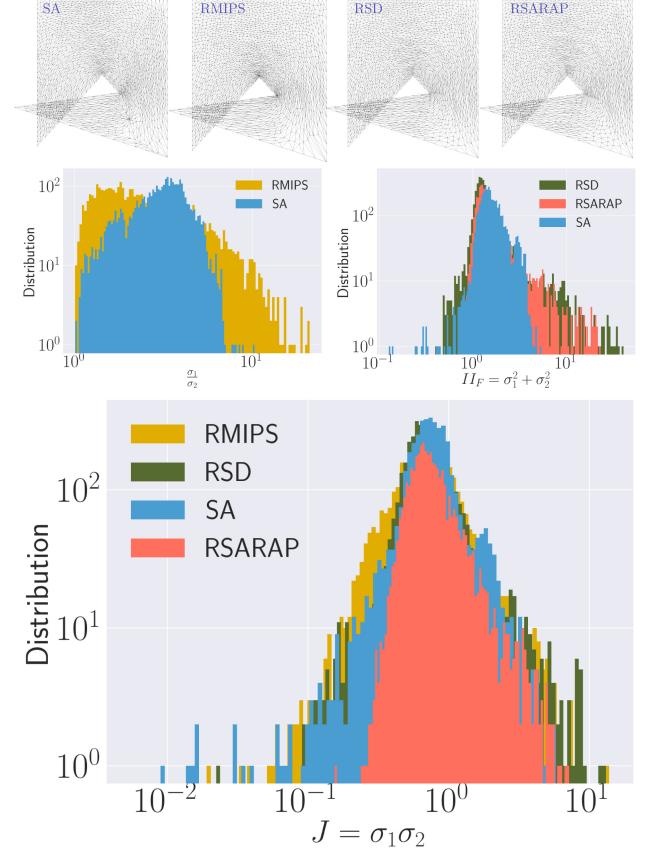


Fig. 8. Comparison with **Simplex Assembly (SA)** [Fu and Liu 2016]. SA minimises AMIPS energy, which results in better conformal quality, while our RSARAP and RSD energies result in better isometric distortion, while creating inversion-free maps similar to SA. Generally, SA creates elements with low J . Denser histograms about $J = 1$ are better.

We chose to analyse quality with the example in Weber and Zorin [2014], where a convex polygon is morphed into a concave shape using a harmonic mapping, resulting in 28 fold-overs (see inset). We also compared the results of our framework using RSD and RSARAP. As seen at the bottom of Figure 8, the quality with respect to J obtained with SA is comparable to our RMIPS and FFM.

While SA creates a few elements with extremely low J , our framework pushes the elements away from this zone. SA’s conformal results are strictly within a bound and better than our RMIPS. Given that RSD and RSARAP create an isometric mapping, the isometric distortion of RSARAP and RSD is lower.

5.2 Locally Injective Mappings Benchmark

We conducted a parametrisation study with SA, ABCD, FFM, TLC, and our regularised energies on the most challenging examples from the recent benchmark of Du et al. [2020b]: the Lucy-to-G and Gargoyle-to-S embeddings (Figure 9). To cover the spectrum from

conformal to isometric, we used RMIPS (and FFM with $\lambda = 1/128$, though quasi-isometric maps can also be obtained by increasing the value of λ), RSD and RSARAP energies. The initial embeddings for these models have $\sim 98K$ triangles and over 3K fold-overs, requiring the optimisation to move a large layer of elements into a concave shape. SA and ABCD (PN) were unable to create inversion-free domains, and respectively failed after ~ 300 and ~ 400 seconds. TLC (PN) succeeded, but yielded nearly collapsed elements ($J \approx 0$) due to stopping criterion. FFM succeeded but took over an hour on the Lucy model.

Our framework minimised the RMIPS energy almost **75 \times faster** than FFM on this model. It successfully minimised RMIPS, RSD, and RSARAP in a few seconds, and yielded high-quality, fixed-boundary, locally injective maps. The results of FFM and our RMIPS energy were visually identical, so only FFM is shown in Figure 9. However, the histograms should give a better representation of quality (J , conformal $\frac{\sigma_1}{\sigma_2}$ and isometric II_F) distribution.

We ran our regularised energies over all the 2D and 3D benchmarks of Du et al. [2020b], as shown in Figure 10. Our framework passed the entire benchmark (similar to TLC and FFM). The benchmark covers a variety of 2D parameterisations, 3D polycube parameterisations and 3D deformations, and range from easy to extremely difficult. We chose $\lambda = 0.01$ in 2D (to be compatible with FFM although, this creates low quality elements) and $\lambda = 1$ in 3D for the RMIPS energy. Unlike Du et al. [2020a]; Garanzha et al. [2021a], we used the significantly more challenging *source mesh* as the rest configuration, instead of an auxiliary “ideal” element.

We also ran a side-by-side comparison with FFM, Modified-FFM, and ABCD by tuning all solver parameters (e.g., number of steps, termination criteria, and ϵ_k decay (for FFM and ours)) to be identical. Figure 11 shows our framework was statistically **12.9 \times faster** than FFM over 10,000 2D meshes and **2 \times faster** over 900 3D meshes. ABCD did not have 100% success rate (91.1% in 2D and 88.3% in 3D) but, it was the second fastest method in 2D for cases that it succeeded (within 1.8 \times of our analytic PN). The performance of ABCD and our framework confirms that for 2D problems, PN solvers are preferred.

In many 3D cases, ABCD stalled at the repair phase and in general performed over **10 \times slower** (excluding failure times) compared to ours. For some 3D cases when the initialisation was close to the rest shape, FFM’s L-BFGS was faster than our framework. However, it had a more non-uniform convergence pattern and dispersed performance profile over the entire dataset. As expected, FFM’s performance was closer to ours in 3D since, unlike L-BFGS, the cost of the linear solver factors in for our framework. The second fastest method in 3D was our implementation of Modified-FFM, which was 43% slower than our analytic PN. This is because Modified-FFM, despite doing more iterations than analytic PN, does not require an SVD which dominates the assembly cost in 3D.

Overall, our framework was strictly faster than other methods on 72% (7728/10706) of the 2D cases and 65% (593/904) of the 3D cases. Such performance disparities between different optimisers have been well documented in the context of locally injective methods [Kovalsky et al. 2016; Rabinovich et al. 2017; Zhu et al. 2018]. Quality-wise, best performing methods were our RSD (see

Figure 10 instead) and ABCD, although all frameworks yielded comparable results.

5.3 Volumetric Polycube Parametrisation

Fixed boundary volumetric polycube parametrisation is an ideal candidate for our regularised energies, since fold-overs are common. In Figure 12 we show several examples with many initially folded tetrahedra, which we then optimised with the RMIPS energy. Given the fixed boundaries, we did not optimise the individual faces, which can lead to higher distortion. The exception is the Max Planck model, where we processed the individual faces (the smaller cube in Figure 12) using 2D RMIPS. This yielded a more uniform distribution of conformal distortion along the surface, and improved the quality of initial parametrisation.

5.4 Free Boundary Surface Parametrisation

In Figure 13 we embed multiple 3D models onto star-shaped domains, resulting in $\sim 10K$ fold-overs. We used RSARAP to optimise our map and obtained results similar to previous works [Claici et al. 2017; Rabinovich et al. 2017; Shtengel et al. 2017; Smith et al. 2019] in approximately the same number of iterations as Shtengel et al. [2017] and Smith et al. [2019]. Our RSARAP energy is stabler, and converges quickly despite starting from an invalid Tutte embedding. Since the energy has a barrier, regularising J decreases the residual, which accelerates convergence. These problems each took 10–12 iterations to converge.

In Figure 1, we start from an invalid mapping and use our RSD energy. As described in Section 2, using regularised barrier energies and filtered Hessians within PN solvers can cause oscillatory convergence, (see Figure 13), while unfiltered Hessians lead to stalls. Over a wide range of examples, we observed that our solver did not have this issue.

5.5 Globally Injective Surface Parametrisation

The purpose of our framework is to generate inversion-free mappings; a first step towards local injectivity. However, many frameworks (including ours), cannot ensure a locally injective map due to overwinding issue and k -covering traps (see Figure 4). Overlaps can occur under free boundaries, which means the parametrisation may not be globally injective as well. The authors of TLC fixed this issue with the **Smooth Excess Area (SEA)** proposal [Du et al. 2021], while FFM proposed using phantom triangulation [Garanzha et al. 2021b].

Our regularisation scheme allows for the complete re-use of existing techniques for locally injective mappings. To circumvent both overwinding and boundary self-intersection, we can use a “curve-repulsive” energy on the boundaries, such as those proposed in Smith and Schaefer [2015] and Su et al. [2020]. In Figure 14 we combined a simplified implementation of Smith and Schaefer [2015] contact penalty with our RSD energy to obtain a globally injective mesh parametrisation. We used the triangle inequality enhancement of Su et al. [2020] but did not use a surrounding shell mesh. In some hard cases, overwinding can still occur around interior vertices [Weber and Zorin 2014]. Furthermore, such boundary energies do not work when the initialisation has a self-intersecting

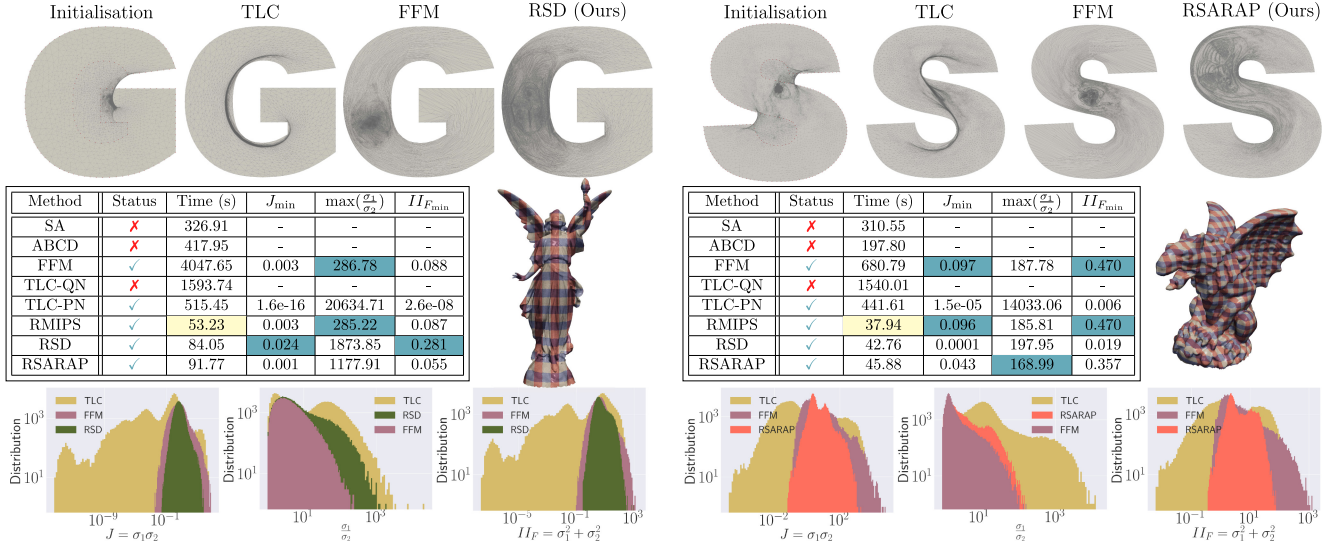


Fig. 9. The most challenging examples from Du et al. [2020b]. Of all tested frameworks, only TLC-PN and FFM recovered local injectivity. Our framework succeeded and minimised polyconvex MIPS and non-convex SD and SARAP an order of magnitude faster, creating equal or superior quality mappings. Denser histograms around $J = 1$, $\frac{\sigma_1}{\sigma_2} = 1$ and $II_F = 2$ are better.

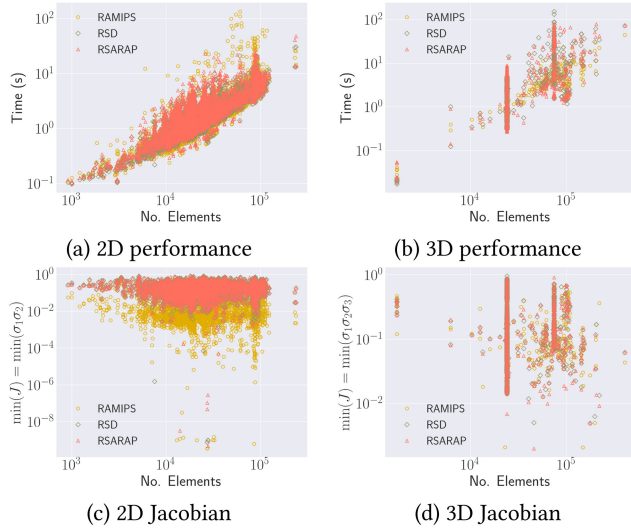


Fig. 10. Performance of our regularised energies over the benchmarks in Du et al. [2020b] showing (a) 10,706 meshes in 2D (b) 904 meshes in 3D.

boundary. However, this is a limitation of Smith and Schaefer [2015], and we leave a more thorough investigation to future work.

5.6 3D Mesh Deformation

5.6.1 Comparison with ABCD. We compare with **Adaptive Block Coordinate Descent (ABCD)** [Naitat et al. 2020] and FFM in 3D over some challenging test cases. ABCD follows a two-stage procedure in the presence of invalid elements, i.e., map repair and then optimisation. For repair, it uses a flip penalty distinct from the energy in the rest of the domain. In contrast, we minimise the *same* energy over all elements.

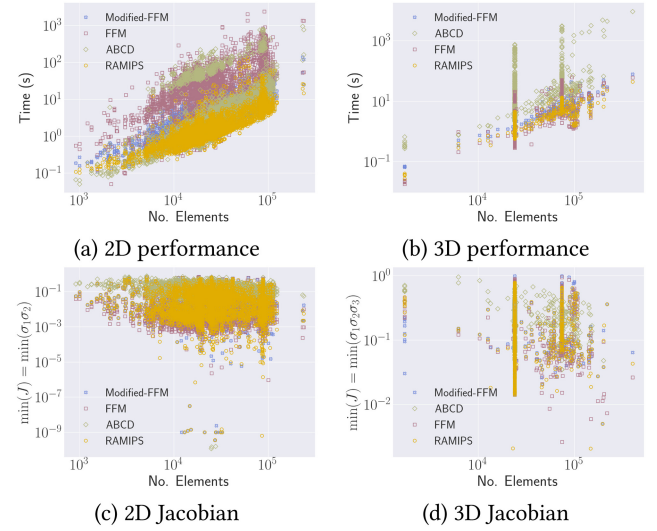


Fig. 11. Comparing against FFM, Modified-FFM, and ABCD over 2D and 3D benchmarks of Du et al. [2020b]. We used our RAMIPS model, which is closest to FFM. FFM and our framework passed the benchmark with 100% success rate. ABCD had 950 failures in 2D and 106 failures in 3D. The mean timing over 2D benchmarks were: RAMIPS 2.34s, Modified-FFM: 4.53s (1.93× slower), FFM: 30.19s (12.90× slower) and ABCD: [including failure cases - 11.49s (4.91× slower); excluding failure cases - 4.22s (1.80× slower)]. The mean timing over 3D benchmarks were: RAMIPS 2.84s, Modified-FFM: 4.05s (1.43× slower), FFM: 5.94s (2.09× slower), and ABCD: [including failure cases - 112.96s (39.77× slower); excluding failure cases - 29.82s (10.57× slower)].

In Figure 15 we compare against ABCD with modified Symmetric Dirichlet. For a twisting bar, ABCD and our framework create inversion-free maps with almost identical isometric quality; the histograms almost overlap. However, our RSD energy creates



Fig. 12. Fixed boundary conformal polycube parametrisation: Red meshes show the model and white-to-red shows conformal distribution. Labels indicate # of tetrahedrons, # of initial fold-overs, # of final fold-overs, and maximum conformal σ_1/σ_3 deviation.

elements with better Jacobian quality (J_{\min} in Figure 16). We traced this difference to ABCD’s termination criteria.

For these stress tests, our framework was generally faster than ABCD’s combined map repair and optimisation. With the exception of one example (Figure 16) where our framework was marginally slower (15% for wrench example) our framework converged before ABCD’s map repair completed. For a severely twisted Armadillo, ABCD (PN) failed under fixed-boundary, while our framework minimised both RSD and RAMIPS. As discussed in (Figure 11) however, our framework was statistically over $10\times$ faster than ABCD over a large 3D dataset. We also ran FFM for these 3D cases, and while it succeeded for fixed-boundary problems, it failed for the free-boundary Armadillo and wrench.

5.7 Pathological Test Cases

The 3D benchmark in (Figure 11) covers a wide range of practical problems, and our framework passes 100%. Nevertheless, it is possible to handcraft pathological tests that are more difficult to optimise. Following on from Figure 15, we optimise a severely twisted bar (Figure 17) by fixing its ends and using FFM, ABCD, and our RAMIPS and RSD energies. This is a corner case where both our RAMIPS energy and FFM generate inversion-free maps, but the final mesh overwinds around itself. ABCD generates a superior map, but with multiple twists and our RSD unfolds the bar to the least feasible number of twists with the same termination criterion. This difference is due to ABCD’s block-wise repair. Our framework instead needs to “blow” the whole mesh in order to un-flip hence, resulting in a global minimum. However, the difference between RAMIPS and RSD, lies in the underlying energy rather than our regularisation scheme.

We further stress test our framework with two randomised ogre heads that are then shrunk by $100\times$. In Figure 18, the second initialisation is challenging because intermediate configurations develop around two lines (fixed vertices around ears) and the optimiser has to step through low curvature regions. Neither FFM nor TLC are suitable for such cases, as they perform best under fixed boundary simulations. With our regularisation scheme we were able to

minimise RAMIPS/RSD/RSARAP and recover the rest shape, albeit under a much tighter 10^{-6} residual tolerance.

5.8 Comparison with Invertible Elasticity

Models that intentionally remove barriers in an attempt to extend energies to the infeasible zone are known as *invertible* [Irving et al. 2004; Smith et al. 2018; Stomakhin et al. 2012]. Irving et al. [2004] proposed an invertible model at the gradient level (PK1), resulting in an inconsistent treatment of gradients. Energetic consistency was restored by Stomakhin et al. [2012] via C^2 extension, while a state-of-the-art **Stable Neo-Hookean (SNH)** energy was proposed by Smith et al. [2018] which removed barriers altogether.

Our regularisation also extends distortion energies into the infeasible zone at the singular-value level, so how is it related? Invertible elasticity *removes* barriers, making it unsuitable for generating inversion-free mappings in the presence of positional constraints. Our regularisation *includes* the barrier for repaired elements, and only regularises degenerate elements. Once the map is repaired, our approach reduces to those of locally injective methods.

Invertible elasticity is not a suitable substitute for locally injective methods. We deform an elephant in Figure 19 by pushing the boundary vertices to extremes, and compare our RAMIPS energy with SNH [Smith et al. 2018]. While SNH is suitable for invertible simulations, it fails to generate an inversion-free mapping. With a traditional line search, the energy is minimised at a non-injective state, and with a Smith and Schaefer [2015] line search, the simulation stalls.

6 DISCUSSION AND CONCLUSION

We have presented a regularisation approach to geometric optimisation of locally non-injective mappings using popular flip-preventing energy. Our regularisation scheme is based on spectral shift in singular values, results in consistent formulation, and preserves convexity when applied on distortion energies (particularly, in 2D).

In addition, we have discussed the analytic eigensystems of the most important energies, such as conformal and isometric distortions. When expressed in terms of regularised stretch invariants. Through numerous examples in 2D and 3D, we have shown that our approach is extremely robust, and its second-order nature allows it to outperform state-of-the-art algorithms. For 2D surface parametrisation problems, our method appears to recover/optimize maps quicker than other methods we tested, while maintaining similar or superior quality. Given that, our framework can be tuned like FFM and ABCD its success rate is no less than these frameworks however, spectral shifting naturally also yields more stable Hessian eigensystems than Quasi-Newton methods which in turn eliminates numerical and divergence issues.

Our framework works well in practice, but has limitations. We do not guarantee bounded distortions. Further, there is not a theoretical guarantee that the global minimum of energy will always be an injective map. As discussed in Section 5.7, it can fail on pathological tests depending on the energy. Regularisation counters the fundamental assumptions of elasticity such as growth condition. The complexity of this issue was explored in Garanzha et al.

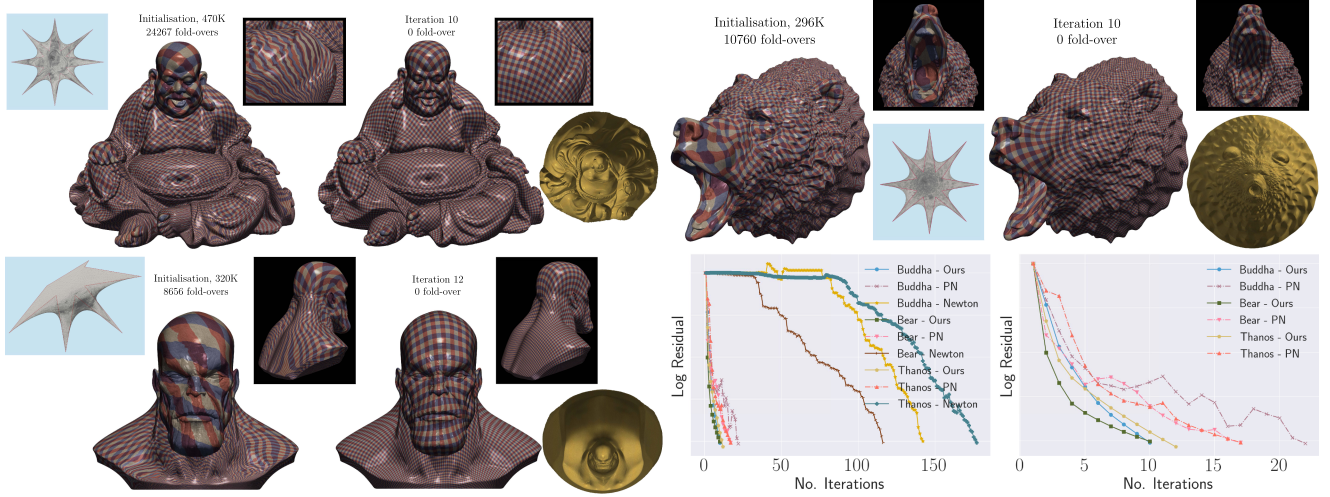


Fig. 13. Free boundary surface parametrisation starting from an invalid embeddings using regularised Symmetric ARAP. The original energy is non-convex and infinite, but our approach converges as quickly as Shtengel et al. [2017]; Smith et al. [2019]. Regularised infinite energies in PN solvers can have oscillatory convergence, and standard Newton experiences many stalls. The lower right graph compares PN to our approach.

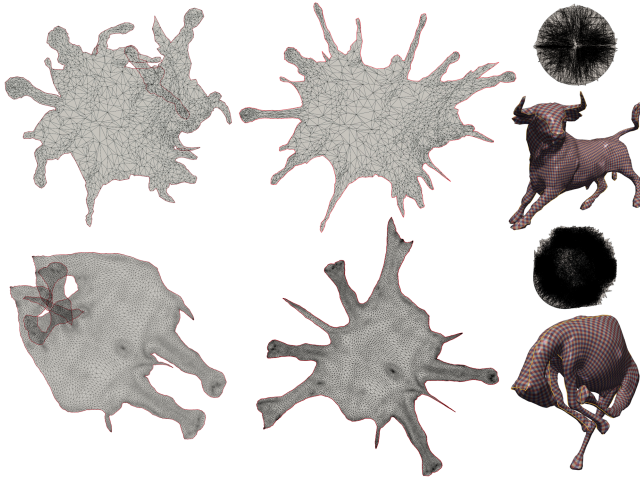


Fig. 14. We combine a prototype of Smith and Schaefer [2015] with our RSD energy and apply it to locally non-injective initialisations (Tutte + interior nodes randomised) to obtain globally injective parameterisations.

[2021a] as FFM shares the same limitation. Moreover, most barrier energies where singular value shifting applies are also non-convex. Nevertheless, the shift is a linear transform that does not deter convexity or nonlinearity.

Since our scheme generalises over the regularisation parameter, a future direction is to investigate better estimation of the regularisation parameter for other numerical optimisation problems.

APPENDICES

A WHY THE COFACTOR INVARIANTS?

Invariants that can be expressed in terms other invariants of the same family are common (e.g. $II_C^* = 1/2(I_C^2 - II_C)$ with Cauchy-Green) and can be useful when they have clean physical interpre-

tations. In terms of our extended invariants, there are multiple reasons to include the two additional cofactor invariants, I_H and II_H :

- A complete set of invariants should complete the characteristic polynomial of F (note I_H): $\sigma_i^3 - I_F \sigma_i^2 + I_H \sigma_i - J = 0$
- II_H corresponds to the Cauchy-Green invariant II_C , which does not appear in the Smith et al. [2019] invariants
- They appear in well-known energies such as Symmetric Dirichlet, Symmetric ARAP (see Table 1)
- They appear in our β parameter (see Algorithm 1)
- They enable closed-form study of polyconvexity; our supplement proves lack of polyconvexity for Symmetric Dirichlet
- They significantly simplify the algebra for gradient and Hessian expressions which become forbiddingly long when expressed in terms of singular-values or Smith et al. [2019].

B GRADIENT AND HESSIAN TENSORS

B.1 Gradient and Hessian of Distortion Energies

We can explicitly write generic expressions for the gradient (Equation (3)) and Hessian (Equation (4)) of distortion energies in terms of the extended stretch invariants (Equation (10)) without relying on analytic eigensystems. This allows us to study convexity, and perform convex-concave decompositions in closed-form. To obtain these, we first need the gradients of the invariants

$$\begin{aligned} G_{I_F} &= \frac{\partial I_F}{\partial F} = R, & G_{II_F} &= \frac{\partial II_F}{\partial F} = 2F, & G_J &= \frac{\partial J}{\partial F} = H \\ G_{I_H} &= \frac{\partial I_H}{\partial F} = I_F R - G, & G_{II_H} &= \frac{\partial II_H}{\partial F} = 2H \times F \end{aligned}$$

as well as the Hessians

$$\begin{aligned} \mathbb{H}_{I_F} &= \frac{\partial^2 I_F}{\partial F \partial F} = \sum_i \tilde{\lambda}_i \mathbb{T}_i, & \mathbb{H}_{II_F} &= \frac{\partial^2 II_F}{\partial F \partial F} = 2\mathbb{I}, & \mathbb{H}_J &= \frac{\partial^2 J}{\partial F \partial F} = \mathbb{I} \times F \\ \mathbb{H}_{I_H} &= \frac{\partial^2 I_H}{\partial F \partial F} = R \otimes R + I_F \sum_i \tilde{\lambda}_i \mathbb{T}_i - \mathbb{I}, & \mathbb{H}_{II_H} &= \frac{\partial^2 II_H}{\partial F \partial F} = 2F \times \mathbb{I} \times F, \end{aligned}$$

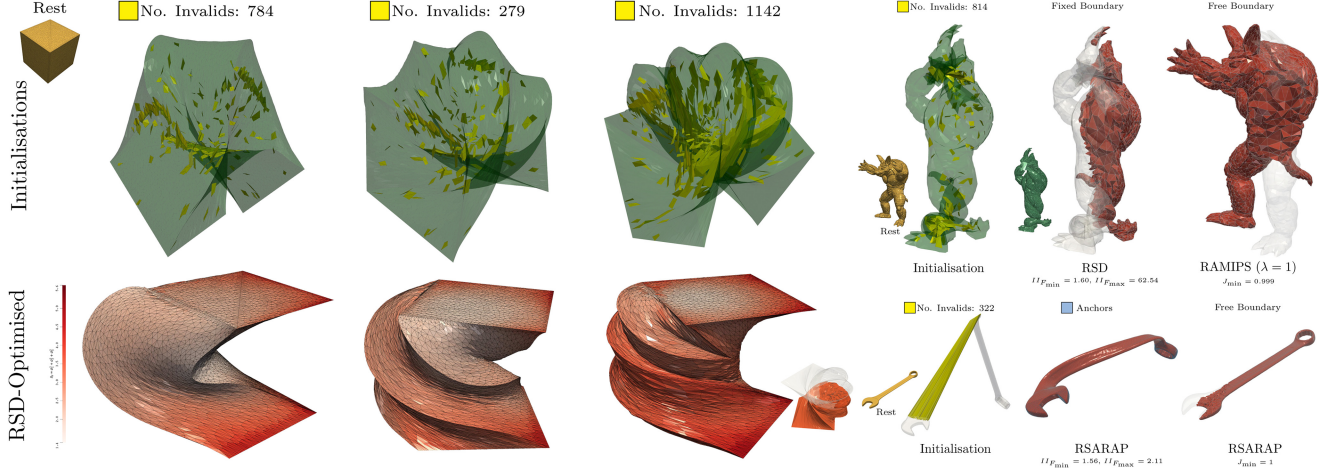


Fig. 15. 3D stress tests with RSD, RSARAP and RAMIPS energies: Twisted bar examples were initialised to: π , 1.8π and 4.25π with fixed boundaries. Armadillo with warped legs and arms (free boundary with one random point fixed). Wrench was sliced then oriented negatively (free boundary with one random point fixed). We compare our results with ABCD (PN). Our framework produces better quality distribution more quickly in all but one case (see Figure 16).

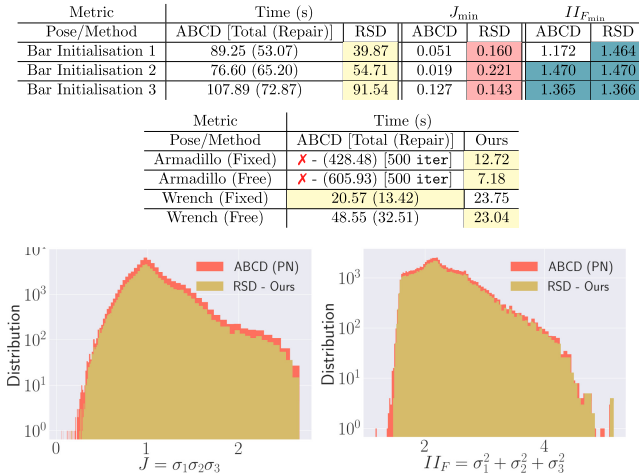


Fig. 16. Performance and quality metrics corresponding to Figure 15. In the top table RSD corresponds to our regularised Symmetric Dirichlet. Histograms show distribution of J and isometric distortion II_F for the first bar problem where most significant differences were observed in the quality.

where \mathbb{T}_i s are the twist tensor [Smith et al. 2019], $H \times F$ is the tensor cross product between two 3×3 tensors, and $\mathbb{I} \times F$ and $F \times \mathbb{I}$ are higher-order tensor cross products. Eigen code is given in the supplement. The gradient (Equation (3)) of a distortion energy can be obtained using the chain rule

$$\begin{aligned}
 P &= \frac{\partial \mathcal{D}}{\partial F} = \nabla_F \hat{\mathcal{D}}(I_F, II_F, I_H, II_H, J) \\
 &= \frac{\partial \hat{\mathcal{D}}}{\partial I_F} R + 2 \frac{\partial \hat{\mathcal{D}}}{\partial II_F} F + \frac{\partial \hat{\mathcal{D}}}{\partial I_H} (I_F R - F) + 2 \frac{\partial \hat{\mathcal{D}}}{\partial II_H} H \times F + \frac{\partial \hat{\mathcal{D}}}{\partial J} H.
 \end{aligned}$$

This is a $d \times d$ matrix and can be flattened in to a $d^2 \times 1$ vector. The Hessian can be expressed as

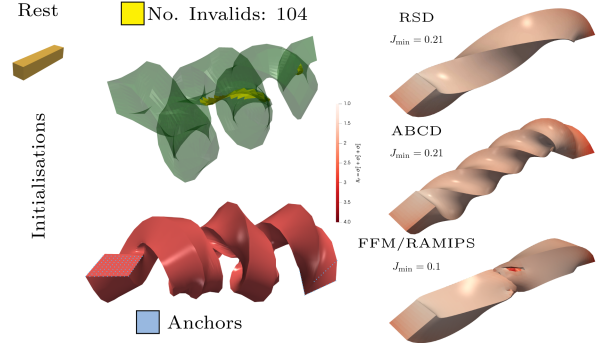


Fig. 17. Pathological tests: severely twisted bar with positional constraints. Our RSD model, ABCD, FFM (and RAMIPS model) all generate inversion-free maps (zero fold-overs) but with significantly different profiles. Both FFM and our RAMIPS create overwinding (not locally injective maps).

$$\begin{aligned}
 \mathbb{H} &= \frac{\partial^2 \mathcal{D}}{\partial F \partial F} = \nabla_F^2 \hat{\mathcal{D}}(I_F, II_F, I_H, II_H, J) \\
 &= \frac{\partial^2 \hat{\mathcal{D}}}{\partial I_F \partial I_F} R \otimes R + 4 \frac{\partial^2 \hat{\mathcal{D}}}{\partial II_F \partial II_F} F \otimes F + \frac{\partial^2 \hat{\mathcal{D}}}{\partial I_H \partial I_H} (I_F R - F) \otimes (I_F R - F) \\
 &\quad + 4 \frac{\partial^2 \hat{\mathcal{D}}}{\partial II_H \partial II_H} (H \times F) \otimes (H \times F) + \frac{\partial^2 \hat{\mathcal{D}}}{\partial J \partial J} H \otimes H \\
 &\quad + 2 \frac{\partial^2 \hat{\mathcal{D}}}{\partial I_F \partial I_H} \mathbb{I} + \frac{\partial^2 \hat{\mathcal{D}}}{\partial I_F \partial I_H} \mathbb{H}_{I_F} + \frac{\partial^2 \hat{\mathcal{D}}}{\partial I_H \partial I_H} \mathbb{H}_{I_H} + 2 \frac{\partial^2 \hat{\mathcal{D}}}{\partial II_H} F \times \mathbb{I} \times F + \frac{\partial^2 \hat{\mathcal{D}}}{\partial J} \mathbb{I} \times F \\
 &\quad + 2 \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial I_F \partial II_F} R \otimes F \right)^{\text{sym}} + \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial I_F \partial I_H} R \otimes (I_F R - F) \right)^{\text{sym}} \\
 &\quad + 2 \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial I_F \partial II_H} R \otimes (H \times F) \right)^{\text{sym}} + \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial I_F \partial J} R \otimes H \right)^{\text{sym}} \\
 &\quad + 2 \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial II_F \partial I_H} F \otimes (I_F R - F) \right)^{\text{sym}} + 4 \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial II_F \partial II_H} F \otimes (H \times F) \right)^{\text{sym}} \\
 &\quad + 2 \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial II_F \partial J} F \otimes H \right)^{\text{sym}} + 2 \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial I_H \partial II_H} (I_F R - F) \otimes (H \times F) \right)^{\text{sym}} \\
 &\quad + \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial I_H \partial J} (I_F R - F) \otimes H \right)^{\text{sym}} + 2 \left(\frac{\partial^2 \hat{\mathcal{D}}}{\partial II_H \partial J} (H \times F) \otimes H \right)^{\text{sym}} \quad (15)
 \end{aligned}$$

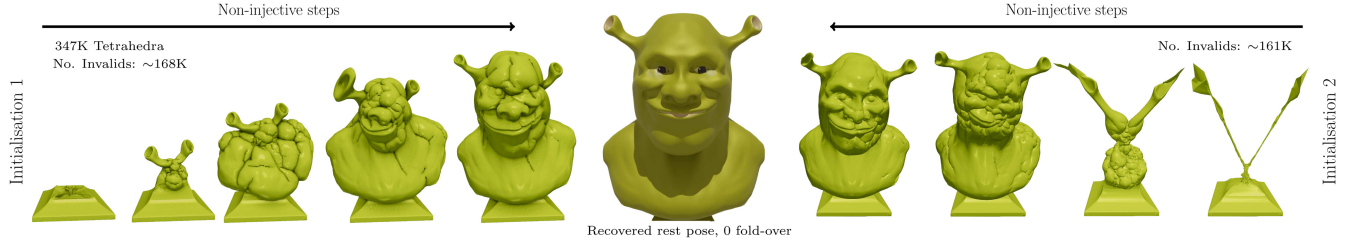


Fig. 18. Exploding ogre: We randomise and then shrink an ogre head by 100× from the rest pose and our algorithm recovers the original shape.

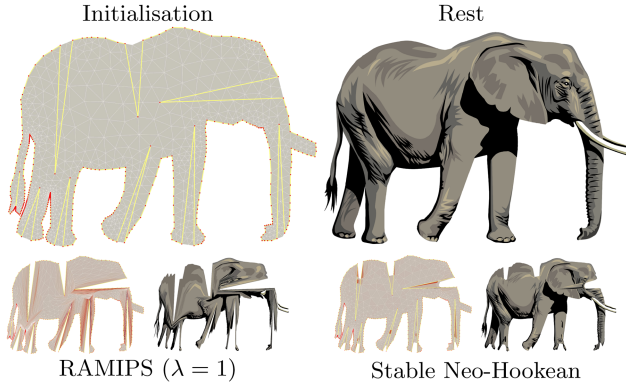


Fig. 19. Comparison with invertible elasticity: We compare our RAMIPS energy with Stable Neo-Hookean (SNH) with hard positional constraints; fixed boundary (red dots) and vertices moved to extremes (yellow lines). SNH energy fails to create an inversion-free mapping as it has no barrier, but we succeed because we only temporarily regularise barriers.

Blue terms are diagonal entries of the Hessian, and are second derivatives of the energy with respect to the same invariant. *Green* terms are off-diagonal entries that are second derivatives of the energy with respect to different invariants. *Red* terms are first derivatives of the energy with respect to the invariants, and form the geometric stiffness term [Bonet et al. 2016b]. The “sym” superscript indicates off-diagonal blocks with symmetric counterparts s.t. for every product $A \odot B$, there exists $B \odot A$, wherein \odot is any one of the tensor products along the off-diagonal. Equation (15) can be flattened in to a $d^2 \times d^2$ matrix.

For polyconvex energies, all terms except the last three *red* ones are SPD (the first two, \mathbb{I} and \mathbb{H}_{I_F} are SPD), so a convex-concave decomposition can be performed by discarding the last three red terms. Modified-FFM was implemented using this approach.

B.2 Eigenmatrices of the Invariants

Eigenmatrices of stretch invariants have been covered in detail [Kim and Eberle 2022; Kim et al. 2019; Smith et al. 2019], and three types arise: twist (\mathbb{T}), flip (\mathbb{L}), and scaling (D_i) tensors.

B.2.1 General Isotropic Energies. In 2D, to obtain the analytic eigensystem of an arbitrary isotropic energy, we have

$$\mathbb{H} = \lambda_1 E_1 \otimes E_1 + \lambda_2 E_2 \otimes E_2 + \lambda_3 \mathbb{L} + \lambda_4 \mathbb{T} \quad (16)$$

where \mathbb{T} and \mathbb{L} are twist and flip tensors and E_1 and E_2 are coupled scaling modes. These can be expressed using decoupled

scaling modes D_1 and D_2 as

$$E_1 = \frac{1}{\sqrt{1 + \tau^2}} (\tau D_1 + D_2), \quad E_2 = \frac{1}{\sqrt{1 + \tau^2}} (D_1 - \tau D_2) \quad (17)$$

where $\tau = 0$ if the scaling modes decouple. In 3D, the analytic eigensystem can be written as

$$\mathbb{H} = \sum_{i=1}^3 \left(\lambda_i E_i \otimes E_i + \lambda_{i+3} \mathbb{L}_i + \lambda_{i+6} \mathbb{T}_i \right). \quad (18)$$

Here, E_i can be computed by solving the 3×3 eigenvalue problem for the coupling matrix A

$$E_i = U \text{diag}(Q_i) V^T \quad (19)$$

where Q_i are the eigenvectors of A (see Smith et al. [2018, 2019]). We can now present the analytic eigensystem of our two new co-factor invariants, I_H and II_H .

B.2.2 Analytic Eigensystem of I_H . For the twist modes we have

$$\lambda_{i+3} = \frac{I_F + \sigma_i}{\sigma_j + \sigma_k} \quad E_{i+3} = T_i \quad i, j, k = 1, 2, 3, j \neq i, k \neq i, j$$

whereas flip modes have constant eigenvalues

$$\lambda_{i+6} = -1 \quad E_{i+6} = L_i \quad i = 1, 2, 3$$

and the coupled scaling modes

$$\lambda_1 = 2, \quad \lambda_{2,3} = -1 \quad E_i = U \text{diag}(Q_i) V^T \quad i = 1, 2, 3$$

where Q_i are the eigenvectors of $A = [0, 1, 1; 1, 0, 1; 1, 1, 0]$.

B.2.3 Analytic Eigensystem of II_H . For the twist modes we have

$$\lambda_{i+3} = 2(\sigma_j^2 + \sigma_i \sigma_k) \quad E_{i+3} = T_i \quad i, j, k = 1, 2, 3, j \neq i, k \neq i, j$$

and the flip modes have the same structure

$$\lambda_{i+6} = 2(\sigma_j^2 - \sigma_i \sigma_k) \quad E_{i+6} = L_i \quad i, j, k = 1, 2, 3, j \neq i, k \neq i, j.$$

The coupled scaling modes are from the eigensystem of:

$$A = 2 \begin{bmatrix} \sigma_2^2 + \sigma_3^2 & 2\sigma_1\sigma_2 & 2\sigma_1\sigma_3 \\ \text{sym} & \sigma_1^2 + \sigma_3^2 & 2\sigma_2\sigma_3 \\ & & \sigma_1^2 + \sigma_2^2 \end{bmatrix}$$

where we then apply Equation (19).

REFERENCES

- Noam Aigerman and Yaron Lipman. 2013. Injective and bounded distortion mappings in 3D. *ACM Trans. Graph.* 32, 4, Article 106 (July 2013), 14 pages.
- John M. Ball. 1976. Convexity conditions and existence theorems in nonlinear elasticity. *Archive for Rational Mechanics and Analysis* 63, 4 (1976), 337–403.
- John M. Ball. 1981. Global invertibility of Sobolev functions and the interpenetration of matter. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences* 88 (1981), 315–328.
- John M. Ball. 1983. Energy-minimising configurations in nonlinear elasticity. In *Proceedings of the International Congress of Mathematicians*. Warsaw.
- J. Bonet, A. J. Gil, and R. Ortigosa. 2015. A computational framework for polyconvex large strain elasticity. *Computer Methods in Applied Mechanics and Engineering* 283 (2015), 1061–1094.
- J. Bonet, A. J. Gil, and R. Ortigosa. 2016a. On a tensor cross product based formulation of large strain solid mechanics. *International Journal of Solids and Structures* 84 (2016), 49–63.
- J. Bonet, A. J. Gil, and R. D. Wood. 2016b. *Nonlinear Solid Mechanics for Finite Element Analysis: Statics* (3rd ed.). Cambridge University Press, Cambridge, UK.
- J. Brackbill and J. Saltzman. 1982. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.* 46 (1982), 342–368.
- Marcel Campen, Ryan Capouellez, Hanxiao Shen, Leyi Zhu, Daniele Panozzo, and Denis Zorin. 2021. Efficient and robust discrete conformal equivalence with boundary. *ACM Trans. Graph.* 40, 6, Article 261 (Dec. 2021), 16 pages. DOI: <https://doi.org/10.1145/3478513.3480557>
- Renjie Chen and O. Weber. 2017. GPU-accelerated locally injective shape deformation. *ACM Trans. Graph.* 36, 6 (2017), 1–13. DOI: <https://doi.org/10.1145/3130800.3130843>
- Sebastian Claici, Mikhail Bessmeltsev, S. Schaefer, and J. Solomon. 2017. Isometry-aware preconditioning for mesh parameterization. *Computer Graphics Forum* 36, 5 (2017), 37–47. DOI: <https://doi.org/10.1111/cgf.13243>
- R. de Boer. 1982. *Vektor- und Tensorrechnung für Ingenieure*. Springer.
- Xingyi Du, Noam Aigerman, Qingnan Zhou, Shahar Z. Kovalsky, Yajie Yan, Danny M. Kaufman, and Tao Ju. 2020a. Lifting simplices to find injectivity. *ACM Trans. Graph.* 39, 4, Article 120 (July 2020), 17 pages. DOI: <https://doi.org/10.1145/3386569.3392484>
- Xingyi Du, Noam Aigerman, Qingnan Zhou, Shahar Z. Kovalsky, Yajie Yan, Danny M. Kaufman, and Tao Ju. 2020b. Locally injective mappings benchmark (version 1.0). *Zenodo* (2020). DOI: <https://doi.org/10.5281/zenodo.3827969>
- Xingyi Du, Danny M. Kaufman, Qingnan Zhou, Shahar Z. Kovalsky, Yajie Yan, Noam Aigerman, and Tao Ju. 2021. Optimizing global injectivity for constrained parameterization. *ACM Trans. Graph.* 40, 6, Article 260 (Dec. 2021), 18 pages. DOI: <https://doi.org/10.1145/3478513.3480556>
- Xiao-Ming Fu and Yang Liu. 2016. Computing inversion-free mappings by simplex assembly. 35, 6, Article 216 (Nov. 2016), 12 pages.
- Xiao-Ming Fu, Yang Liu, and Baining Guo. 2015. Computing locally injective mappings by advanced MIPS. *ACM Trans. Graph.* 34, 4, Article 71 (July 2015), 12 pages.
- Xiao-Ming Fu, Jian-Ping Su, Qing Fang Zheng-Yu Zhao, Chunyang Ye, and Ligang Liu. 2021. Inversion-free geometric mapping construction: A survey. *Comp. Visual Media* 7, 58 (2021), 289–318. DOI: <https://doi.org/10.1007/s41095-021-0233-9>
- Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, Nicolas Ray, and Dmitry Sokolov. 2021a. Foldover-free maps in 50 lines of code. *ACM Trans. Graph.* 40, 4, Article 102 (July 2021), 16 pages.
- Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, Nicolas Ray, and Dmitry Sokolov. 2021b. On local invertibility and quality of free-boundary deformations. In *Proceedings of 29th International Meshing Roundtable*. 12.
- Abel Gargallo-Peiró, Xevi Roca, Jaime Peraire, and Josep Sarrate. 2015. Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *Internat. J. Numer. Methods Engrg.* 103, 5 (2015), 342–363.
- Mark Gillespie, Boris Springborn, and Keenan Crane. 2021. Discrete conformal equivalence of polyhedral surfaces. *ACM Trans. Graph.* 40, 4 (2021).
- Gaël Guennebaud and Benoît Jacob. 2010. Eigen V3. (2010). <http://eigen.tuxfamily.org>.
- S. Hartmann and P. Neff. 2003. Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility. *International Journal of Solids and Structures* 40 (2003), 2767–2791.
- G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible finite elements for robust simulation of large deformation. In *Proceedings of SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 131–140.
- Theodore Kim and David Eberle. 2022. Dynamic deformables: Implementation and production practicalities (now with code!). In *ACM SIGGRAPH 2022 Courses (SIGGRAPH'22)*. Association for Computing Machinery, New York, NY, USA, Article 7, 259 pages. DOI: <https://doi.org/10.1145/3532720.3535628>
- T. Kim, F. De Goes, and H. Iben. 2019. Anisotropic elasticity for inversion-safety and element rehabilitation. *ACM Trans. Graph.* 38, 4, Article 69 (July 2019), 15 pages.
- S. Kovalsky, N. Aigerman, R. Basri, and Y. Lipman. 2015. Large-scale bounded distortion mappings. *ACM Trans. Graph.* 34, 6, Article 191 (Oct. 2015), 10 pages.
- Shahar Z. Kovalsky, Meirav Galun, and Yaron Lipman. 2016. Accelerated quadratic proxy for geometric optimization. *ACM Trans. Graph.* 35, 4, Article 134 (July 2016), 11 pages. DOI: <https://doi.org/10.1145/2897824.2925920>
- Wentao Liao, Renjie Chen, Yuchen Hua, Ligang Liu, and Ofir Weber. 2021. Real-time locally injective volumetric deformation. *ACM Trans. Graph.* 40, 4, Article 74 (July 2021), 16 pages. DOI: <https://doi.org/10.1145/3450626.3459794>
- Yaron Lipman. 2014. Bijective mappings of meshes with boundary and the degree in mesh processing. *SIAM J. Imaging Sci.* 7 (2014), 1263–1283.
- Ligang Liu, Chunyang Ye, Ruiqi Ni, and Xiao-Ming Fu. 2018. Progressive parameterizations. *ACM Trans. Graph.* 37, 4, Article 41 (July 2018), 12 pages.
- A. Naitat, Y. Zhu, and Y. Zeevi. 2020. Adaptive block coordinate descent for distortion minimization. *Computer Graphics Forum* 39, 6 (2020), 360–376.
- Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization (second ed.)*. Springer, New York, NY. DOI: <https://doi.org/10.1007/978-0-387-40065-5>
- Roman Poya, Antonio J. Gil, and Rogelio Ortigosa. 2017. A high performance data parallel tensor contraction framework: Application to coupled electro-mechanics. *Computer Physics Communications* 216 (2017), 35–52.
- Roman Poya, Ruben Sevilla, and Antonio J. Gil. 2016. A unified approach for a posteriori high-order curved mesh generation using solid mechanics. *Computational Mechanics* 58, 3 (2016), 457–490. DOI: <https://doi.org/10.1007/s00466-016-1302-2>
- M. Rabinovich, R. Poranne, D. Panozzo, and O. Sorkine-Hornung. 2017. Scalable locally injective mappings. *ACM Trans. Graph.* 36, 2, Article 16 (April 2017), 16 pages.
- Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. 2013. Locally injective mappings. *Computer Graphics Forum (Proceedings of Symposium on Geometry Processing)* 32, 5 (2013).
- Hanxiao Shen, Zhongshi Jiang, Denis Zorin, and Daniele Panozzo. 2019. Progressive embedding. *ACM Trans. Graph.* 38, 4, Article 32 (July 2019), 13 pages.
- Anna Shtengel, Roi Poranne, Olga Sorkine-Hornung, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Geometric optimization via composite majorization. *ACM Trans. Graph.* 36, 4, Article 38 (July 2017), 11 pages.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean flesh simulation. *ACM Trans. Graph.* 37, 2, Article 12 (March 2018), 15 pages.
- B. Smith, F. De Goes, and T. Kim. 2019. Analytic eigensystems for isotropic distortion energies. *ACM Trans. Graph.* 38, 1, Article 3 (Feb. 2019), 15 pages.
- J. Smith and S. Schaefer. 2015. Bijective parameterization with free boundaries. *ACM Trans. Graph.* 34 (2015), 1–9. DOI: <https://doi.org/10.1145/2766947>
- Alexey Stomakhin, Russell Howes, Craig Schroeder, and Joseph M. Teran. 2012. Energetically consistent invertible elasticity. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*. The Eurographics Association.
- Jian-Ping Su, Xiaoming Fu, and L. Liu. 2019. Practical foldover-free volumetric mapping construction. *Computer Graphics Forum* 38 (2019).
- Jian-Ping Su, Chunyang Ye, Ligang Liu, and Xiao-Ming Fu. 2020. Efficient bijective parameterizations. *ACM Trans. Graph.* 39, 4, Article 111 (July 2020), 8 pages.
- Joseph Teran, Efthychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proceedings of SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, New York, NY, USA, 181–190.
- W. T. Tutte. 1963. How to draw a graph. *Proceedings of the London Mathematical Society* s3-13, 1 (1963), 743–767. DOI: <https://doi.org/10.1112/plms/s3-13.1.743>
- Ofir Weber, Ashish Myles, and Denis Zorin. 2012. Computing extremal quasiconformal maps. *Computer Graphics Forum* 31, 5 (Aug. 2012), 1679–1689.
- Ofir Weber and Denis Zorin. 2014. Locally injective parameterization with arbitrary fixed boundaries. *ACM Trans. Graph.* 33, 4, Article 75 (July 2014), 12 pages.
- Alan M. Winslow. 1966. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.* 1, 2 (1966), 149–172.
- H. Xu, F. Sin, Y. Zhu, and Jernej J. Barbic. 2015. Nonlinear material design using principal stretches. *ACM Trans. Graph.* 34, 4, Article 75 (July 2015), 11 pages.
- Yufeng Zhu. 2021. Eigen space of mesh distortion energy Hessian. (2021). <https://arxiv.org/pdf/2103.08141.pdf>.
- Yufeng Zhu, Robert Bridson, and Danny M. Kaufman. 2018. Blended cured Quasi-Newton for distortion optimization. *ACM Trans. Graph.* 37, 4 (2018), 14.

Received 21 July 2022; revised 12 January 2023; accepted 1 February 2023