Supplemental Material

Somigliana Coordinates: an elasticity-derived approach for cage deformation

JIONG CHEN, LI X, Ecole Polytechnique, IP Paris, France
FERNANDO DE GOES, Pixar Animation Studios, USA
MATHIEU DESBRUN, Inria / Ecole Polytechnique, France

A EVALUATING SOMIGLIANA COORDINATES
As explained in our main paper [Chen et al. 2023], Somigliana coordinates for each face i and node j are defined as

\[
\begin{align*}
T_j(x) &= \int_{\Omega} T(y, x) \phi_j(y) \, d\sigma_y, \\
K_j(x) &= \int_{\Omega} K(y, x) \psi_j(y) \, d\sigma_y,
\end{align*}
\]

(1a) (1b)

Finding efficiently-computable closed-form expressions for these integrals is challenging. Instead, we apply quadratures to numerically evaluate them in practice. Here, we provide details on integrating Gauss-Legendre quadrature over the interval \([0, 1]\) to numerically compute both \(T_j(x)\) and \(K_j(x)\) by iterating over each element of the domain boundary.

2D CASE. Given an edge \(L_{pq} = (y_p, y_q)\), a point \(y\) on this edge can be expressed as \(y = y_p + \alpha(y_q - y_p)\) for a barycentric coordinate \(\alpha \in [0, 1]\). We define \(e = y_q - y_p, d = y_p - x, \) and \(r = e + d\) (see inset), while \(r = \|r\|\) denotes the norm of \(r\). Since \(\psi_j(y)\) are piecewise constant per edge, face coordinates are easily computed through quadrature as:

\[
\begin{align*}
K_{L_{pq}}(x) &= \int_{L_{pq}} K(y, x) \, d\sigma_y = \|e\| \int_0^1 K(y(\alpha), x) \, d\alpha \\
&= \|e\| \int_0^1 (a - b)(1 - \alpha) \ln \left( \frac{1}{r(\alpha)} \right) I + b \frac{r(\alpha) r'(\alpha)}{r^2(\alpha)} \, d\alpha \\
&\approx \|e\| \sum_k w_k \left( a - b \ln \left( \frac{1}{r(\alpha_k)} \right) I + b \frac{r(\alpha_k) r'(\alpha_k)}{r^2(\alpha_k)} \right),
\end{align*}
\]

where \(a = 1/\mu(2^{d-1} - \pi)\) and \(b = a/4(1 - \nu)\), while \(\{w_k, \alpha_k\}\) denote the pairs of weights and barycentric coordinates of points of a Gauss-Legendre quadrature over the interval \([0, 1]\).

As for the vertex coordinates, integrating over the line segment \(L_{pq}\) contributes \(T_{L_{pq}}^p\) (resp. \(T_{L_{pq}}^q\)) to \(T_p(x)\) (resp. \(T_q(x)\)) with:

\[
T_p^{L_{pq}}(x) = \|e\| \int_0^1 T(y(\alpha), x)(1 - \alpha) \, d\alpha
\]

\[
= \|e\| \int_0^1 \left[ \frac{\mu(a - 2b)}{r^2(\alpha)} \left( (n^r r'(\alpha)) I + nr'(\alpha) - r(\alpha)n' \right) + \frac{4\mu b}{r^4(\alpha)} (n^r r'(\alpha) r'(\alpha)) (1 - \alpha) \right] \, d\alpha.
\]

Similarly, we have \(T_q^{L_{pq}}(x)\) and both of them are approximately using a Gauss-Legendre quadrature. We then assemble each \(T_j(x)\) from its two neighboring edges \(L_{pi}\) and \(L_{iq}\) via:

\[
T_i(x) = T_i^{L_{pq}}(x) + T_i^{L_{pi}}(x).
\]

3D CASE. The 3D case proceeds similarly: given a triangular cage facet \(\triangle pqs = (y_p, y_q, y_s)\), a point inside this triangle can be expressed using barycentric coordinates as \(y = y_p + \alpha(y_q - y_p) + \beta(y_s - y_p)\), where \(\alpha \in [0, 1]\), and \(\beta \in [0, 1 - \alpha]\). Denote \(v = y_q - y_p, w = y_s - y_p, d = y_p - x\) (see inset), so that \(r = d + \alpha v + \beta w\), with still \(r = \|r\|\). Then one has:

\[
K_{\triangle pqs}(x) = \int_{\triangle pqs} K(y, x) \, d\sigma_y
\]

\[
= 2|\triangle pqs| \int_0^1 \frac{1}{|\partial(\alpha)|} \int_0^{1 - \alpha} d\beta \, K(y(\alpha, \beta), x)
\]

\[
= 2|\triangle pqs| \int_0^1 \frac{1}{|\partial(\alpha)|} \int_0^{1 - \alpha} \left[ \frac{a - b}{r(\alpha, \beta)} I + \frac{b}{r(\alpha, \beta)} r'(\alpha, \beta) \right] \, d\beta
\]

\[
= 2|\triangle pqs| \sum_k w_k \left( \frac{a - b}{r(\alpha_k, \beta_k)} I + \frac{b}{r(\alpha_k, \beta_k)} r'(\alpha_k, \beta_k) \right),
\]

where \(|\triangle pqs|\) is the area of \(\triangle pqs\), while \{\(w_k, \alpha_k, \beta_k\}\) are the quadrature weights of their associated 2D quadrature points \{(\(\alpha_k, \beta_k\))\} on a canonical triangle in \(\mathbb{R}^2\) with nodes located at \((0, 0), (0, 1)\) and \((1, 0)\). Taking into account both the cost and the generality of quadrature calculations to support high-order precision, we choose to subdivide each triangle into three quads to apply standard Gauss-Legendre quadratures — please refer to Sec. 5 of our paper for an illustration.

Now, since the basis function \(\phi_j(y)\) is a hat function, \(T_j(x)\) is a sum of components coming from all its neighboring triangle facets, i.e., \(T_j(x) = \sum_{\triangle \in N_j} T_{\triangle}^j(x)\). The contribution to \(T_j(x)\) of a neighboring triangle \(\triangle pqs\), for instance, reads:

\[ T_{i}^{\alpha pq} = \int_{0}^{1} d\alpha \int_{0}^{1-\alpha} d\beta \mathcal{T}(y(\alpha, \beta), x)(1 - \alpha - \beta) \]

\[ = 2|\Delta p q| \int_{0}^{1} d\alpha \int_{0}^{1-\alpha} d\beta \left\{ \mu(a - 2b) \left( n'(r(\alpha, \beta))I + nr'(\alpha, \beta) \right) - r(\alpha, \beta)n' \right\} + \frac{6\mu b}{r^5(\alpha, \beta)}(n'r(\alpha, \beta))r(\alpha, \beta)r(\alpha, \beta)' \left( 1 - \alpha - \beta \right). \]

Similarly, we have

\[ T_{j}^{\beta pq} = \int_{0}^{1} d\alpha \int_{0}^{1-\alpha} d\beta \mathcal{T}(y(\alpha, \beta), x)\alpha, \]

\[ T_{k}^{\gamma pq} = \int_{0}^{1} d\alpha \int_{0}^{1-\alpha} d\beta \mathcal{T}(y(\alpha, \beta), x)\beta. \]

### B CONDITIONAL EQUIVALENCE TO CAUCHY-GREEN COORDINATES (2D GREEN COORDINATES)

In the special 2D case when \( \nu = \pm \infty \) and \( \gamma = 0 \), the traction terms given by Eq. (14) from our paper is zero, so we only have the deformed boundary \( \tilde{y} \) with traction kernel \( T_{i}(y, x) \) that contributes to the actual deformation \( \tilde{x} \). Since \( \lim_{\nu \to \pm \infty} b = 0 \), traction kernel becomes \( \mathcal{T}(r) = \frac{1}{2\pi r^2}(n'r)I + nr' - rn' \). As a result, the deformation computed by summing over all edges reduces to

\[ \tilde{x}(x) = \frac{1}{2\pi} \sum_{e} \int_{0}^{L_e} \left( \frac{r'n}{r^2} + 2 \frac{r'}{r^2}(nr' - rn') \right) \tilde{y} \, d\sigma_y \]

\[ = \frac{1}{2\pi} \sum_{e} \int_{0}^{L_e} \frac{1}{r^2} \begin{pmatrix} r_1 n_1 + r_2 n_2 & 0 & r_1 n_1 + r_2 n_2 \\ 0 & r_1 n_1 + r_2 n_2 & 0 \end{pmatrix} \tilde{y} \, d\sigma_y. \]

(2)

For each edge, we can rewrite the above integral using complex numbers denoted with hollow letters, yielding

\[ \frac{1}{2\pi} \int_{0}^{L_e} \frac{r_1 n_1 + r_2 n_2 + (i(r_1 n_2 - r_2 n_1)+i(r_1 n_2 - r_2 n_1))}{r^* r} \tilde{y} \, d\sigma_y \]

\[ = \frac{1}{2\pi} \int_{0}^{L_e} \frac{r_1 n_1 + i(r_1 n_2 - r_2 n_1)}{r^* r} \tilde{y} \, d\sigma_y \]

\[ = \frac{1}{2\pi} \int_{0}^{L_e} \frac{r_1 n_1}{r^* r} \tilde{y} \, d\sigma_y = \frac{1}{2\pi i} \int_{0}^{L_e} \frac{r_1 n_1}{r} \tilde{y} \, d\sigma_y \]

\[ = \frac{1}{2\pi i} \int_{1}^{L_e} \frac{\tilde{y} r}{r} \, d\sigma_y, \]

where \( r^* \) denotes the conjugate of \( r \). This proves that when \( \nu = \pm \infty \) and \( \gamma = 0 \), our 2D coordinates exactly reproduce the Cauchy-Green coordinates [Weber et al. 2009] derived from Cauchy’s integral.

### REFERENCES
