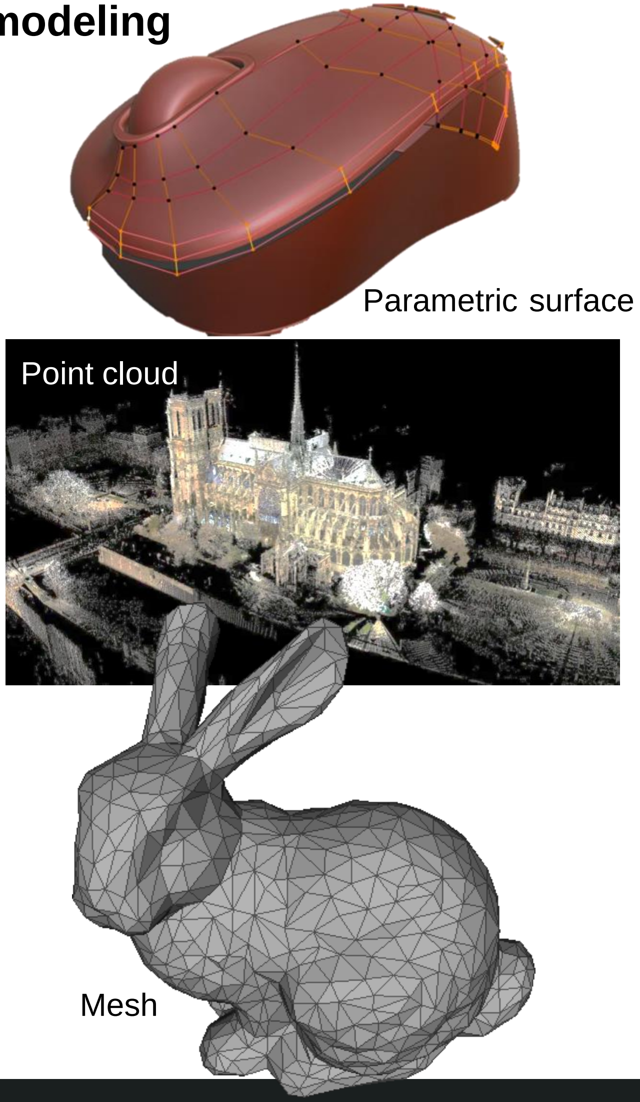
THE FASCINATION OF GREEN’S FUNCTIONS AND THEIR APPLICATIONS IN COMPUTER GRAPHICS

JIONG CHEN
INRIA
OCT 24, 2024, LIX

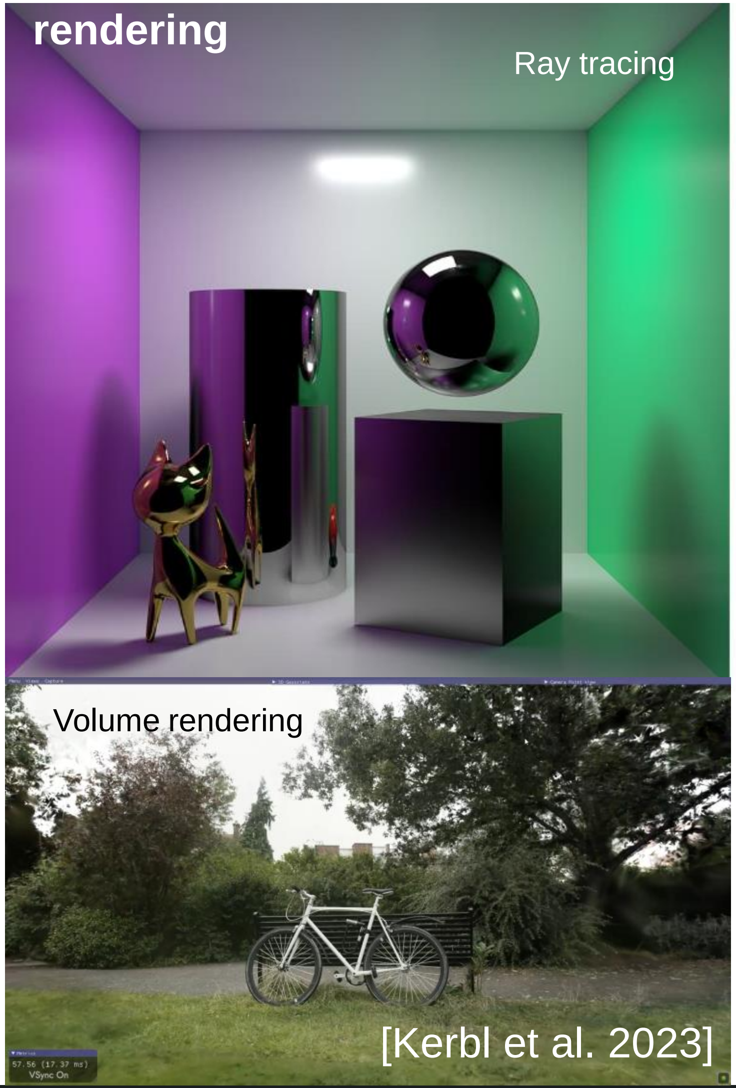
CORE TASKS OF COMPUTER GRAPHICS

modeling



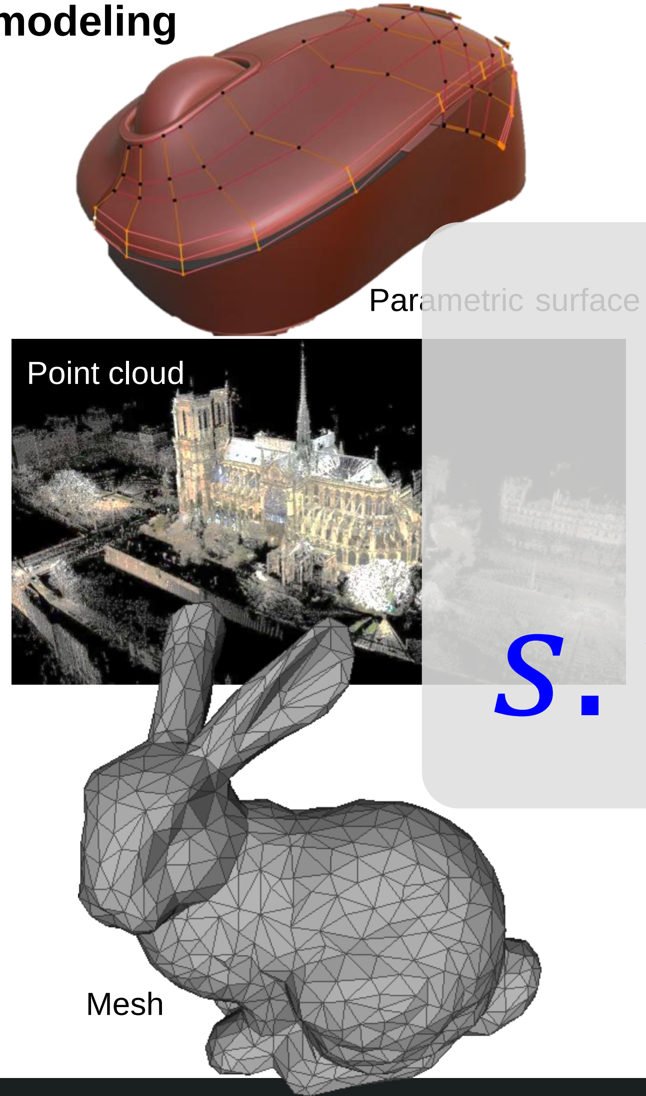
simulation

rendering

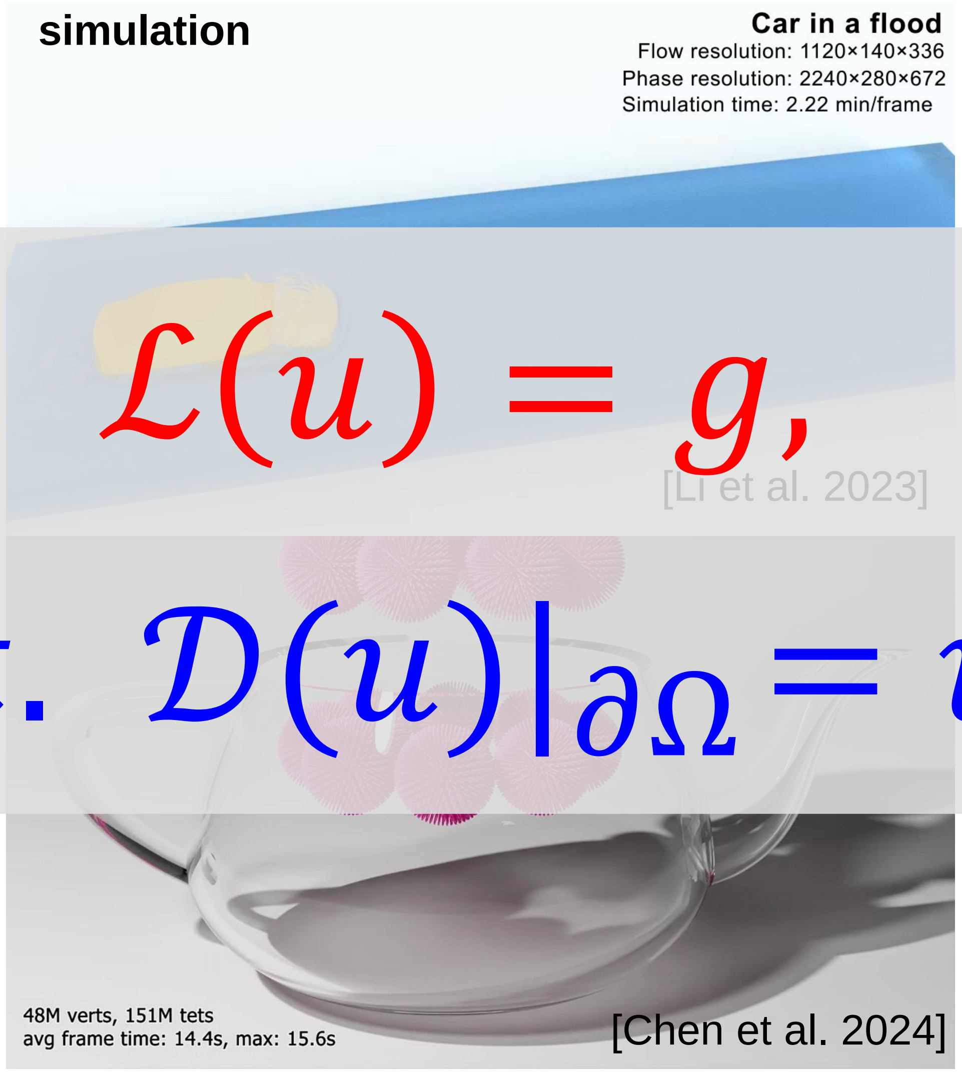


CORE TASKS OF COMPUTER GRAPHICS

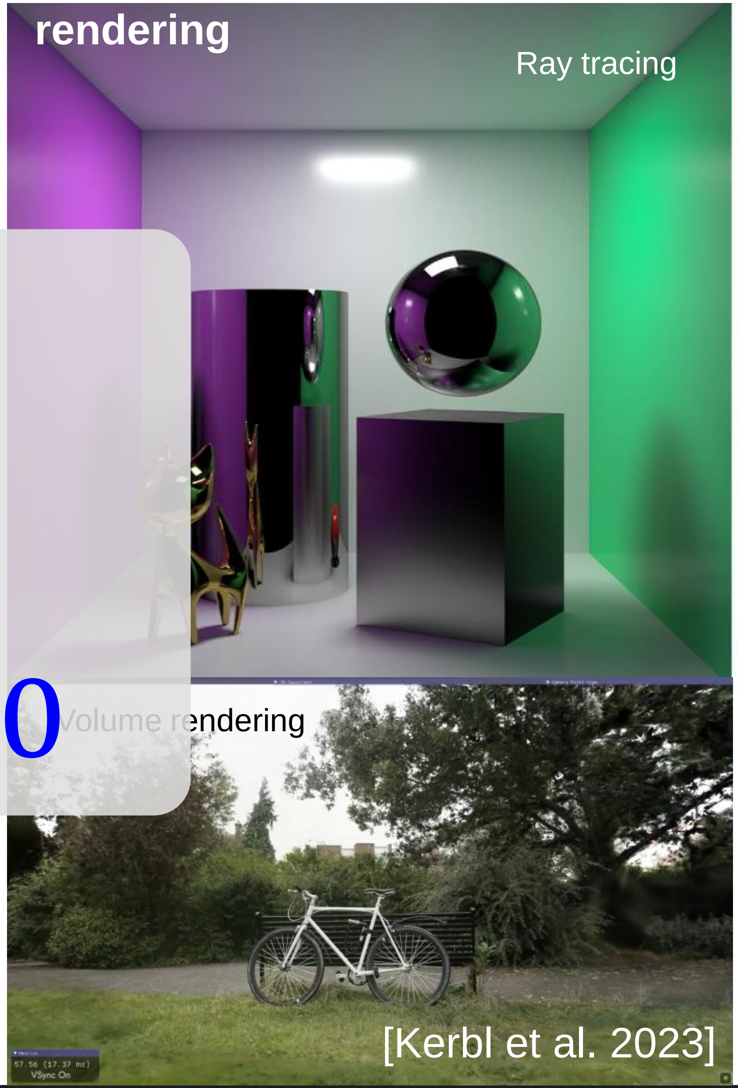
modeling



simulation



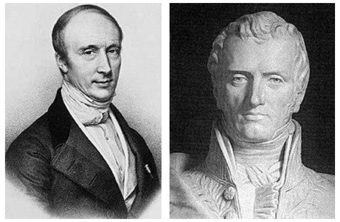
rendering



$$\mathcal{L}(u) = g,$$

$$s.t.\ \mathcal{D}(u)|_{\partial\Omega} = u_0$$

FAMILY OF PDES



Cauchy-Navier equation

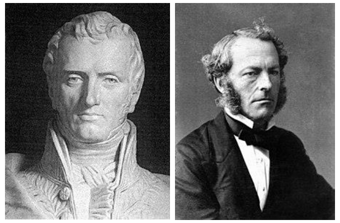
$$\mu\Delta \boldsymbol{u} + \frac{\mu}{(1-2\nu)}\nabla(\nabla\cdot\boldsymbol{u}) + \boldsymbol{b} = 0.$$

Maxwell equation

$$\nabla\cdot\boldsymbol{B} = 0\,,$$
$$\boldsymbol{B} = \mu_0(\boldsymbol{H}+\boldsymbol{M})\,,$$
$$\nabla\times\boldsymbol{H} = \boldsymbol{J}_f + \frac{\partial \boldsymbol{D}}{\partial t}\,,$$

Schrödinger equation

$$i\hbar\dot{\psi} = -\frac{\hbar^2}{2}\Delta\psi + p\psi$$



Navier-Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \mathbf{f}$$
$$\nabla\cdot\mathbf{u} = \mathbf{0}\,,$$

Helmholtz equation

$$\nabla^2 p + k^2 p = 0,$$

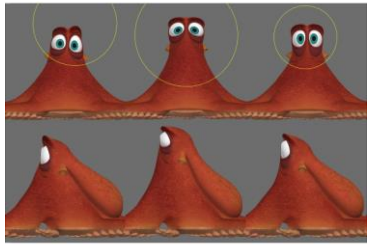
d’Alembert equation

$$\left(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)u(\boldsymbol{x},t) = 0,$$

Boltzmann transport equation

$$\frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\nabla f = \Omega(f - f^{\mathrm{eq}}) + \boldsymbol{F}\cdot\nabla_{\boldsymbol{v}} f,$$

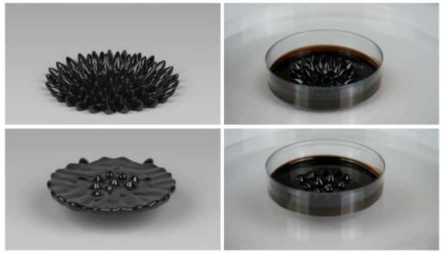
APPLICATIONS IN COMPUTER GRAPHICS



Digital sculpting

$$\mu\Delta\boldsymbol{u} + \frac{\mu}{(1-2\nu)}\nabla(\nabla\cdot\boldsymbol{u}) + \boldsymbol{b} = 0.$$

[de Goes et al. 2017]



Ferrofluids

$$\boldsymbol{B} = \mu_0(\boldsymbol{H} + \boldsymbol{M}),$$

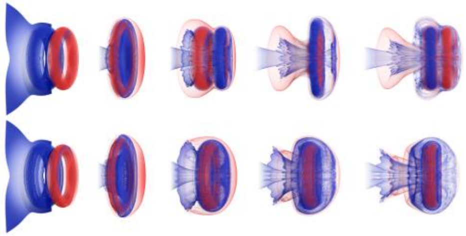
$$\nabla\times\boldsymbol{H} = \boldsymbol{J}_f + \frac{\partial\boldsymbol{D}}{\partial t},$$

[Huang et al. 2019]

Smoke

$$i\hbar\dot{\psi} = -\frac{\hbar^2}{2}\Delta\psi + p\psi$$

[Chern et al. 2016]

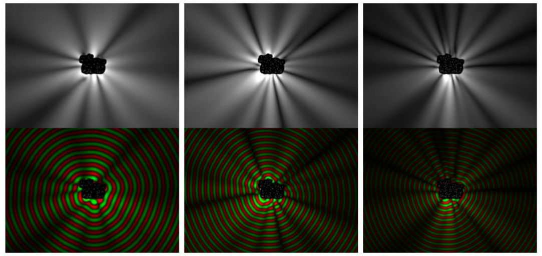


Incompressible flow

$$\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \mathbf{f}$$

$$\nabla\cdot\mathbf{u} = \mathbf{0},$$

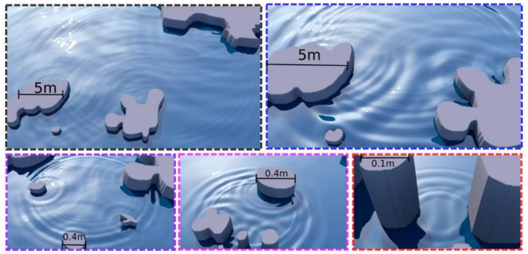
[Zehnder et al. 2018]



Sound propagation

$$\nabla^2 p + k^2 p = 0,$$

[James et al. 2006]



Water waves

$$\left(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)u(\boldsymbol{x},t) = 0,$$

[Schreck et al. 2006]

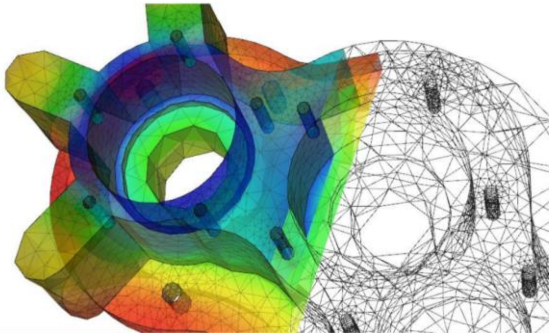
Turbulence

$$\frac{\partial f}{\partial t} + \boldsymbol{v}\cdot\nabla f = \Omega(f - f^{\mathrm{eq}}) + \boldsymbol{F}\cdot\nabla_{\boldsymbol{v}} f,$$

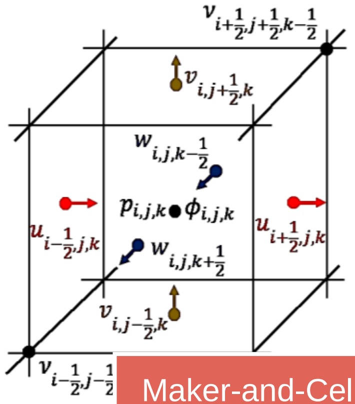
[Li et al. 2022]

FIND NUMERICAL SOLUTIONS

- Classical discretization schemes
  - Finite element method
  - Finite difference method
  - Finite volume method
- Safe choices if you know little about your problems
  - Linear or nonlinear PDEs
  - Spatially varying or constant coefficients
- Challenges
  - Accuracy issues
    - Numerical locking, numerical dissipation, etc.
  - High-quality volumetric meshes are often hard and costly to obtain
  - The problem size increases quickly in the complexity of geometry

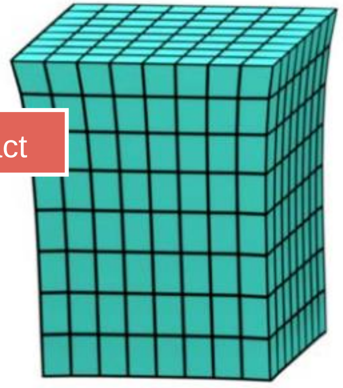


Tetrahedral mesh



Maker-and-Cell grid

Locking artifact



Fine simulation

Using a 2x coarse mesh

REVISIT ANALYTICAL METHODS

Green's function – **analytical** solution to **homogenous** & **"boundless"** PDEs *w.r.t.* a **singular** impulse

- Given a linear and homogeneous PDE

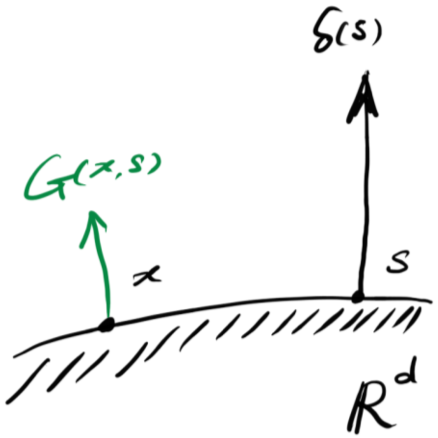
$$\mathcal{L}u(x) = f(x)$$

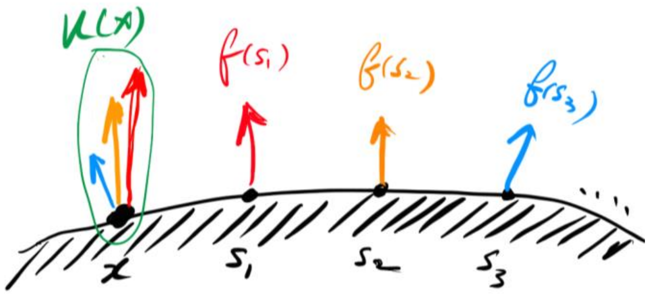
- A Green's function $G(x,s)$ is defined as

$$\mathcal{L}G(x, s) = \delta(x - s)$$

- Solution expressed via convolution

$$u(x) = \int G(x, s) f(s)\, \mathrm{d}s$$



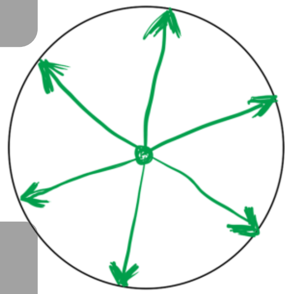


No equation solves, cheap to evaluate!

GREEN'S FUNCTIONS

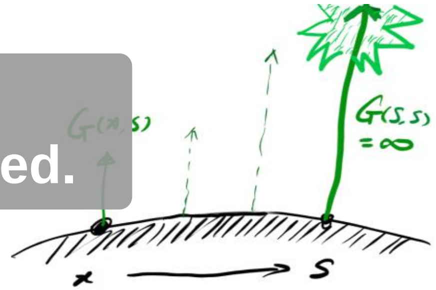
Differential operator L	Green's function G	Example of application
∂_t^{n+1}	$\frac{t^n}{n!}\Theta(t)$	
$\partial_t + \gamma$	$\Theta(t)e^{-\gamma t}$	
$(\partial_t + \gamma)^2$	$\Theta(t)te^{-\gamma t}$	
$\partial_t^2 + 2\gamma\partial_t + \omega_0^2$ where $\gamma < \omega_0$	$\Theta(t)e^{-\gamma t}\frac{\sin(\omega t)}{\omega}$ with $\omega = \sqrt{\omega_0^2 - \gamma^2}$	1D underdamped harmonic oscillator
$\partial_t^2 + 2\gamma\partial_t + \omega_0^2$ where $\gamma > \omega_0$	$\Theta(t)e^{-\gamma t}\frac{\sinh(\omega t)}{\omega}$ with $\omega = \sqrt{\gamma^2 - \omega_0^2}$	1D overdamped harmonic oscillator
$\partial_t^2 + 2\gamma\partial_t + \omega_0^2$ where $\gamma = \omega_0$	$\Theta(t)e^{-\gamma t}t$	1D critically damped harmonic oscillator
2D Laplace operator $\nabla_{\text{2D}}^2 = \partial_x^2 + \partial_y^2$	$\frac{1}{2\pi}\ln\rho$ with $\rho = \sqrt{x^2 + y^2}$	2D Poisson equation
3D Laplace operator $\nabla_{\text{3D}}^2 = \partial_x^2 + \partial_y^2 + \partial_z^2$	$\frac{-1}{4\pi r}$ with $r = \sqrt{x^2 + y^2 + z^2}$	Poisson equation
Helmholtz operator $\nabla_{\text{3D}}^2 + k^2$	$\frac{-e^{-ikr}}{4\pi r} = i\sqrt{\frac{k}{32\pi r}}H_{1/2}^{(2)}(kr) = i\frac{k}{4\pi}h_0^{(2)}(kr)$	stationary 3D Schrödinger equation for free particle
$\nabla^2 - k^2$ in n dimensions	$-(2\pi)^{-n/2}\left(\frac{k}{r}\right)^{n/2-1}K_{n/2-1}(kr)$	Yukawa potential, Feynman propagator
$\partial_t^2 - c^2\partial_x^2$	$\frac{1}{2c}\Theta(t - \|x/c\|)$	1D wave equation
$\partial_t^2 - c^2\nabla_{\text{2D}}^2$	$\frac{1}{2\pi c\sqrt{c^2t^2 - \rho^2}}\Theta(t - \rho/c)$	2D wave equation
D'Alembert operator $\square = \frac{1}{c^2}\partial_t^2 - \nabla_{\text{3D}}^2$	$\frac{\delta(t - \frac{r}{c})}{4\pi r}$	3D wave equation
$\partial_t - k\partial_x^2$	$\Theta(t)\left(\frac{1}{4\pi kt}\right)^{1/2}e^{-x^2/4kt}$	1D diffusion
$\partial_t - k\nabla_{\text{2D}}^2$	$\Theta(t)\left(\frac{1}{4\pi kt}\right)e^{-\rho^2/4kt}$	2D diffusion
$\partial_t - k\nabla_{\text{3D}}^2$	$\Theta(t)\left(\frac{1}{4\pi kt}\right)^{3/2}e^{-r^2/4kt}$	3D diffusion
$\frac{1}{c^2}\partial_t^2 - \partial_x^2 + \mu^2$	$\frac{1}{2}\left[(1 - \sin\mu ct)(\delta(ct - x) + \delta(ct + x)) + \mu\Theta(ct - \|x\|)J_0(\mu u)\right]$ with $u = \sqrt{c^2t^2 - x^2}$	1D Klein–Gordon equation
$\frac{1}{c^2}\partial_t^2 - \nabla_{\text{2D}}^2 + \mu^2$	$\frac{1}{4\pi}\left[(1 + \cos(\mu ct))\frac{\delta(ct - \rho)}{\rho} + \mu^2\Theta(ct - \rho)\operatorname{sinc}(\mu u)\right]$ with $u = \sqrt{c^2t^2 - \rho^2}$	2D Klein–Gordon equation
$\square + \mu^2$	$\frac{1}{4\pi}\left[\frac{\delta\left(t - \frac{r}{c}\right)}{r} + \mu c\Theta(ct - r)\frac{J_1(\mu u)}{u}\right]$ with $u = \sqrt{c^2t^2 - r^2}$	3D Klein–Gordon equation
$\partial_t^2 + 2\gamma\partial_t - c^2\partial_x^2$	$\frac{1}{2}e^{-\gamma t}\left[\delta(ct - x) + \delta(ct + x) + \Theta(ct - \|x\|)\left(\frac{\gamma}{c}I_0\left(\frac{\gamma u}{c}\right) + \frac{\gamma t}{u}I_1\left(\frac{\gamma u}{c}\right)\right)\right]$ with $u = \sqrt{c^2t^2 - x^2}$	telegrapher's equation
$\partial_t^2 + 2\gamma\partial_t - c^2\nabla_{\text{2D}}^2$	$\frac{e^{-\gamma t}}{4\pi}\left[(1 + e^{-\gamma t} + 3\gamma t)\frac{\delta(ct - \rho)}{\rho} + \Theta(ct - \rho)\left(\frac{\gamma\sinh\left(\frac{\gamma u}{c}\right)}{cu} + \frac{3\gamma t\cosh\left(\frac{\gamma u}{c}\right)}{u^2} - \frac{3ct\sinh\left(\frac{\gamma u}{c}\right)}{u^3}\right)\right]$ with $u = \sqrt{c^2t^2 - \rho^2}$	2D relativistic heat conduction
$\partial_t^2 + 2\gamma\partial_t - c^2\nabla_{\text{3D}}^2$	$\frac{e^{-\gamma t}}{20\pi}\left[(8 - 3e^{-\gamma t} + 2\gamma t + 4\gamma^2t^2)\frac{\delta(ct - r)}{r^2} + \frac{\gamma^2}{c}\Theta(ct - r)\left(\frac{1}{cu}I_1\left(\frac{\gamma u}{c}\right) + \frac{4t}{u^2}I_2\left(\frac{\gamma u}{c}\right)\right)\right]$ with $u = \sqrt{c^2t^2 - r^2}$	3D relativistic heat conduction

PROBLEM I: Operators & GFs are isotropic: they behave the same in all directions.



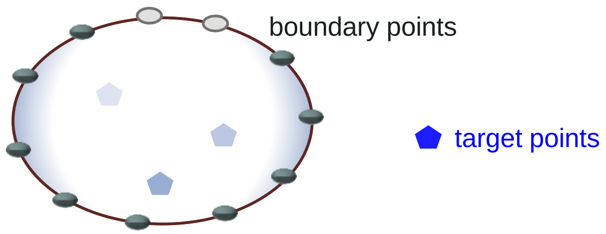
PROBLEM II: GFs are singular at the impulse.

PROBLEM III: Boundary conditions are not considered.

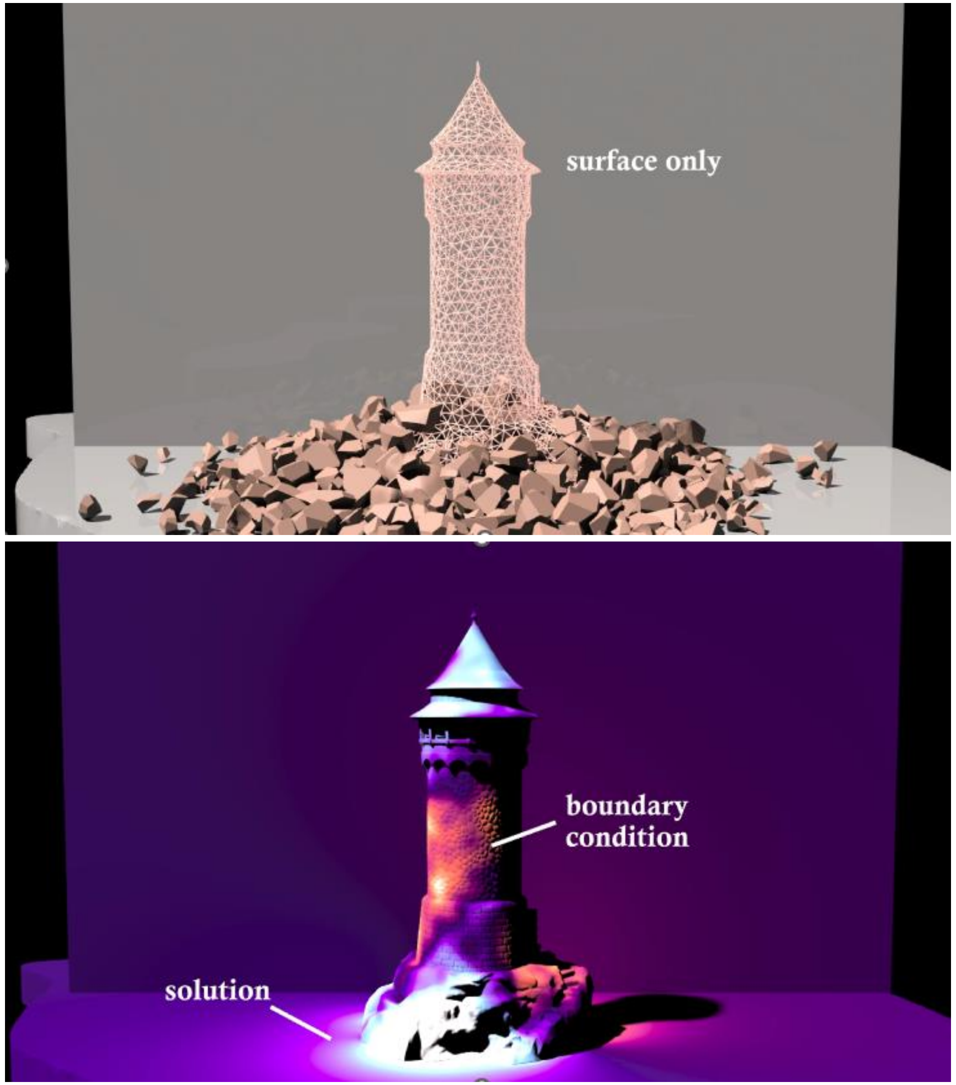


ENFORCING BOUNDARY CONDITIONS

- Boundary element method (BEM)
  - Turn volumetric differential equations to **boundary integral equations (BIE)**
    - Solve **unknown** boundary data based on **given** boundary conditions
    - Diffuse out the boundary data to get solutions at arbitrary target points



- No need for volumetric tessellation
  - Meshing a surface is simpler and faster in general
  - Slower growth of the problem size
  - Works for infinite large domains



ENFORCING BOUNDARY CONDITIONS

- Yet, we have to deal with dense linear systems

$$\begin{aligned} \Delta u &= 0 \quad \text{in a domain } \Omega \subset R^n \\ u &= g \quad \text{on the boundary } \Gamma := \partial\Omega. \end{aligned}$$

PDE & boundary conditions

→

Different approaches to build BIEs

$$u(x) = -\int_\Gamma \partial_{n(y)}\gamma(x,y)u(y)do(y) + \int_\Gamma \gamma(x,y)\partial_n u(y)do(y)$$

$$u(x) = \int_\Gamma \gamma(x,y)\psi(y)do(y)$$

$$u(x) = \int_\Gamma v(y)\partial_n\gamma(x,y)do(y)$$

[Costabel 1987]

→

$$\left(\frac{1}{2} + K\right) g = V(\partial_n u)$$

$$g = V\psi$$

$$\left(\frac{1}{2} - K\right) v = g$$

Dense linear equations of boundary unknowns

- Direct solvers
  - LU, Cholesky, QR…
  - Not scalable in time and memory cost
- Iterative solvers
  - Krylov subspace methods
  - Often suffer from poor convergence

Solve $\mathcal{L}(u) = 0$ using the method of Green's functions

$$u(x) = \int G(y,x)g(y)dS_y$$

RPESENTATION of the fundamental solutions to general linear operators

$$s.t.\ \mathcal{D}(u)|_{\partial\Omega} = u_0$$

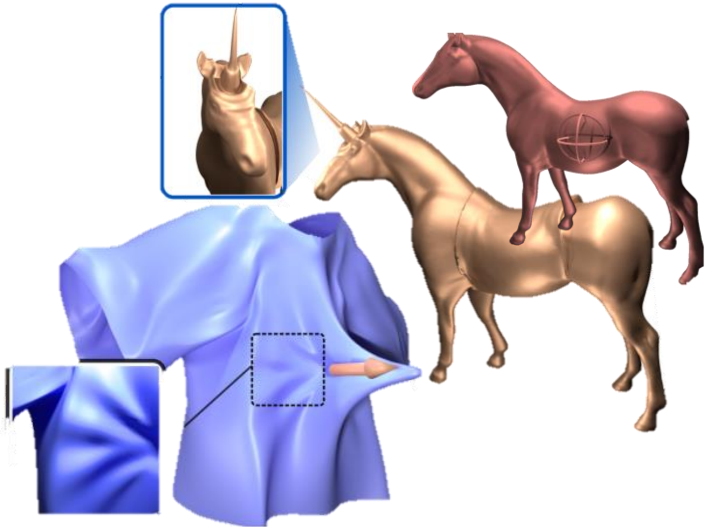
SCALABILITY of enforcing boundary conditions for large scale problems

OUR ATTEMPTS

Applications

Boundaryless deformation tool

SIGGRAPH '22



- **Mathematical tool**
  - **Generalized, regularized Green's functions** using series expansion
- **Tradeoff**
  - No equations solves, real-time performance
  - **Not aware** of any boundaries

Meshless solvers for BIE

SIGGRAPH '24

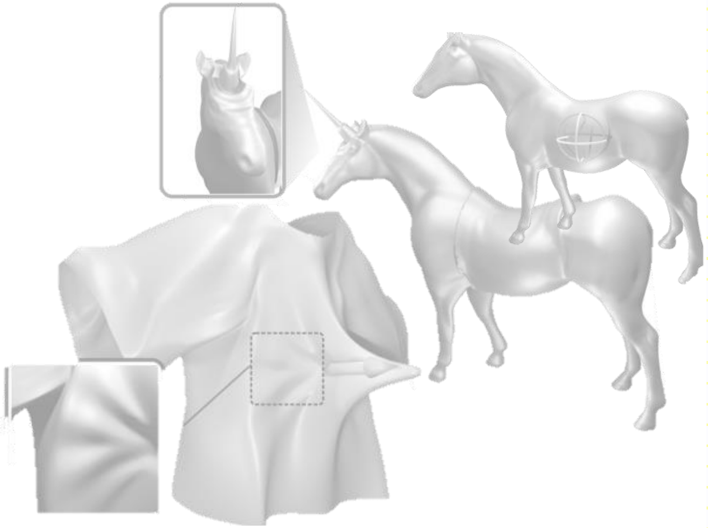
- **Mathematical tool**
  - **Inverse Cholesky decomposition** for accelerating PCG convergence
- **Tradeoff**
  - Fast, but not real-time due to solving dense systems
  - Boundary conditions are **strictly** satisfied

OUR ATTEMPTS

Applications

Boundaryless deformation tool

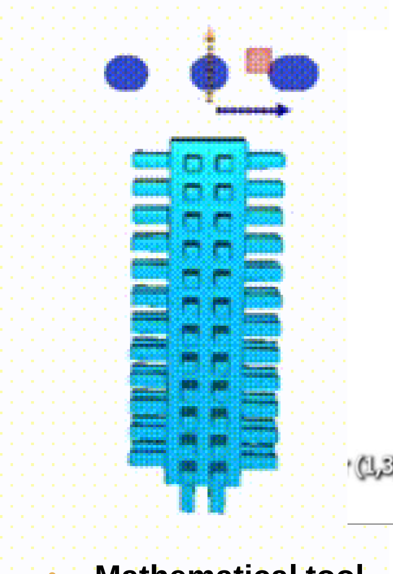
SIGGRAPH ‘22

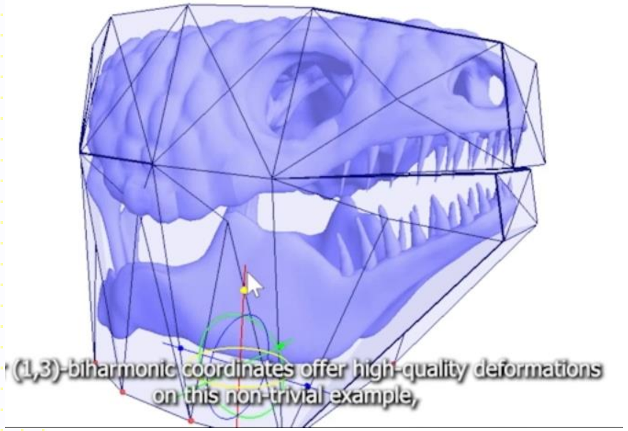


- Mathematical tool
  - **Generalized Green’s functions** using series expansion
- Tradeoff
  - No equations solves, real-time performance
  - **Not aware** of any boundaries

Cage controlled deformation tools

SIGGRAPH ‘23 ‘24



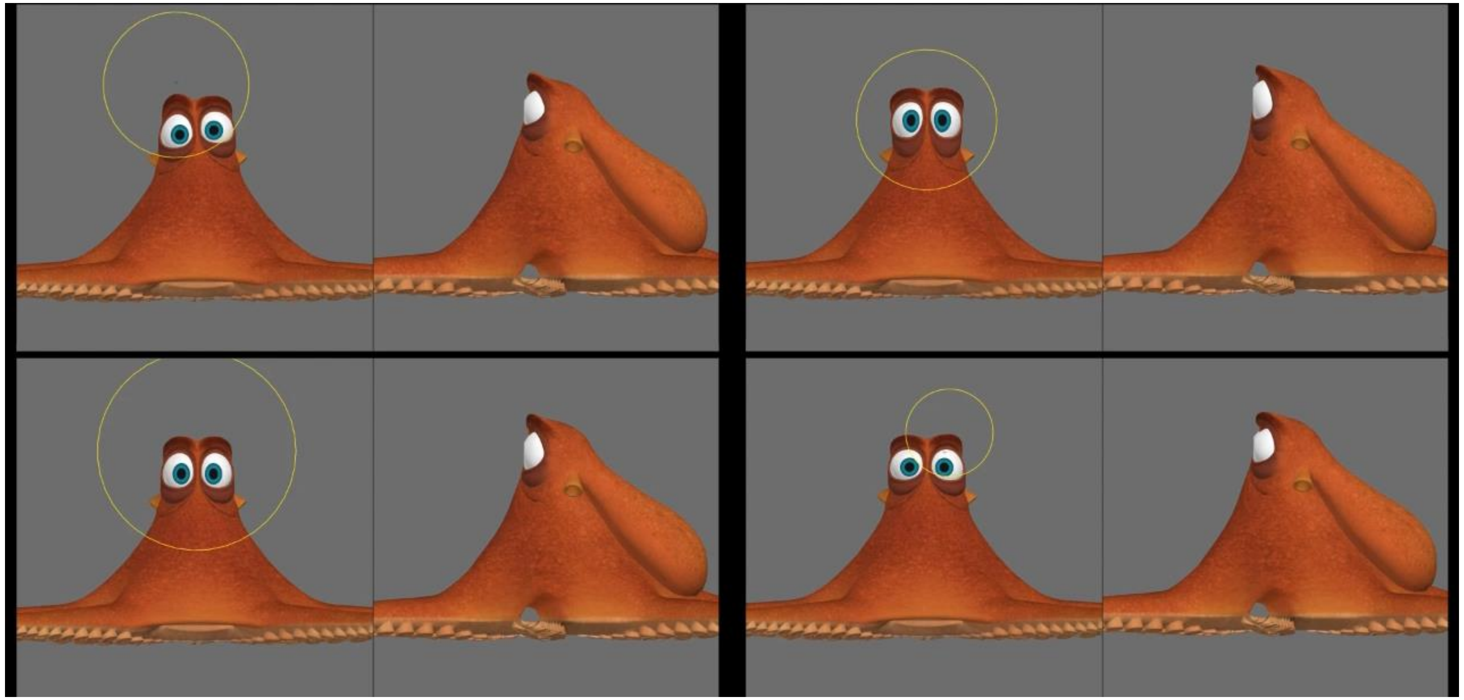


- **Mathematical tool**
  - **Generalized barycentric coordinates** w.r.t. the controlling cage
- **Tradeoff**
  - Some precomputation, no equations solves, real-time performance
  - Aware of any boundary conditions, but only **approximately** fulfill them

Meshless solvers for BIE

SIGGRAPH ‘24

- Mathematical tool
  - **Inverse Cholesky decomposition** for accelerating PCG convergence
- Tradeoff
  - Fast, but not real-time due to solving dense systems
  - Boundary conditions are **strictly** satisfied

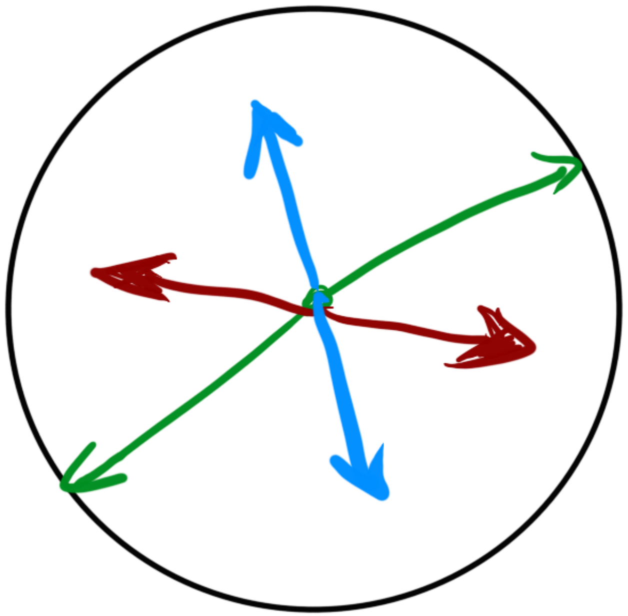


EXTENDED GREEN’S FUNCTIONS

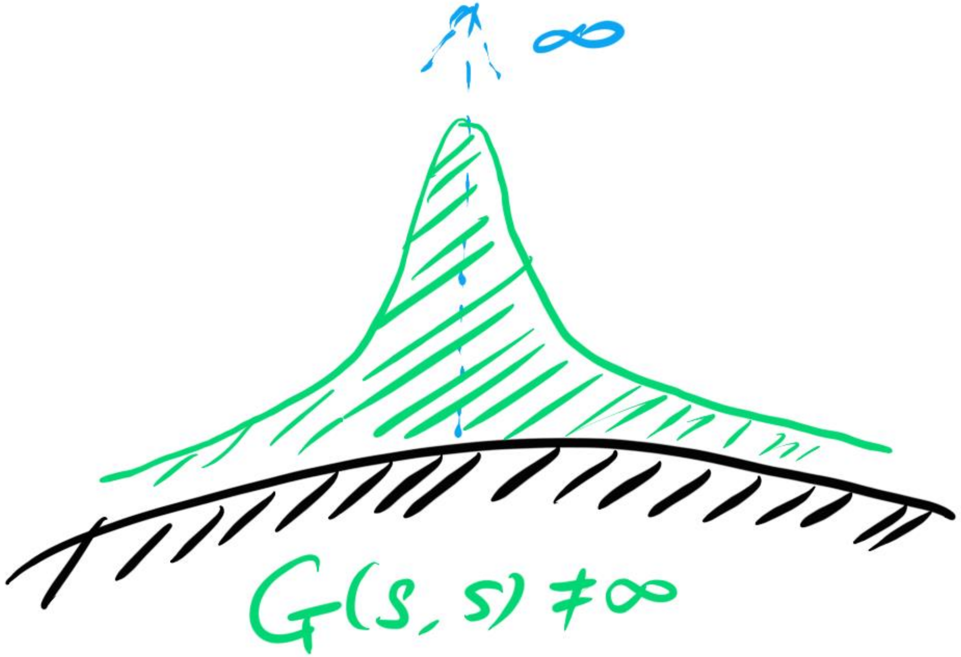
A REAL-TIME & CONTROLLABLE TOOL FOR MESH SCULPTING

OUR MOTIVATION

Extend Green's function to support…



Anisotropy



Arbitrary regularization

ELASTICITY EQUATION

- Isotropic strain-stress relationship

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon} + \lambda \mathrm{tr}(\boldsymbol{\varepsilon})\mathbf{I}$$

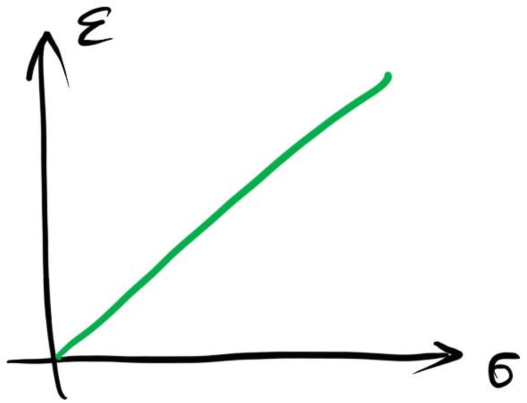
- Linear anisotropic strain-stress relationship

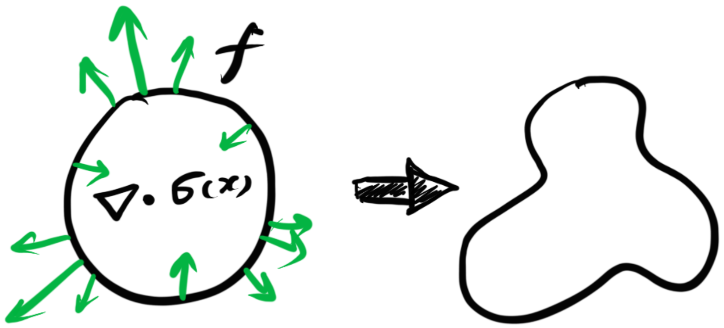
$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}$$

21 independent variables in 3D

- Equation of elasticity

$$\boxed{C_{ijkl}\frac{\partial^2 u_k}{\partial x_l \partial x_j} + f_i = 0,}$$

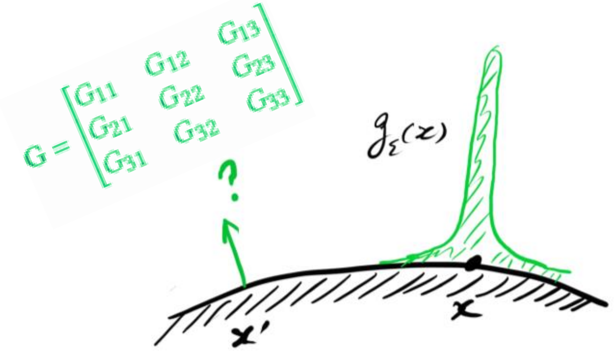




GREEN'S FUNCTION OF ELASTICITY

- **GF** to elasticity equation satisfies

$$\boxed{C_{ijkl}\frac{\partial^2 G_{km}}{\partial x_l \partial x_j} + \delta_{im} g_\varepsilon(\mathbf{x}) = 0,}$$



C_{ijkl} ⇒ Isotropic

Isotropic ⇒ $g_\varepsilon(\mathbf{x}) = \delta(\mathbf{x})$ ⟹ $\mathbf{u}(\mathbf{r}) = \left[\frac{(a-b)}{r}\mathbf{I} + \frac{b}{r^3}\mathbf{r}\mathbf{r}^t\right]\mathbf{f}$ Kelvin solution

[Kelvin 1848]

Isotropic ⇒ $g_\varepsilon(r) = 15\varepsilon^4/(8\pi)(r^2+\varepsilon^2)^{-\frac{7}{2}}$ ⟹ $\mathbf{u}_\varepsilon(\mathbf{r}) = \left[\frac{(a-b)}{r_\varepsilon}\mathbf{I} + \frac{b}{r_\varepsilon^3}\mathbf{r}\mathbf{r}^t + \frac{a}{2}\frac{\varepsilon^2}{r_\varepsilon^3}\mathbf{I}\right]\mathbf{f} \equiv \mathcal{K}_\varepsilon(\mathbf{r})\mathbf{f}.$

[Cortez 2001; de Goes 2017]

C_{ijkl} ⇒ Anisotropic ⟹ $g_\epsilon(x)$ is a more general radially symmetric function

???

GENERAL REGULARIZED GREEN'S FUNCTION

To derive **GF** from $\boxed{C_{ijkl}\dfrac{\partial^2 G_{km}}{\partial x_l \partial x_j} + \delta_{im} g_\varepsilon(\mathbf{x}) = 0,}$

- we use Fourier transform

$$\widehat{G}_{km}(\xi) = (C_{ijkl}\xi_l\xi_j)^{-1}\delta_{im}\boxed{\widehat{g}_\varepsilon(\xi).}$$

radial symmetry

$$\widehat{g}_\varepsilon(\xi) = \widehat{g}_\varepsilon(|\xi|) = \frac{(2\pi)^{\frac{3}{2}}}{\sqrt{|\xi|}} \int_0^\infty J_{\frac{1}{2}}(|\mathbf{x}||\xi|) g_\varepsilon(|\mathbf{x}|)|\mathbf{x}|^{\frac{3}{2}}\, \mathrm{d}|\mathbf{x}|,$$

Spatial domain		Frequency domain
$f(x)$	$\rightarrow$	$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i\xi x}\, dx$
$a \cdot f(x)$	$\rightarrow$	$a \cdot \hat{f}(\xi)$
$\dfrac{d^n f(x)}{dx^n}$	$\rightarrow$	$(i\xi)^n \hat{f}(\xi)$

- and its inverse Fourier transform

$$\boxed{\mathbf{G}(\mathbf{x}) = \frac{1}{8\pi^3}\int_{\mathbb{R}^3} \widehat{\mathbf{G}}(\xi)\exp(\mathrm{i}\,\mathbf{x}\cdot\xi)\,\mathrm{d}\xi,}$$

How to evaluate this integral?

- Thus, given a load function $f(x) = g_\varepsilon(x - x')h$

$$\mathbf{u}(\mathbf{x}) = \mathrm{Re}[\mathbf{G}(\mathbf{x} - \mathbf{x}')]\,\mathbf{h},$$

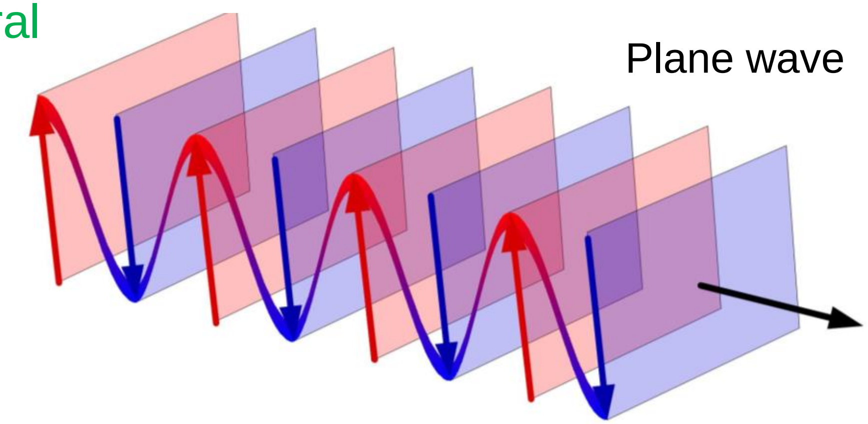
SPHERICAL HARMONIC EXPANSION

- For arbitrary material C_{ijkl}, $\mathbf{G}(\mathbf{x})$ has no analytical expressions in general
- Plane-wave expansion, or Rayleigh expansion

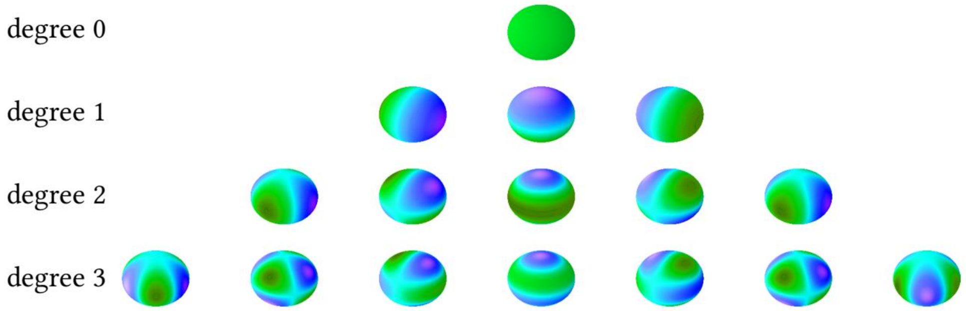
$$\exp(\mathbb{i}\,\mathbf{x}\cdot\xi) = 4\pi \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \mathbb{i}^l j_l(|\mathbf{x}||\xi|)\,Y_l^m(\widetilde{\mathbf{x}})\overline{Y}_l^m(\widetilde{\xi}),$$

- Recall: $\widehat{G}_{km}(\xi) = (C_{ijkl}\xi_l\xi_j)^{-1}\delta_{im}\widehat{g}_\varepsilon(\xi).$

$$\mathbf{G}(\mathbf{x}) = \frac{1}{8\pi^3}\int_{\mathbb{R}^3} \widehat{\mathbf{G}}(\xi)\exp(\mathbb{i}\,\mathbf{x}\cdot\xi)\,\mathrm{d}\xi,$$



Plane wave



Spherical harmonics

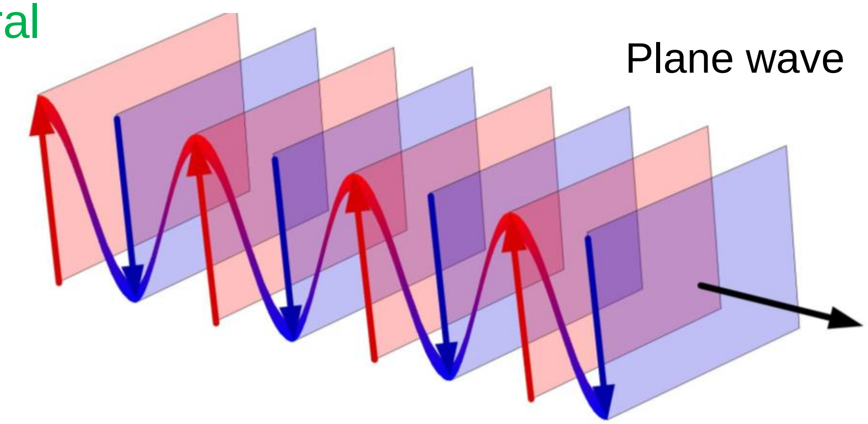
SPHERICAL HARMONIC EXPANSION

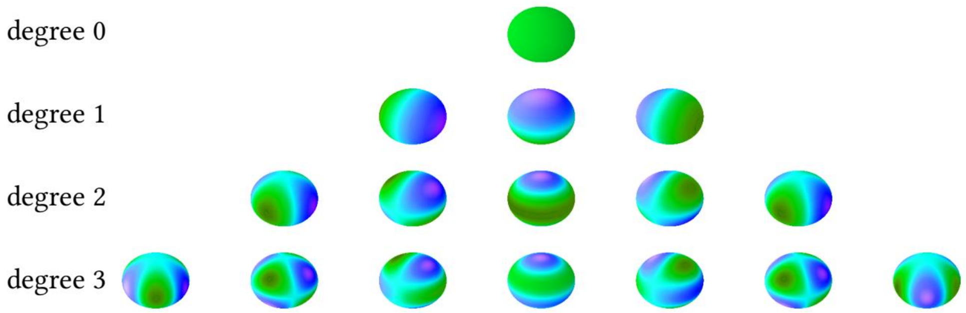
- For arbitrary material C_{ijkl}, $\mathbf{G}(\mathbf{x})$ has no analytical expressions in general
- Plane-wave expansion, or Rayleigh expansion

$$\exp(\mathbb{i}\,\mathbf{x}\cdot\xi) = 4\pi \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \mathbb{i}^l j_l(|\mathbf{x}||\xi|) Y_l^m(\widetilde{\mathbf{x}})\overline{Y}_l^m(\widetilde{\xi}),$$

- Recall: $\widehat{G}_{km}(\xi) = (C_{ijkl}\xi_l\xi_j)^{-1}\delta_{im}\widehat{g}_\varepsilon(\xi)$.

$$\mathbf{G}(\mathbf{x}) = \frac{1}{2\pi^2}\sum_{l=0}^{\infty}\sum_{m=-l}^{l} \mathbb{i}^l\, Y_l^m(\widetilde{\mathbf{x}}) \int_0^\infty \widehat{g}_\varepsilon(|\xi|) j_l(|\mathbf{x}||\xi|)\,\mathrm{d}|\xi| \cdot \int_{\mathbb{S}^2} (C_{ikjl}\widetilde{\xi}_k\widetilde{\xi}_l)^{-1}\overline{Y}_l^m(\widetilde{\xi})\,\mathrm{d}\mathbb{S}(\widetilde{\xi}),$$





Spherical harmonics

GREEN'S FUNCTION IN SERIES

- Expressed in spherical coordinates, $\mathbf{G}(\mathbf{x})$ is decomposed as

Directional term

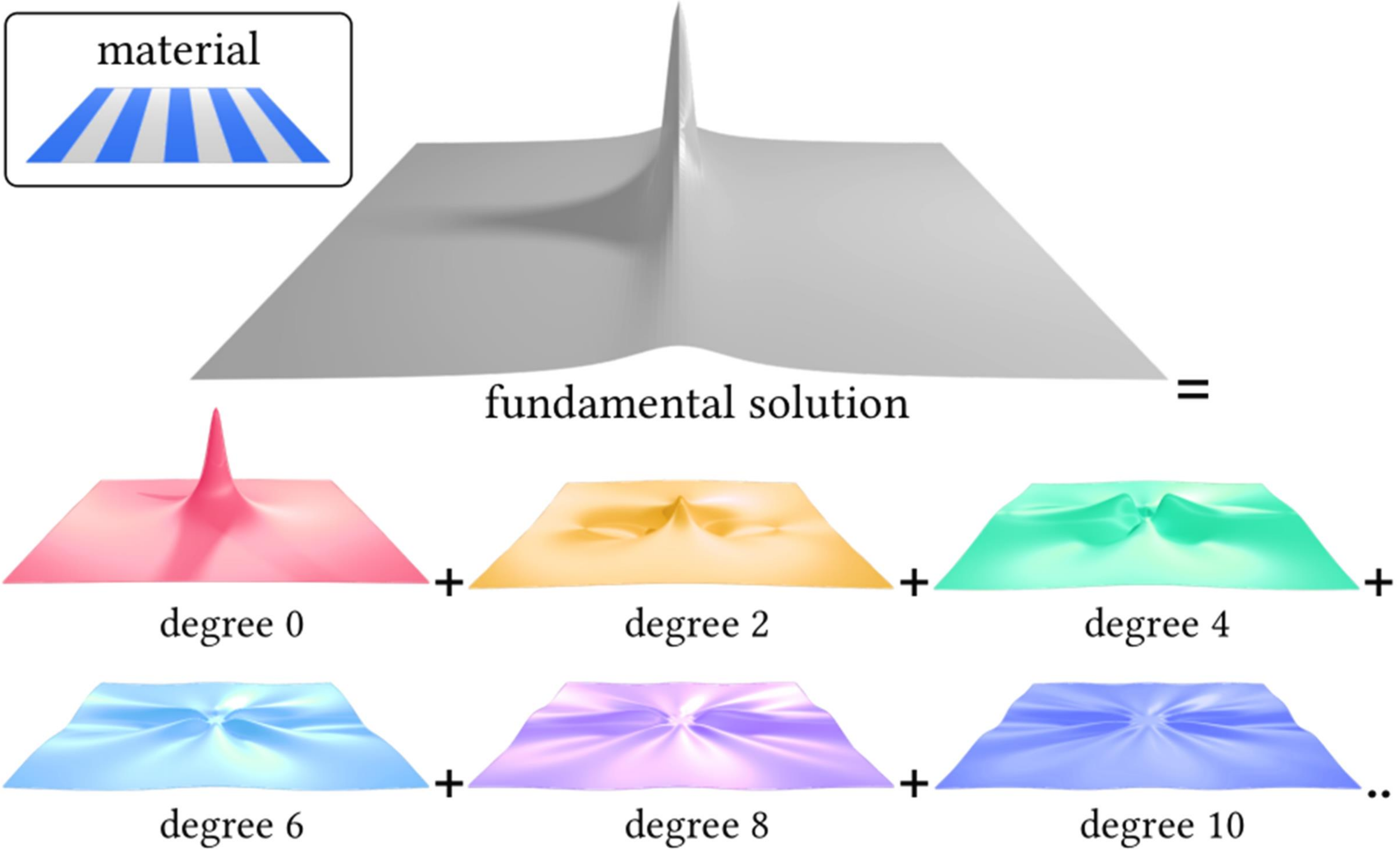
Radial term

$$\int_0^\infty \widehat{g}_\varepsilon(|\xi|) j_l(|\mathbf{x}||\xi|)\,\mathrm{d}|\xi|$$

$$\mathbf{G}(r,\theta,\varphi) = \frac{1}{2\pi^2}\sum_{l=0}^{\infty}\sum_{m=-l}^{l} \mathbb{i}^l\, Y_l^m(\theta,\varphi)\, R_l(r)\, \mathbf{P}_l^m(\mathbf{C}),$$

$$\int_{\mathbb{S}^2} (C_{ikjl}\widetilde{\xi}_k\widetilde{\xi}_l)^{-1}\overline{Y}_l^m(\widetilde{\xi})\,\mathrm{d}\mathbb{S}(\widetilde{\xi})$$

Material term



material
fundamental solution
=
degree 0
+
degree 2
+
degree 4
+
degree 6
+
degree 8
+
degree 10
..

ISOTROPIC CASE — KELVINLETS
[DE GOES & JAMES 2017]

- For isotropic material, $C_{ijkl} \Rightarrow \mu, \lambda$
- $\mathbf{G}(\mathbf{x})$ are only non-zero at degree **0** and **2**.

$$g_\varepsilon(r) = \delta(r)$$

$$G_{11}(r,\theta,\varphi) = \frac{\sin^2(\theta)(\lambda+\mu)\cos^2(\varphi)+\lambda+3\mu}{8\pi\mu r(\lambda+2\mu)},$$

$$G_{12}(r,\theta,\varphi) = G_{21}(r,\theta,\varphi) = \frac{\sin^2(\theta)(\lambda+\mu)\sin(\varphi)\cos(\varphi)}{8\pi\mu r(\lambda+2\mu)},$$

$$G_{13}(r,\theta,\varphi) = G_{31}(r,\theta,\varphi) = \frac{\sin(\theta)\cos(\theta)(\lambda+\mu)\cos(\varphi)}{8\pi\mu r(\lambda+2\mu)},$$

$$G_{22}(r,\theta,\varphi) = \frac{\sin^2(\theta)(\lambda+\mu)\sin^2(\varphi)+\lambda+3\mu}{8\pi\mu r(\lambda+2\mu)},$$

$$G_{23}(r,\theta,\varphi) = G_{32}(r,\theta,\varphi) = \frac{\sin(\theta)\cos(\theta)(\lambda+\mu)\sin(\varphi)}{8\pi\mu r(\lambda+2\mu)},$$

$$G_{33}(r,\theta,\varphi) = \frac{\lambda\,c\cdots(\theta)+\mu\cos^2(\theta)+\lambda\cdots}{8\pi\mu r(\lambda+2\mu)}$$

$$\boldsymbol{u}(\boldsymbol{r}) = \boxed{\left[\frac{(a-b)}{r}\boldsymbol{I} + \frac{b}{r^3}\boldsymbol{r}\boldsymbol{r}^t\right]}\boldsymbol{f}$$

$$g_\varepsilon(r) = 15\varepsilon^4/(8\pi)(r^2+\varepsilon^2)^{-\frac{7}{2}}$$

$$G_{11}(r,\theta,\varphi) = \frac{r^2\sin^2(\theta)(\lambda+\mu)\cos^2(\varphi)+r^2(\lambda+3\mu)+\varepsilon^2(2\lambda+5\mu)}{8\pi\mu(\lambda+2\mu)\left(r^2+\varepsilon^2\right)^{3/2}},$$

$$G_{12}(r,\theta,\varphi) = G_{21}(r,\theta,\varphi) = \frac{r^2\sin^2(\theta)(\lambda+\mu)\sin(\varphi)\cos(\varphi)}{8\pi\mu(\lambda+2\mu)\left(r^2+\varepsilon^2\right)^{3/2}},$$

$$G_{13}(r,\theta,\varphi) = G_{31}(r,\theta,\varphi) = \frac{r^2\sin(\theta)\cos(\theta)(\lambda+\mu)\cos(\varphi)}{8\pi\mu(\lambda+2\mu)\left(r^2+\varepsilon^2\right)^{3/2}},$$

$$G_{22}(r,\theta,\varphi) = \frac{r^2\sin^2(\theta)(\lambda+\mu)\sin^2(\varphi)+r^2(\lambda+3\mu)+\varepsilon^2(2\lambda+5\mu)}{8\pi\mu(\lambda+2\mu)\left(r^2+\varepsilon^2\right)^{3/2}},$$

$$G_{23}(r,\theta,\varphi) = G_{32}(r,\theta,\varphi) = \frac{r^2\sin(\theta)\cos(\theta)(\lambda+\mu)\sin(\varphi)}{8\pi\mu(\lambda+2\mu)\left(r^2+\varepsilon^2\right)^{3/2}},$$

$$G_{33}(r,\theta,\varphi) = \frac{r^2\cos^2(\theta)(\lambda+\cdots r^2(\lambda+3\mu)+\varepsilon^2(2\lambda+5\mu)}{8\pi\mu(\lambda\cdots 2\mu)\left(r^2+\varepsilon^2\right)^{3/2}}.$$

$$\boldsymbol{u}_\varepsilon(\boldsymbol{r}) = \boxed{\left[\frac{(a-b)}{r_\varepsilon}\boldsymbol{I} + \frac{b}{r_\varepsilon^3}\boldsymbol{r}\boldsymbol{r}^t + \frac{a}{2}\frac{\varepsilon^2}{r_\varepsilon^3}\boldsymbol{I}\right]}\boldsymbol{f}$$

Equivalent to Kelvinlet

EXTENSION TO GRADIENT

- Fourier transform of partial derivative

$$\widehat{G_{ij,p}} = \mathbb{i}\,\xi_p\,\widehat{G}_{ij}, \quad p=0,1,2,$$

$$\mathcal{R}_l(|\mathbf{x}|) = \int_0^\infty \widehat{g}_\varepsilon(|\xi|)\, j_l(|\mathbf{x}||\xi|)\,|\xi|\ \mathrm{d}|\xi|,$$

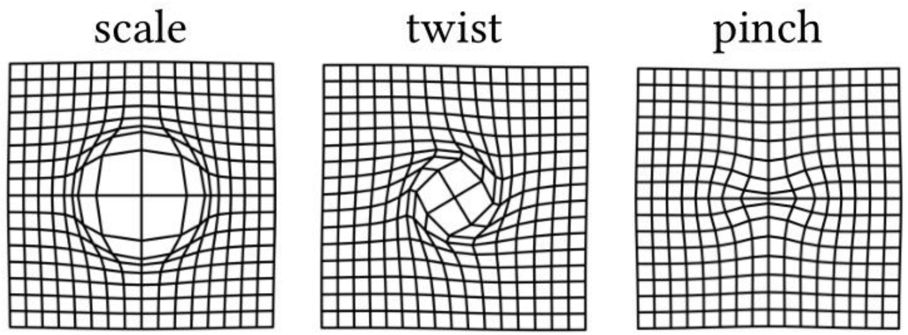
- SH expansion of gradient

$$\mathcal{P}^m_{l,p}(\mathrm{C}) = \int_{\mathbb{S}^2} (C_{ikjl}\widetilde{\xi}_k\widetilde{\xi}_l)^{-1}\overline{Y}^m_l(\widetilde{\xi})\,\widetilde{\xi}_p\ \mathrm{d}\mathbb{S}(\widetilde{\xi}).$$

$$\boxed{\nabla_p \mathrm{G}(\mathbf{x}) = \frac{1}{2\pi^2}\sum_{l=0}^{\infty}\sum_{m=-l}^{l} \mathbb{i}^{l+1}\, Y^m_l(\widetilde{\mathbf{x}})\,\mathcal{R}_l(|\mathbf{x}|)\,\mathcal{P}^m_{l,p}(\mathrm{C})}$$

- Then given an affine load $F(x) = \delta(x - x')H$

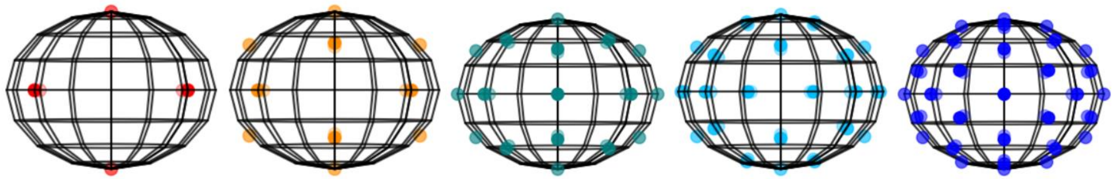
$$\mathbf{u}(\mathbf{x}) = \mathrm{Re}\left[\nabla \mathrm{G}(\mathbf{x}-\mathbf{x}')\right] : \mathbf{H},$$



IMPLEMENTATION

Offline computations

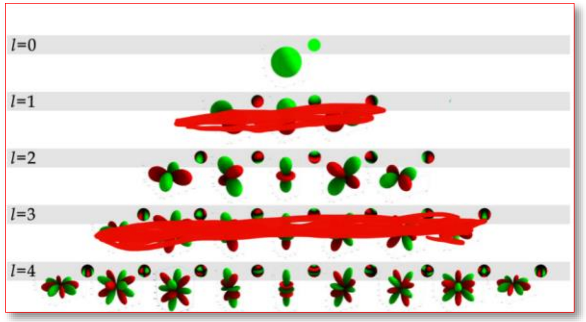
- $\mathbf{P}_l^m(\mathbf{C}) = \int_{\mathbb{S}^2} (C_{ikjl}\widetilde{\xi}_k\widetilde{\xi}_l)^{-1}\overline{Y}_l^m(\widetilde{\xi})\, \mathrm{d}\mathbb{S}(\widetilde{\xi})$
- and $\mathcal{P}_{l,p}^m(\mathrm{C}) = \int_{\mathbb{S}^2} (C_{ikjl}\widetilde{\xi}_k\widetilde{\xi}_l)^{-1}\overline{Y}_l^m(\widetilde{\xi})\,\widetilde{\xi}_p\, \mathrm{d}\mathbb{S}(\widetilde{\xi}).$
- No analytical expressions in general
- Can be precomputed using *Lebedev* quadrature

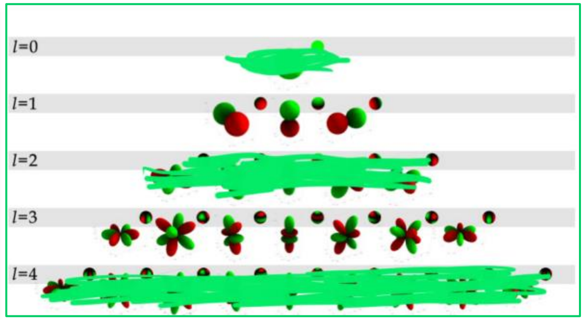


order = 6 order = 14 order = 26 order = 38 order = 50

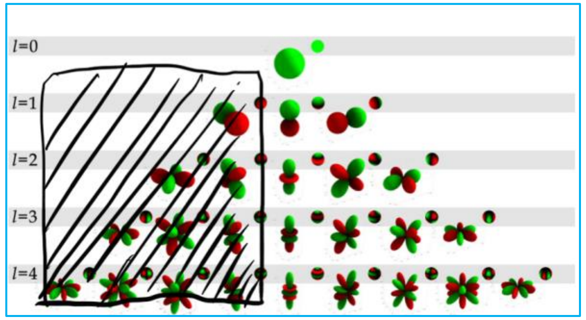
Online computations

- $\mathbf{G}_l(\mathbf{x}) \equiv 0$ for $l = 1,3,5 \ldots$
- $\nabla\mathbf{G}_l(\mathbf{x}) \equiv 0$ for $l = 0,2,4 \ldots$





- $\mathrm{Re}[\mathbf{G}_l^{-m}(\mathbf{x})] \equiv \mathrm{Re}[\mathbf{G}_l^{m}(\mathbf{x})]$
- $\mathrm{Re}[\nabla\mathbf{G}_l^{-m}(\mathbf{x})] \equiv \mathrm{Re}[\nabla\mathbf{G}_l^{m}(\mathbf{x})]$

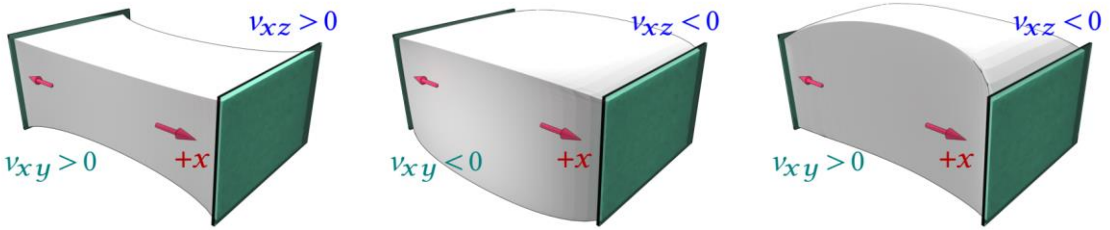


ANISOTROPY CONTROL

PARTICULAR TYPES OF MATERIAL

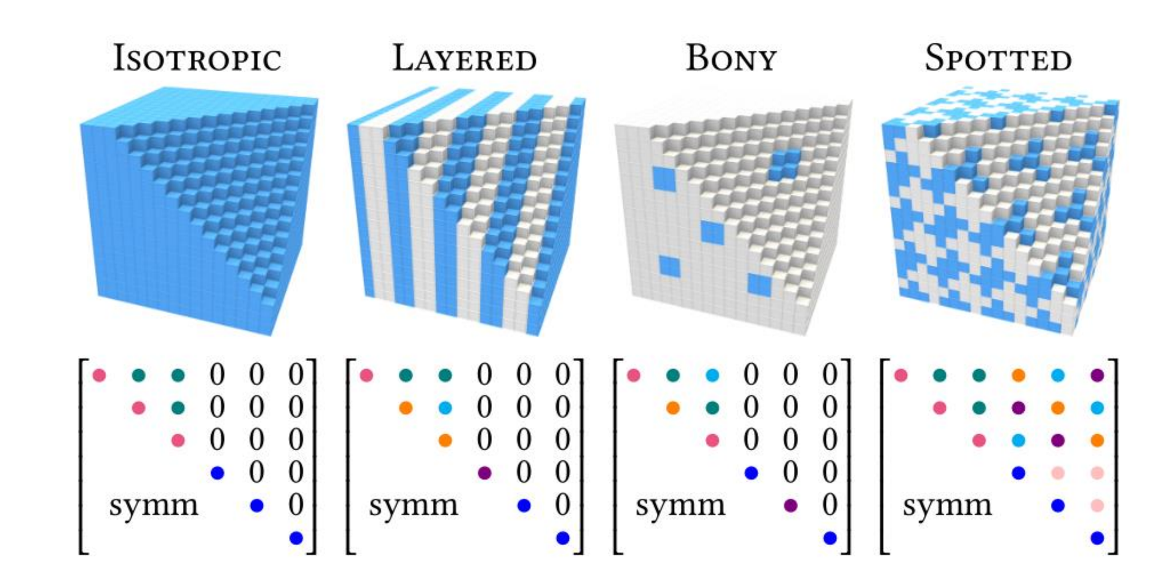
- e.g., orthotropic material with 3 Young's moduli, 3 shear moduli and 3 Poisson ratios

$$
S^V_{\text{orth}} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{yx}}{E_y} & -\frac{\nu_{zx}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xy}}{E_x} & \frac{1}{E_y} & -\frac{\nu_{zy}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xz}}{E_x} & -\frac{\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{zx}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xy}} \end{bmatrix},
$$



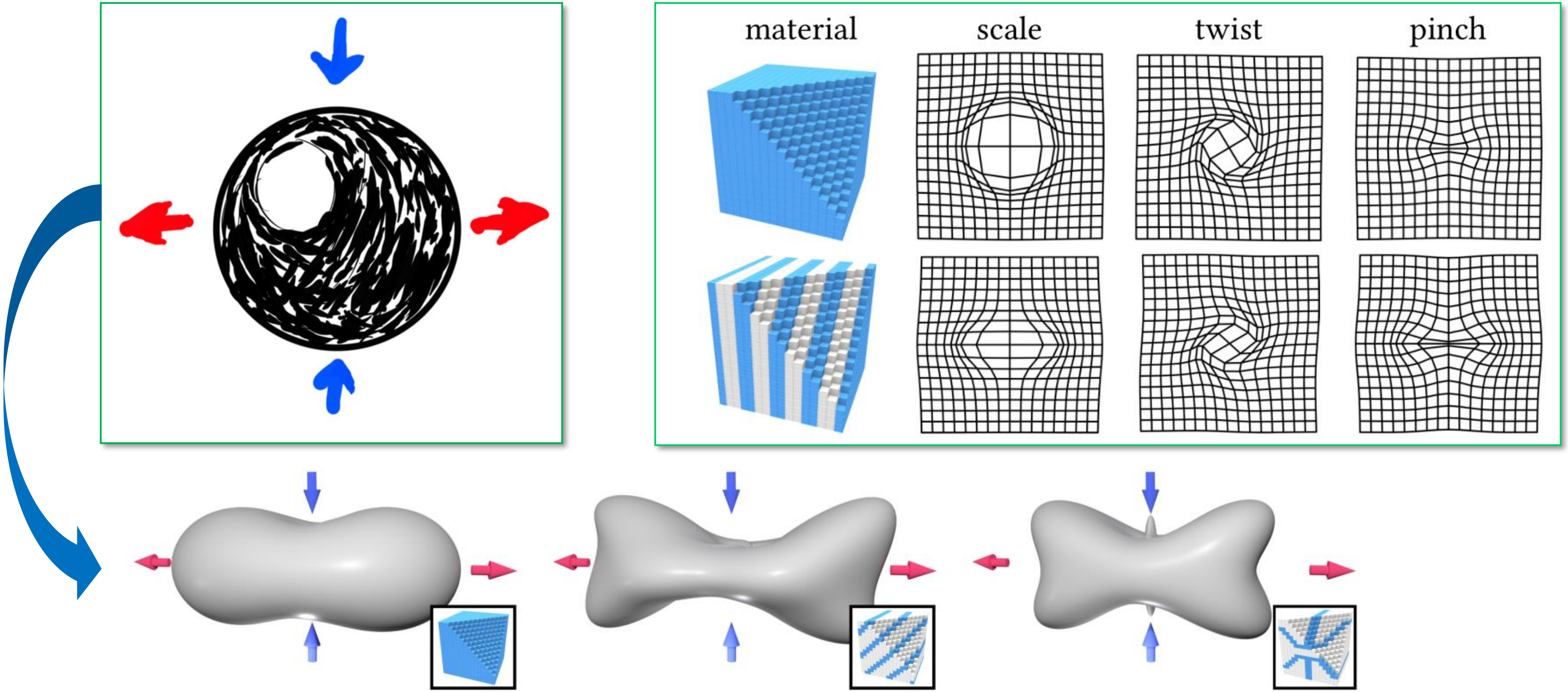
GENERAL CASES

- Homogenize bi-materials on a regular grid [Kharevych et al. 2008]



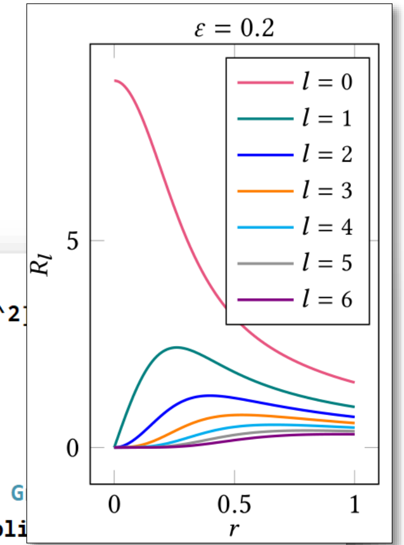
C_{ijkl} in Voigt notation

ANISOTROPY CONTROL



DEFORMATION PROPAGATION CONTROL

- Recall: $R_l(|x|) = \int_0^\infty \hat{g}_\varepsilon(|\xi|)\, j_l(|\mathbf{x}||\xi|)\, \mathrm{d}|\xi|$ which controls deformation propagation
- For specific smooth impulses..




```
In[3]:= (* smooth load [de Goes and James 2017] *)
        g[r_] := 15 ϵ^4 / (8 Pi) Power[r^2 + ϵ^2, -7 / 2]
Out[4]= 15 ϵ^4 / (8 π (r^2 + ϵ^2)^(7/2))

        (* Fourier transform of g(r): for regularized Green's function *)
        hatg = Power[2 Pi, 3 / 2] / Sqrt[ξ] Integrate[Simplify[BesselJ[1 / 2, r ξ]
            g[r] Power[r, 3 / 2]],
          {r, 0, Infinity}, Assumptions → ξ ≥ 0 && ϵ > 0]
Out[5]= 1/2 ϵ^2 ξ^2 BesselK[2, ϵ ξ]

        (* compute the radial term R_l *)
        R = Integrate[hatg * SphericalBesselJ[l, r ξ], {ξ, 0, Infinity},
          Assumptions → ϵ > 0 && l ≥ 0 && r > 0]
Out[6]= 1/2 √π r^l ϵ^(-1-l) Gamma[(1 + l)/2] Gamma[(5 + l)/2] Hypergeometric2F1Regularized[(1 + l)/2, (5 + l)/2, 3/2 + l, -r^2/ϵ^2]
```

$g_\epsilon(r)$

$\hat{g}_\epsilon(r)$

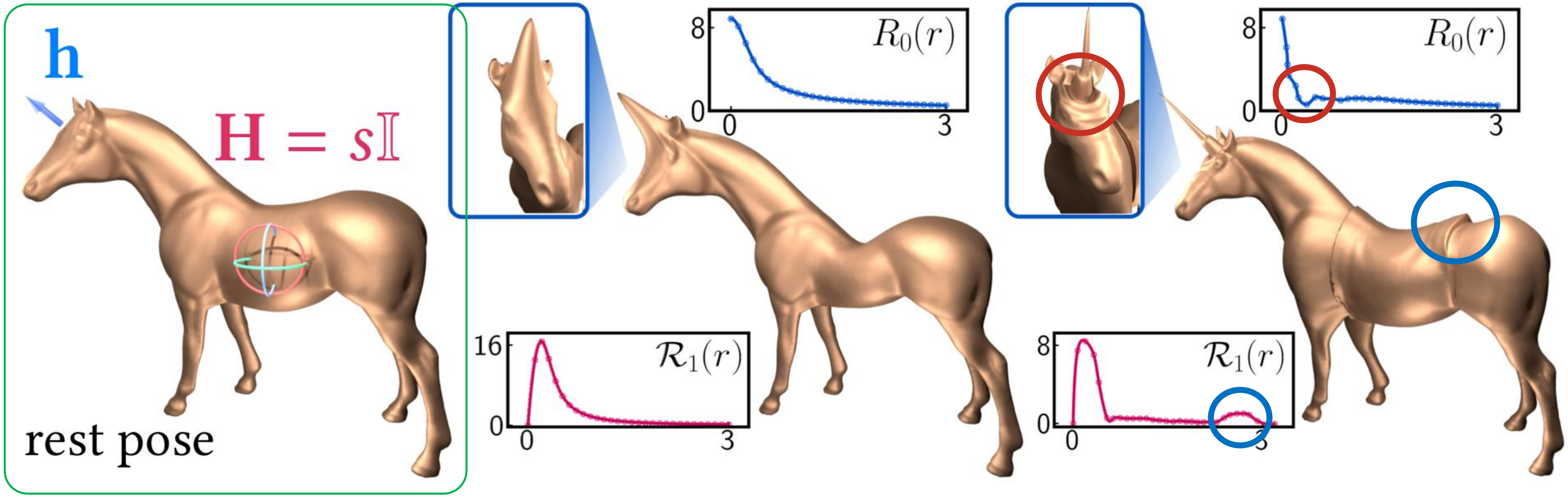
$R_l(r)$

```
         (* Gaussian load *)
         g[r_] := 1 / (ϵ^3 Power[Pi, 3 / 2]) Exp[-r^2 / ϵ^2]
Out[14]= e^(-r^2/ϵ^2) / (π^(3/2) ϵ^3)

In[15]:= (* Fourier transform of g(r): for regularized G
         hatg = Power[2 Pi, 3 / 2] / Sqrt[ξ] Integrate[Simpli
            g[r] Power[r, 3 / 2]],
          {r, 0, Infinity}, Assumptions → ξ ≥ 0 && ϵ > 0]
Out[15]= e^(-1/4 ϵ^2 ξ^2)

In[16]:= (* compute the radial term R_l *)
         R = Integrate[hatg * SphericalBesselJ[l, r ξ], {ξ, 0, Infinity},
          Assumptions → ϵ > 0 && l ≥ 0 && r > 0]
Out[16]= 1/2 √π r^l ϵ^(-1-l) Gamma[(1 + l)/2] Hypergeometric1F1Regularized[(1 + l)/2, 3/2 + l, -r^2/ϵ^2]
```

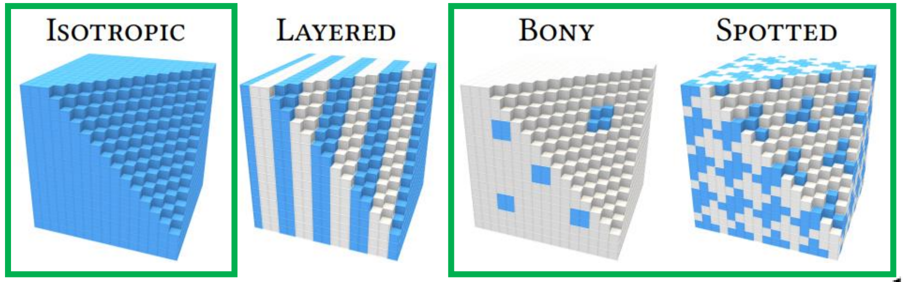
DEFORMATION PROPAGATION CONTROL

- **Limitations:**
  - $g_\epsilon(r)$ is neither intuitive nor flexible
  - the integral $R_l(r)$ (or $\mathcal{R}_l(r)$) is hard to evaluate given $g_\epsilon(r)$
    - may not even exist!

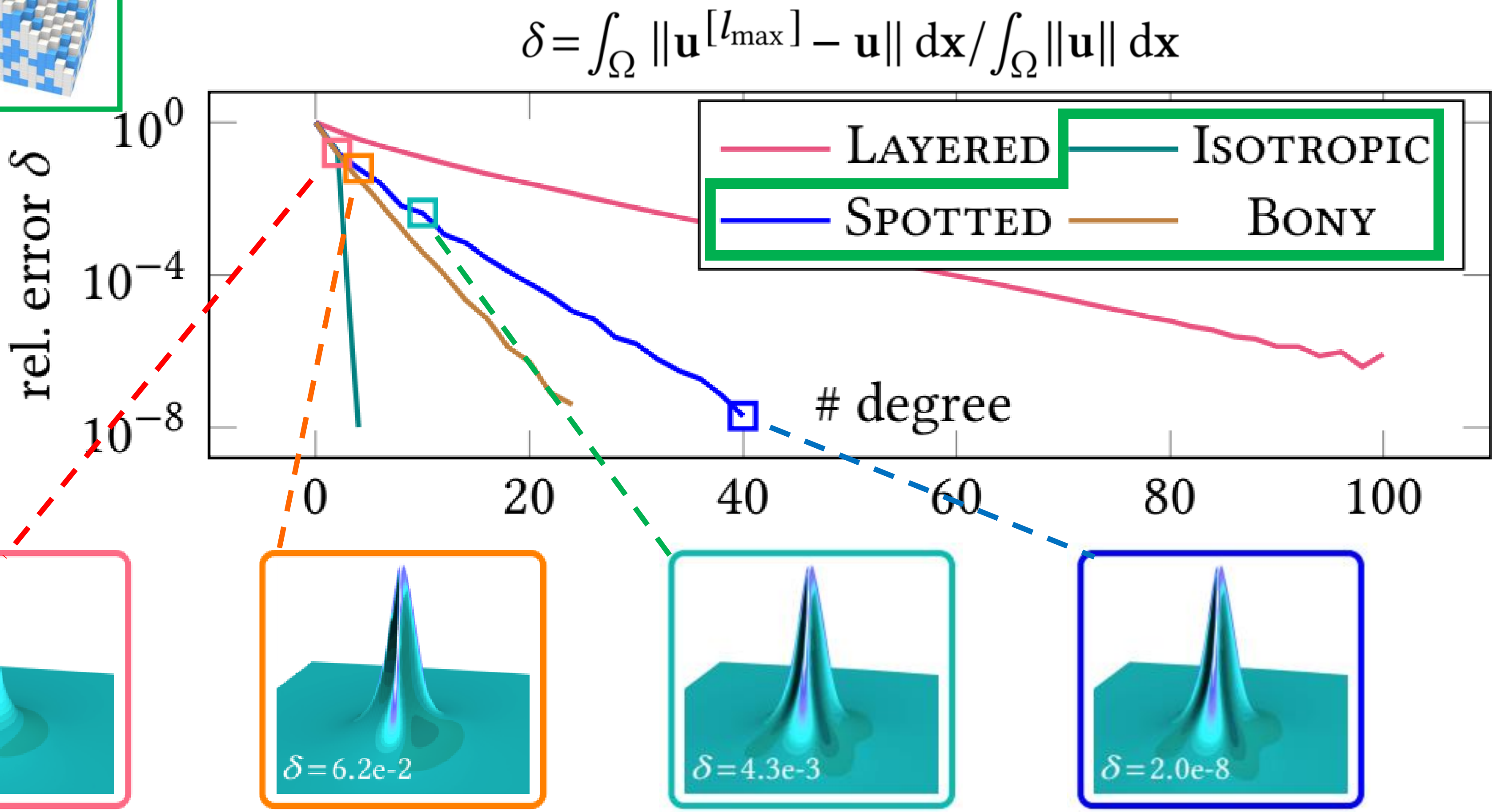
- **Our approach**:
  - Edit $R_l(r)$ (or $\mathcal{R}_l(r)$) directly via cubic splines instead of constructing an integrable $g_\epsilon(r)$



TRUNCATING SPHERICAL HARMONIC SERIES



- $l_{\max} = 20$ for strong anisotropy
- $l_{\max} = 8$, otherwise

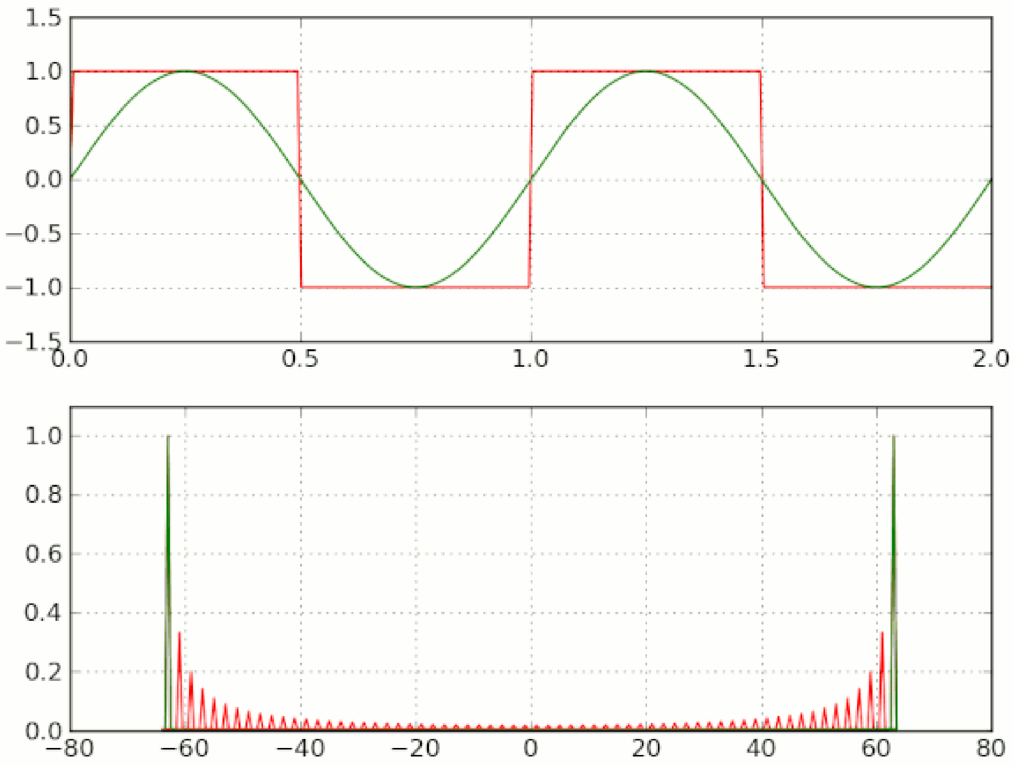


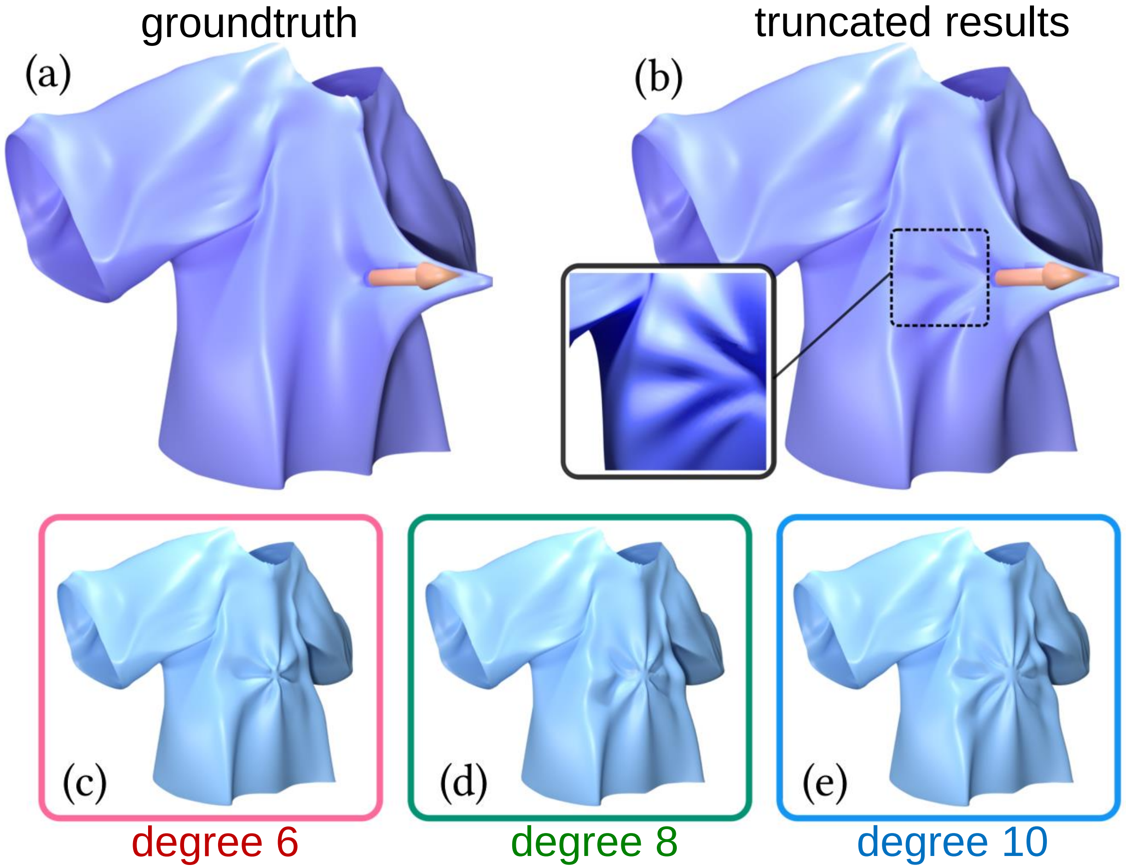
EXPLOIT EARLY TRUNCATION

Turn it into a feature!

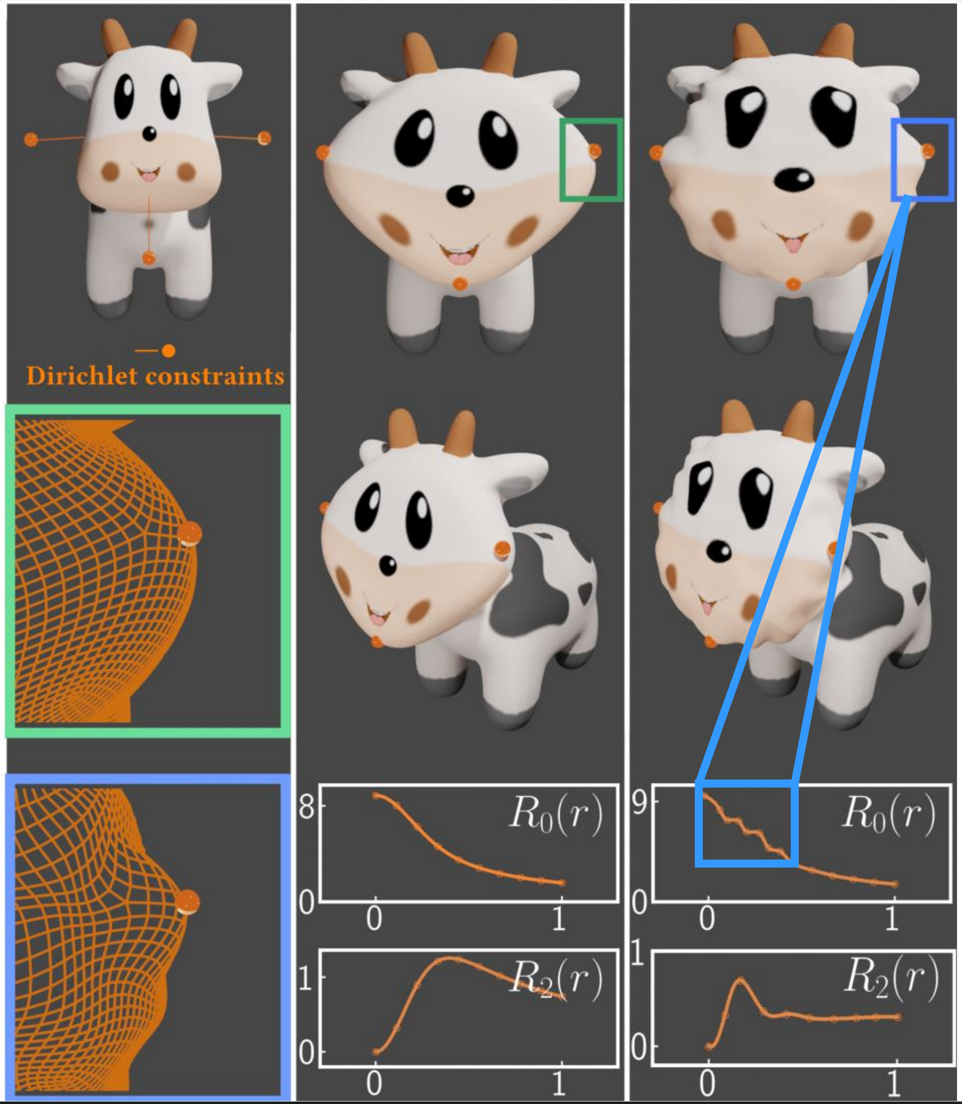
Gibbs phenomenon:
discontinuity generates ringing artifacts!

Wrinkle generation





CONSTRAINED DEFORMATION

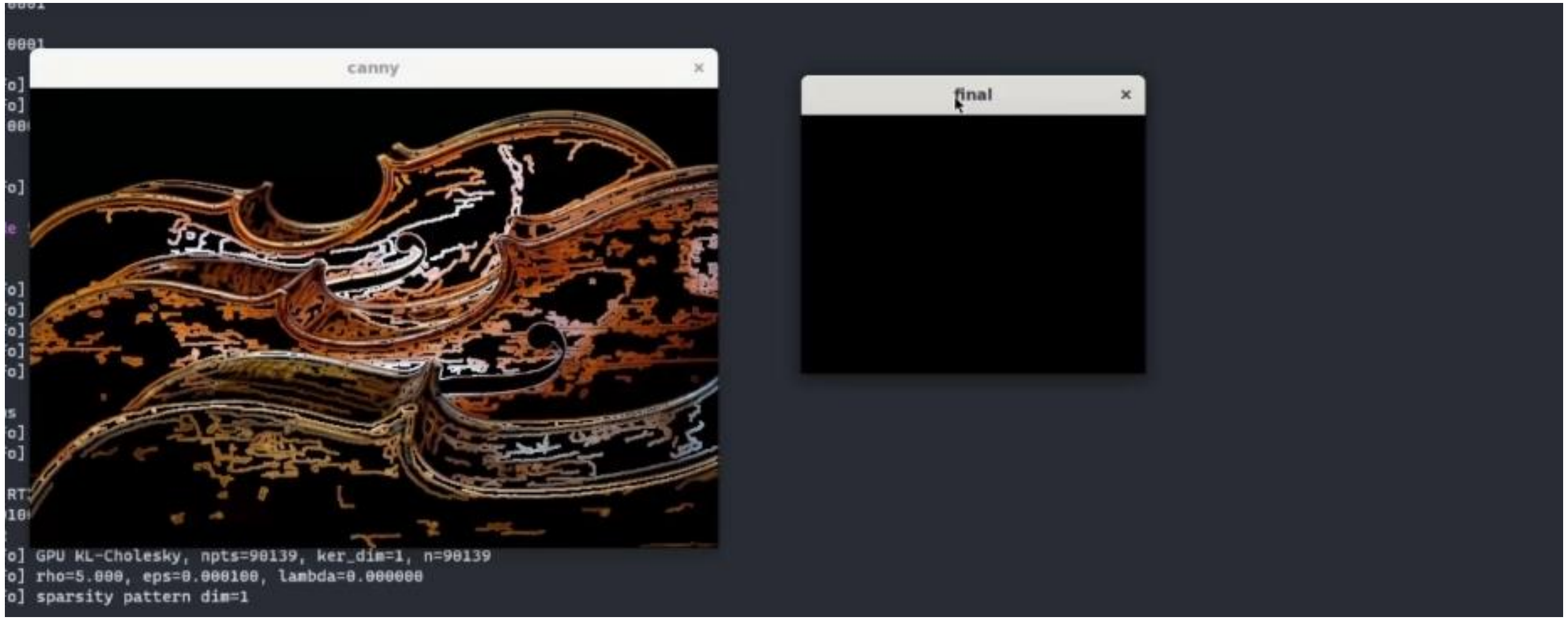


- Solve a dense linear system

$$\begin{bmatrix} \mathrm{Re}[\mathbf{G}(\mathbf{x}_0 - \mathbf{x}_0)] & \dots & \mathrm{Re}[\mathbf{G}(\mathbf{x}_0 - \mathbf{x}_{k-1})] \\ \vdots & \ddots & \vdots \\ \mathrm{Re}[\mathbf{G}(\mathbf{x}_{k-1} - \mathbf{x}_0)] & \dots & \mathrm{Re}[\mathbf{G}(\mathbf{x}_{k-1} - \mathbf{x}_{k-1})] \end{bmatrix} \begin{bmatrix} \mathbf{h}_0 \\ \vdots \\ \mathbf{h}_{k-1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{k-1} \end{bmatrix}$$

- By superposition:

$$\boxed{\mathbf{u}(\mathbf{x}) = \sum_{i=0}^{k} \mathbf{G}(\mathbf{x} - \mathbf{x}_i)\mathbf{h}_i}$$



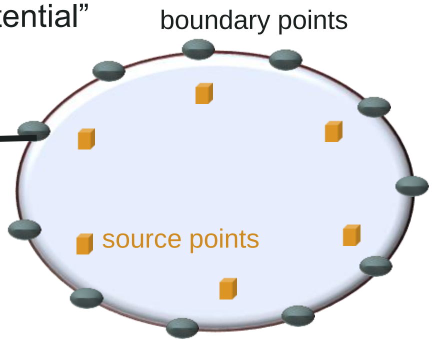
SCALABLE INVERSE CHOLESKY PRECONDITIONER

SPEEDUP ITERATIVE SOLVERS FOR MESHLESS BIES

RECAP BEM

- Methods to build Boundary Integral Equation (BIE) systems [Costabel 1984]
  - Direct approaches: solve for Dirichlet or Neumann boundary conditions
    - based on Green's third identity or its variants
  - Indirect approaches: solve for an unknown density on the boundary
    - E.g., “charges” for potential problems, “forces” for elasticity
- Indirect approach: single layer potential for Dirichlet problems
    - **Solve stage:** solve for “charges” that enforce a set of given boundary “potential”

$$\int_{\mathcal{M}} G(\boldsymbol{z},\boldsymbol{y})\sigma(\boldsymbol{y})\,\mathrm{d}v_{\boldsymbol{y}} = b(\boldsymbol{z})\ \forall \boldsymbol{z} \in \mathcal{M}.$$



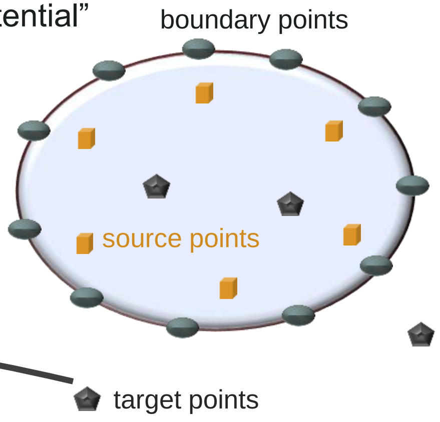
RECAP BEM

- Methods to build Boundary Integral Equation (BIE) systems [Costabel 1984]
  - Direct approaches: solve for Dirichlet or Neumann boundary conditions
    - based on Green's third identity or its variants
  - Indirect approaches: solve for an unknown density on the boundary
    - E.g., "charges" for potential problems, "forces" for elasticity
- Indirect approach: single layer potential for Dirichlet problems
    - **Solve stage:** solve for "charges" that enforce a set of given boundary "potential"

$$\boxed{\int_{\mathcal{M}} G(\boldsymbol{z},\boldsymbol{y})\sigma(\boldsymbol{y})\,\mathrm{d}v_{\boldsymbol{y}} = b(\boldsymbol{z}) \quad \forall \boldsymbol{z}\in\mathcal{M}.}$$

    - **Evaluation stage:** evaluate the "potential" at any target point in space

$$\boxed{u(\boldsymbol{x}) = \int_{\mathcal{M}} G(\boldsymbol{x},\boldsymbol{y})\sigma(\boldsymbol{y})\,\mathrm{d}v_{\boldsymbol{y}}.}$$



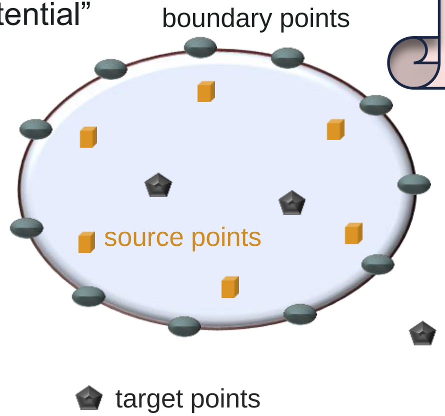
RECAP BEM

- Methods to build Boundary Integral Equation (BIE) systems [Costabel 1984]
  - Direct approaches: solve for Dirichlet or Neumann boundary conditions
    - based on Green's third identity or its variants
  - Indirect approaches: solve for an unknown density on the boundary
    - E.g., "charges" for potential problems, "forces" for elasticity
- Indirect approach: single layer potential for Dirichlet problems
  - **Solve stage:** solve for "charges" that enforce a set of given boundary "potential"

$$\int_{\mathcal{M}} G(\boldsymbol{z}, \boldsymbol{y})\sigma(\boldsymbol{y})\, \mathrm{d}v_{\boldsymbol{y}} = b(\boldsymbol{z}) \quad \forall \boldsymbol{z} \in \mathcal{M}.$$

  - **Evaluation stage:** evaluate the "potential" at any target point in space

$$u(\boldsymbol{x}) = \int_{\mathcal{M}} G(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y})\, \mathrm{d}v_{\boldsymbol{y}}.$$



- Results in *Fredholm integral equation of the first kind*, more Ill-posed than the *second kind*
- Need efficient preconditioners
- ***Any symmetric and sparse structures to leverage to get a stable and scalable solver?***

EXPLOIT SYMMETRY

Discretize boundary integral equations (BIE)

$$K\ s = b$$

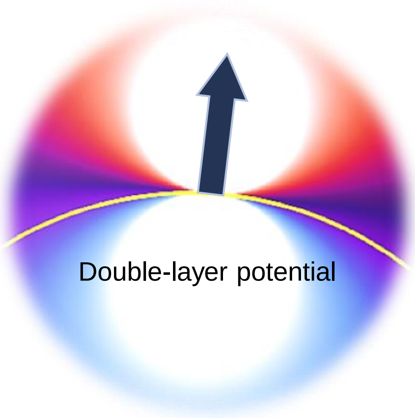
$$\int_{\mathcal{M}} G(\mathbf{z}, \mathbf{y})\sigma(\mathbf{y})\,\mathrm{d}v_{\mathbf{y}} = b(\mathbf{z}) \quad \forall \mathbf{z} \in \mathcal{M}.$$

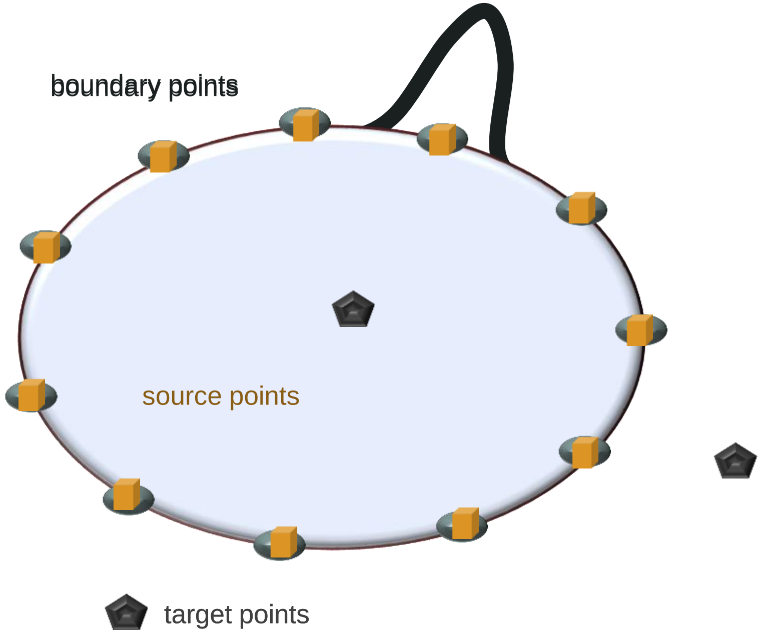
$$\sum_{j=1}^{S}\left(\iint_{\mathcal{M}\times\mathcal{M}} \phi_i(\mathbf{y})G(\mathbf{y},\mathbf{z})\psi_j(\mathbf{z})\,\mathrm{d}v_{\mathbf{y}}\mathrm{d}v_{\mathbf{z}}\right) s_j = \int_{\mathcal{M}} b(\mathbf{y})\phi_i(\mathbf{y})\mathrm{d}v_{\mathbf{y}}$$

Discretize boundary data $b(z) = \sum_i \psi_i(z) b_i$

Discretize sources $\sigma(y) = \sum_j \phi_j(y) s_j$

- To obtain a symmetric discrete BIE
  - Either identical basis functions for collocated source and boundary points
  - Or solve least-squares problem $\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{s} = \boldsymbol{K}^T\boldsymbol{b}$
    - e.g., Fredholm integral equations of the second kind
      - Double-layer potential for Dirichlet problems
      - Single-layer potential for Neumann problems





EXPLOIT SPARSITY

- Directly applying incomplete Cholesky to factorize K

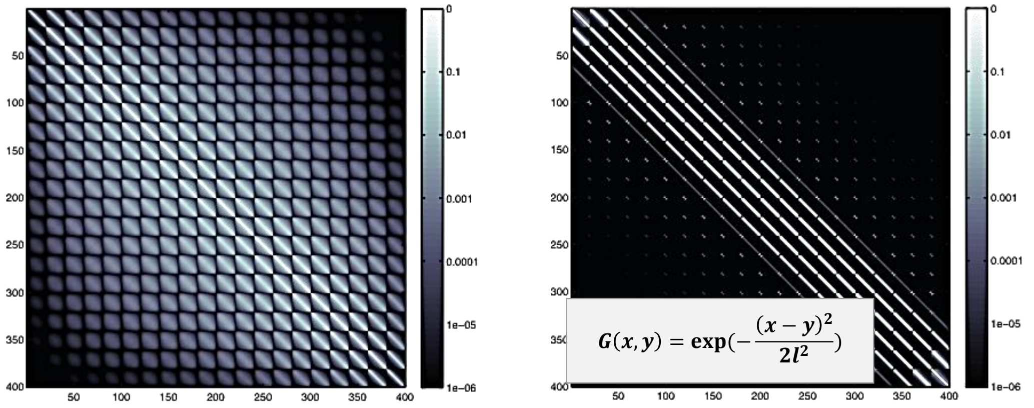
  [Chen et al. 2021]
  - *Accuracy issue:* Numerous entries must be dropped out for efficiency
  - *Stability issue:* Loss of positive definiteness causes breakdowns
- However, boundary integral operators are conceptually close to the inverse of their differential operator
  - Green function is the solution subject to a singular impulse
  - E.g., in elasticity, a BIE matrix acts like the inverse of stiffness, or compliance
- So, the inverse of BIE matrices could be sparse
  - True for many covariance matrices assembled by fast-decaying kernel functions in Gaussian Process
  - Similar for Green's functions as well

Compliance

$$Ks = b$$

Forces Displacements

$$K \approx LL^T$$



(a) Inverse Laplacian matrix (b) Inverse exponential covariance matrix

[Chow and Saad 2014]

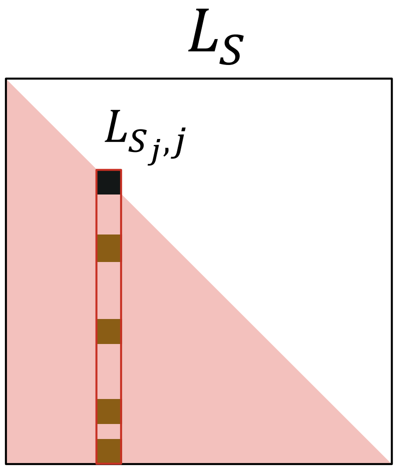
INVERSE CHOLESKY PRECONDITIONER

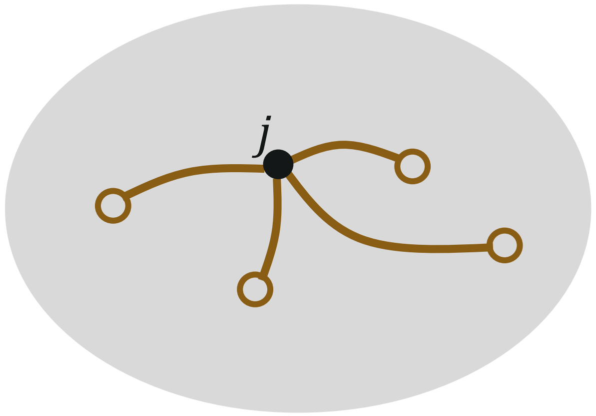
- We leverage inverse Cholesky factorization to precondition BIE matrices

$$Ks = b \longrightarrow \boxed{K^{-1} \approx L_S L_S^T} \longrightarrow s \approx L_S L_S^T b$$

- Kaporin's construction for L_S [Kaporin 1994]

$$\mathbf{L}_{s_j,j} = \frac{\mathbf{K}^{-1}_{s_j,s_j} \mathbb{e}_j}{\sqrt{\mathbb{e}_j^{\mathsf{T}} \mathbf{K}^{-1}_{s_j,s_j} \mathbb{e}_j}}, \qquad \forall j = 1..B,$$





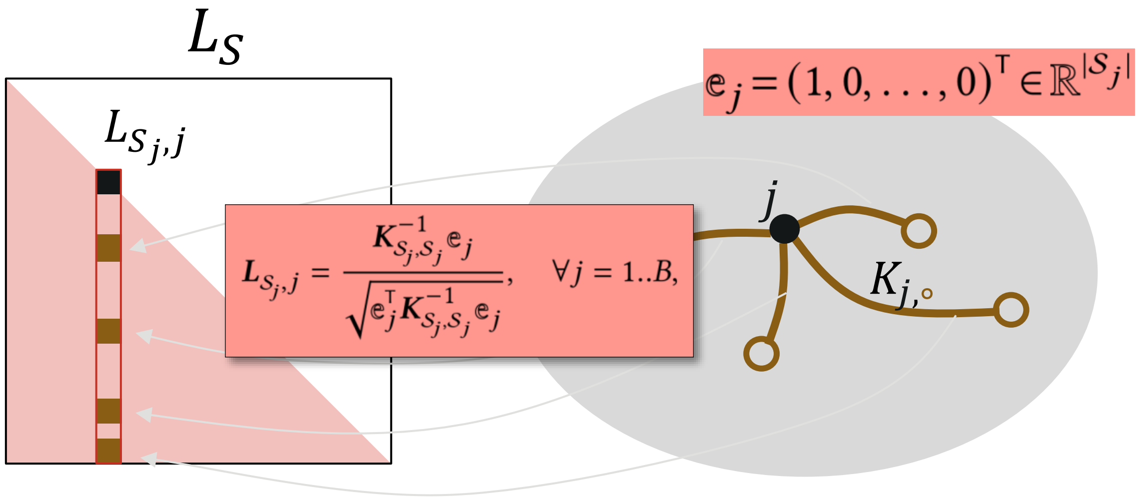
INVERSE CHOLESKY PRECONDITIONER

- We leverage inverse Cholesky factorization to precondition BIE matrices

$$Ks = b \rightarrow \boxed{K^{-1} \approx L_S L_S^T} \rightarrow s \approx L_S L_S^T b$$

- Kaporin's construction for L_S [Kaporin 1994]

$$\boldsymbol{L}_{S_j,j} = \frac{\boldsymbol{K}_{S_j,S_j}^{-1} \mathbb{e}_j}{\sqrt{\mathbb{e}_j^{\mathsf{T}} \boldsymbol{K}_{S_j,S_j}^{-1} \mathbb{e}_j}}, \qquad \forall j = 1..B,$$



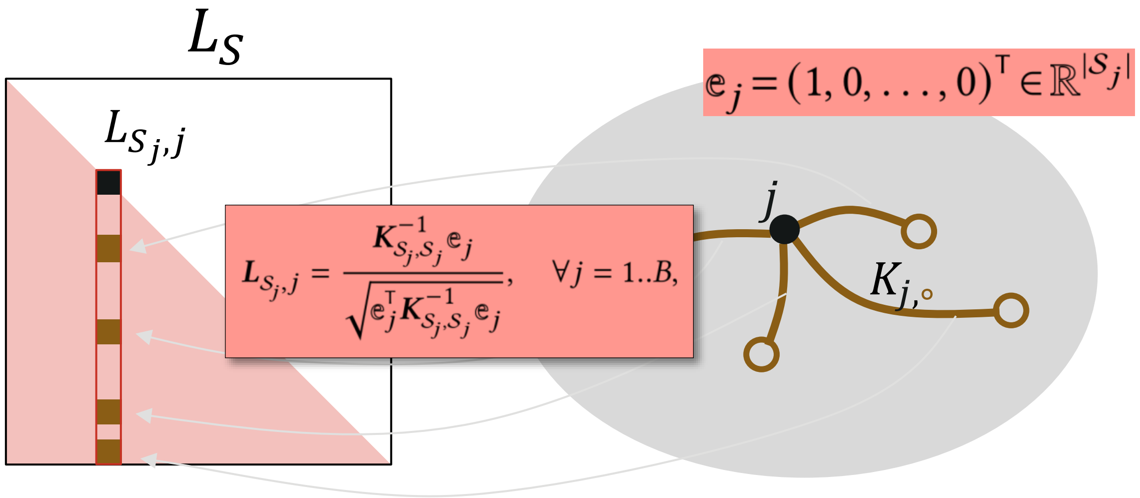
INVERSE CHOLESKY PRECONDITIONER

- We leverage inverse Cholesky factorization to precondition BIE matrices

$$Ks = b \rightarrow \boxed{K^{-1} \approx L_S L_S^T} \rightarrow s \approx L_S L_S^T b$$

- Kaporin's construction for L_S [Kaporin 1994]

$$\boxed{\mathbf{L}_{S_j,j} = \frac{\mathbf{K}^{-1}_{S_j,S_j}\mathbb{e}_j}{\sqrt{\mathbb{e}_j^{\mathsf{T}}\mathbf{K}^{-1}_{S_j,S_j}\mathbb{e}_j}}, \qquad \forall j = 1..B,}$$

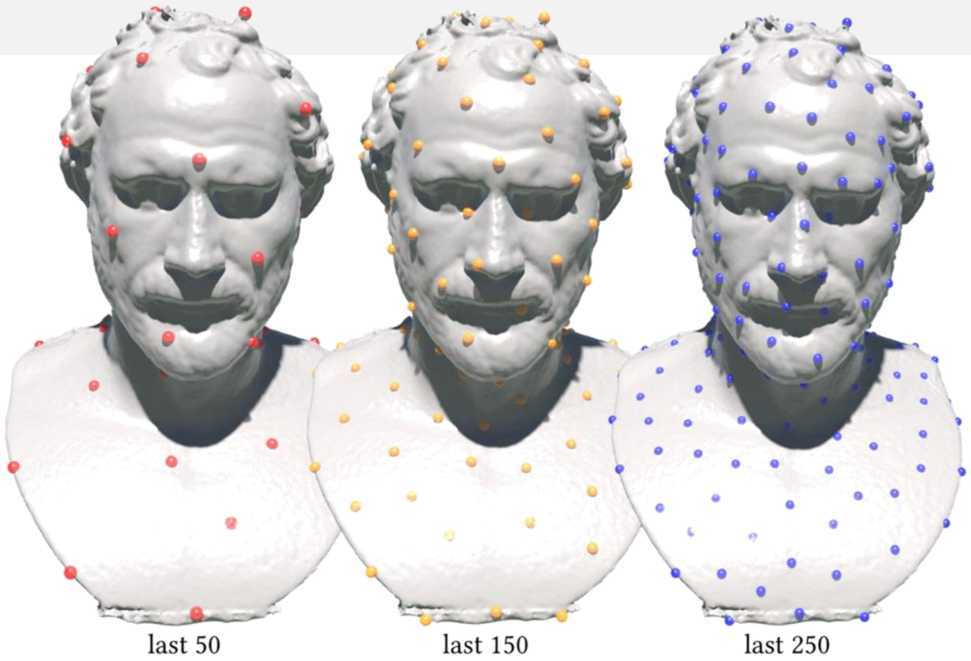


- Properties
  - **Massively parallel:** each column of L_S is computed independently to others. Good for GPUs!
  - **Memory efficient:** no need to assemble the global BIE matrix.
  - **Stable:** no breakdowns will occur
  - **Variational interpretation(s):** minimizing Kaporin's condition number*, KL-divergence, and a constrained quadratic form

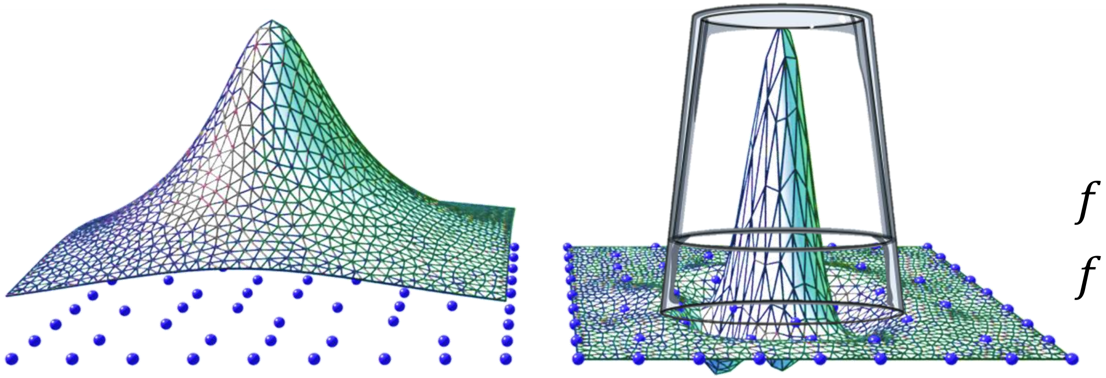
$$^{*}\ \kappa_{\text{Kap}}(M) = \frac{1}{B}\frac{\operatorname{tr}(M)}{\det(M)^{1/B}}$$

REORDERING DOFS

- Fine-to-coarse ordering by farthest point sampling [Chen et al. 2021]
  - Max-min ordering $i_k = \underset{q}{\operatorname{argmax}} \min_{p \in \{0, k-1\}} \operatorname{dist}(\boldsymbol{y}_q, \boldsymbol{y}_{i_p}),$
  - Reverse max-min ordering $\boldsymbol{P} = \{i_{B-1}, \ldots, i_1, i_0\}$, i.e., fine-to-coarse
- Intuition
  - Make sampling points space uniformly within each scale
  - The screening effect in kriging [Stein 2002]
    - GP: conditioning a subset of points results in localized correlations



A fine-to-coarse reordering

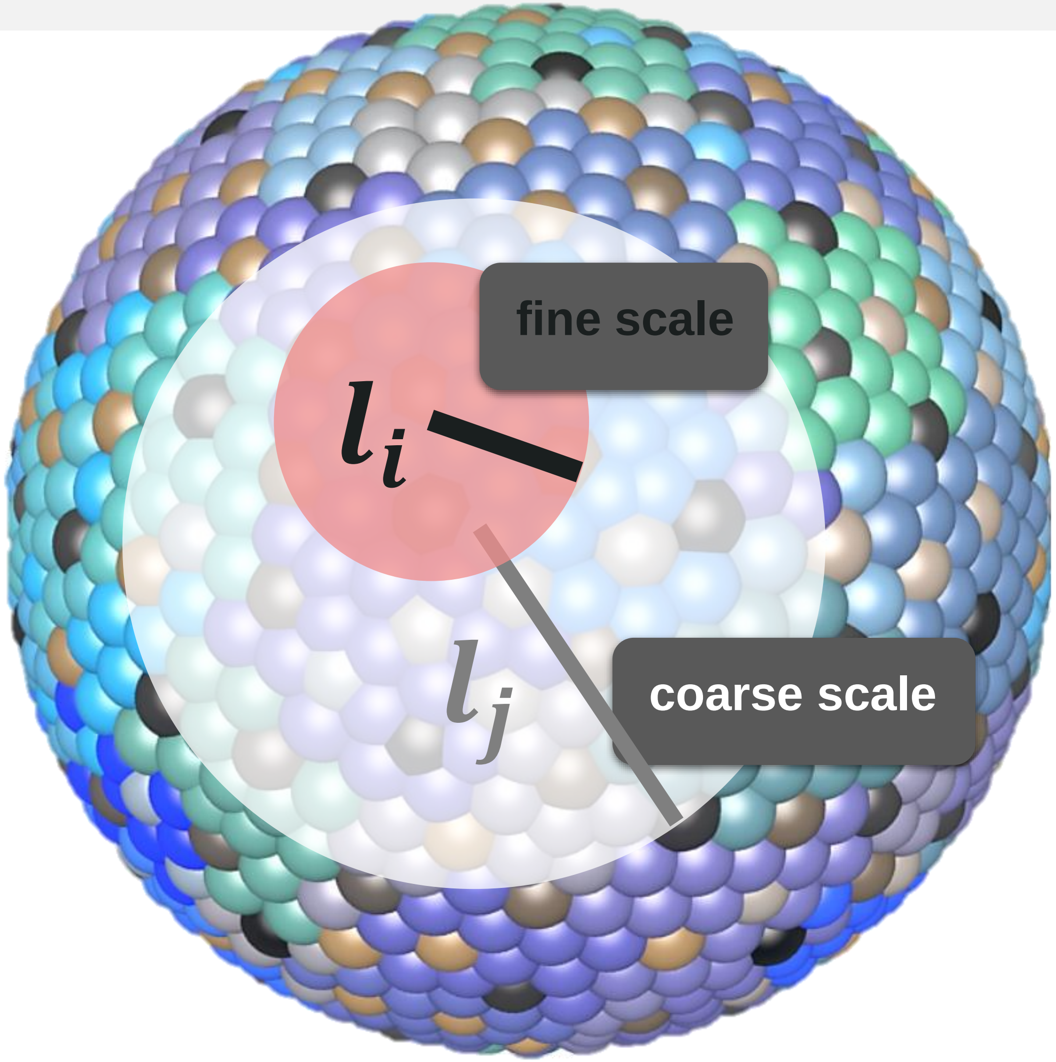


$$f(A,B,C,D) = f(A)f(B|A)f(C|A,B)f(D|A,B,C) = N(0,\Sigma)$$

$$f(A,B,C,D) \approx f(A)f(B|A)f(C|A,\cancel{B})f(D|A,B,\cancel{C}) = N\left(0,\left(LL^{\mathrm{T}}\right)^{-1}\right)$$

Too far Too far

CONSTRUCTING SPARSITY PATTERN



- Length scale returned in coarse-to-fine ordering

$$\ell_{i_k} = \min_{p \in \{0,k-1\}} \mathrm{dist}(\boldsymbol{y}_{i_k}, \boldsymbol{y}_{i_p})$$

- Lower-triangular sparsity patter

$$\mathcal{S} := \{(i,j) | i \geq j \text{ and } \mathrm{dist}(x_i, x_j) \leq \rho \boxed{\min(\ell_i, \ell_j)}\}$$

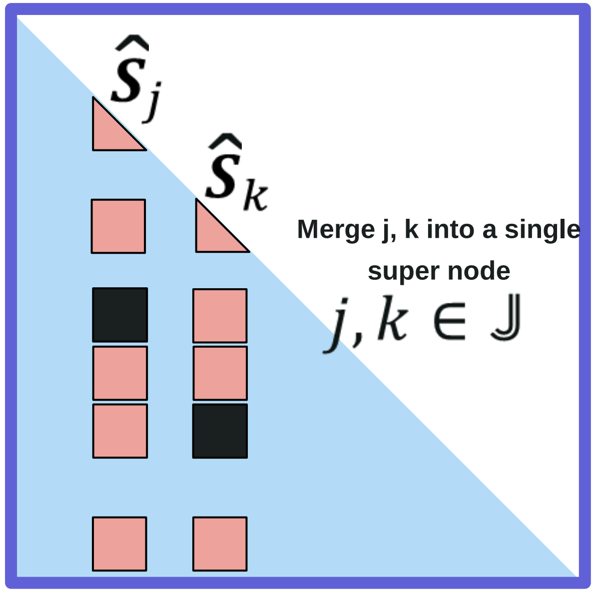
- **Again, screening effect:** a fine-scale point is unlikely to be correlated to distant points on coarser scales

EFFICIENT IMPLEMENTATION

FOR PRECONDTIONER

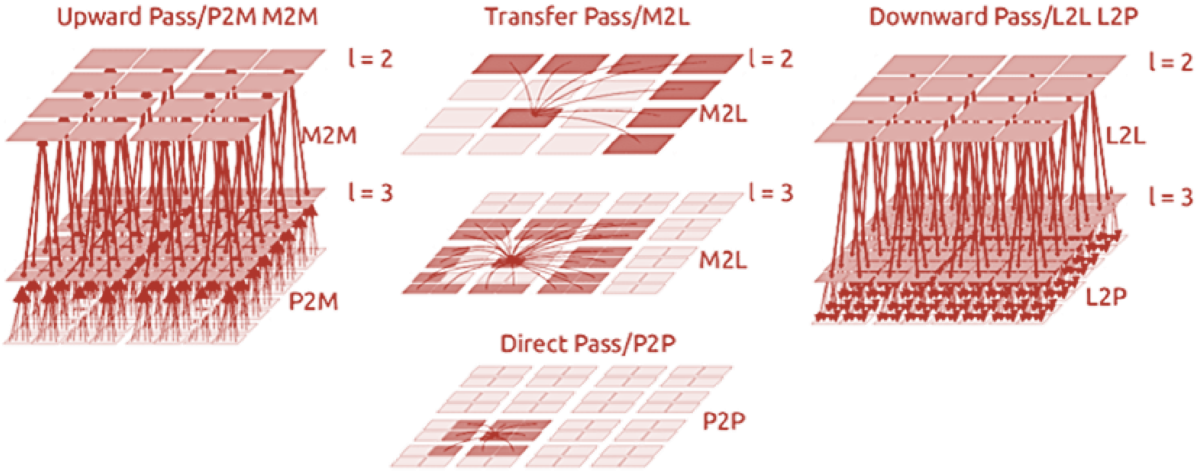
- Supernode mode to reuse local factorizations as much as possible

Supernodal sparsity pattern



FOR PCG ITERATIONS

- Fast Multipole Method to evaluate matrix-vector products



EXAMPLES OF APPLICATION

LAPLACE'S EQUATION

$$\Delta u = 0$$

$$G(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} -\dfrac{1}{2\pi} \ln(r), & \text{in 2D} \\ \dfrac{1}{4\pi r}, & \text{in 3D} \end{cases}$$

LINEAR ELASTICITY

$$\Delta u + \frac{1}{1-2\nu} \nabla(\nabla \cdot u) = 0,$$

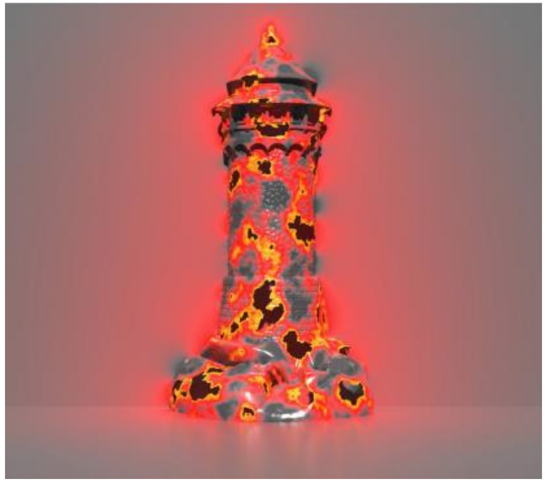
$$G(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \dfrac{a-b}{r} \ln(1/r) \boldsymbol{I} + \dfrac{b}{r^2} \boldsymbol{r}\boldsymbol{r}^{\mathsf{T}}, & \text{in 2D} \\ \dfrac{a-b}{r} \boldsymbol{I} + \dfrac{b}{r^3} \boldsymbol{r}\boldsymbol{r}^{\mathsf{T}}, & \text{in 3D} \end{cases}$$

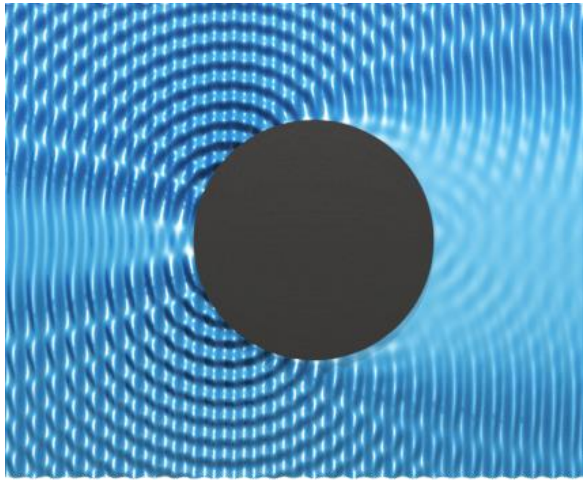
HELMHOLTZ EQUATION

$$\Delta u + k^2 u = 0,$$

$$G(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \dfrac{\mathbb{i}}{4} H_0^{(1)}(kr), & \text{in 2D,} \\ \dfrac{\exp(\mathbb{i}kr)}{4\pi r}, & \text{in 3D,} \end{cases}$$

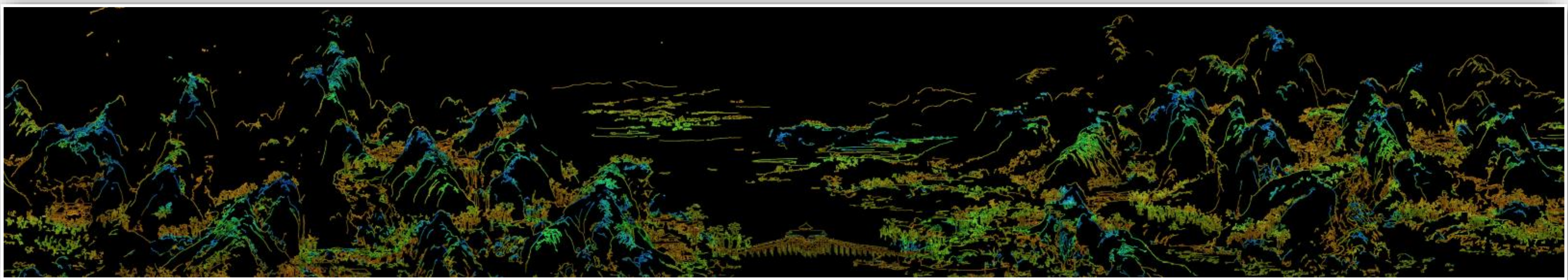
The method of fundamental solutions (MFS)



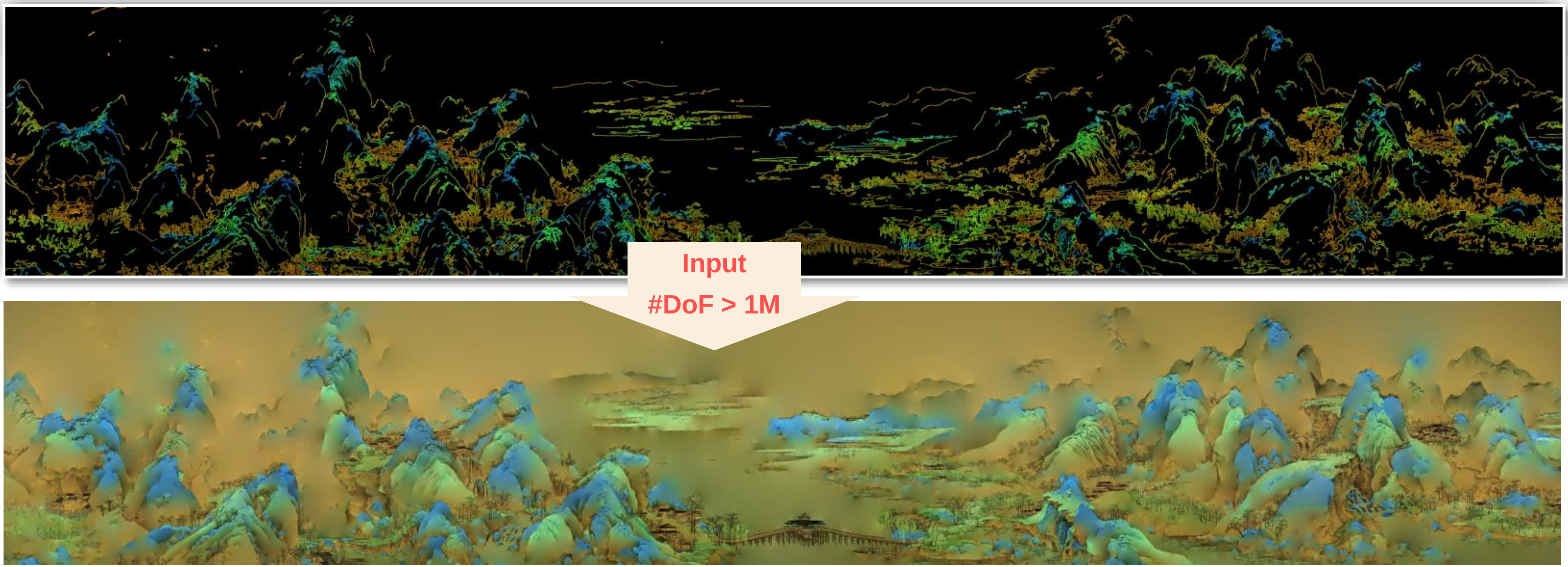


LAPLACE'S EQUATION

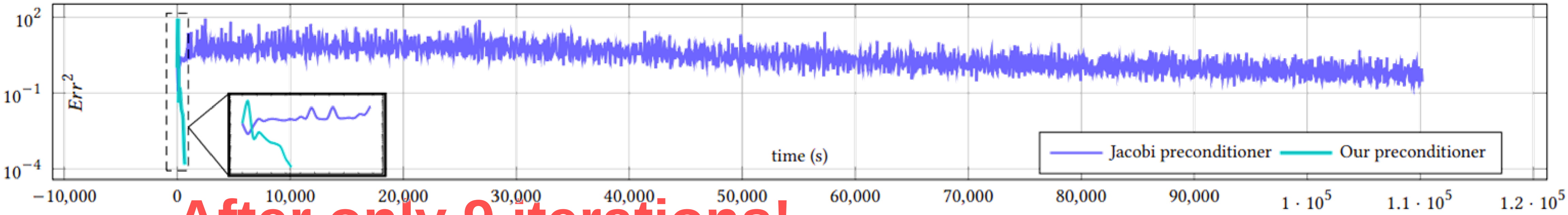
LAPLACE'S EQUATION



LAPLACE’S EQUATION



LAPLACE’S EQUATION



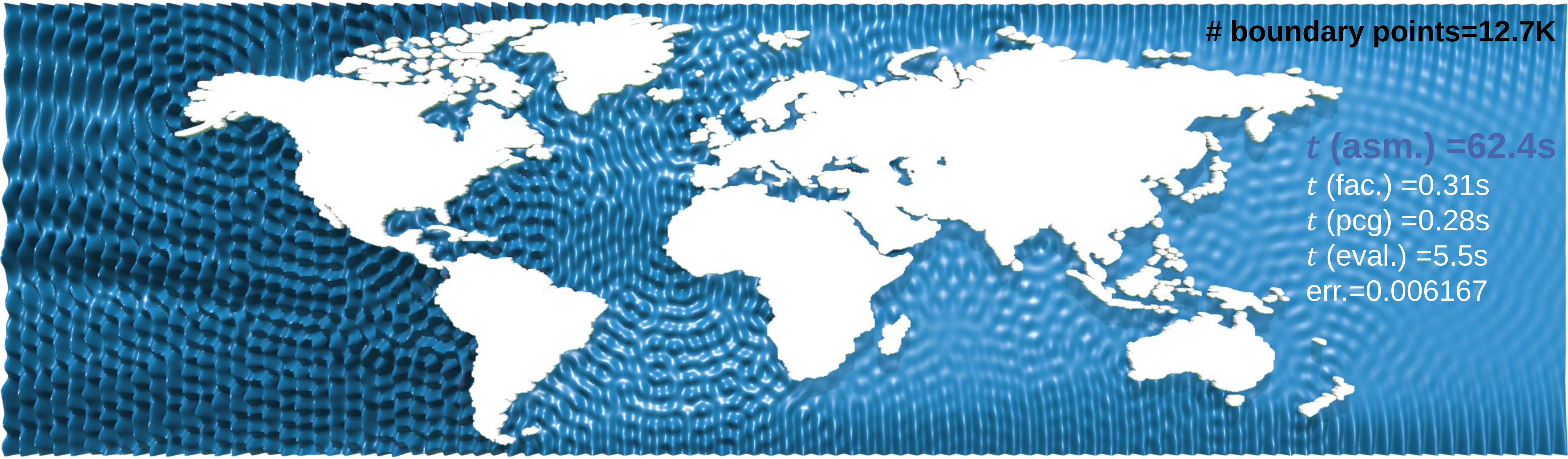
After only 9 iterations!

LINEAR ELASTICITY

Constrained Kelvinlet deformer [de Goes and James 2017]

HELMHOLTZ EQUATION

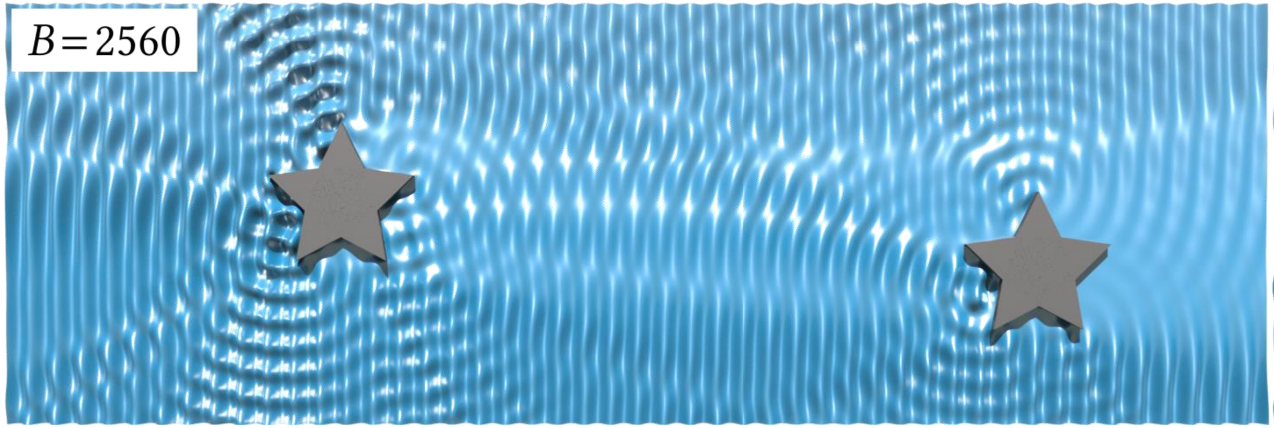
- Least-squares solves are always needed for Helmholtz equations
  - The BIE/MFS matrices are complex symmetric, but not Hermitian
  - Cholesky factorization does no exist

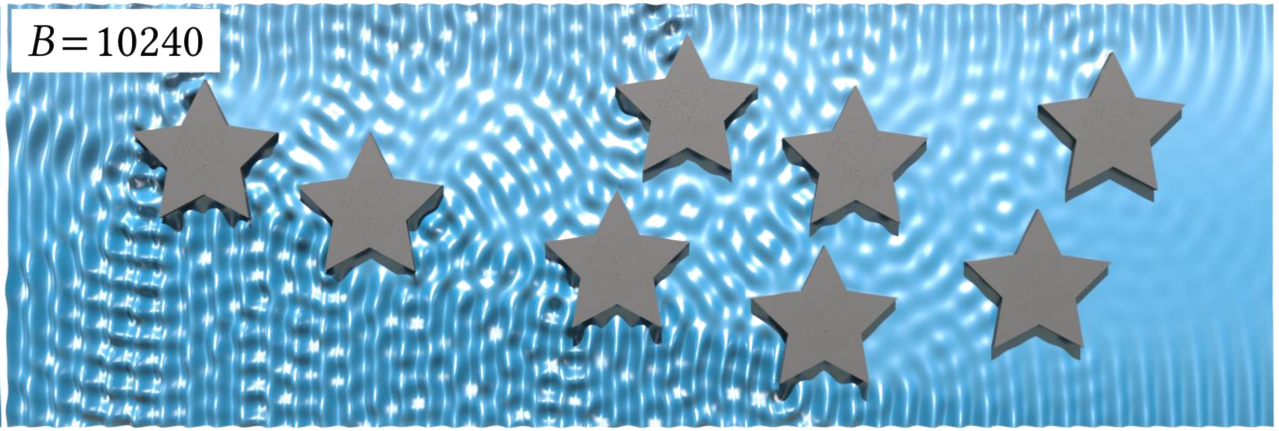


COMPARISON WITH SVD

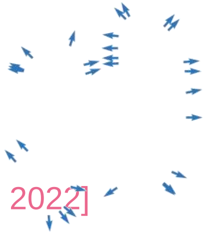
Boundary size

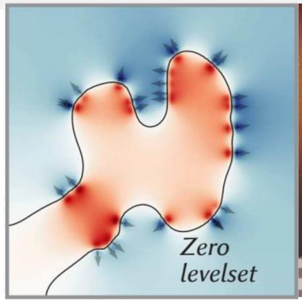
B	SVD		Ours					
	t(fac.)	t(slv.)	t(precomp.)	t(comp.)	#iters	t(pcg)	t(total)	Err
1280	4.864	0.003	0.004	0.419	15	0.005	0.427	0.000706
2560	33.757	0.011	0.007	0.715	15	0.013	0.735	0.000679
5120	261.454	0.045	0.013	1.270	15	0.048	1.331	0.004405
7680	911.212	0.156	0.023	3.478	15	0.099	3.600	0.003497
10240	2405.59	0.303	0.032	7.170	15	0.167	7.369	0.003665

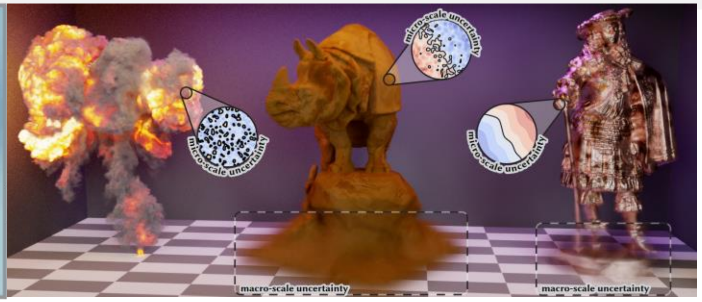




BEM FROM GAUSSIAN PROCESS VIEWPOINT







- Formulate stochasticity in Computer Graphics
  - Geometry processing, e.g., surface reconstruction [Sellán and Jacobson 2022]
  - Rendering, e.g., light transport [Seyb et al. 2024]
- Boundary value problems from a statistical point of view
  - Investigate the distribution of all possible solutions, not just a single one!
- Gaussian-process based inference *v.s.* MFS
  - Beyond conditional mean

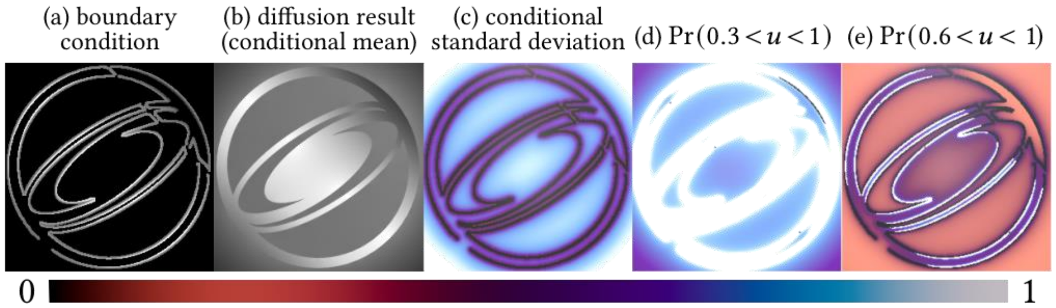
$$\mu\left(f(\boldsymbol{x}) \mid \boldsymbol{y}, f(\boldsymbol{y})\right) = \boldsymbol{K}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{K}(\boldsymbol{y},\boldsymbol{y})^{-1} f(\boldsymbol{y}),$$

  - Conditional variance for uncertainty quantification

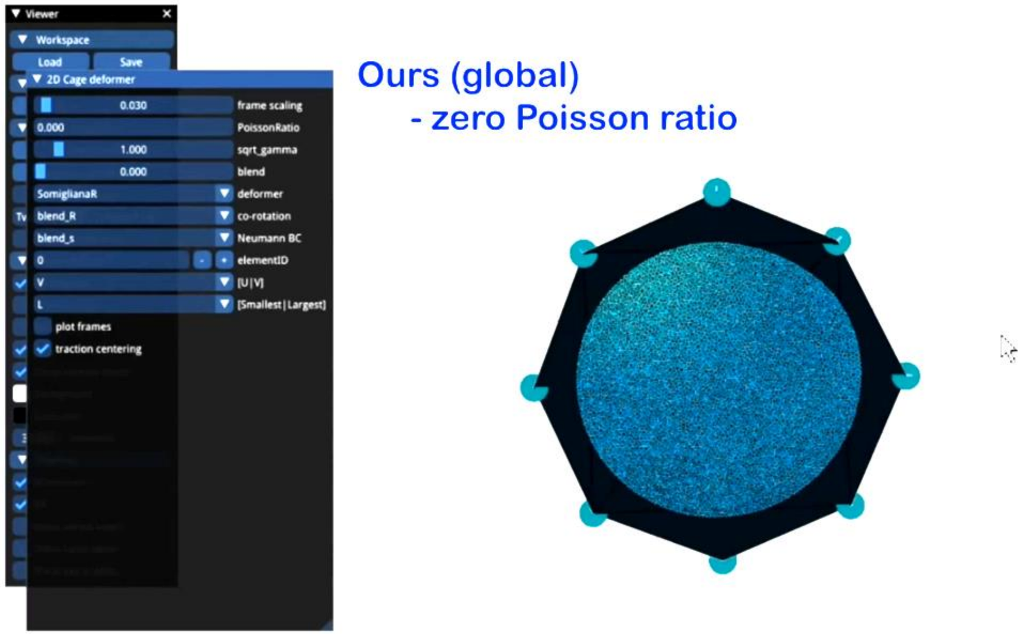
$$\sigma_{\boldsymbol{y}_i}^2 = \boldsymbol{K}(\boldsymbol{y}_i,\boldsymbol{y}_i) - \boldsymbol{K}(\boldsymbol{y}_i,\boldsymbol{x})\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x})^{-1}\boldsymbol{K}(\boldsymbol{x},\boldsymbol{y}_i).$$

  - Tell the probability of the solution falling within a given range

Gaussian Process	MFS
Kernel function	Green's function
Observation	Boundary condition
Conditional mean	Solution
Prediction	Evaluation



Uncertainty quantification of BIE solves

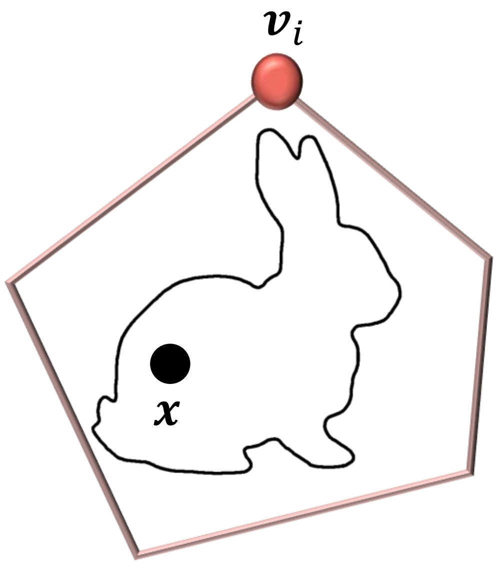


SOMIGLIANA COORDINATES

CONTROL VOLUMETRIC DEFORMATION USING A CAGE

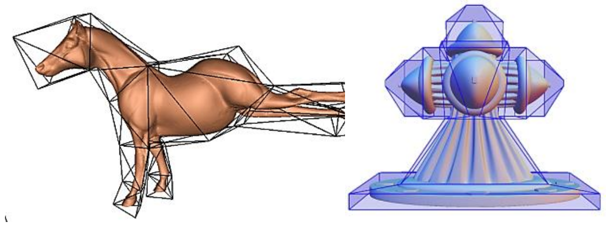
CAGE DEFORMER

- Cage deformer
  - based on generalized barycentric coordinates
    - many options available now (see our survey [Ströter et al. 2024])
  - **Boundary-aware, meshfree, extremely fast**
  - **Purely geometric, no elastic feel or volume control**

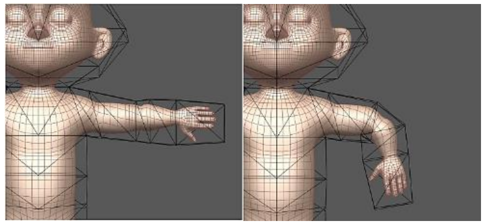


$$x = \sum_i \phi_i(x) v_i$$

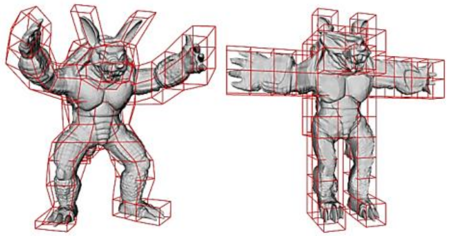
$$\tilde{x}(x) = \sum_i \phi_i(x) \tilde{v}_i$$



Mean-value coords [Floater 2003; Ju et al. 2005; Thiery et al. 2018]



Harmonic coords [Joshi et al. 2007]

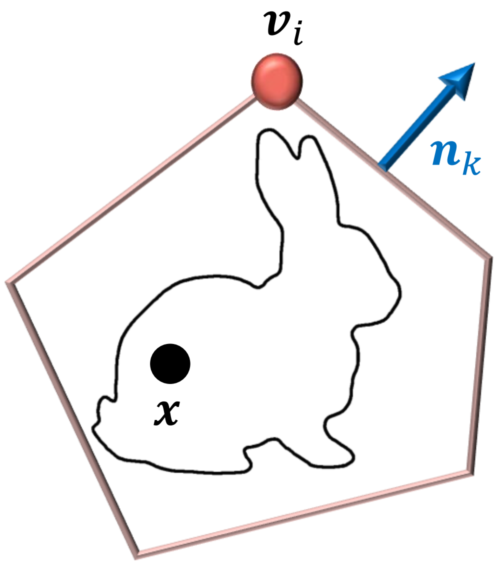


Maximum entropy coords [Hormann and Sukumar 2008]

Complex coords [Weber et al. 2009]

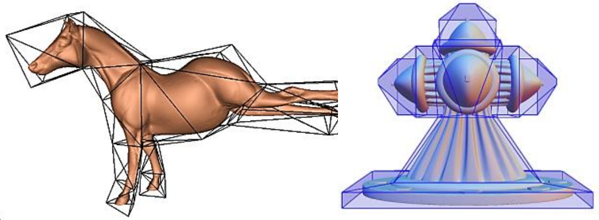
CAGE DEFORMER

- Cage deformer
  - based on generalized barycentric coordinates
    - many options available now (see our survey [Ströter et al. 2024])
  - **Boundary-aware, meshfree, extremely fast**
  - **Purely geometric, no elastic feel or volume control**

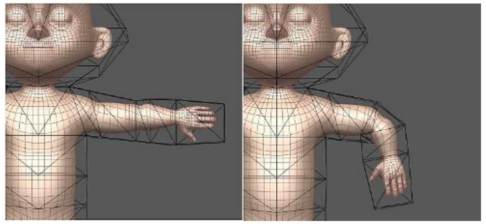


$$\boldsymbol{x} = \sum_i \phi_i(\boldsymbol{x})\boldsymbol{v}_i + \sum_k \psi_k(\boldsymbol{x})\boldsymbol{n}_k$$

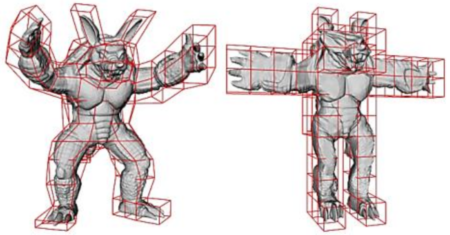
$$\tilde{\boldsymbol{x}}(\boldsymbol{x}) = \sum_i \phi_i(\boldsymbol{x})\tilde{\boldsymbol{v}}_i + \sum_k \psi_k(\boldsymbol{x})\,(c_k\tilde{\boldsymbol{n}}_k)$$



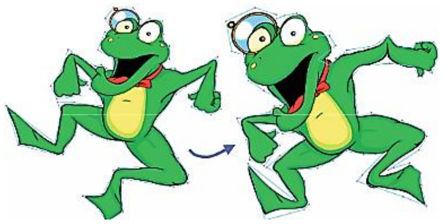
Mean-value coords [Floater 2003; Ju et al. 2005; Thiery et al. 2018]



Harmonic coords [Joshi et al. 2007]

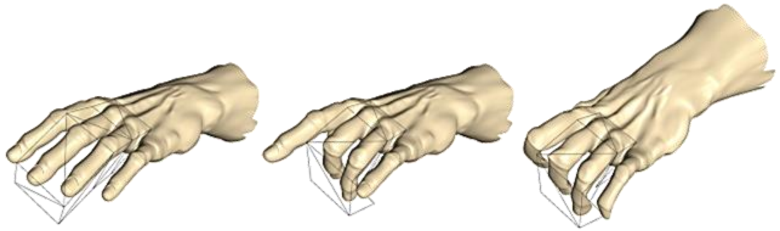


Maximum entropy coords [Hormann and Sukumar 2008]



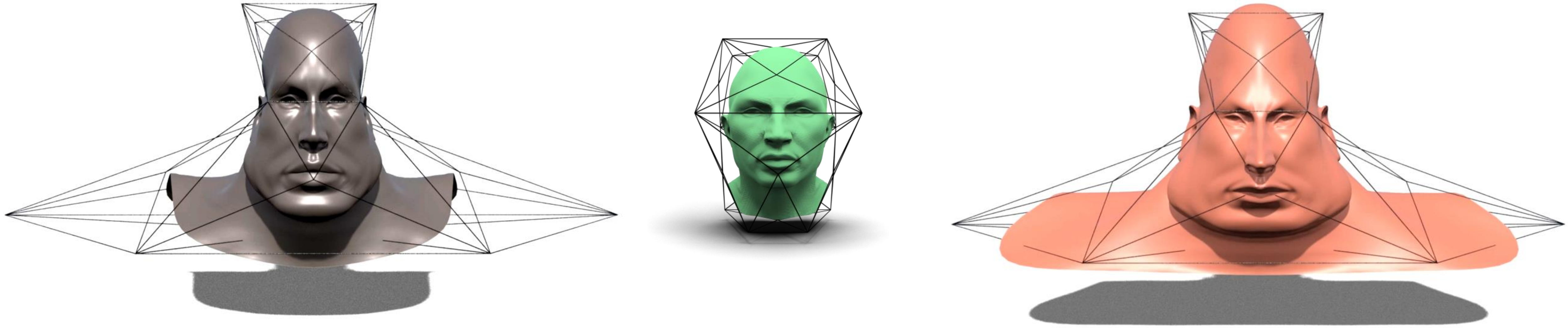
Complex coords [Weber et al. 2009]

+ Green coordinates [Lipman et al. 2008]



CONTRIBUTIONS

- Inject *elasticity* into cage deformers for fast volumetric deformation
  - **Matrix-valued coordinates**, extending Green coordinates
  - Derived from **linear elasticity** and mimicking elastic behaviors
  - **Invariant** under similarity transformations through **corotational** formulation
  - Control over **volume change** and **local bulge**

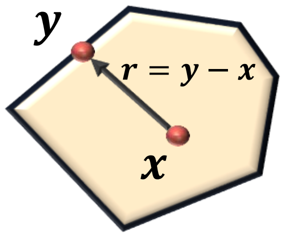


Green coordinates (GC)

PDE: $$\Delta \boldsymbol{u} = \boldsymbol{0}$$

Fundamental solutions:

$$G(\boldsymbol{y},\boldsymbol{x}) = \begin{cases} -\dfrac{1}{4\pi r}, & d = 3, \\ \dfrac{1}{2\pi}\log(r), & d = 2. \end{cases}$$



Boundary reformulation:

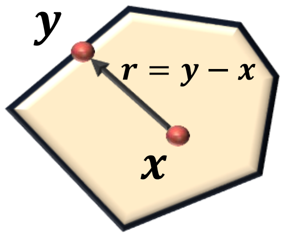
$$\boldsymbol{u}(\boldsymbol{x}) = \int_{\partial\Omega} \left[\boxed{\nabla_{\boldsymbol{n}} G(\boldsymbol{y},\boldsymbol{x})}\,\boldsymbol{u}(\boldsymbol{y}) - G(\boldsymbol{y},\boldsymbol{x})\boxed{\nabla_{\boldsymbol{n}}\boldsymbol{u}(\boldsymbol{y})}\right] \mathrm{d}\sigma_y$$

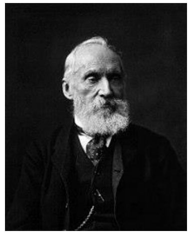
$$\boxed{\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{x}}$$

$$\boxed{\{\phi_i(\boldsymbol{x}), \psi_k(\boldsymbol{x})\} \in \mathbf{R}}$$

FROM GREEN TO SOMIGLIANA

	Green coordinates (GC)	Somigliana coordinates (SC)
PDE:	$\Delta \boldsymbol{u} = \boldsymbol{0}$	$\Delta \boldsymbol{u} + \frac{1}{1-2\nu}\nabla(\nabla \cdot \boldsymbol{u}) = \boldsymbol{0}$
Fundamental solutions:	$G(\boldsymbol{y},\boldsymbol{x}) = \begin{cases} -\frac{1}{4\pi r}, & d=3, \\ \frac{1}{2\pi}\log(r), & d=2. \end{cases}$	$\mathcal{K}(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \frac{a-b}{r}\boldsymbol{I} + \frac{b}{r^3}\boldsymbol{r}\boldsymbol{r}^t, & d=3, \\ (b-a)\log(r)\boldsymbol{I} + \frac{b}{r^2}\boldsymbol{r}\boldsymbol{r}^t, & d=2. \end{cases}$
Boundary reformulation:	$\boldsymbol{u}(\boldsymbol{x}) = \int_{\partial\Omega} [\boxed{\nabla_n G(\boldsymbol{y},\boldsymbol{x})}\boldsymbol{u}(\boldsymbol{y}) - G(\boldsymbol{y},\boldsymbol{x})\boxed{\nabla_n \boldsymbol{u}(\boldsymbol{y})}]\,\mathrm{d}\sigma_y$	$\boldsymbol{u}(\boldsymbol{x}) = \int_{\partial\Omega} [\boxed{\mathcal{T}(\boldsymbol{y},\boldsymbol{x})}\boldsymbol{u}(\boldsymbol{y}) + \mathcal{K}(\boldsymbol{y},\boldsymbol{x})\boxed{\boldsymbol{\tau}(\boldsymbol{y})}]\,\mathrm{d}\sigma_y$





Lord Kelvin

Carlo Somigliana

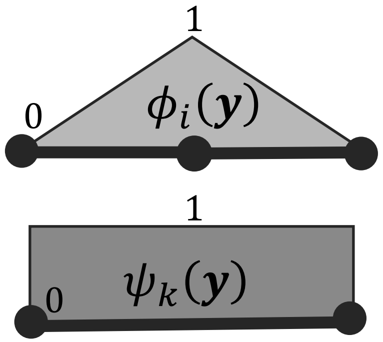
$$\boxed{\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{x}}$$

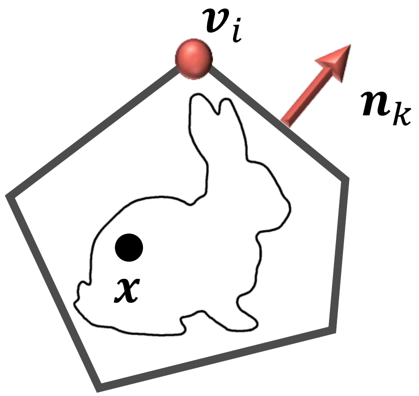
$$\boxed{\{T_i(\boldsymbol{x}), K_k(\boldsymbol{x})\} \in \mathbf{R}^{d\times d}}$$

SOMIGLIANA COORDINATES

Compute SC w.r.t. a triangulated cage

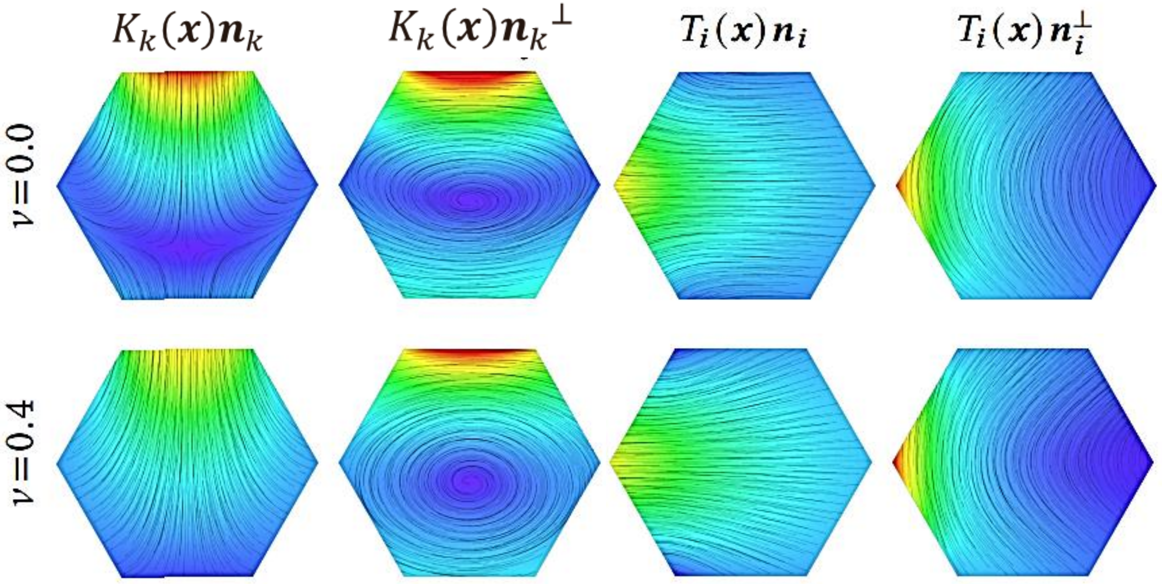
$$\begin{cases} T_i(\boldsymbol{x}) = \int_{\partial\Omega} \mathcal{T}(\boldsymbol{y},\boldsymbol{x})\phi_i(\boldsymbol{y})\mathrm{d}\sigma_{\boldsymbol{y}}, \\ K_k(\boldsymbol{x}) = \int_{\partial\Omega} \mathcal{K}(\boldsymbol{y},\boldsymbol{x})\psi_k(\boldsymbol{y})\mathrm{d}\sigma_{\boldsymbol{y}}. \end{cases}$$





$$\boldsymbol{x} = \sum_i T_i(\boldsymbol{x})\boldsymbol{v}_i + \sum_k K_k(\boldsymbol{x})(c\boldsymbol{n}_k)$$

T_i and K_k as matrix functions of Poisson ratio ν



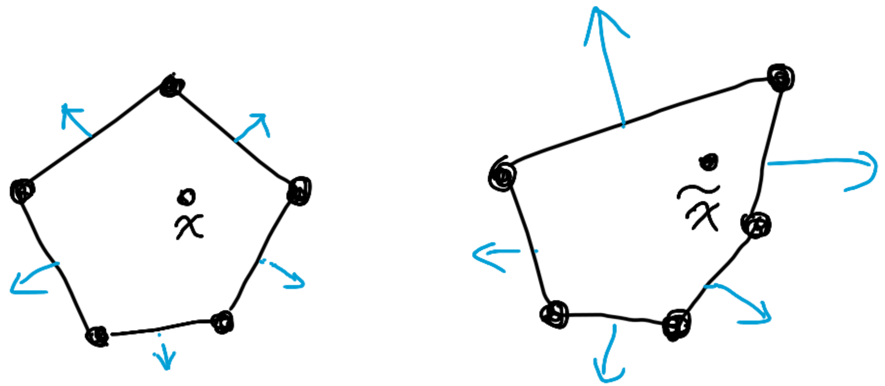
THE CONNECTION TO BEM

- When the cage is deformed, i.e., with new specified vertex positions

$$\tilde{\boldsymbol{x}}(\boldsymbol{x}) = \sum_i T_i(\boldsymbol{x})\tilde{\boldsymbol{v}}_i + \sum_k K_k(\boldsymbol{x})\,(c_k\tilde{\boldsymbol{n}}_k)$$

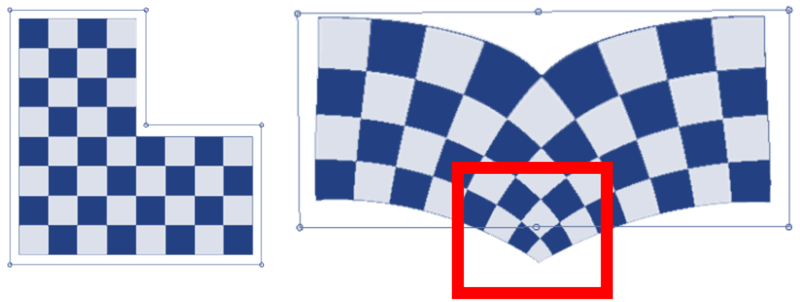
Should be $\partial_{\boldsymbol{n}}\tilde{\boldsymbol{x}}$ in BEM

- In BEM, $\partial_{\boldsymbol{n}}\tilde{\boldsymbol{x}}|_{\partial\Omega}$ is solved from $\tilde{\boldsymbol{x}}|_{\partial\Omega}$
  - Dirichlet and Neumann boundary conditions are compatible
- In cage deformation, we “guess” the boundary normal derivatives
  - Efficient for real-time manipulation
  - Parameterize Neumann conditions to support flexible control over the interior deformation
- Price to pay: normal terms are not compatible with cage vertex positions
  - The interior deformation could be not intuitive, e.g., not following the cage tightly



undeformed

deformed

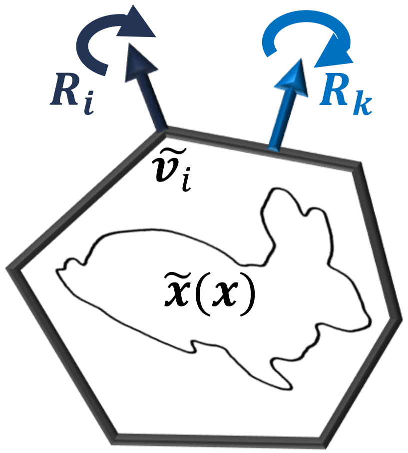


COROTATIONAL FORMULATION

$T_i(\boldsymbol{x})$ and $K_k(\boldsymbol{x})$ are not rotationally invariant

- Typical remedy: corotational formulation

$$\tilde{\boldsymbol{x}}(\boldsymbol{x}) = \left(\sum_i R_i T_i(\boldsymbol{x}) R_i^t\right)^{-1} \left[\sum_i R_i T_i(\boldsymbol{x}) R_i^t \tilde{\boldsymbol{v}}_i + \sum_k R_k K_k(\boldsymbol{x}) R_k^t \tilde{\boldsymbol{\tau}}_k\right]$$

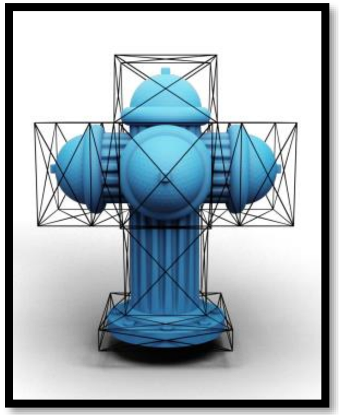


$$\tilde{\boldsymbol{\tau}}_k = s_k R_k \boldsymbol{n}_k$$

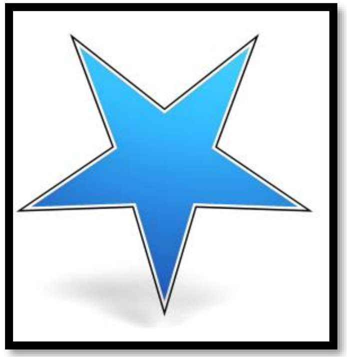
$$= \left[\frac{2(1-\nu)}{1-2\nu}\eta_k + \frac{2\nu(d-1)}{1-2\nu}\lambda_k\right] R_k \boldsymbol{n}_k$$

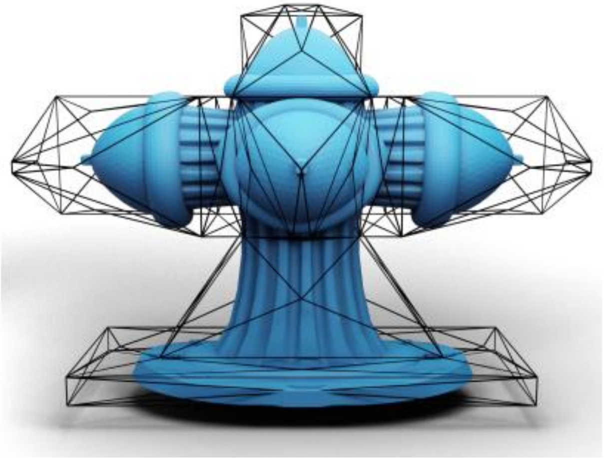
Estimate $\{R_k, \lambda_k, \eta_k\}$ for each boundary facet

GLOBAL VS. LOCAL ROTATION & TANGENT STRETCHES



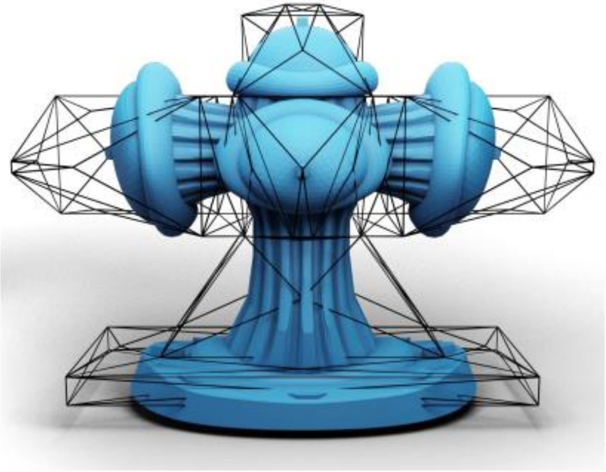
Rest pose





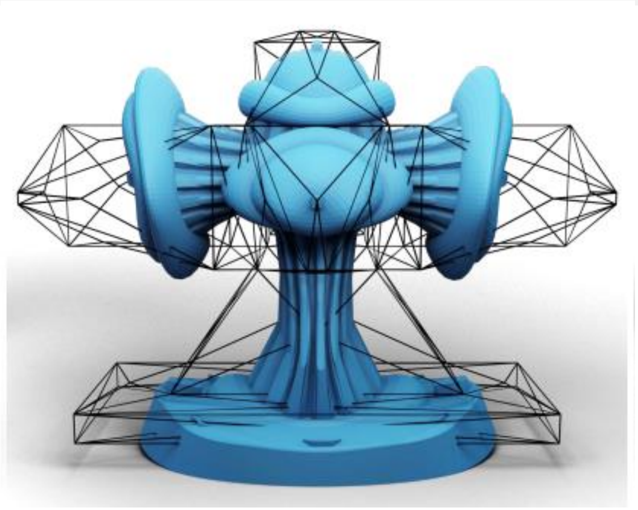
Global variant

$R_k = R_{global},\ \lambda_k = \lambda_{global}$ from the optimal similarity transformation



In between

blend global and local R_k, λ_k



Local variant

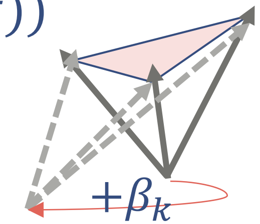
R_k and λ_k are decuded on per facet basis

CURVATURE-BASED NORMAL STRETCHES

- Normal stretching factor η_k for each cage facet
  - No information about out-of-plane deformation
  - E.g., account for curvature change for local bulging

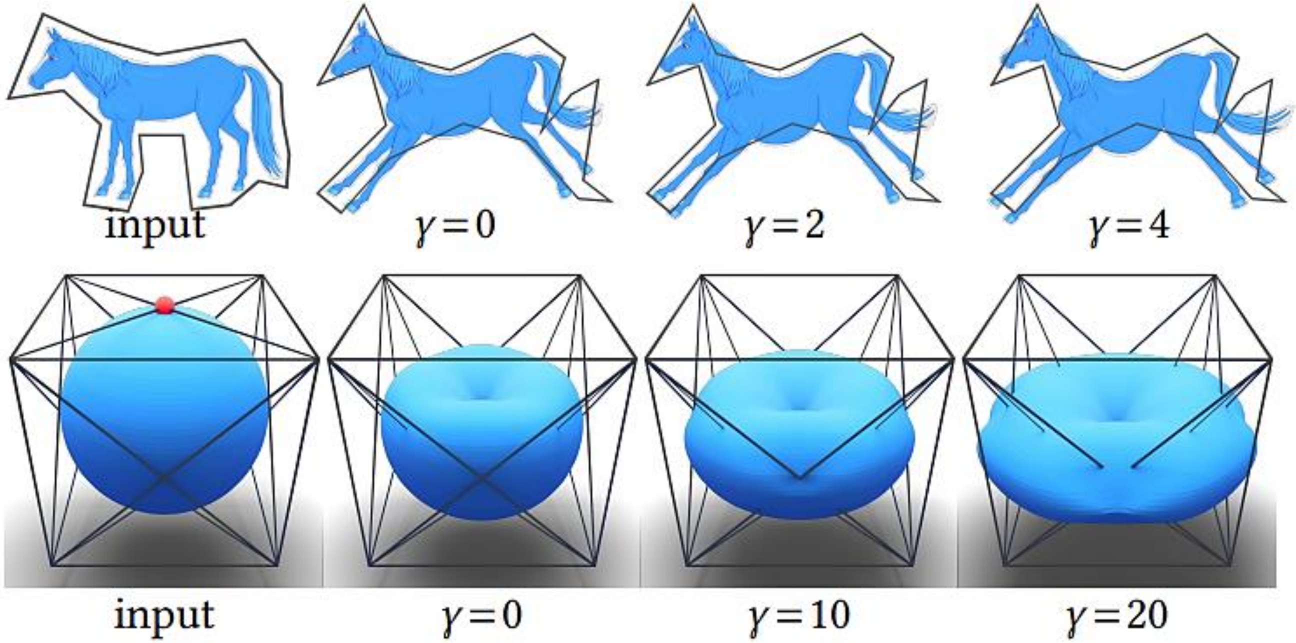
$$\eta_k = \lambda_k \exp(\gamma \beta_k / (2^{d-1}\pi))$$

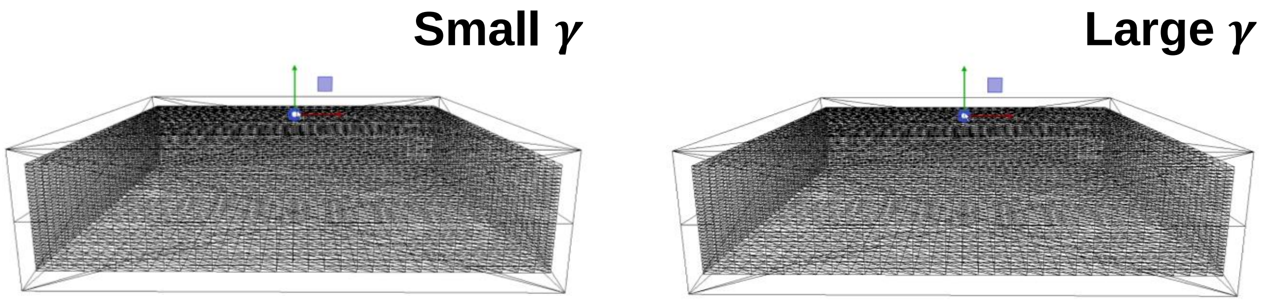
  - Compute on-the-fly



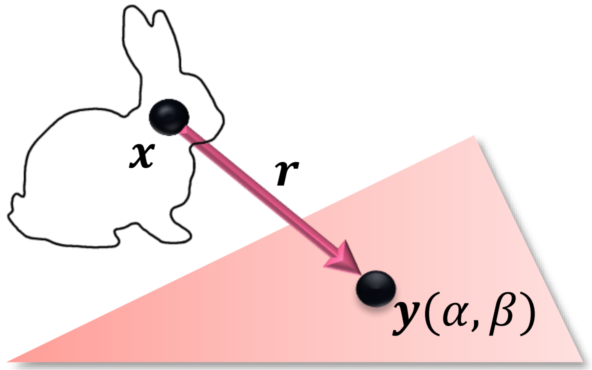
- Our choice of R_j, λ_j, η_j keeps the deformation invariant under similarity transformations

$$\tilde{\boldsymbol{x}}(sR\boldsymbol{x} + \boldsymbol{t}) = sR\tilde{\boldsymbol{x}}(\boldsymbol{x}) + \boldsymbol{t}$$





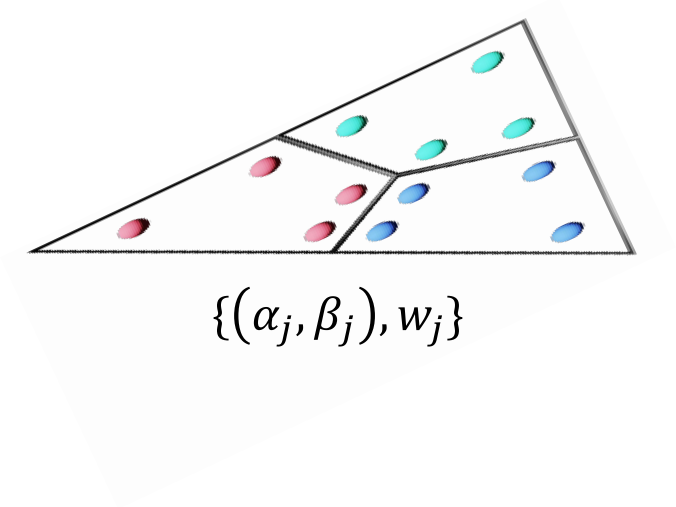
IMPLEMENTATION



$$K_k(\boldsymbol{x}) = \int_{\Delta_k} \mathcal{K}(\boldsymbol{y}, \boldsymbol{x}) \mathrm{d}\sigma_{\boldsymbol{y}} = 2|\Delta_k| \int_0^1 \mathrm{d}\alpha \int_0^{1-\alpha} \mathrm{d}\beta \, \mathcal{K}(\boldsymbol{y}(\alpha, \beta), \boldsymbol{x})$$

- #query points: **39k**
- #quadratures per face: **7500**
- #cage faces: **184**

$\approx$ Coord. computation time: **1.5s**



$$2|\Delta_k| \sum_j w_j \left(\frac{a-b}{r(\alpha_j, \beta_j)} \boldsymbol{I} + \frac{b}{r^3(\alpha_j, \beta_j)} \boldsymbol{r}(\alpha_j, \beta_j) \boldsymbol{r}^t(\alpha_j, \beta_j) \right)$$

ANALYTICAL EXPRESSIONS OF SOMIGLIANA COORDS.

- Biharmonic coordinates for 3D **triangular** cages [Thiery et al. 2024]

Biharmonic coordinates

$$f(\eta) = \sum_{i\in\mathcal{V}} \phi_i(\eta)\boldsymbol{a}_i + \sum_{j\in\mathcal{T}} \psi_j(\eta)\boldsymbol{b}_j + \sum_{i\in\mathcal{V}} \bar{\phi}_i(\eta)\boldsymbol{c}_i + \sum_{j\in\mathcal{T}} \bar{\psi}_j(\eta)\boldsymbol{d}_j,$$

Boundary integral formulation of biharmonic functions $\Delta^2 f = 0$

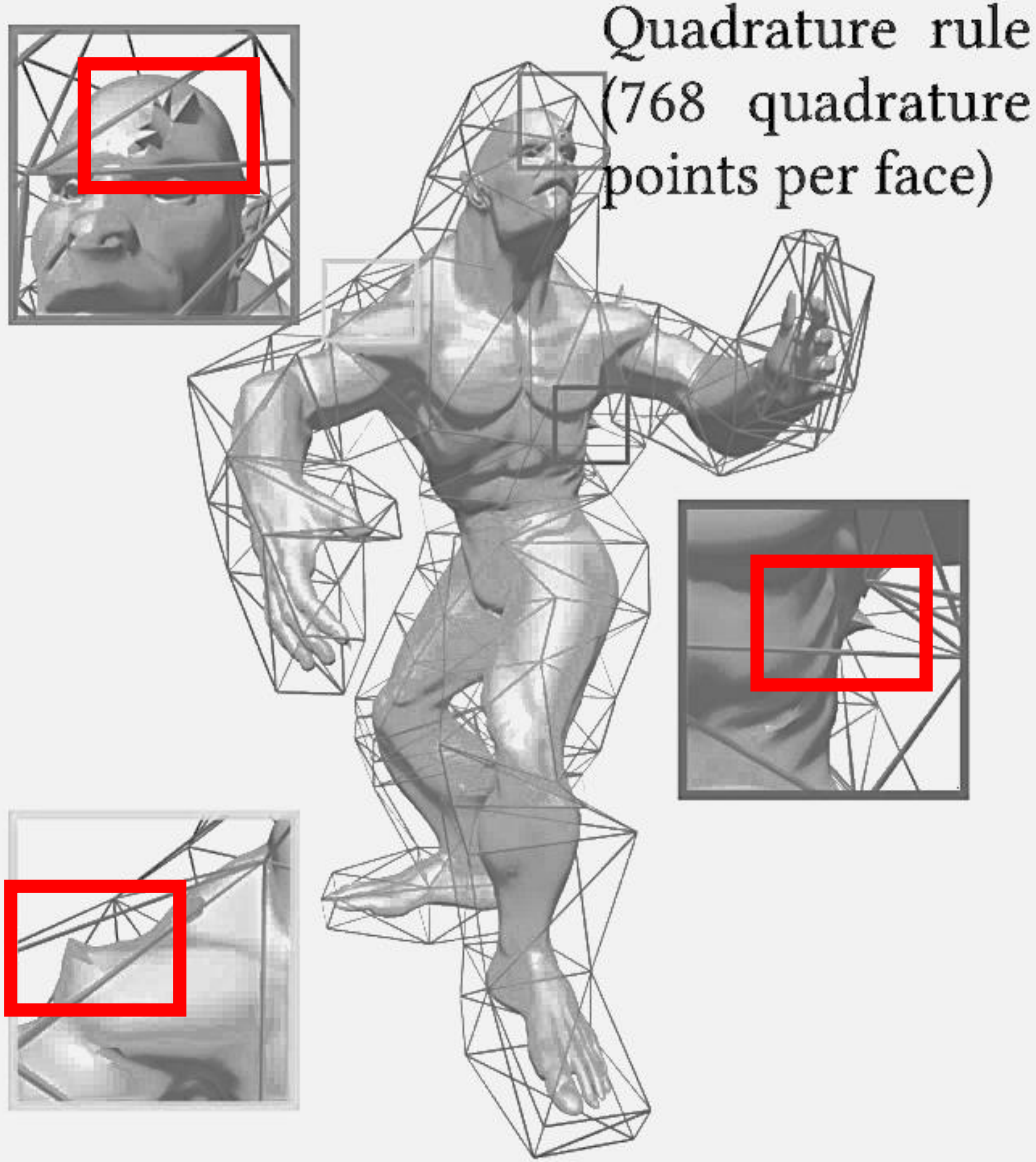
$$f(\eta) = \int_{\xi\in\partial\Omega} f(\xi)\frac{\partial_1 G_1}{\partial n}(\xi,\eta)d\xi - \int_{\xi\in\partial\Omega} G_1(\xi,\eta)\frac{\partial f}{\partial n}(\xi)d\xi$$
$$+ \int_{\xi\in\partial\Omega} \Delta f(\xi)\frac{\partial_1 G_2}{\partial n}(\xi,\eta)d\xi - \int_{\xi\in\partial\Omega} G_2(\xi,\eta)\frac{\partial \Delta f}{\partial n}(\xi)d\xi,$$

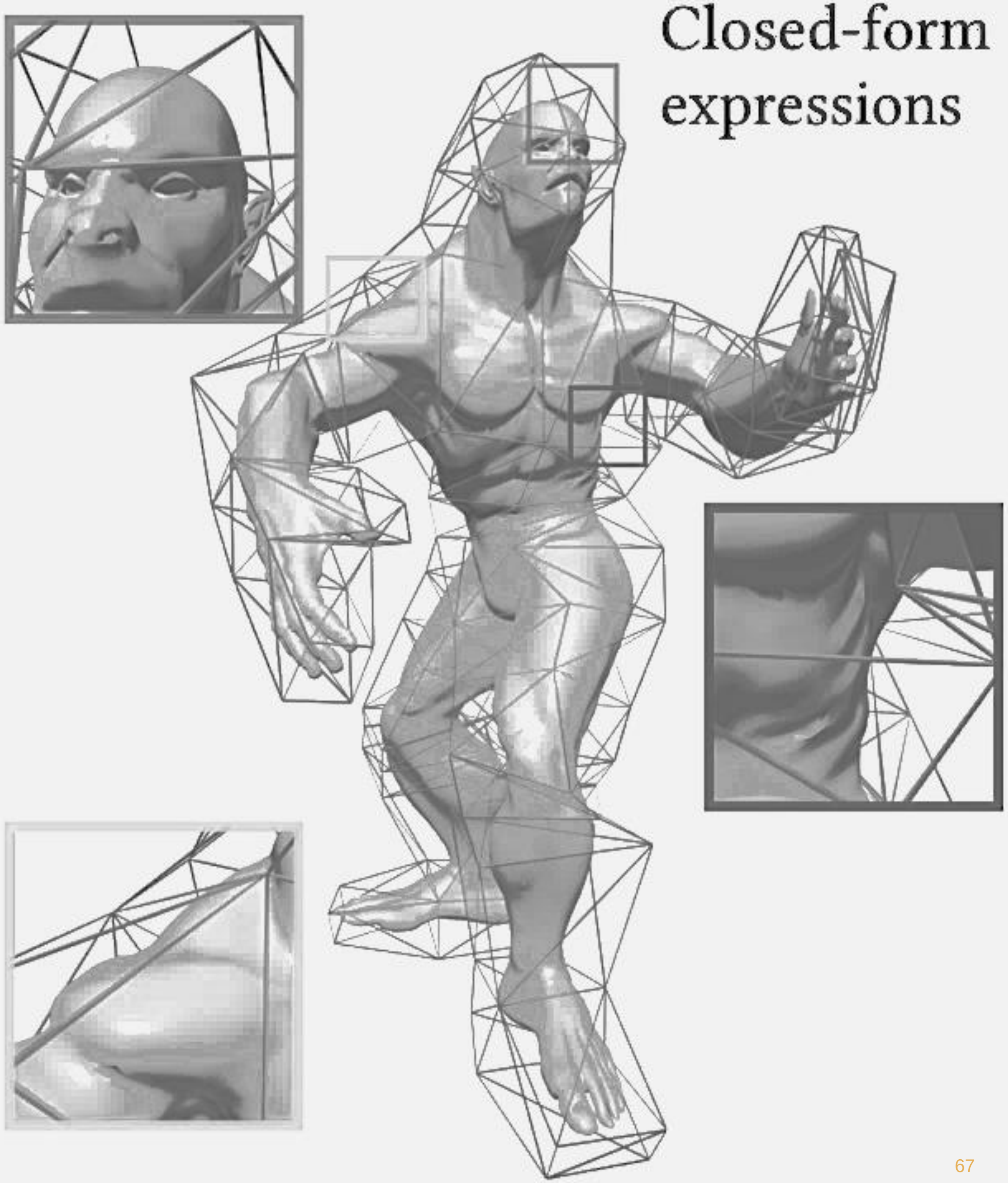
$$\int_{\xi\in t} \|\xi-\eta\| d\xi = \frac{d_t^3\omega_t(\eta)}{3}$$
$$+ \sum_{e\in t}\frac{a_e^t}{6}\left((2d_t^2 + D_e^2)\log\left(\frac{l_{e_1}-\zeta_{e_1}}{l_{e_0}-\zeta_{e_0}}\right) - l_{e_1}\zeta_{e_1} + l_{e_0}\zeta_{e_0}\right)$$

Expressing Somigliana coords. with biharmonic coords.

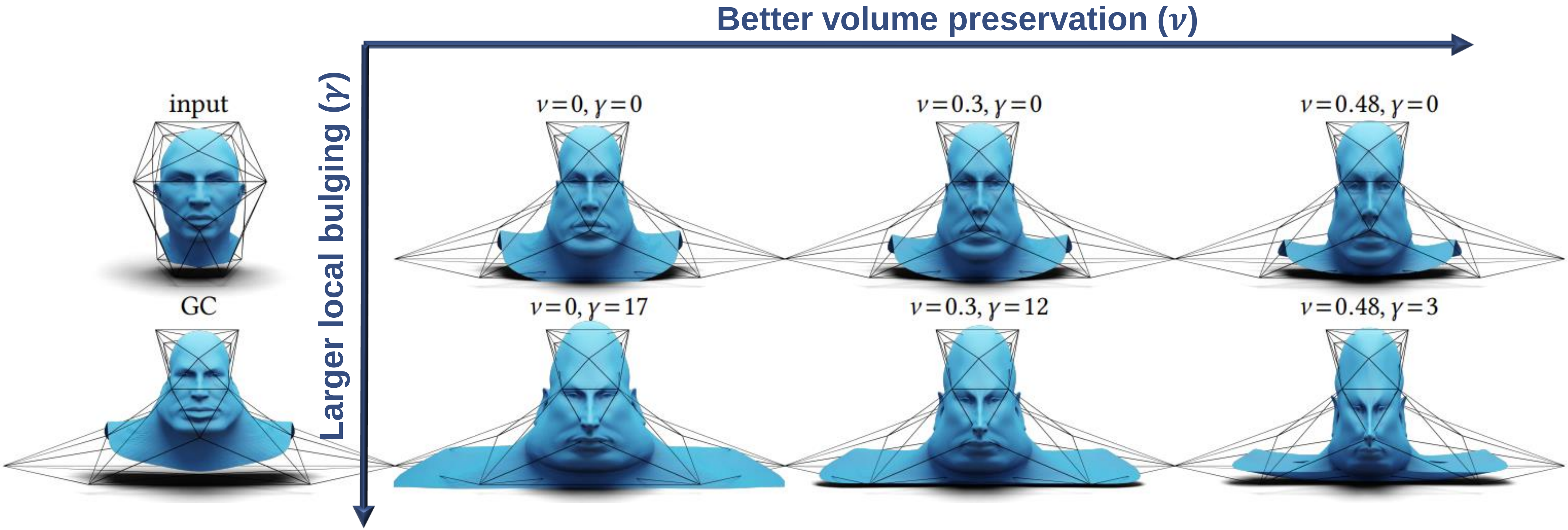
$$K_t(\eta) = \frac{b}{\lambda_2}\underbrace{H_\eta(\bar{\psi}_t)(\eta)}_{} - \frac{a}{\lambda_1}\psi_t(\eta)I_3$$

$$T_{ti}^t(\eta) = \frac{2bd_t}{\lambda_1}\underbrace{H_\eta\left(\psi_t^i\right)(\eta)}_{} - \frac{a}{\lambda_1}\phi_{ti}^t(\eta)I_3 - \frac{a-2b}{\lambda_1}\left[n_t\underbrace{\nabla_\eta^T(\psi_t^i)}_{} - \underbrace{\nabla_\eta(\psi_t^i)n_t^T}_{}\right].$$

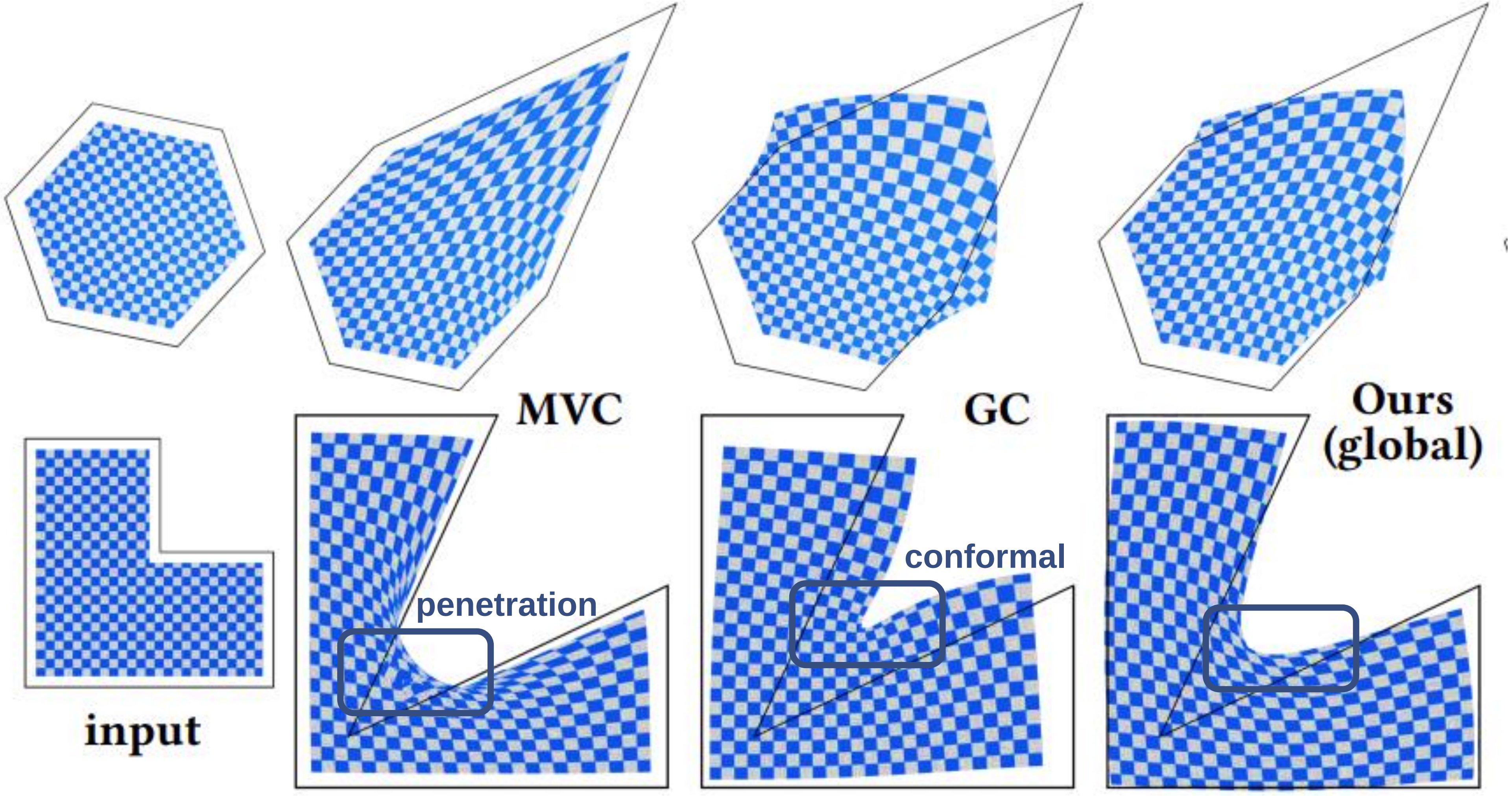

Quadrature rule (768 quadrature points per face)


Closed-form expressions

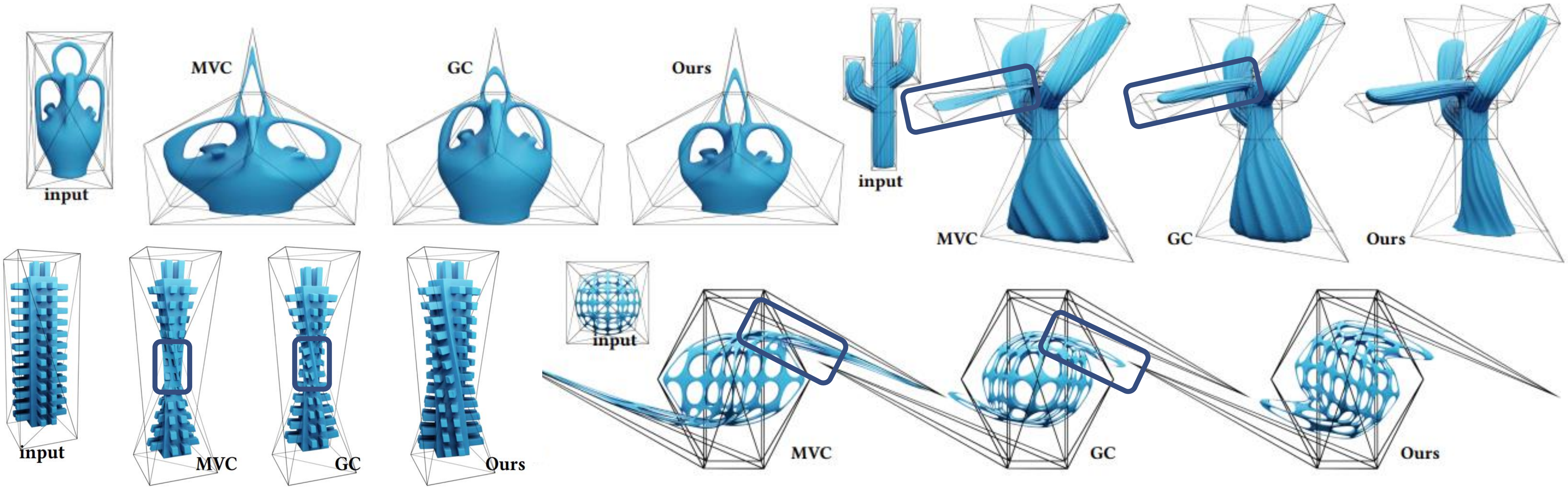
VOLUME PRESERVING VS. LOCAL BULGING



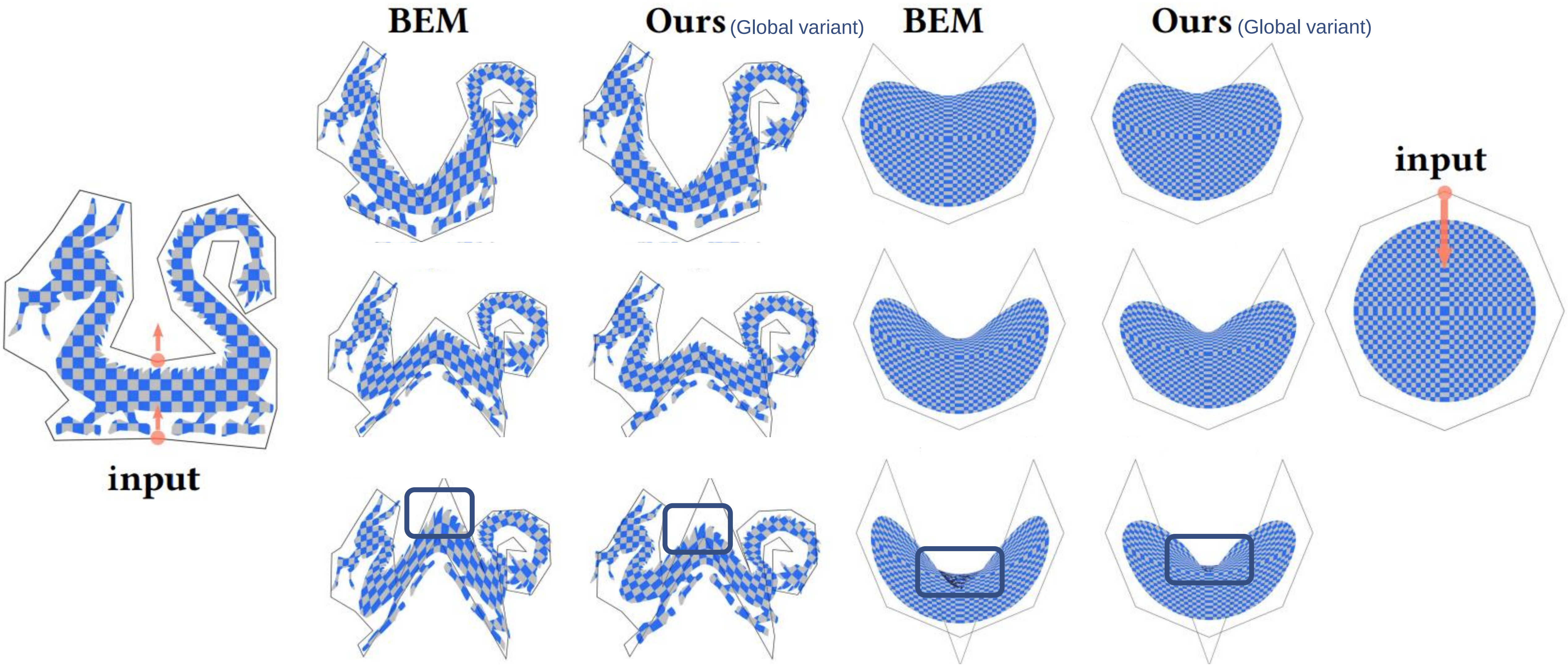
2D COMPARISONS



3D COMPARISONS



COMPARISONS

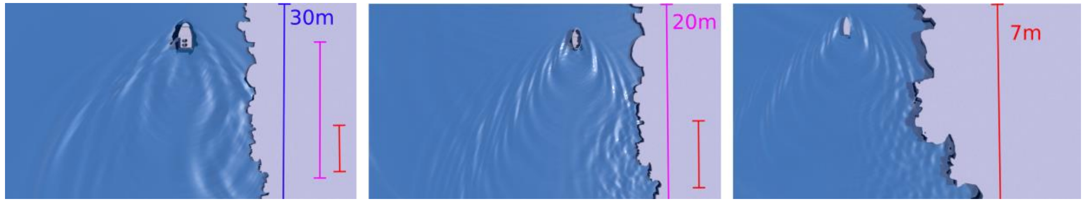


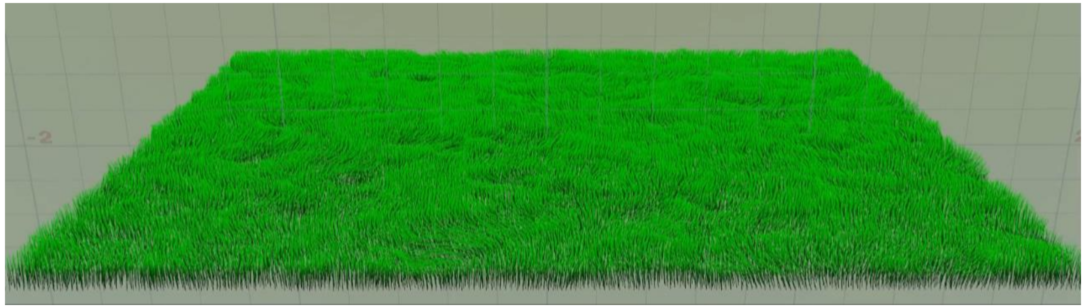
SUMMARY

- Extended Green’s functions
  - A systematical approach to represent and derive Green’s functions for general linear operators
  - Regularization for practical use
- Meshless solver for boundary integral equations
  - Massively parallel preconditioner based on inverse Cholesky factorization
  - Scalable to large scale problems
- Matrix-valued barycentric coordinates for cage deformation
  - Compromise between pointwise deformers and boundary element methods
  - Flexible control over the “fake” Neumann boundary conditions

FUTURE WORKS

- Generalize the idea to …
  - Asymmetric systems from elliptic PDEs or non-elliptic PDEs without least-squares solves, e.g., wave equations [Schreck et al. 2019]
  - Nonlinear problems, e.g., Gaussian process hydrodynamics [Owhadi 2023]
- The art of preconditioning
  - Dilemma between convergence and cost per iteration
  - Analogue of the uncertainty principle of Fourier transform
- Simulation meets stochasticity
  - Make use of uncertainty quantification for adaptive simulation
  - Develop stochastic representation to account for the uncertainty of a complex dynamical system






THANKS