Analytic Eigensystems for Isotropic Distortion Energies
Supplemental Material

BREANNAN SMITH, FERNANDO DE GOES, and THEODORE KIM, Pixar Animation Studios

ACM Reference Format:
Breannan Smith, Fernando de Goes, and Theodore Kim. 2018. Analytic Eigensystems for Isotropic Distortion Energies **Supplemental Material** . *ACM Trans. Graph.* 1, 1 (September 2018), 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 2D DEFORMATIONS VIA COMPLEX NUMBERS

The derivatives of distortion energies in $\mathbb{R}^2$ were recently considered by Chen and Weber [2017]. In contrast to our work, this method analyzed energies using Wirtinger derivatives obtained by expressing 2D deformations with complex numbers. We now review this complex-based formulation and compare its results to ours. Importantly, we show that Wirtinger derivatives are restricted to 2D and, therefore, the results in Chen and Weber [2017] *do not* extend to 3D. In contrast, ours is suited to both 2D and 3D.

1.1 Wirtinger Derivatives

Any 2D point (x, y) can be written as a complex number $\mathbf{z} = x + iy$. A 2D deformation can then be expressed by a complex function $\Phi(\mathbf{z}) = u(\mathbf{z}) + iv(\mathbf{z})$, where u and v map complex numbers to scalars. The Wirtinger derivatives of Φ are defined by:

$$\mathbf{w}_1 \equiv \frac{\partial \Phi}{\partial \mathbf{z}} = {}^1\!/_2 \left(\partial_x \Phi - i \partial_y \Phi\right) = {}^1\!/_2 \left(\partial_x u + \partial_y v\right) + {}^i\!/_2 \left(\partial_x v - \partial_y u\right),$$

$$\mathbf{w}_2 \equiv \frac{\partial \Phi}{\partial \overline{\mathbf{z}}} = {}^1\!/_2 \left(\partial_x \Phi + i \partial_y \Phi\right) = {}^1\!/_2 \left(\partial_x u - \partial_y v\right) + {}^i\!/_2 \left(\partial_x v + \partial_y u\right).$$

Using Euclidean coordinates (x, y), the 2D mapping Φ also defines a deformation gradient $\mathbf{F}$ of the form:

$$\begin{bmatrix} f_1 & f_3 \\ f_2 & f_4 \end{bmatrix} = \mathbf{F} \equiv \nabla \Phi = \begin{bmatrix} \partial_x u & \partial_y u \\ \partial_x v & \partial_y v \end{bmatrix}.$$

We can further decompose $\mathbf{F}$ into similarity and anti-similarity parts:

$$\mathbf{F} = {}^1\!/_2 \left(f_1 + f_4\right) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + {}^1\!/_2 \left(f_2 - f_3\right) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$+ {}^1\!/_2 \left(f_1 - f_4\right) \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + {}^1\!/_2 \left(f_2 + f_3\right) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

and then conclude that:

$$\mathbf{w}_1 = {}^1\!/_2 \left(f_1 + f_4\right) + {}^i\!/_2 \left(f_2 - f_3\right),$$
$$\mathbf{w}_2 = {}^1\!/_2 \left(f_1 - f_4\right) + {}^i\!/_2 \left(f_2 + f_3\right).$$

Therefore, the first Wirtinger derivative $\mathbf{w}_1$ corresponds to the similarity deformation encoded by $\mathbf{F}$, while the second Wirtinger derivative $\mathbf{w}_2$ contains the anti-similarity transformation. More concisely, we can express the Wirtinger derivatives as a simple change of variables applied to the deformation gradient. To this end, we flatten the deformation gradient into a vector, $\mathbf{f} = \text{vec}(\mathbf{F})$, and set the vector $\mathbf{w}$ to the concatenation of $\mathbf{w}_1$ and $\mathbf{w}_2$, i.e.:

$$\mathbf{w} = [w_i] = {}^1\!/_2 \begin{bmatrix} f_1 + f_4 & f_2 - f_3 & f_1 - f_4 & f_2 + f_3 \end{bmatrix}^\top.$$

It is then trivial to verify that $\mathbf{w} = {}^1\!/_2\, \mathbf{O}\, \mathbf{f}$, where $\mathbf{O}$ is the matrix

$$\mathbf{O} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Note, however, that the Wirtinger derivatives and the similarity decomposition of $\mathbf{F}$ have no equivalents in 3D. Consequently, the results in Chen and Weber [2017] are restricted to 2D deformations. We overcome this limitation by employing $\mathbf{f}$ as our primary representation, which leads to a unified formulation suited to eigenanalysis both in 2D and 3D.

1.2 Invariants

Isotropic distortion energies are determined entirely by rotation-invariant measures extracted from the deformation gradient $\mathbf{F}$, and can thus be expressed by functions of the singular values (σ_1, σ_2) of $\mathbf{F}$. As detailed in our main document, we compute the singular values of $\mathbf{F}$ using the SVD reflection convention from [Irving et al. 2004; Twigg and Kačić-Alesić 2010], which ensures that $\sigma_1 \geq |\sigma_2|$.

Some methods (e.g. [Stomakhin et al. 2012; Teran et al. 2005]) employed the singular values (σ_1, σ_2) as the energy invariants directly. This approach, however, leads to energy Hessians with no known analytical eigenstructure. Our work advocates instead the use of the invariants (I_1, I_2, I_3) derived from the stretch part $\mathbf{S}$ of $\mathbf{F}$. As shown in Section 4 of the main text, the eigensystem for these $\mathbf{S}$-based invariants can be expressed in closed-form. Observe that our 2D formulation intentionally includes a third, redundant invariant so that our results are consistent in 2D and 3D. In contrast, the work of Chen and Weber [2017] considered invariants computed based on the Wirtinger derivatives. These complex-based invariants can be written in terms of our $\mathbf{S}$-based invariants as follows:

$$\begin{cases} a_1 = \|\mathbf{w}_1\|^2 = {}^1\!/_4 \left(\sigma_1 + \sigma_2\right)^2 = {}^1\!/_4 \left(I_2 + 2I_3\right) = {}^1\!/_4\, I_1^2, \\ a_2 = \|\mathbf{w}_2\|^2 = {}^1\!/_4 \left(\sigma_1 - \sigma_2\right)^2 = {}^1\!/_4 \left(I_2 - 2I_3\right). \end{cases}$$

1.3 Analytical Eigensystem

Similar to our work, Chen and Weber [2017] considered the optimization of 2D isotropic distortion energies Ψ using a variant of the Newton method that projects the Hessian matrix at every quadrature point to positive semi-definiteness. In contrast to ours, Chen and Weber [2017] computed the derivatives of Ψ with respect to $\mathbf{w}$, instead of using the deformation gradient $\mathbf{f}$. As a side note, we point

Authors' address: Breannan Smith; Fernando de Goes; Theodore Kim, Pixar Animation Studios.

2018. 0730-0301/2018/9-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

out that [Chen and Weber 2017] adopted harmonic coordinates as their specific choice of basis functions so that quadrature points could be placed solely on the boundary of the 2D domain. Yet, their Hessian construction and eigenanalysis are agnostic to the basis choice and follow the same steps in Section 3 of our main document.

Analogous to Eqn. (9) of our main paper, Chen and Weber [2017] expanded the energy derivatives via the invariants (a_1, a_2), yielding:

$$\begin{cases} \dfrac{\partial \Psi}{\partial \mathbf{w}} = \sum_i \alpha_i \dfrac{\partial a_i}{\partial \mathbf{w}}, \\ \dfrac{\partial^2 \Psi}{\partial \mathbf{w}^2} = \sum_i \alpha_i \dfrac{\partial^2 a_i}{\partial \mathbf{w}^2} + \sum_{i,j} \beta_{ij} \left(\dfrac{\partial a_i}{\partial \mathbf{w}}\right)\left(\dfrac{\partial a_j}{\partial \mathbf{w}}\right)^\top, \end{cases}$$

where $\alpha_i = \partial\Psi/\partial a_i$ and $\beta_{ij} = \partial^2\Psi/\partial a_i \partial a_j$. Therefore, the Hessian eigensystem requires the analysis of the $\mathbf{w}$-based derivatives of (a_1, a_2). Based on the definitions in Section 1.2, we obtain:

$$\frac{\partial a_1}{\partial \mathbf{w}} = 2\begin{bmatrix} w_1 & w_2 & 0 & 0 \end{bmatrix}^\top \quad \frac{\partial a_2}{\partial \mathbf{w}} = 2\begin{bmatrix} 0 & 0 & w_3 & w_4 \end{bmatrix}^\top$$

$$\frac{\partial^2 a_1}{\partial \mathbf{w}^2} = 2\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \frac{\partial^2 a_2}{\partial \mathbf{w}^2} = 2\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and, consequently, the energy derivatives reduce to:

$$\begin{cases} \dfrac{\partial \Psi}{\partial \mathbf{w}} = 2\mathbf{A}\mathbf{w} \qquad \mathbf{A} = \begin{bmatrix} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_1 & 0 & 0 \\ 0 & 0 & \alpha_2 & 0 \\ 0 & 0 & 0 & \alpha_2 \end{bmatrix}, \\ \dfrac{\partial^2 \Psi}{\partial \mathbf{w}^2} = 2\mathbf{A} + 4\mathbf{B} \quad \mathbf{B} = \begin{bmatrix} \beta_{11}\mathbf{w}_1\mathbf{w}_1^\top & \beta_{12}\mathbf{w}_1\mathbf{w}_2^\top \\ \beta_{12}\mathbf{w}_2\mathbf{w}_1^\top & \beta_{22}\mathbf{w}_2\mathbf{w}_2^\top \end{bmatrix}. \end{cases}$$

By analyzing these derivatives, Chen and Weber [2017] found the eigenvalues λ_i and eigenvectors $\mathbf{e}_i$ of $\partial^2\Psi/\partial\mathbf{w}^2$ in closed-form for any 2D energy (see Eqns. (23) and (24) of their main text). For the purposes of comparison, we list their analytical expressions below:

$$\begin{aligned} \lambda_1 &= 2\alpha_1 & \mathbf{e}_1 &= \begin{bmatrix} w_2 & -w_1 & 0 & 0 \end{bmatrix}^\top \\ \lambda_2 &= 2\alpha_2 & \mathbf{e}_2 &= \begin{bmatrix} 0 & 0 & w_4 & -w_3 \end{bmatrix}^\top \\ \lambda_3 &= s_1 + \sqrt{s_2^2 + 16\beta_{12}^2 a_1 a_2} & \mathbf{e}_3 &= \begin{bmatrix} w_1 & w_2 & t_1 w_3 & t_1 w_4 \end{bmatrix}^\top \\ \lambda_4 &= s_1 - \sqrt{s_2^2 + 16\beta_{12}^2 a_1 a_2} & \mathbf{e}_4 &= \begin{bmatrix} w_1 & w_2 & t_2 w_3 & t_2 w_4 \end{bmatrix}^\top. \end{aligned}$$

Note that the eigenvectors are not normalized. Also, observe that these expressions depend on auxiliary variables given by:

$$\begin{aligned} s_1 &= \alpha_1 + 2\beta_{11}a_1 + \alpha_2 + 2\beta_{22}a_2 \\ s_2 &= \alpha_1 + 2\beta_{11}a_1 - \alpha_2 - 2\beta_{22}a_2 \\ t_1 &= \left(\lambda_3 - 2\alpha_1 - 4\beta_{11}a_1\right) / \left(4\beta_{12}a_2\right) \\ t_2 &= \left(\lambda_4 - 2\alpha_1 - 4\beta_{11}a_1\right) / \left(4\beta_{12}a_2\right) \end{aligned}$$

Crucially, the variables t_1 and t_2 may be ill-defined due to the division by $\beta_{12}a_2$, which can be zero for any energy with $\beta_{12} = 0$ or when $a_2 = 0$ (which is equivalent to $\sigma_1 = \sigma_2$). Moreover, the case of $a_2 = 0$ implies $\mathbf{w}_2 = 0$ and thus $\mathbf{e}_2 = 0$, which is not a valid eigenvector. These numerical issues can be resolved by rederiving the Hessian eigensystem for these special cases, but to our knowledge these special configurations must be addressed in a case-by-case basis in Chen and Weber [2017]. Therefore, the most generic form of the eigensystem presented by Chen and Weber [2017] requires numerical surgery for any energy, even if the shape is at rest. In sharp contrast, our $\mathbf{S}$-based formulation is well-defined for any state of the deformation gradient and for any energy both in 2D and 3D.

2 COMPOSITE MAJORIZATION FOR ARAP

In this section, we describe two derivations of composite majorization for the ARAP energy in 2D. As noted in Shtengel et al. [2017], composite majorization depends on the choice of a function's composite and convex-concave decomposition. Both of these choices are not unique, and distinct choices may lead to different approximations. Here, we present two of these choices and demonstrate how the differences can be indeed significant. We then employ these expressions for comparisons against our projected Newton solver. We report these results in §6.1 of the main text and in Table 16.

2.1 Comp. Majorization for 2D ARAP: Version 1

Our first approach to derive composite majorization for the ARAP energy uses the invariants of the stretch tensor $\mathbf{S}$. In this case, the ARAP energy per quadrature point is written as $\Psi_{\text{ARAP}} = I_2 - 2I_1 + 2$. Using the notation from Shtengel et al. [2017], the pair of invariants I_1 and I_2 corresponds to the their function "g" and Ψ_{ARAP} is their function "h". Note that Ψ_{ARAP} is linear in terms of I_1 and I_2, so its second derivatives are zero and the convex majorizer includes only the second derivatives of the invariants. While the Hessian of I_2 is always positive definite, the Hessian of I_1 is positive definite if and only if $I_1 \geq 0$. Using the SVD reflection convention from [Irving et al. 2004; Twigg and Kačić-Alesić 2010], which combines reflections with rotations, we can ensure that $\sigma_1 \geq |\sigma_2|$ and, consequently, $I_1 \geq 0$. Therefore no convex-concave decomposition is needed. Given that the first derivatives of Ψ_{ARAP} are -2 for I_1 and 1 for I_2, we can employ Eqn. (9) of [Shtengel et al. 2017] and conclude:

$$\frac{\partial^2 \Psi_{\text{ARAP}}}{\partial \mathbf{f}^2} \approx 2\mathbf{I}. \tag{1}$$

As discussed in §5.4 of our main text, an identity $\mathbf{F}$-based Hessian contributes to the full Hessian with a Laplacian matrix. This implies that this specific majorizer approximation is equivalent to [Kovalsky et al. 2016], and thus degrades the convergence of the ARAP optimization to first-order.

2.2 Comp. Majorization for 2D ARAP: Version 2

Shtengel et al. [2017] proposed to set the composite function g to the singular values (σ_1, σ_2) of each quadrature point. The ARAP energy is then expressed as $\Psi_{\text{ARAP}} = (\sigma_1 - 1)^2 + (\sigma_2 - 1)^2$. The derivatives of Ψ_{ARAP} with respect to the singular values are: $\partial\Psi_{\text{ARAP}}/\partial\sigma_i = 2(\sigma_i - 1)$ and $\partial^2\Psi_{\text{ARAP}}/\partial\sigma_i\partial\sigma_j = 2\delta_{ij}$, where δ_{ij} is the Kronecker delta. To address inverted elements, we again adopt the SVD reflection convention presented in [Irving et al. 2004; Twigg and Kačić-Alesić 2010], which guarantees that $\sigma_1 \geq |\sigma_2|$ and, consequently, $I_1 \geq 0$.

The formulation of Shtengel et al. [2017] expanded the expressions for the singular values based on the similarity decomposition of the deformation gradient $\mathbf{F}$, similar to Chen and Weber [2017]. Using the notation from Section 1.1, we denote with $\mathbf{w}_1$ the similarity part of $\mathbf{F}$ and use $\mathbf{w}_2$ for its anti-similarity part. We then have the following identities:

$$\begin{cases} b_1 \equiv \|\mathbf{w}_1\| = 1/2\,(\sigma_1 + \sigma_2) = 1/2\sqrt{I_2 + 2I_3} = 1/2\; I_1, \\ b_2 \equiv \|\mathbf{w}_2\| = 1/2\,(\sigma_1 - \sigma_2) = 1/2\sqrt{I_2 - 2I_3}. \end{cases}$$

Using the eigensystems of the S-based invariants presented in our main text, it can be shown that:

$$\frac{\partial b_1}{\partial \mathbf{f}} = \frac{1}{2}\mathbf{r} \qquad \frac{\partial^2 b_1}{\partial \mathbf{f}^2} = \frac{1}{2b_1}\mathbf{t}\mathbf{t}^\top$$
$$\frac{\partial b_2}{\partial \mathbf{f}} = \frac{1}{4b_2}(\mathbf{f} - \mathbf{g}) = \frac{\sqrt{2}}{2}\mathbf{p} \qquad \frac{\partial^2 b_2}{\partial \mathbf{f}^2} = \frac{1}{2b_2}\mathbf{l}\mathbf{l}^\top.$$

Note that the Hessians of both b_1 and b_2 are always positive definite because $b_1 \geq 0$ and $b_2 \geq 0$.

Shtengel et al. [2017] also proposed to decompose the singular values into convex functions $g^+ = (b_1 + b_2, b_1)$ and concave functions $g^- = (0, -b_2)$ so that $g^+ + g^- = (\sigma_1, \sigma_2)$. By directly substituting these expressions into Eqn. (9) of [Shtengel et al. 2017], the convex majorizer produces:

$[\sigma_1 \geq \sigma_2 \geq 1]$

$$\frac{\partial^2 \Psi_{\text{ARAP}}}{\partial \mathbf{f}^2} \approx \mathbf{r}\mathbf{r}^\top + 2\mathbf{p}\mathbf{p}^\top + 2(\sigma_1 - 1)\left(\frac{1}{2b_1}\mathbf{t}\mathbf{t}^\top + \frac{1}{2b_2}\mathbf{l}\mathbf{l}^\top\right) + 2(\sigma_2 - 1)\frac{1}{2b_1}\mathbf{t}\mathbf{t}^\top$$

$[\sigma_1 \geq 1 > \sigma_2]$

$$\frac{\partial^2 \Psi_{\text{ARAP}}}{\partial \mathbf{f}^2} \approx \mathbf{r}\mathbf{r}^\top + 2\mathbf{p}\mathbf{p}^\top + 2(\sigma_1 - 1)\left(\frac{1}{2b_1}\mathbf{t}\mathbf{t}^\top + \frac{1}{2b_2}\mathbf{l}\mathbf{l}^\top\right) - 2(\sigma_2 - 1)\frac{1}{2b_2}\mathbf{l}\mathbf{l}^\top$$

$[1 > \sigma_1 \geq \sigma_2]$

$$\frac{\partial^2 \Psi_{\text{ARAP}}}{\partial \mathbf{f}^2} \approx \mathbf{r}\mathbf{r}^\top + 2\mathbf{p}\mathbf{p}^\top - 2(\sigma_2 - 1)\frac{1}{2b_2}\mathbf{l}\mathbf{l}^\top.$$

The left-most $(\mathbf{r}, \mathbf{p})$ terms correspond to the Gauss-Newton-like part of Eqn. (9) from [Shtengel et al. 2017], while the right-most terms include the Hessians of the convex and concave functions (g^+, g^-). After algebraic manipulation, we can rewrite this Hessian approximation in terms of the exact ARAP Hessian presented in §5.1 of our main text:

$[\sigma_1 \geq \sigma_2 \geq 1]$

$$\frac{\partial^2 \Psi^{\text{CM}}_{\text{ARAP}}}{\partial \mathbf{f}^2} = \frac{\partial^2 \Psi_{\text{ARAP}}}{\partial \mathbf{f}^2} + 2(\sigma_2 - 1)\frac{1}{2b_2}\mathbf{l}\mathbf{l}^\top$$

$[\sigma_1 \geq 1 > \sigma_2]$

$$\frac{\partial^2 \Psi^{\text{CM}}_{\text{ARAP}}}{\partial \mathbf{f}^2} = \frac{\partial^2 \Psi_{\text{ARAP}}}{\partial \mathbf{f}^2} - 2(\sigma_2 - 1)\frac{1}{2b_1}\mathbf{t}\mathbf{t}^\top$$

$[1 > \sigma_1 \geq \sigma_2]$

$$\frac{\partial^2 \Psi^{\text{CM}}_{\text{ARAP}}}{\partial \mathbf{f}^2} = \frac{\partial^2 \Psi_{\text{ARAP}}}{\partial \mathbf{f}^2} - 2(\sigma_1 - 1)\left(\frac{1}{2b_1}\mathbf{t}\mathbf{t}^\top + \frac{1}{2b_2}\mathbf{l}\mathbf{l}^\top\right) - 2(\sigma_2 - 1)\frac{1}{2b_1}\mathbf{t}\mathbf{t}^\top.$$

By inspecting the exact ARAP Hessian, it can be verified that the case of $\sigma_1 \geq \sigma_2 \geq 1$ guarantees positive semi-definiteness. However, composite majorization returns just Hessian approximations. In sharp contrast, our method reproduces positive semi-definite Hessians exactly. This discrepancy justifies our superior performance compared to composite majorization, as reported in Table 16.

3 EIGENSYSTEM OF II_C

The eigensystem of the 3D second Cauchy-Green invariant $II_C = \|\mathbf{C}\|^2$ is as follows:

$$\begin{aligned}
\lambda_1 &= 12\sigma_1^2 & \mathbf{q}_1 &= \mathbf{d}_1\\
\lambda_2 &= 12\sigma_2^2 & \mathbf{q}_2 &= \mathbf{d}_2\\
\lambda_3 &= 12\sigma_3^2 & \mathbf{q}_3 &= \mathbf{d}_3\\
\lambda_4 &= 4(\sigma_2^2 + \sigma_3^2 - \sigma_2\sigma_3) & \mathbf{q}_4 &= \mathbf{t}_1\\
\lambda_5 &= 4(\sigma_2^2 + \sigma_3^2 + \sigma_2\sigma_3) & \mathbf{q}_5 &= \mathbf{l}_1\\
\lambda_6 &= 4(\sigma_1^2 + \sigma_3^2 - \sigma_1\sigma_3) & \mathbf{q}_6 &= \mathbf{t}_2\\
\lambda_7 &= 4(\sigma_1^2 + \sigma_3^2 + \sigma_1\sigma_3) & \mathbf{q}_7 &= \mathbf{l}_2\\
\lambda_8 &= 4(\sigma_1^2 + \sigma_2^2 - \sigma_1\sigma_2) & \mathbf{q}_8 &= \mathbf{t}_3\\
\lambda_9 &= 4(\sigma_1^2 + \sigma_2^2 + \sigma_1\sigma_2) & \mathbf{q}_9 &= \mathbf{l}_3
\end{aligned}$$

4 3D EIGENSYSTEM OF I_3

For completeness, we list the eigenvalues and eigenvectors for the Hessian of I_3 in 3D. The first three eigenvalues are the roots of a depressed cubic and can be written as:

$$\lambda_i = 2\sqrt{\frac{I_2}{3}}\cos\left[\frac{1}{3}\left(\arccos\left(\frac{3I_3}{I_2}\sqrt{\frac{3}{I_2}}\right) + 2\pi(i-1)\right)\right]. \qquad (2)$$

Note that these three eigenvalues reduce to zero in the special case when $\mathbf{F} = 0$. The corresponding eigenvectors are:

$$\mathbf{e}_i = \frac{1}{\varepsilon_i}\sum_j z_{ij}\mathbf{d}_j \quad \text{with} \quad \begin{cases} z_{i1} = \sigma_1\sigma_3 + \sigma_2\lambda_i \\ z_{i2} = \sigma_2\sigma_3 + \sigma_1\lambda_i \\ z_{i3} = \lambda_i^2 - \sigma_3^2 \end{cases}$$

and $\varepsilon_i = \sqrt{\sum_j z_{ij}^2}$ is a normalization factor. The last six eigenpairs defined over $i \in \{1, 2, 3\}$ are:

$$\begin{aligned} \lambda_{i+3} &= \sigma_i & \mathbf{e}_{i+3} &= \mathbf{t}_i \\ \lambda_{i+6} &= -\sigma_i & \mathbf{e}_{i+6} &= \mathbf{l}_i. \end{aligned}$$

5 MATLAB VERIFICATION

We provide Matlab code to verify the eigensystem expressions arrived at via our approach. We provide verification scripts for the ARAP, co-rotational, Symmetric Dirichlet, and Symmetric ARAP energies in both 2D and 3D. Each energy has a direct implementation of its distortion energy, gradient and Hessian. The correctness of the gradient and Hessian implementations are numerically verified using finite differences.

The numerical distortion energy and Hessian are then converted to symbolic code, and the symbolic eigensystem is constructed by using the expressions from the main paper. The symbolic eigensystem is then multiplied against the symbolic Hessian to confirm that it is indeed a valid eigenpair. All of the provided energies pass this sequence of tests. For further instructions on running these scripts, see the `README.txt` provided with the scripts.

6 PARAMETERIZATION PERFORMANCE

In Figures 6 through 7 we provide detailed plots of the Newton solver's progress for Symmetric Dirichlet-based parameterizations on the meshes from §6.1 of the main text. We also summarize the performance statistics for computing surface parameterizations using the Symmetric Dirichlet (Table 8) and the ARAP (Table 16) energies. We tested each energy and each method against 41 meshes drawn from the Thingi10K model database [Zhou and Jacobson 2016], the McGuire Computer Graphics Archive [McGuire 2017], Keenan Crane's 3D Model Repository [Crane 2018], the corpus of canonical graphics meshes, and a collection of production meshes.

REFERENCES

R. Chen and O. Weber. 2017. GPU-accelerated Locally Injective Shape Deformation. *ACM Trans. Graph.* 36, 6 (2017).

K. Crane. 2018. Keenan's 3D Model Repository. https://www.cs.cmu.edu/ kmcrane/Projects/ModelRepository/. (2018).

G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible Finite Elements for Robust Simulation of Large Deformation. In *SIGGRAPH/Eurog. Symp. on Comp. Anim.* 131–140.

S. Z. Kovalsky, M. Galun, and Y. Lipman. 2016. Accelerated Quadratic Proxy for Geometric Optimization. *ACM Trans. Graph.* 35, 4 (2016).

M. McGuire. 2017. Computer Graphics Archive. (2017). https://casual-effects.com/data

A. Shtengel, R. Poranne, O. Sorkine-Hornung, S. Z. Kovalsky, and Y. Lipman. 2017. Geometric Optimization via Composite Majorization. *ACM Trans. Graph.* 36, 4 (2017).

A. Stomakhin, R. Howes, C. Schroeder, and J. M. Teran. 2012. Energetically Consistent Invertible Elasticity. In *ACM SIGGRAPH/Eurog. Symp. Comp. Anim.* 25–32.

J. Teran, E. Sifakis, G. Irving, and R. Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation. In *ACM SIGGRAPH/Eurog. Symp. on Comp. Anim.* 181–190.

C. D. Twigg and Z. Kačić-Alesić. 2010. Point Cloud Glue: Constraining simulations using the Procrustes transform. In *ACM SIGGRAPH/Eurog. Symp. on Comp. Anim.* 45–54.

Q. Zhou and A. Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).

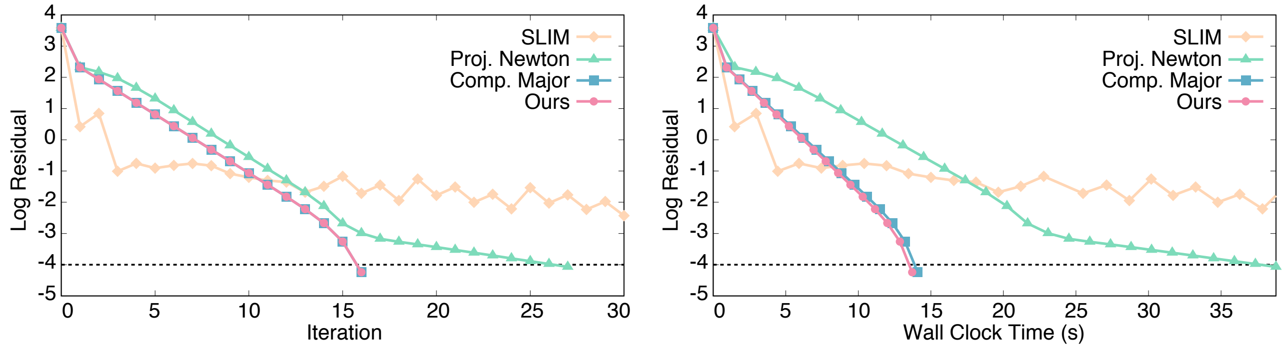

Fig. 1. Solver iterations (left) and wall clock time (right) to compute a parameterization with Symmetric Dirichlet over the bear mesh using our method (pink circles), composite majorization (blue squares), per-element numerical projection (green triangles), and SLIM (yellow diamonds). The termination threshold is denoted by a dashed line.

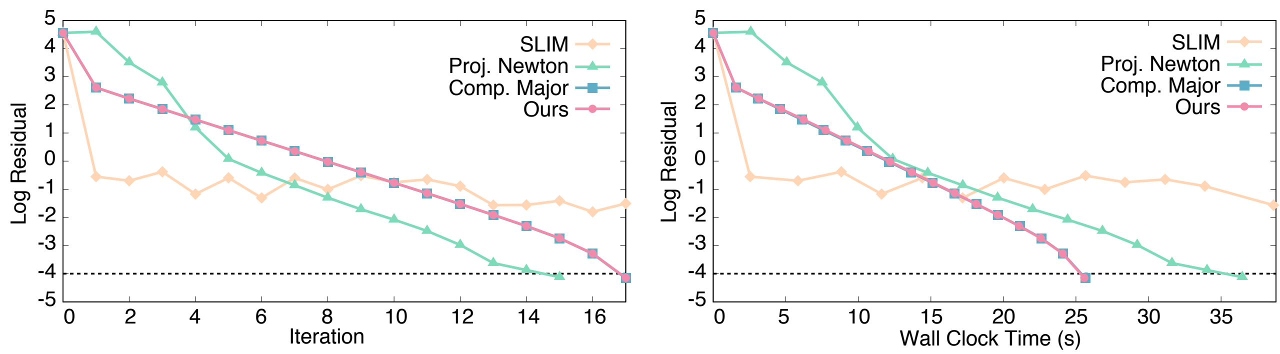

Fig. 2. Solver iterations (left) and wall clock time (right) to compute a parameterization with Symmetric Dirichlet over the Buddha mesh using our method (pink circles), composite majorization (blue squares), per-element numerical projection (green triangles), and SLIM (yellow diamonds). The termination threshold is denoted by a dashed line.

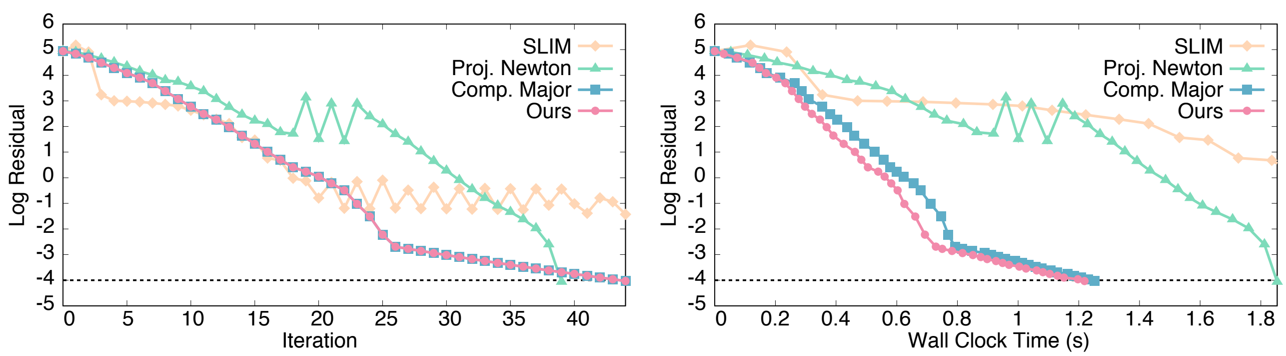

Fig. 3. Solver iterations (left) and wall clock time (right) to compute a parameterization with Symmetric Dirichlet over the car mesh using our method (pink circles), composite majorization (blue squares), per-element numerical projection (green triangles), and SLIM (yellow diamonds). The termination threshold is denoted by a dashed line.

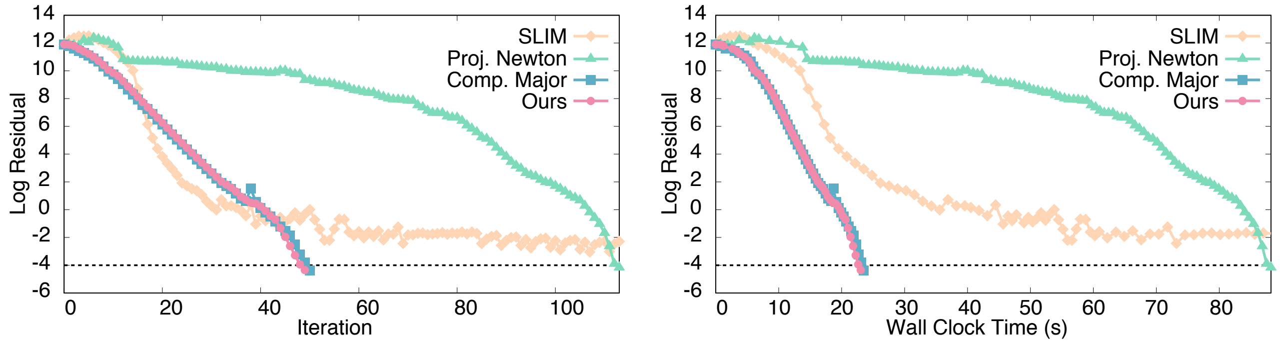

Fig. 4. Solver iterations (left) and wall clock time (right) to compute a parameterization with Symmetric Dirichlet over the subdivided car mesh using our method (pink circles), composite majorization (blue squares), per-element numerical projection (green triangles), and SLIM (yellow diamonds). The termination threshold is denoted by a dashed line.

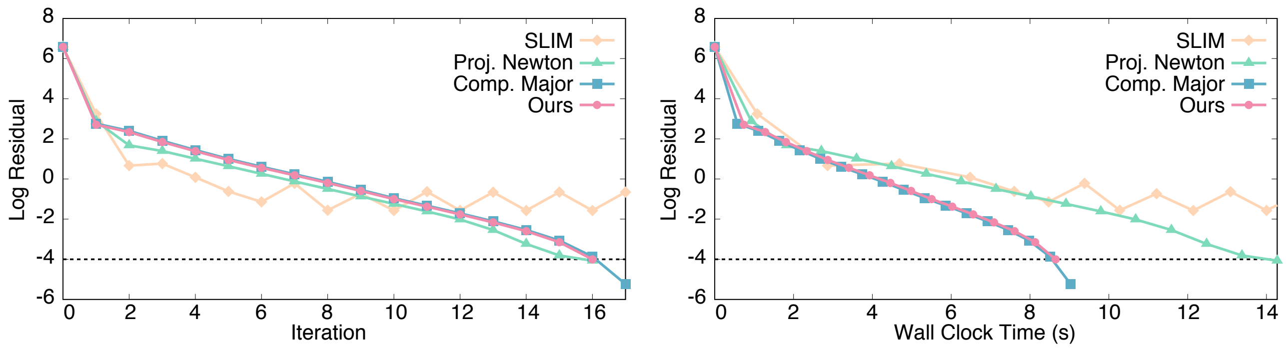

Fig. 5. Solver iterations (left) and wall clock time (right) to compute a parameterization with Symmetric Dirichlet over the man mesh using our method (pink circles), composite majorization (blue squares), per-element numerical projection (green triangles), and SLIM (yellow diamonds). The termination threshold is denoted by a dashed line.

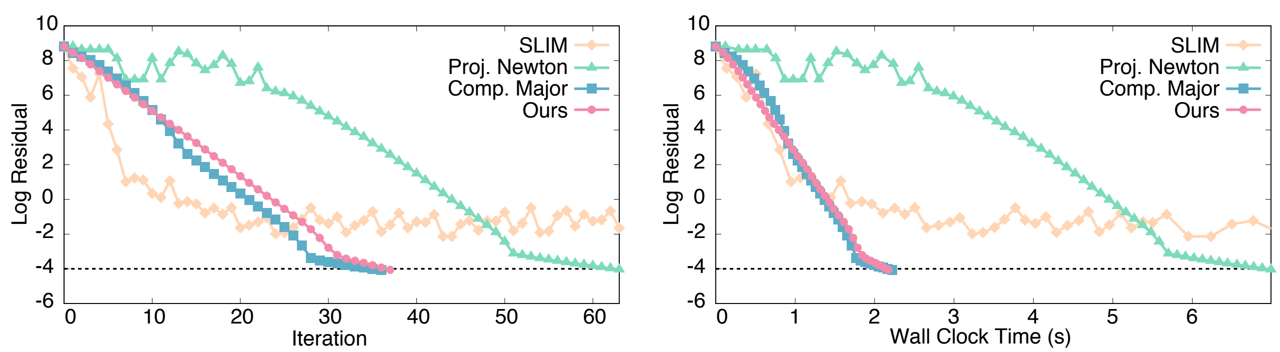

Fig. 6. Solver iterations (left) and wall clock time (right) to compute a parameterization with Symmetric Dirichlet over the octopus mesh using our method (pink circles), composite majorization (blue squares), per-element numerical projection (green triangles), and SLIM (yellow diamonds). The termination threshold is denoted by a dashed line.

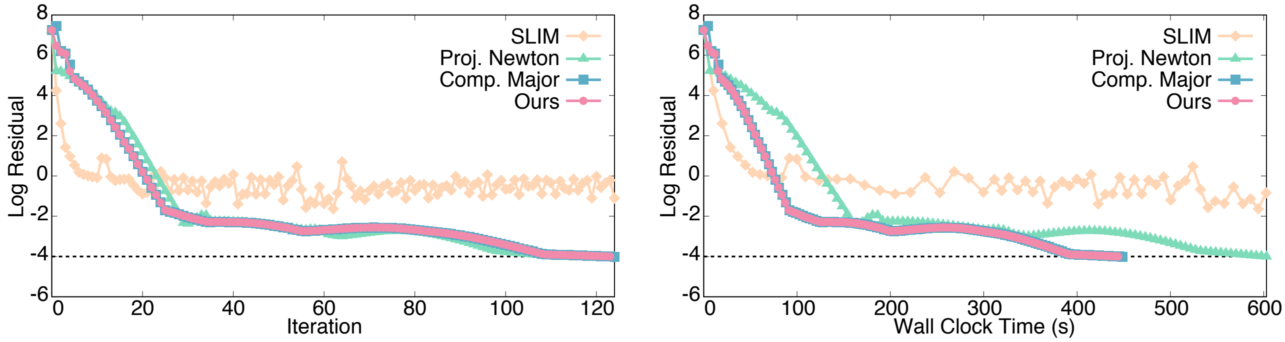

Fig. 7. Solver iterations (left) and wall clock time (right) to compute a parameterization with Symmetric Dirichlet over the Lucy mesh using our method (pink circles), composite majorization (blue squares), per-element numerical projection (green triangles), and SLIM (yellow diamonds). The termination threshold is denoted by a dashed line.

Symmetric Dirichlet

			Ours		Comp. Major.		Proj. Newton		SLIM	
Mesh	**Faces**	**Verts**	**Iters.**	**Time**	**Iters.**	**Time**	**Iters.**	**Time**	**Iters.**	**Time**
Bear	296409	148484	16	13714	16	14092	27	38782	100	164732
Buddha	470507	235771	17	25580	17	25642	15	36458	144	461508
Lucy	1000000	501105	123	443969	124	448814	111	602880	15256	163034065
Man	190471	95712	16	8652	17	9030	16	14272	94	106772
Octopus	26968	15141	37	2173	36	2219	63	6991	1442	270460
Car	5018	5215	44	1219	44	1252	39	1854	651	38866
Car 2	80308	81116	49	23037	50	23511	113	88218	176	178547
Camelhead	22704	11381	22	1279	22	1297	25	2599	527	81868
Cow	1500	762	288	925	288	914	158	979	16754	145891
Duck_Tube	10688	5391	90	3656	90	3681	72	4228	499999	171716036
Fish	7104	7138	48	2530	47	2596	51	3667	712	70360
Hoodie	19122	9715	64	3360	64	3302	89	7859	1553	197164
Hoodie2	21616	10884	568	38137	578	38063	563	53818	76819	12036300
Horse	4030	2038	207	2503	228	2860	294	5160	499999	9169104
LPS_Head	8842	8875	41	2041	41	2088	50	4078	994	109742
Ogre_Smile	39856	19985	44	3450	44	3495	38	5890	150	38168
Pants	2859	1453	99	585	99	571	108	1198	1157	16354
Pig_Body	8864	4453	433	8155	431	8255	538	16809	499999	17539513
Pig_Tounge	768	397	8	20	8	20	7	28	35	204
Rabbit	902	461	81	219	81	218	63	275	593	3437
Spot	5856	2975	20	220	20	220	22	481	192	5086
T10K_101089	6334	3264	29	326	29	330	28	645	819	22023
T10K_101582	107970	54271	39	10017	39	10065	29	13542	343	209736
T10K_127243	30436	15285	219	15722	218	15850	230	30721	8024	1820192
T10K_131969	2874	1468	48	276	48	275	48	540	397	5747
T10K_1324574	17538	8829	29	1277	29	1269	25	1937	250	29767
T10K_134543	4262	2167	35	303	35	309	45	718	481	9248
T10K_200079	34829	17462	12	869	12	868	16	2152	21	3504
T10K_208741	84064	42101	42	8255	42	8198	69	25081	226	104377
T10K_265730	49680	24956	223	22068	223	22432	147	28164	8370	2560303
T10K_308214	394510	197359	225	221605	229	223385	47	84551	x	x
T10K_37384	17374	8775	14	706	14	719	15	1209	55	6153
T10K_59340	786432	393438	21	52795	21	52219	22	87849	89	483920
T10K_65414	622	326	153	271	153	272	83	259	5194	23940
T10K_78319	46936	23518	76	6773	76	6647	136	24772	1284	336563
T10K_79189	6383	3224	32	427	32	432	42	1038	439	13399
T10K_80516	107176	53915	404	140154	513	180430	783	353539	13383	8886716
T10K_81369	47999	24310	33	3476	33	3470	31	6039	189	58897
T10K_998022	23362	11846	281	14252	281	14575	270	26465	10657	1530571
Teapot_Base	6204	6117	116	4983	118	5108	91	5726	499999	75020793
Teapot_Top	1824	1802	12	84	12	83	16	216	107	1774

Fig. 8. Performance statistics for Symmetric Dirichlet parameterization using our method, composite majorization, per-element Hessian projection, and SLIM. The iteration count is capped to 499,999 for each method. Tests that did not complete or reach the maximum iteration count in a reasonable time frame are denoted with an x.

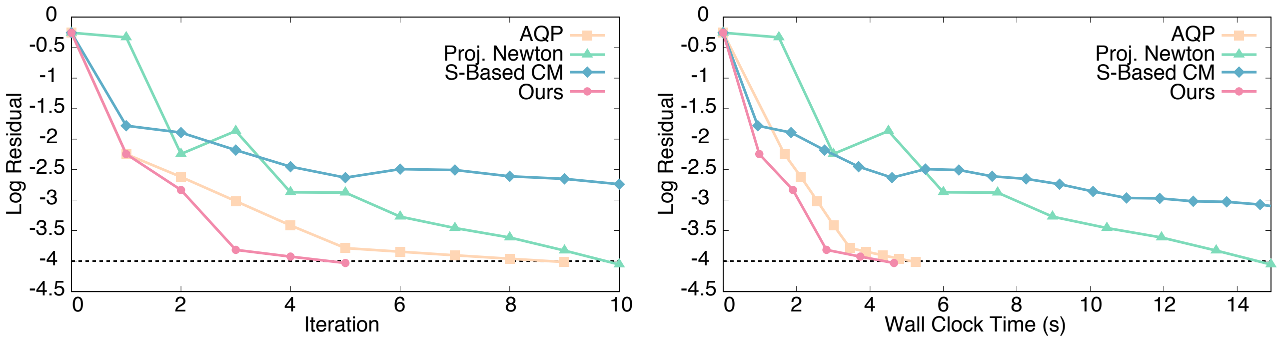

Fig. 9. Solver iterations (left) and wall clock time (right) to compute a parameterization with ARAP over the bear mesh using our method (pink circles), S-based composite majorization (blue diamonds), per-element numerical projection (green triangles), and AQP (yellow squares). The termination threshold is denoted by a dashed line.

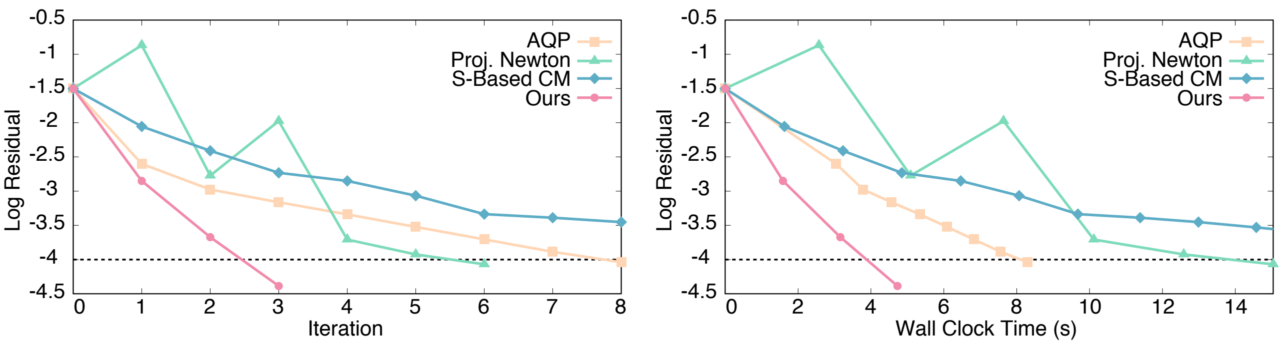

Fig. 10. Solver iterations (left) and wall clock time (right) to compute a parameterization with ARAP over the buddha mesh using our method (pink circles), S-based composite majorization (blue diamonds), per-element numerical projection (green triangles), and AQP (yellow squares). The termination threshold is denoted by a dashed line.

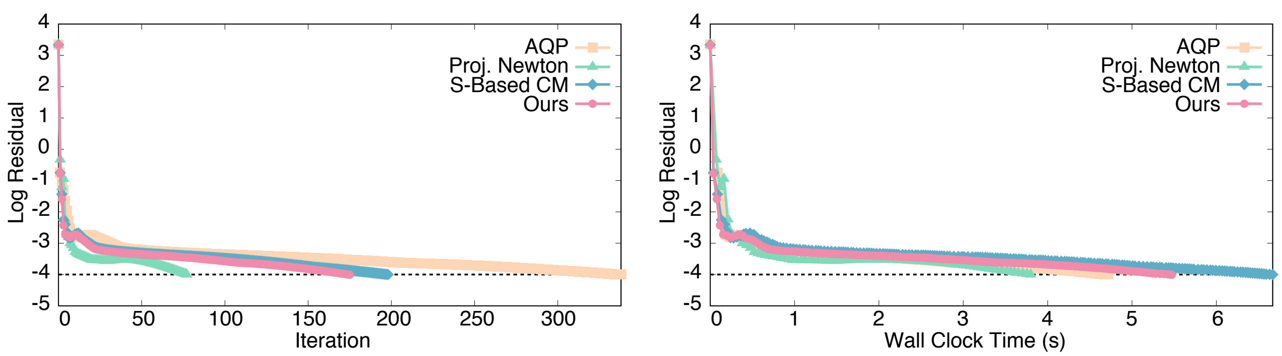

Fig. 11. Solver iterations (left) and wall clock time (right) to compute a parameterization with ARAP over the car mesh using our method (pink circles), S-based composite majorization (blue diamonds), per-element numerical projection (green triangles), and AQP (yellow squares). The termination threshold is denoted by a dashed line.

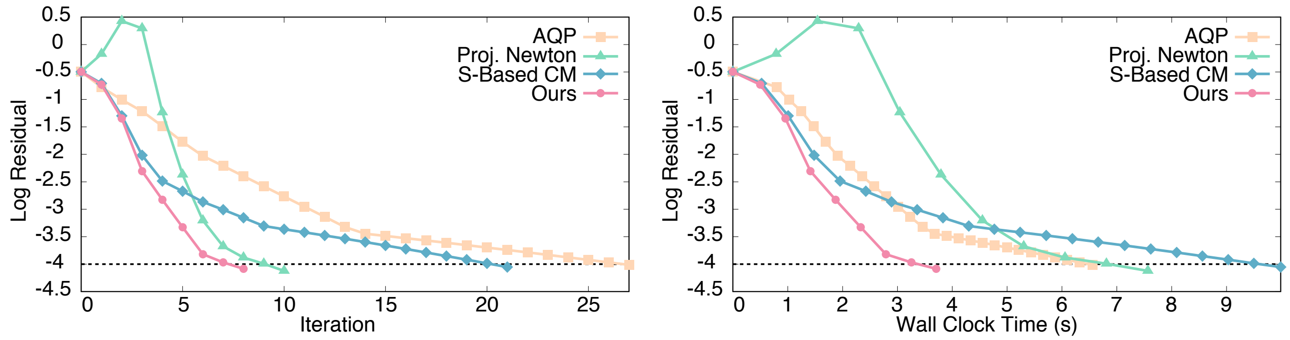

Fig. 12. Solver iterations (left) and wall clock time (right) to compute a parameterization with ARAP over the subdivided car mesh using our method (pink circles), S-based composite majorization (blue diamonds), per-element numerical projection (green triangles), and AQP (yellow squares). The termination threshold is denoted by a dashed line.

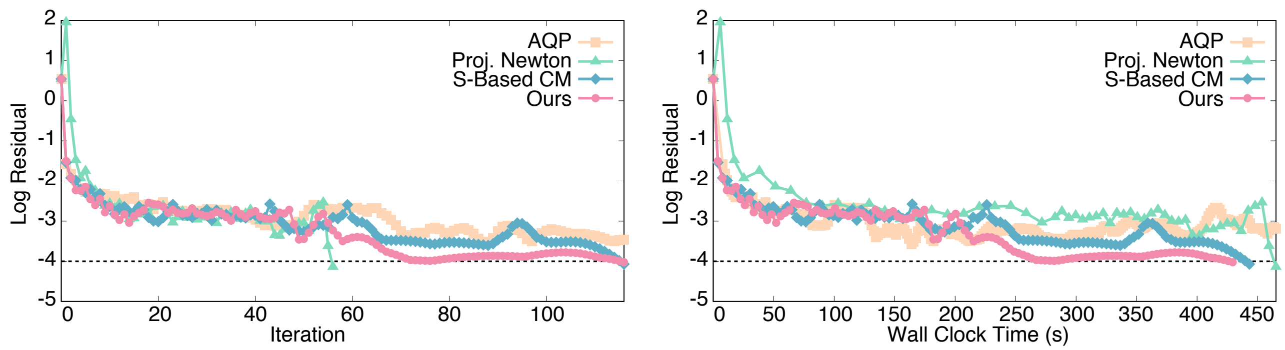

Fig. 13. Solver iterations (left) and wall clock time (right) to compute a parameterization with ARAP over the Lucy mesh using our method (pink circles), S-based composite majorization (blue diamonds), per-element numerical projection (green triangles), and AQP (yellow squares). The termination threshold is denoted by a dashed line.

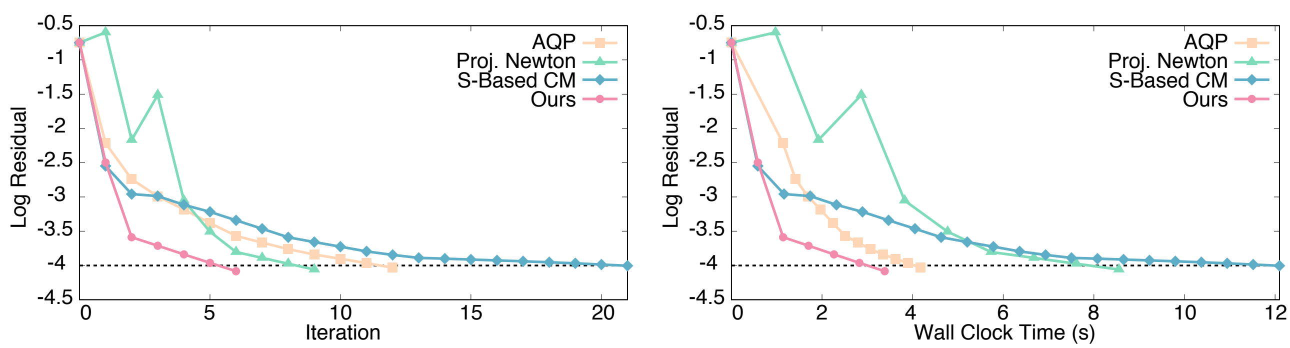

Fig. 14. Solver iterations (left) and wall clock time (right) to compute a parameterization with ARAP over the man mesh using our method (pink circles), S-based composite majorization (blue diamonds), per-element numerical projection (green triangles), and AQP (yellow squares). The termination threshold is denoted by a dashed line.

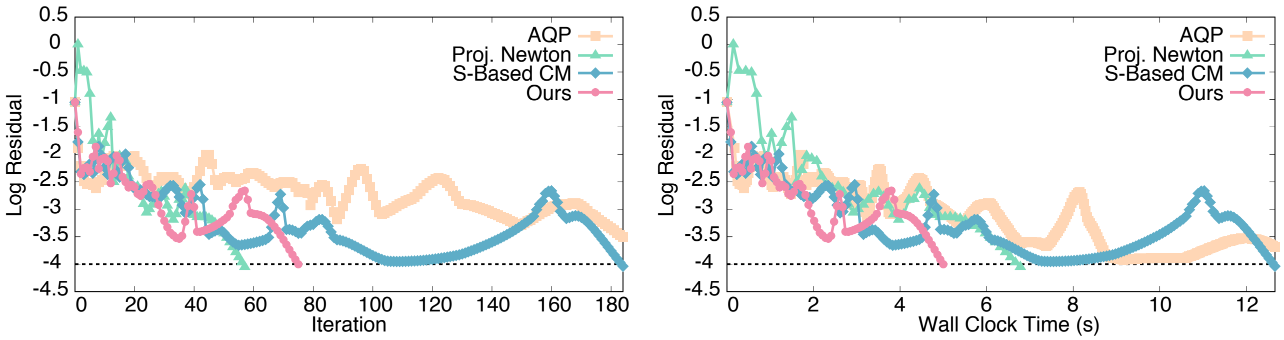

Fig. 15. Solver iterations (left) and wall clock time (right) to compute a parameterization with ARAP over the octopus mesh using our method (pink circles), S-based composite majorization (blue diamonds), per-element numerical projection (green triangles), and AQP (yellow squares). The termination threshold is denoted by a dashed line.

ARAP

			Ours		S-based CM		Proj. Newton		AQP, α/β CM	
Mesh	**Faces**	**Verts**	**Iters.**	**Time**	**Iters.**	**Time**	**Iters.**	**Time**	**Iters.**	**Time**
Bear	296409	148484	5	4653	79	72772	10	14920	9	5244
Buddha	470507	235771	3	4728	24	38290	6	15039	8	8294
Lucy	1000000	501105	116	429373	116	443256	56	465180	616	1177953
Man	190471	95712	6	3384	21	12097	9	8553	12	4178
Octopus	26968	15141	75	4999	184	12661	57	6779	409	14586
Car	5018	5215	175	5472	198	6668	77	3800	338	4722
Car 2	80308	81116	8	3707	21	9988	10	7567	27	6560
Camelhead	22704	11381	51	3411	229	15680	25	2789	83	3045
Cow	1500	762	608	2279	1359	5186	436	2911	1294	3226
Duck_Tube	10688	5391	114	4893	185	8017	181	11603	220	3287
Fish	7104	7138	79	3802	219	10767	81	6172	155	3039
Hoodie	19122	9715	70	4144	402	23749	55	5241	137	4344
Hoodie2	21616	10884	211	14073	933	60420	337	36507	415	15140
Horse	4030	2038	158	1413	563	5174	137	2316	319	1894
LPS_Head	8842	8875	235	12694	305	16827	111	9912	460	10883
Ogre_Smile	39856	19985	15	1450	58	5507	15	2584	27	1708
Pants	2859	1453	279	1899	527	3673	190	2321	582	2545
Pig_Body	8864	4453	73	1339	244	4601	71	2536	144	1733
Pig_Tongue	768	397	5	14	19	51	6	26	12	19
Rabbit	902	461	592	1758	536	1635	331	1579	1212	1715
Spot	5856	2975	13	173	67	890	16	392	25	233
T10K_101089	6334	3264	11	154	26	370	12	310	23	228
T10K_101582	107970	54271	5	1452	41	12238	5	2536	9	1768
T10K_127243	30436	15285	56	4433	457	37339	133	18951	113	5274
T10K_131969	2874	1468	26	180	250	1745	25	314	52	240
T10K_1324574	17538	8829	77	4103	156	8414	60	5199	149	3553
T10K_134543	4262	2167	77	739	424	4166	149	2693	152	968
T10K_200079	34829	17462	8	708	85	7369	55	8290	16	965
T10K_208741	84064	42101	37	8320	98	22542	45	17712	74	9118
T10K_265730	49680	24956	8	929	120	14350	30	6415	14	1201
T10K_308214	394510	197359	7	7583	26	28588	8	15033	15	9780
T10K_37384	17374	8775	8	433	93	5106	13	1139	16	455
T10K_59340	786432	393438	21	56404	116	351814	38	162815	40	51924
T10K_65414	622	326	493	1006	941	1965	535	1794	979	1010
T10K_78319	46936	23518	427	45665	594	64611	890	177181	878	60507
T10K_79189	6383	3224	75	1118	914	13732	118	3220	148	1366
T10K_80516	107176	53915	67	19309	216	64050	68	33716	255	39075
T10K_81369	47999	24310	14	1743	365	45300	23	5047	26	1971
T10K_998022	23362	11846	680	41834	1193	75021	494	52622	1842	62696
Teapot_Base	6204	6117	272	12182	721	32491	196	13257	511	8446
Teapot_Top	1824	1802	3	27	139	1162	6	93	9	63

Fig. 16. Performance statistics for ARAP parameterization using our method, S-based composite majorization (detailed in §2.2), per-element Hessian projection, and AQP (equivalent to α/β-based composite majorization, detailed in §2.1).