SAND2022-7088C

GBCs and the conjugate basis for both element-based and element-free solution of PDEs

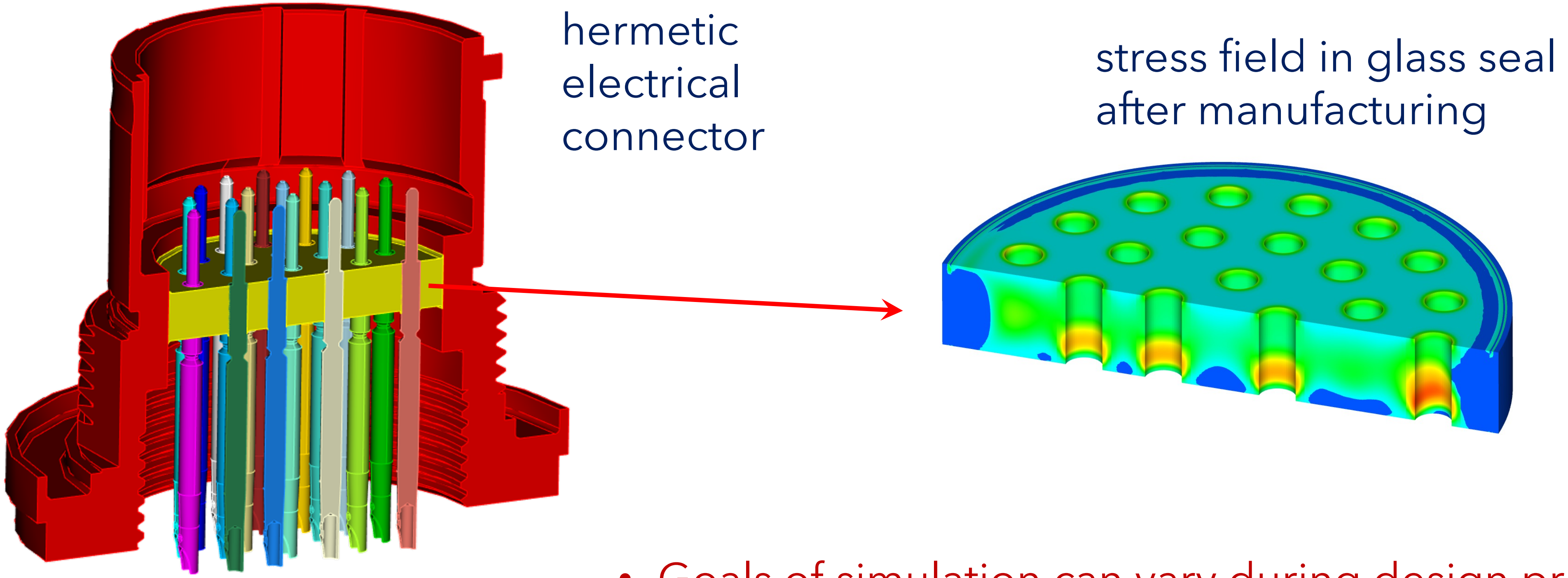
Joe Bishop

Engineering Sciences Center
Sandia National Laboratories
Albuquerque, NM

SAND2022-???? C

Workshop on Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics, June 1-4, 2022, Ascona, Switzerland

Motivation: Agile simulation of complex assemblies

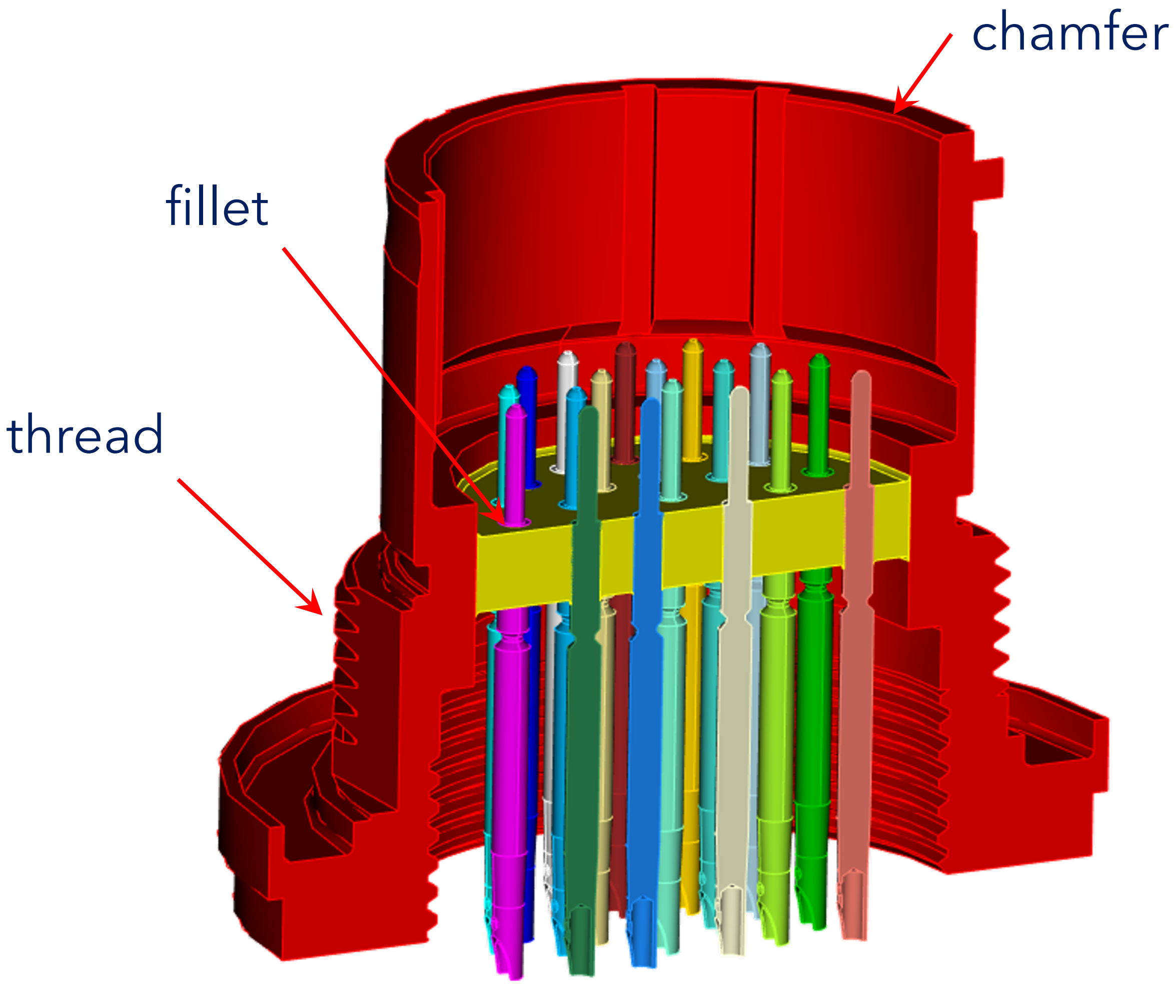


hermetic electrical connector

stress field in glass seal after manufacturing

- Goals of simulation can vary during design process.
- Heuristics are often used to defeature geometry.
- Heuristics are used to construct finite element mesh.

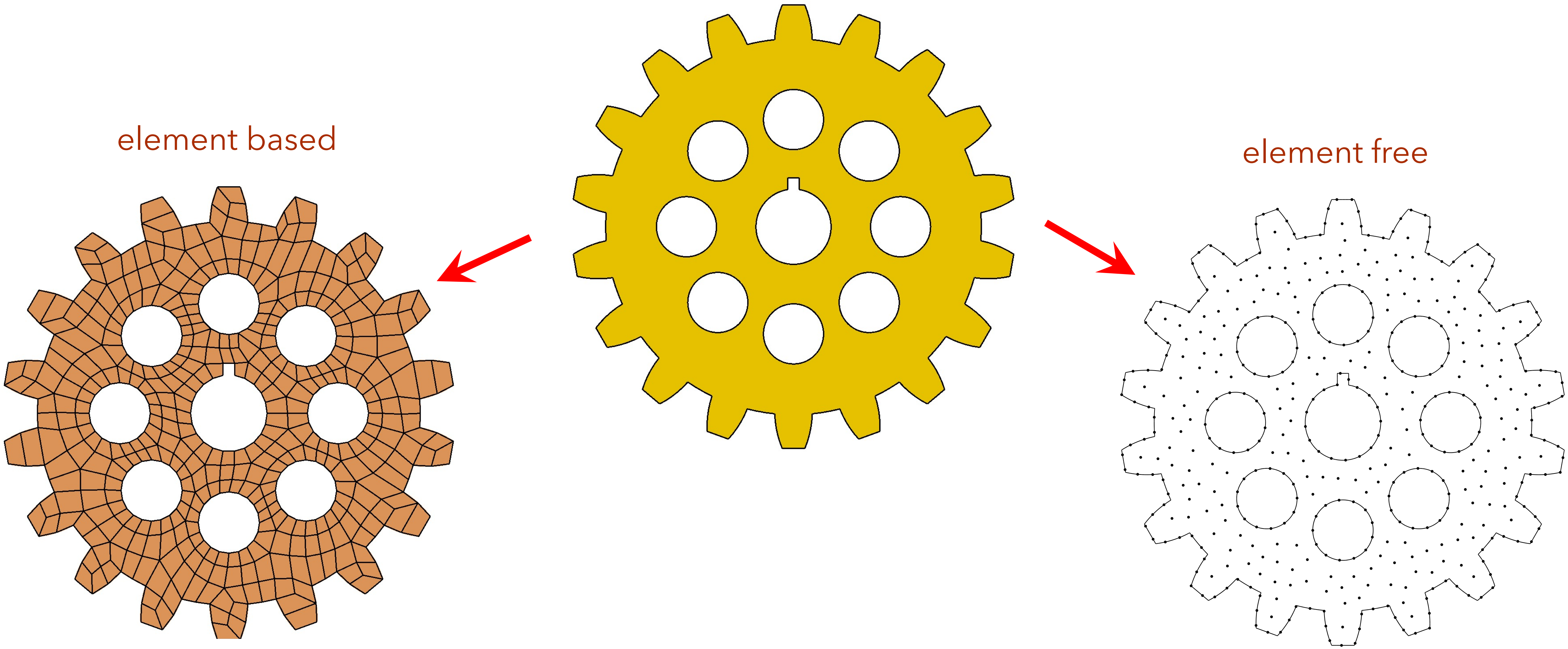
Motivation: Typical domain (geometry) features



In engineering applications, domains typically contain numerous geometric features that are unimportant for the goals of the simulation.

Motivation

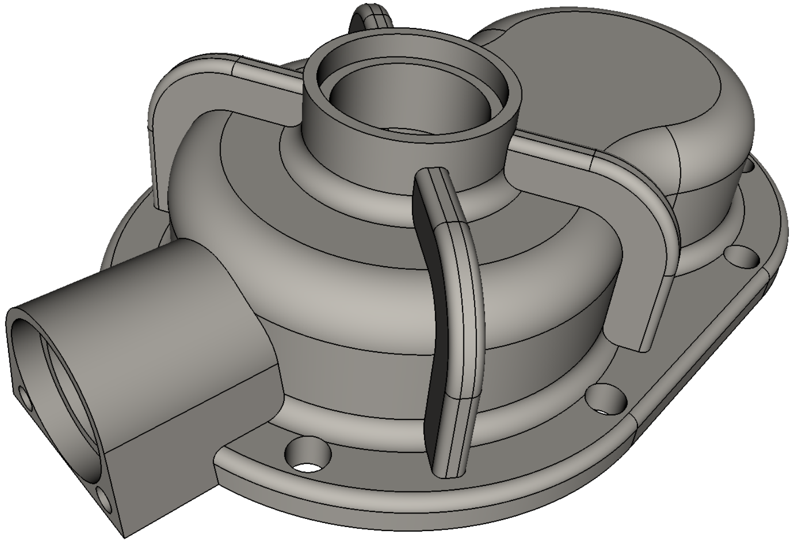
- GBCs provide framework for broad class of discretizations, both element based and element free.
- Can unify both classes of discretizations.

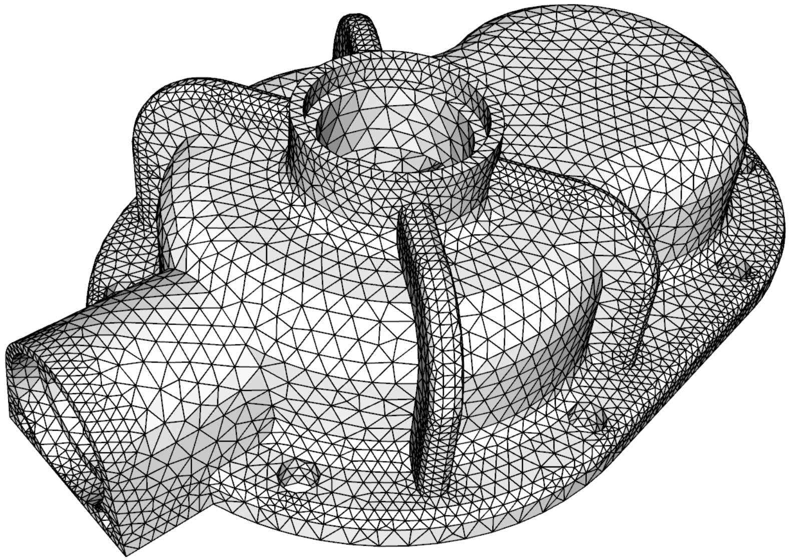


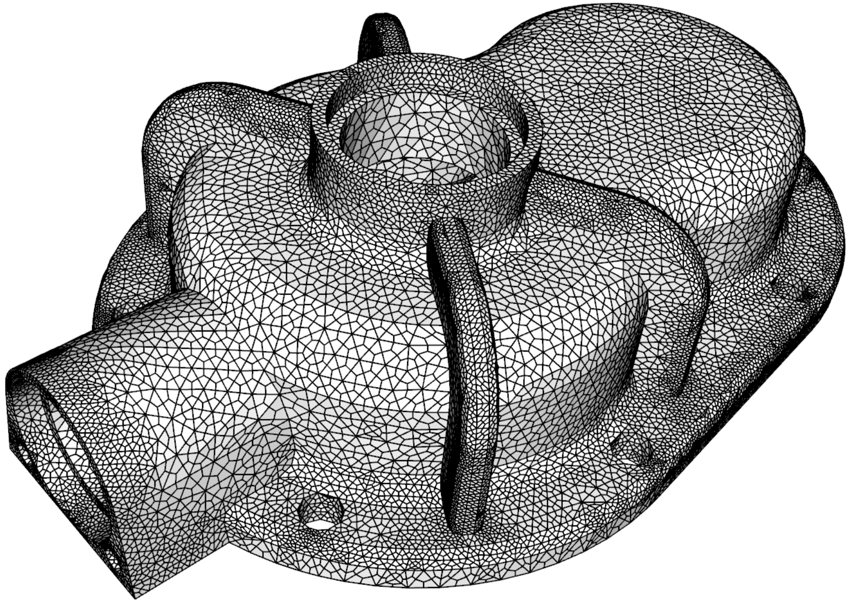
element based

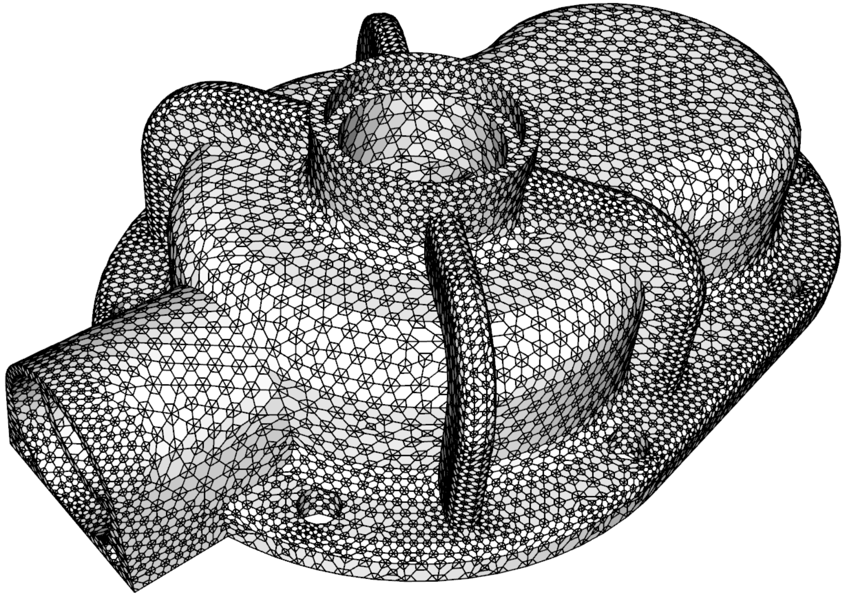
element free

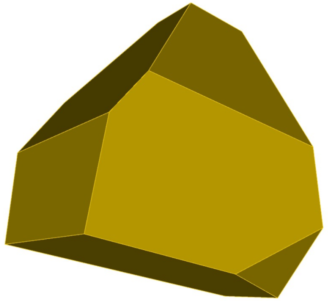
Motivation: element-based polyhedral discretizations

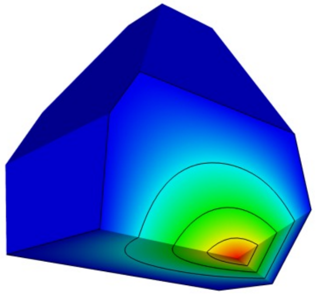


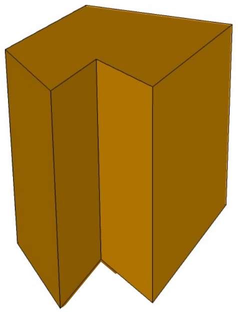


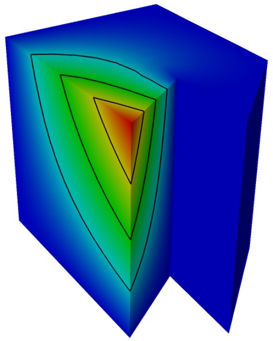




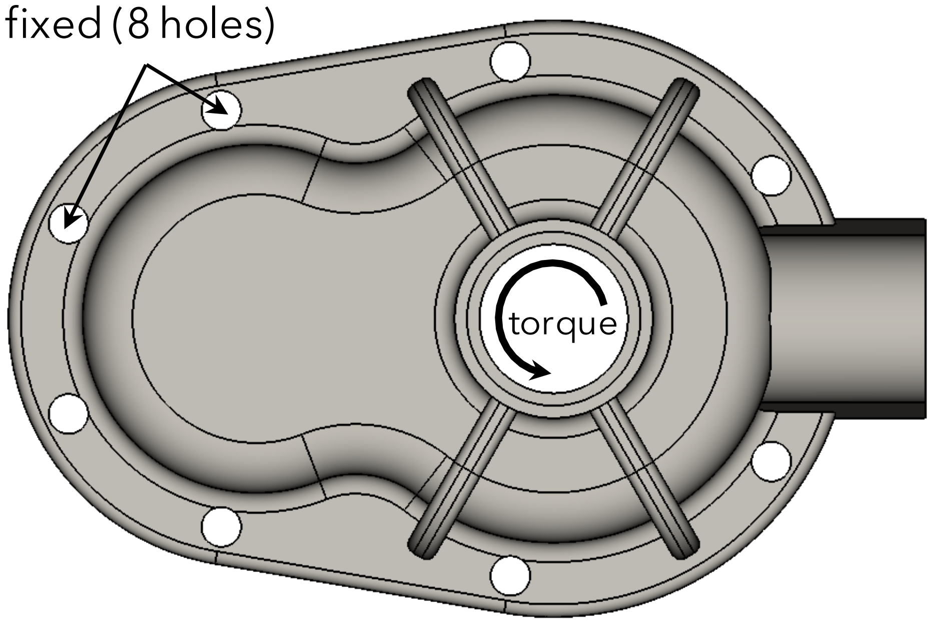




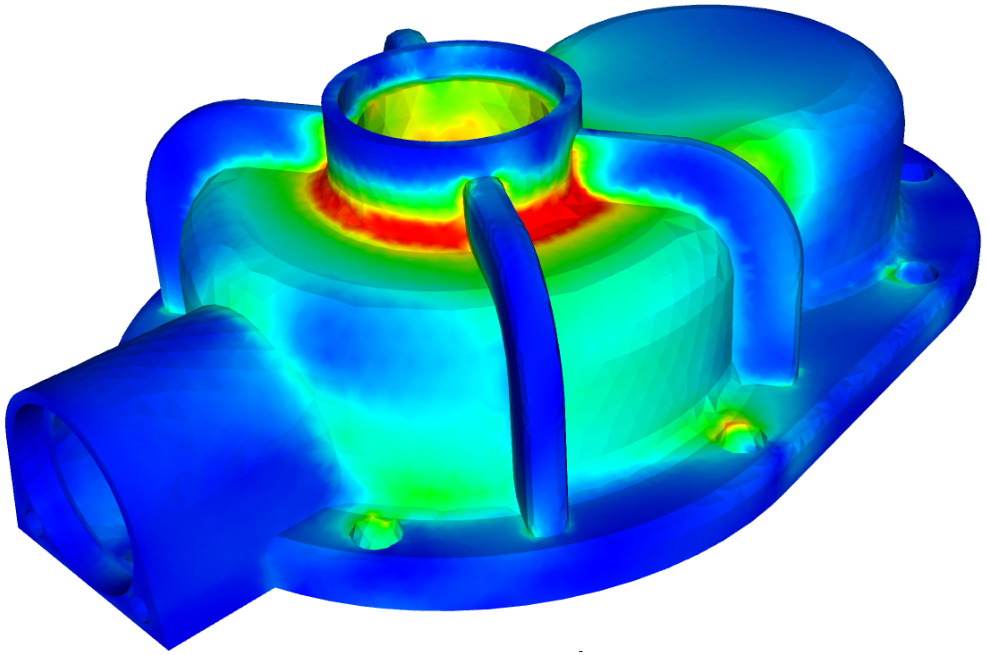




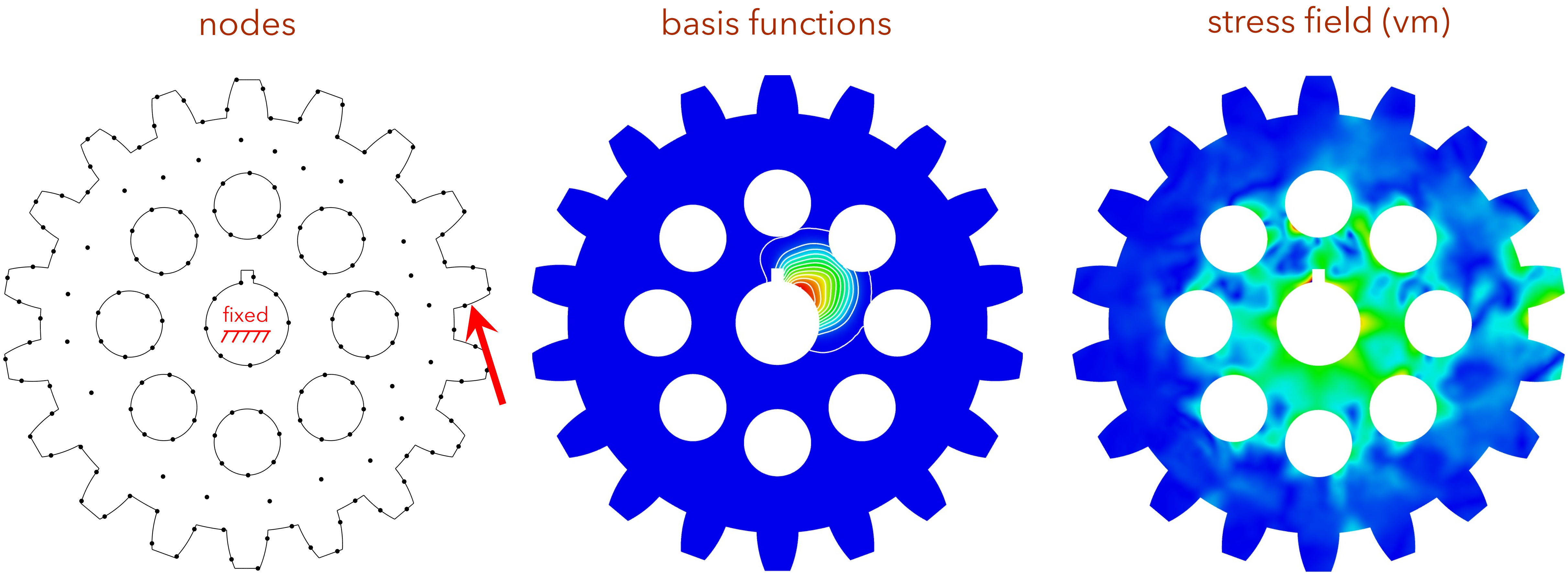
Motivation: element-based polyhedral discretizations



von Mises stress field



Motivation: element-free discretizations

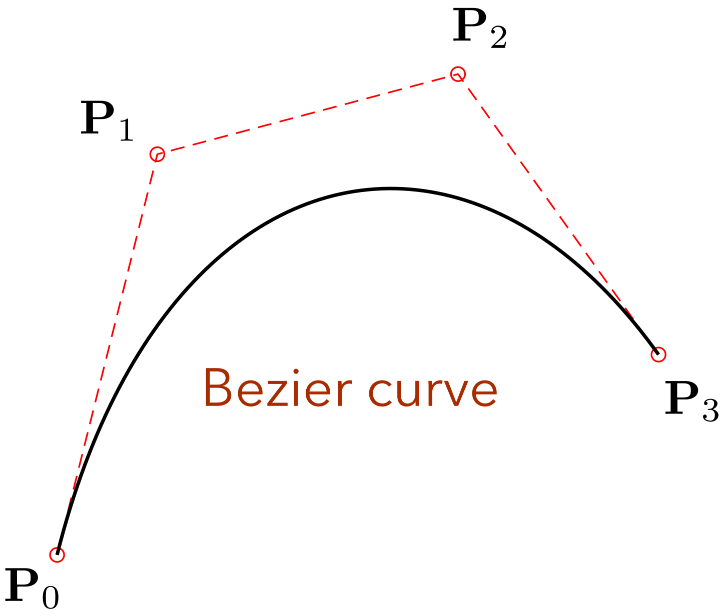


Outline

1. motivation
2. GBC approximation
3. GBC application to quadrature
4. conjugate basis
5. element-based applications
6. element-free applications (weight functions using manifold geodesics)
7. summary

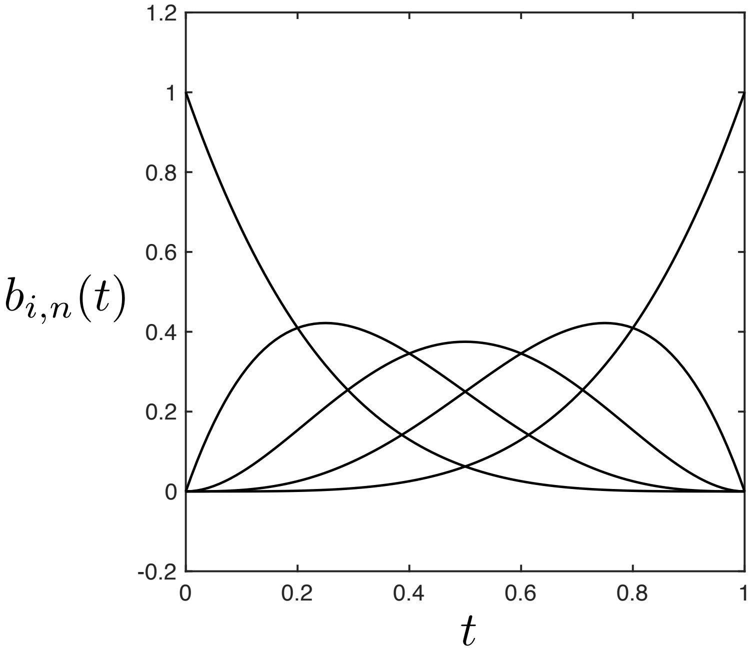
Geometry to function approximation

$$\mathbf{B}(t) := \sum_{i=0}^{n} \mathbf{P}_i \; b_{i,n}(t)$$



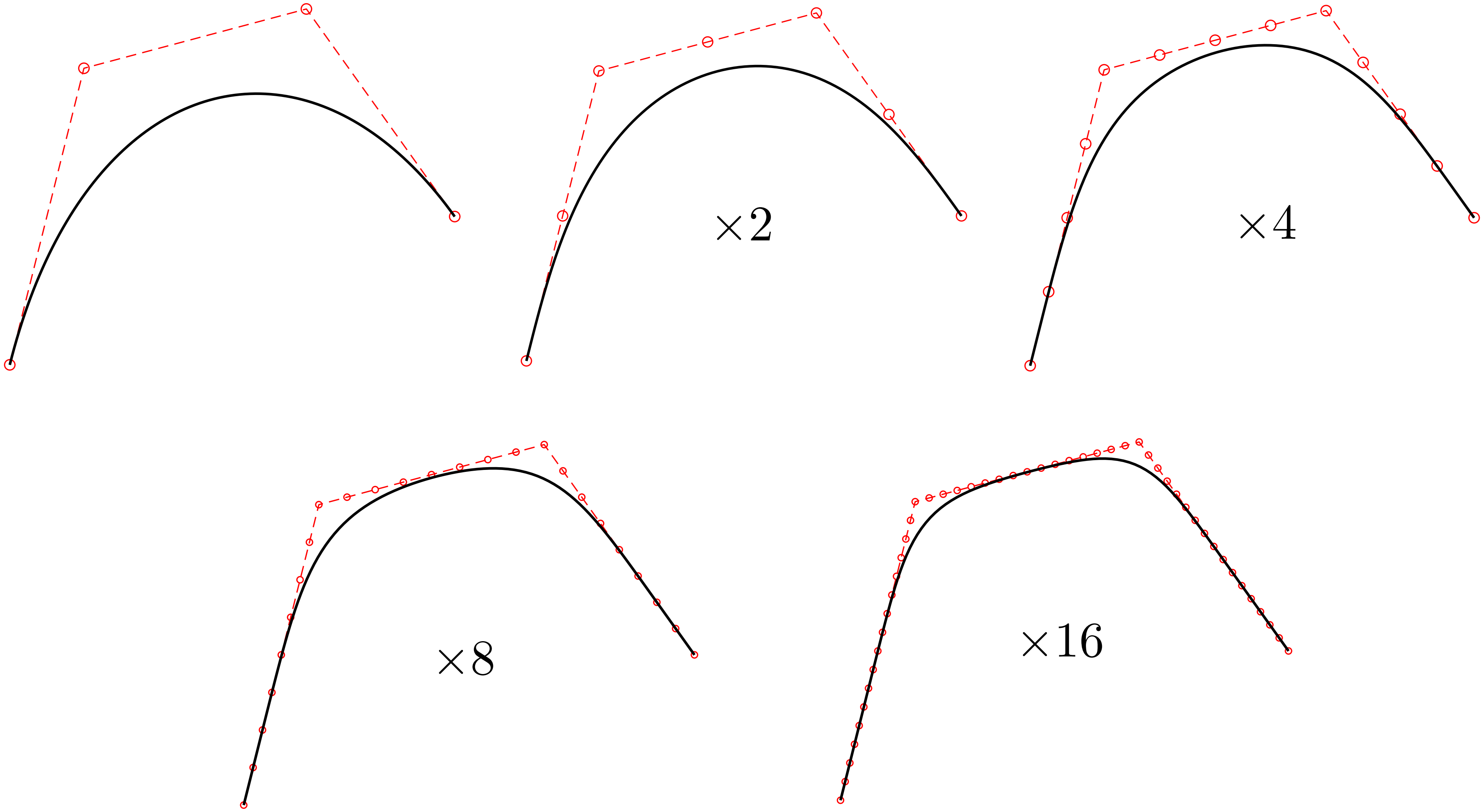
control points $\mathbf{P}_i \quad i = 0, \ldots, n$

Bernstein polynomial $b_{i,n}(t)$



provide affine invariance (scaling, rotation)

Geometry to function approximation



Bernstein polynomial, a GBC-type approximation

Bernstein polynomial $B_n(t) := \sum_{i=0}^{n} \beta_n \, b_{i,n}(t)$

Bernstein basis polynomials of degree n

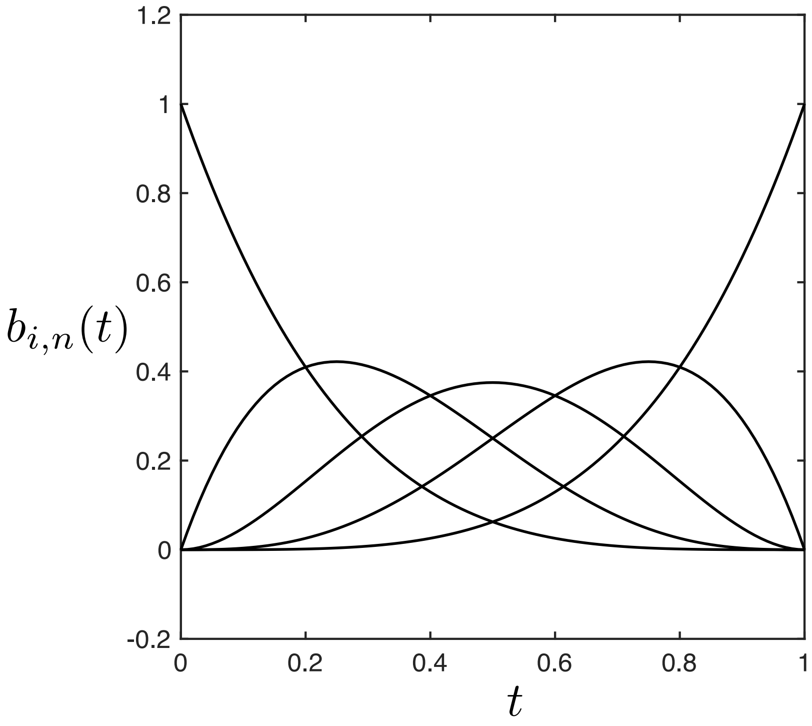
$$b_{i,n}(t) := \binom{n}{i} t^i \, (1-t)^{n-i}, \quad i = 0, \ldots, n$$

$\sum_{i=0}^{n} b_{i,n} = 1$ partition of unity

$\sum_{i=0}^{n} \frac{i}{n} b_{i,n} = t$ linear reproduction

affine invariance

Bernstein polynomial $b_{i,n}(t)$



Note that $b_{i,n}(t)$ are not interpolating except at endpoints.

Function approximation using Bernstein polynomials

Weierstrass approximation theorem: constructive proof that polynomials are dense in C[0,1].

Theorem: Let f be a continuous function on the interval $[0,1]$. Then the Bernstein polynomial $B_n(f)(x)$ given by

$$B_n(f)(x) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) b_{i,n}(x)$$

converges uniformly to $f(x)$ on the interval $[0,1]$.

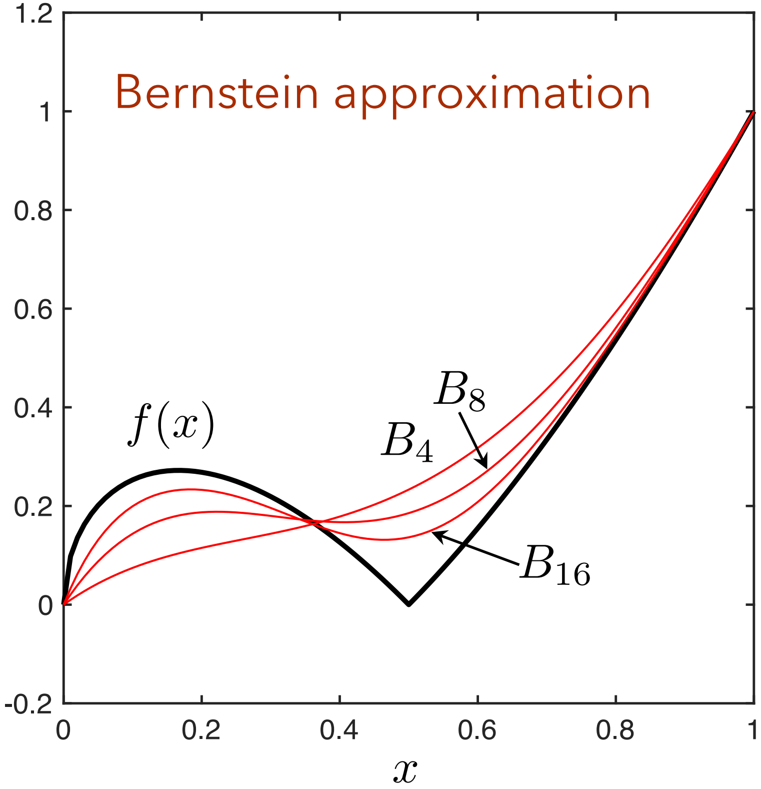
$$\boxed{\lim_{n\to\infty} B_n(f)(x) = f(x)}$$

convergence is uniform

Example: Function approximation using Bernstein polynomials

Let $f(x) = |2x - 1|\sqrt{x}$

$$B_n(x) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) b_{i,n}(x)$$



Key observations:

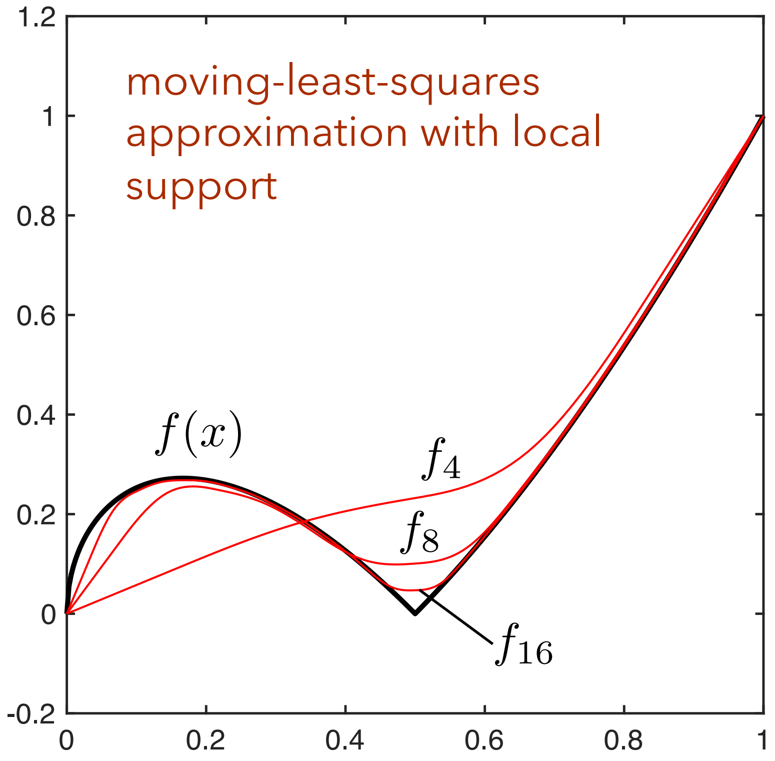
- Bernstein basis polynomials have global support.
- Bernstein polynomials are not interpolatory, but still converge because of linear consistency.
- Approximation involves product of functional and basis.
- Can we generalize to any GBC? (yes!)

Example: Function approximation using moving-least-squares basis

$$f(x) = |2x - 1|\sqrt{x}$$

$$f_N = \sum_{K=1}^{N} f(x_K)\phi_K(x)$$

$\phi_K(x)$ are MLS basis functions with local support



Observations:

- Approximation still converges uniformly.
- Convergence is faster with local support.

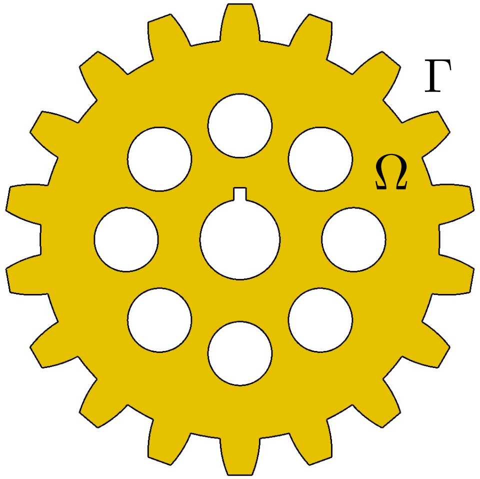
Function approximation using GBCs on general domains

Theorem:

Let f be a continuous function on the domain Ω. Suppose $\{\phi_K(\mathbf{x}), K = 1, \ldots, N\}$ forms a partition of unity and each function ϕ_K has local support of size h. Then the GBC approximation $f_h(\mathbf{x})$ given by

$$f_h(\mathbf{x}) := \sum_K f(\mathbf{x}_K)\phi_K(\mathbf{x})$$

converges uniformly to $f(\mathbf{x})$ on $\overline{\Omega}$.



$$\boxed{\lim_{h\to 0} f_h(x) = f(x)}$$

convergence is uniform

GBC application to quadrature

Will be used in applications to PDE solution.

Application to quadrature

Since $\lim_{h \to 0} f_h(x) = f(x)$ uniformly, it follows that $\int_\Omega f_h(\mathbf{x})\, d\Omega \to \int_\Omega f(\mathbf{x})\, d\Omega$

with $\left| \int f_h(\mathbf{x})\, d\Omega - \int f(\mathbf{x})\, d\Omega \right| \leq \int |\, f_h(\mathbf{x}) - f(\mathbf{x})\,|\, d\Omega \leq \int \varepsilon\, d\Omega = V \cdot \varepsilon$

Can obtain rates of convergence using Taylor's theorem.

Quadrature

$$f_h(\mathbf{x}) := \sum_K f(\mathbf{x}_K)\phi_K(\mathbf{x})$$

then $$\int f_h(\mathbf{x})\,d\Omega = \sum_K f(\mathbf{x}_K)\underbrace{\int \phi_K(\mathbf{x})\,d\Omega}_{}$$

Define quadrature weight as $$\mathrm{w}_K = \int_\Omega \phi_K(\mathbf{x})\,d\Omega$$

$$\boxed{\int f_h(\mathbf{x})\,d\Omega = \sum_K \mathrm{w}_K f(\mathbf{x}_K) \approx \int f(\mathbf{x})\,d\Omega}$$

$$\boxed{\mathrm{w}_K = \int_\Omega \phi_K(\mathbf{x})\,d\Omega}$$

Quadrature

Note that $$\sum_K \mathrm{w}_K = \sum_K \int_\Omega \phi_K(\mathbf{x})\, d\Omega = \int_\Omega \sum_K \phi_K(\mathbf{x})\, d\Omega = \int_\Omega 1\, d\Omega = V$$

Also, since $$\sum_K \mathbf{x}_K\, \phi_K(\mathbf{x}) = \mathbf{x}$$

then $$\sum_K \mathrm{w}_K \mathbf{x}_K = \sum_K \int_\Omega \phi_K(\mathbf{x})\, \mathbf{x}_K\, d\Omega = \int_\Omega \sum_K \mathbf{x}_K \phi_K(\mathbf{x})\, d\Omega = \int_\Omega \mathbf{x}\, d\Omega$$

Now have a second-order integration scheme that can integrate linear functions exactly.

$$\boxed{\sum_K \mathrm{w}_K = V} \qquad \boxed{\sum_K \mathrm{w}_K\, \mathbf{x}_K = \int_\Omega \mathbf{x}\, d\Omega}$$

Can extend to higher-order integration using higher-order reproducing conditions.

Quadrature example in 1D

For Bernstein approximation:

recall $B_n(x) := \sum_{i=0}^{n} f\left(\frac{i}{n}\right) b_{i,n}(x)$

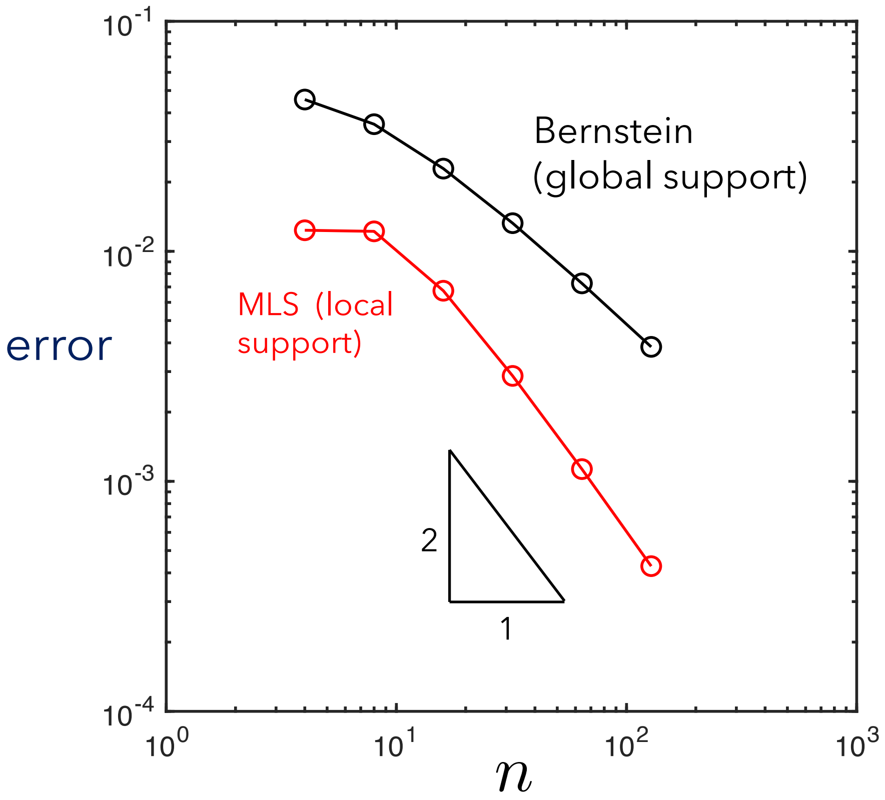
$$\int_0^1 b_i, n(x)\, dx = \frac{1}{n+1} = \mathrm{w}_i$$

$$\int_0^1 f(x)\, dx \approx \int_0^1 B_n(f)(x)\, dx = \frac{1}{n+1} \sum_{i=0}^{n} f\left(\frac{i}{n}\right)$$

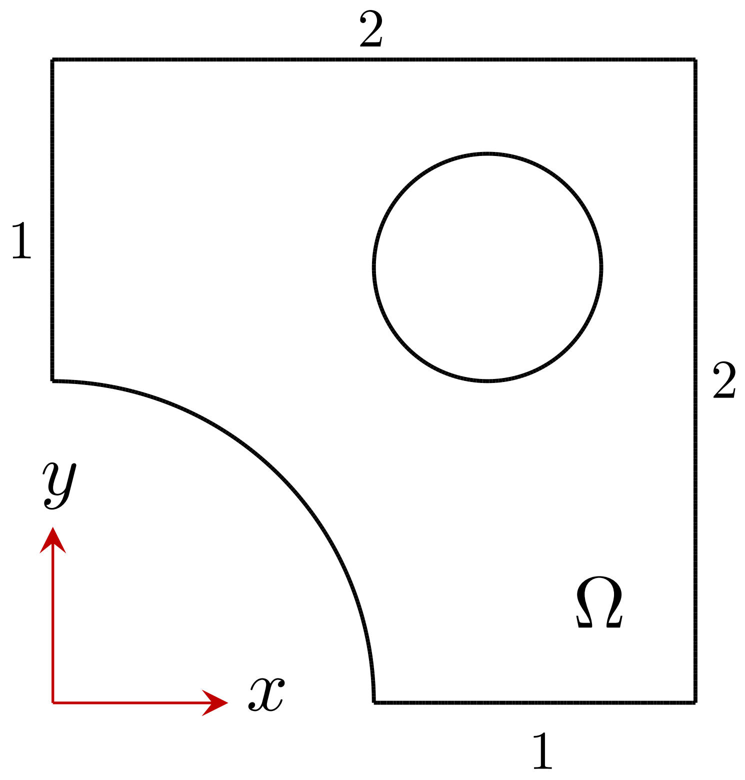
~ midpoint rule

Of course, Gauss rules are much more efficient, but are not generalizable to arbitrary domains.

Let $f(x) = |2x-1|\sqrt{x}$



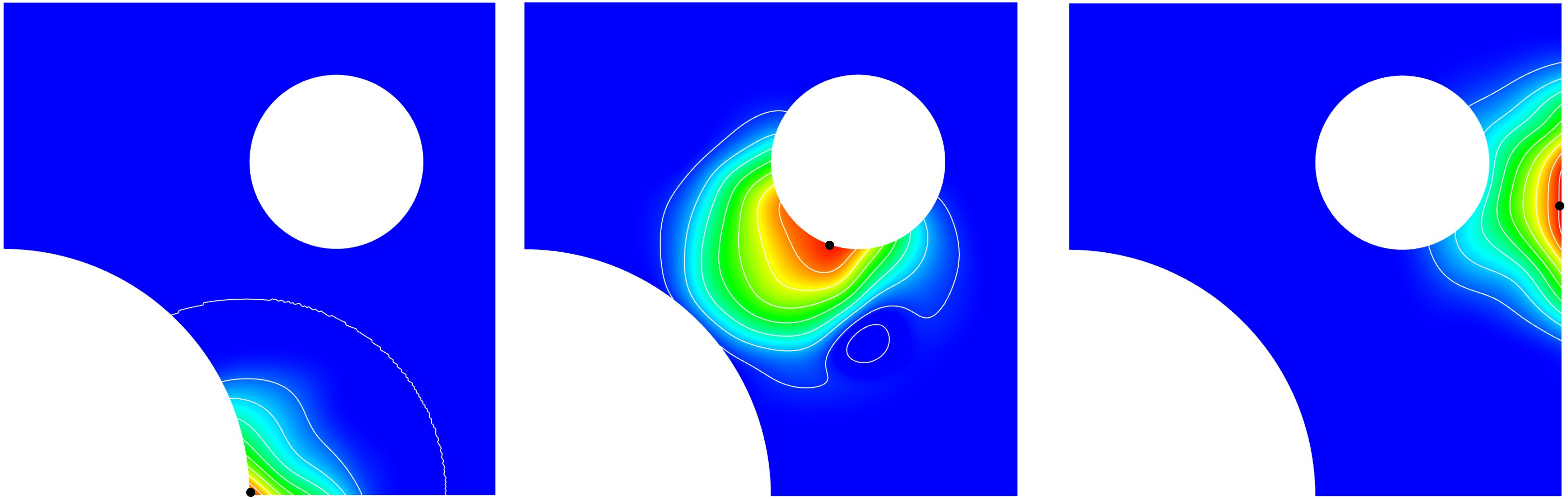
Quadrature example in 2D



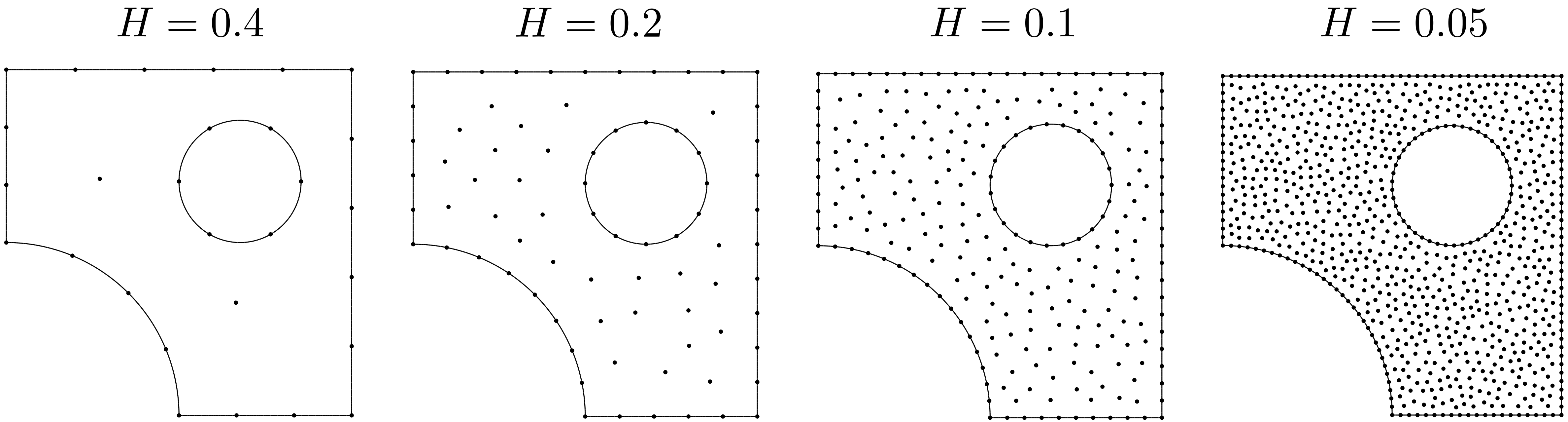
$$f(x, y) = \sin(\pi x/2)\sin(\pi y)$$

$$\text{error} := \left| \sum_K \mathrm{w}_K f(\mathbf{x}_K) - \int_\Omega f \, d\Omega \right|$$

GBC element-free basis functions



Quadrature example

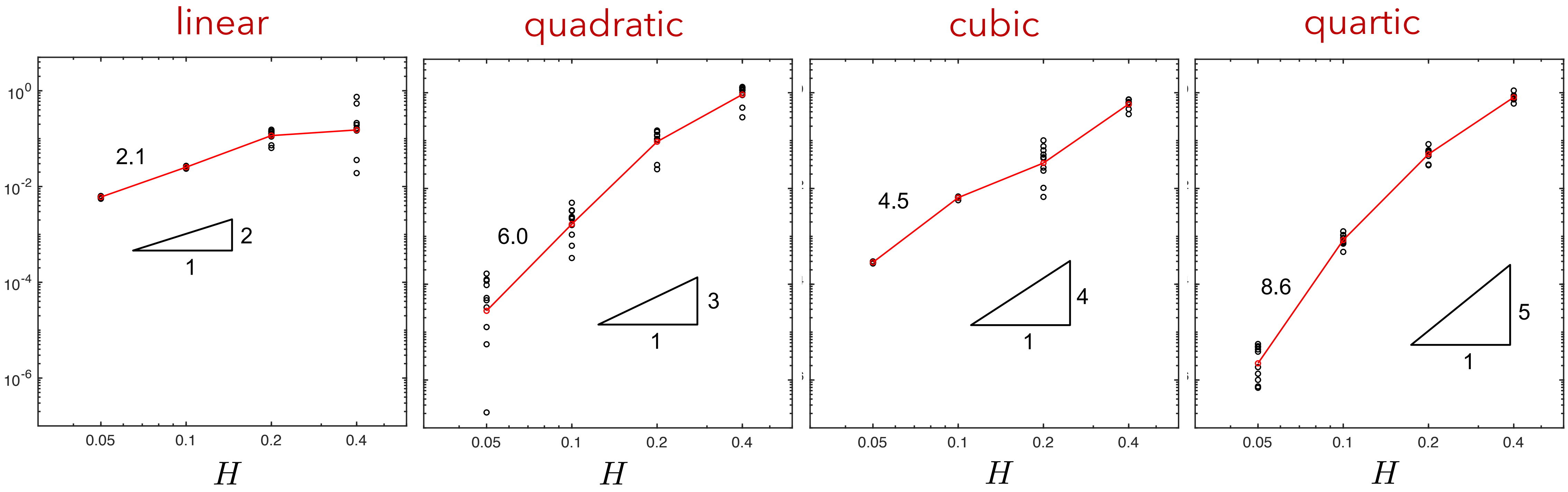


$H = 0.4$ $H = 0.2$ $H = 0.1$ $H = 0.05$

Evaluate error for 10 realizations.

Quadrature convergence

$$\text{error} := \left| \sum_K \mathrm{w}_K f(\mathbf{x}_K) - \int_\Omega f \, d\Omega \right|$$



Note: seeing convergence rates greater than $p+1$

Conjugate (dual) basis

Will be used in applications to PDE solution.

Conjugate basis

Consider a set of linearly independent GBCs $\{\Phi_I\}$

It will be useful in the formulation of both element-based and element-free solutions of PDEs to project function gradients to this basis.

The projection can be written in terms of the conjugate (dual) basis $\{\Phi^J\}$

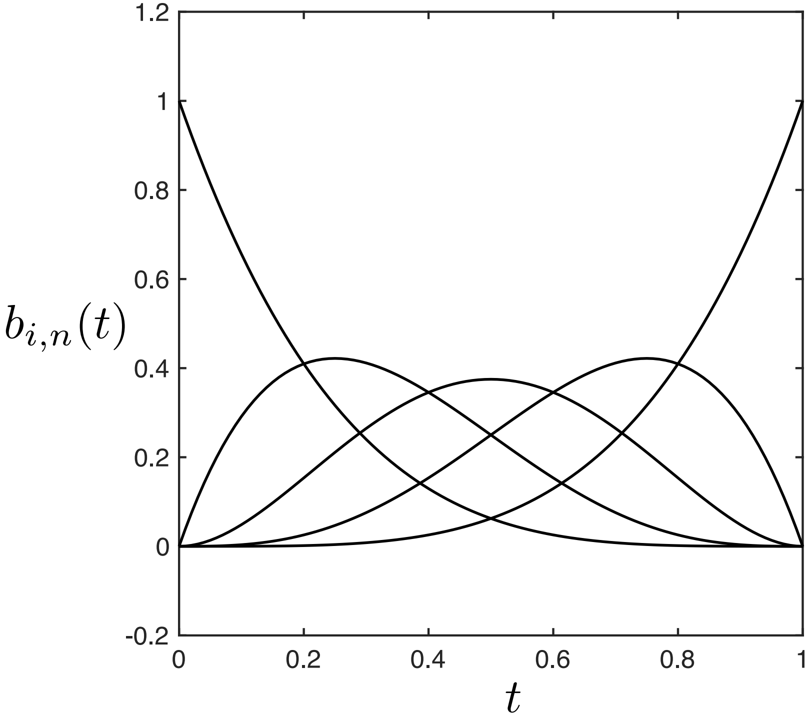
$(\Phi_I, \Phi^J) = \delta_I^J$ *bi-orthogonal*

Define G_{IJ} as the Gram matrix for the basis $\boxed{G_{IJ} := \int_\Omega \Phi_I \Phi_J \, d\Omega}$

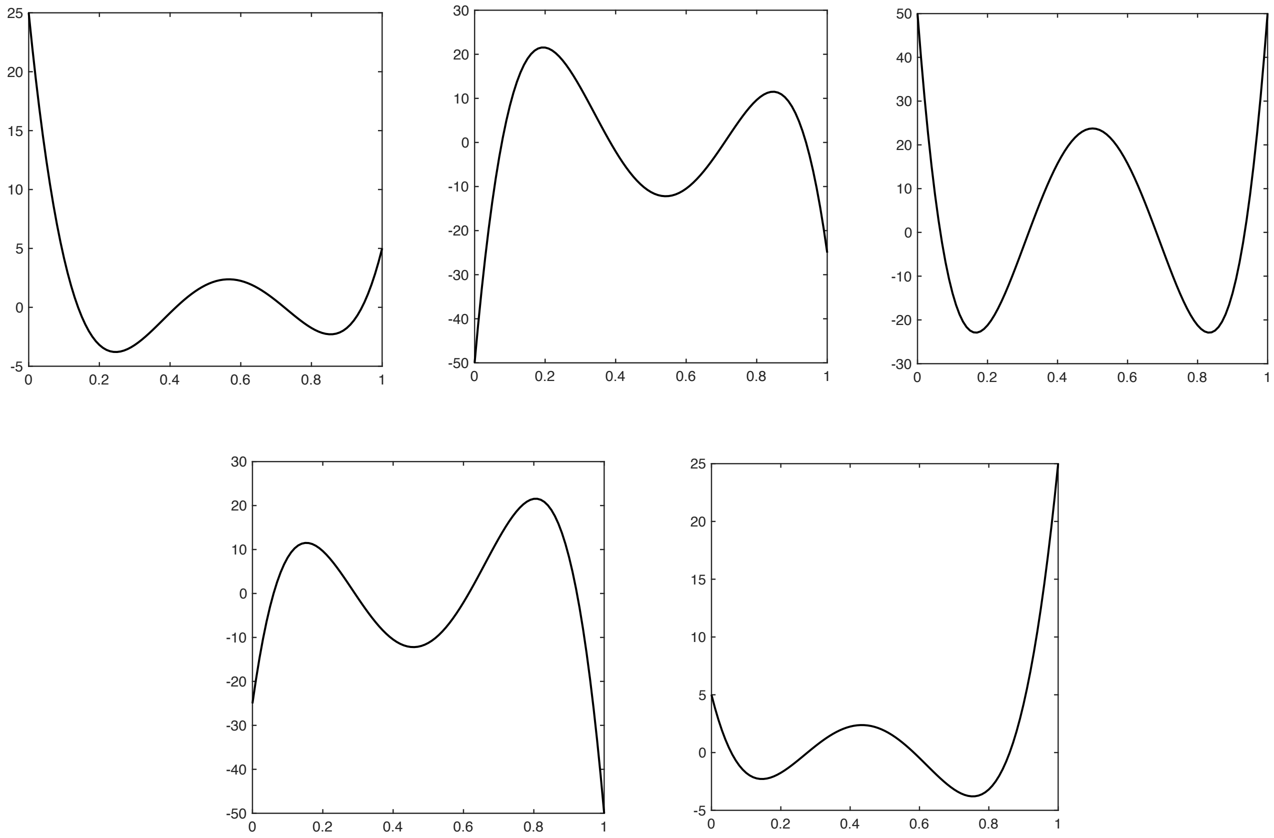
Can show that $\Phi^I = G^{IJ}\Phi_J$ and $\Phi_I = G_{IJ}\Phi^J$ where $G^{IJ} = (G_{IJ})^{-1}$

Conjugate basis example: Bernstein

Bernstein polynomial $b_{i,n}(t)$

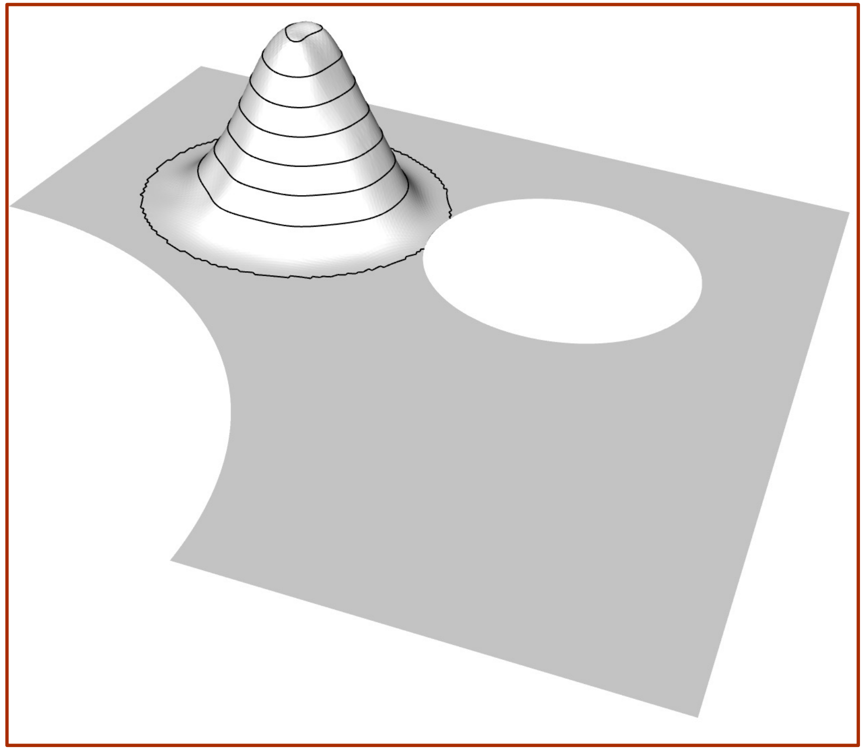


conjugate functions

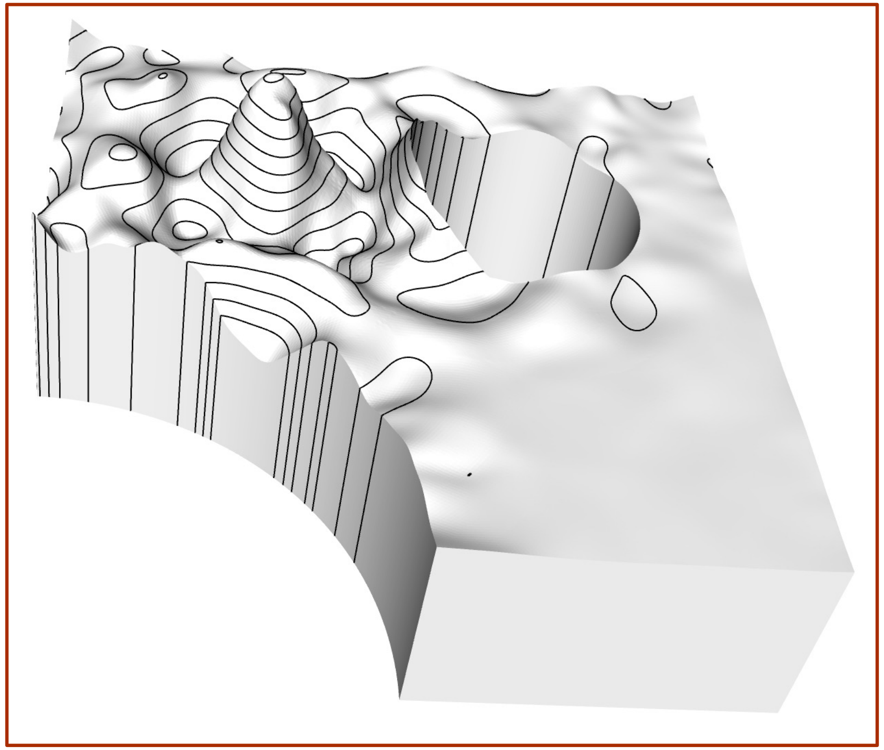


Conjugate basis example: element free

basis vector Φ_K



dual basis vector Φ^K



$$(\Phi_K, \Phi^J) = \delta_K^J$$ *bi-orthogonal*

Note: Φ_K has local support, but Φ^K has global support

Application to PDEs (solid mechanics)

Also applicable to formulations in H(div) and H(curl)?

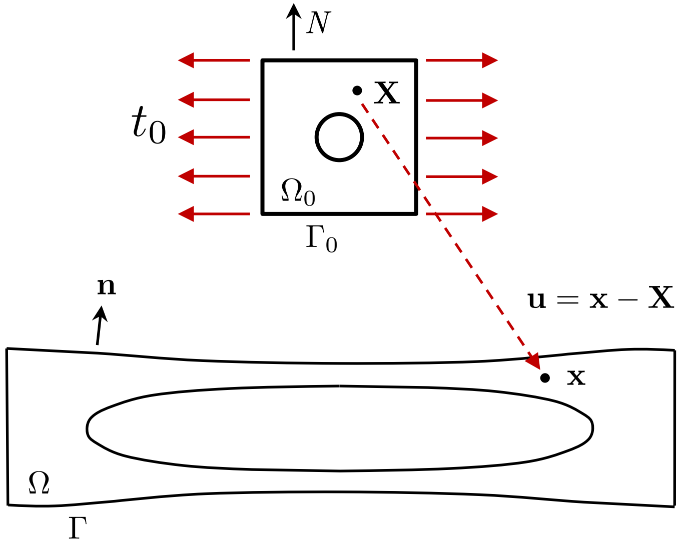
Governing equations (total-Lagrangian formulation)

strong form

$$\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{X}} : \boldsymbol{I} = \rho_0 \, \ddot{\mathbf{u}}$$

$$\boldsymbol{u} = \overline{\boldsymbol{u}} \quad \text{on} \quad \Gamma_0^u \quad \text{and} \quad \boldsymbol{P} \cdot \boldsymbol{N} = \boldsymbol{t}_0 \quad \text{on} \quad \Gamma_0^t$$

P is first Piola-Kirchhoff stress tensor



weak form

find the trial functions $\boldsymbol{u} \in \boldsymbol{H}^1(\Omega_0)$ such that

$$\int_{\Gamma_0^t} \boldsymbol{t}_0 \cdot \boldsymbol{v} \, dS - \int_{\Omega_0} \boldsymbol{P} : (\partial \boldsymbol{v} / \partial \boldsymbol{X}) \, d\boldsymbol{X} = \int_{\Omega_0} \rho_0 \, \ddot{\mathbf{u}} \cdot \boldsymbol{v} \, d\boldsymbol{X}$$

for all test functions $\boldsymbol{v} \in \boldsymbol{H}_0^1(\Omega_0)$

Governing equations for linear elasticity

strong form $$\frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{x}} : \mathbf{I} + \mathbf{f} = \mathbf{0} \qquad \mathbf{u} = \overline{\mathbf{u}} \ \text{ on } \ \Gamma_{\mathrm{u}} \ \text{ and } \ \boldsymbol{\sigma}\,\mathbf{n} = \mathbf{t} \ \text{ on } \ \Gamma_{\mathrm{t}}$$

$$\boldsymbol{\sigma} = \mathbb{C}\,\boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} := \mathrm{sym}\,(\nabla \mathbf{u})$$ *(linear elastic)*

$$\exists\, \alpha_l\,,\, \alpha_u > 0 \ \text{ such that } \ \alpha_l\, \boldsymbol{\epsilon} : \boldsymbol{\epsilon} \leq \boldsymbol{\epsilon} : (\mathbb{C}(\mathbf{x})\,\boldsymbol{\epsilon}) \leq \alpha_u\, \boldsymbol{\epsilon} : \boldsymbol{\epsilon} \qquad \forall \boldsymbol{\epsilon}$$ *(uniform ellipticity)*

weak form find the trial functions $\boldsymbol{u} \in \boldsymbol{H}^1(\Omega_0)$ such that

$$\int_{\Omega} \boldsymbol{\sigma} : (\partial \mathbf{v} / \partial \mathbf{x})\, d\Omega = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}\, d\Omega + \int_{\Gamma_{\mathrm{t}}} \mathbf{t} \cdot \mathbf{v}\, d\Gamma$$

for all test functions $\boldsymbol{v} \in \boldsymbol{H}^1_0(\Omega_0)$

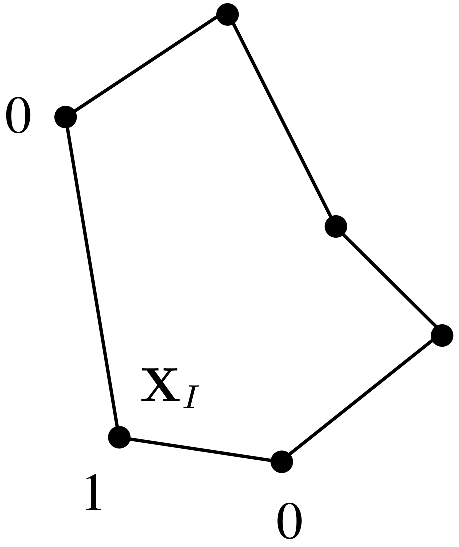
Show bilinear form?

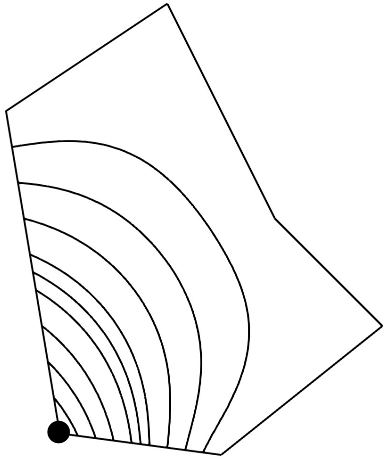
$$a(u, v) = \int_{\Omega_{\mathrm{e}}} \nabla u : \mathbb{C}\, \nabla v\, d\Omega$$

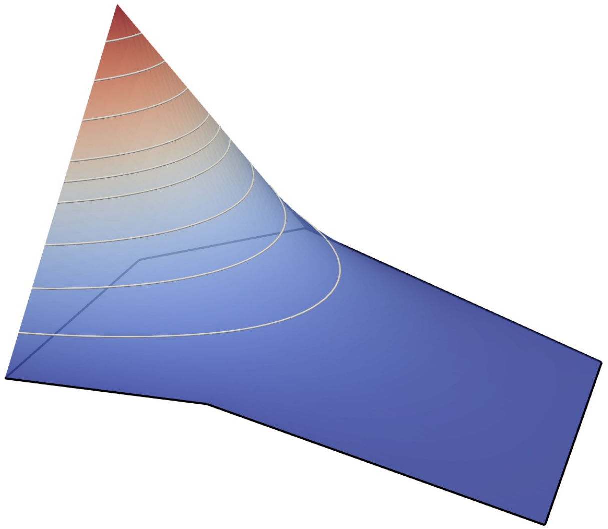
Element-based discretizations (polyhedra)

Harmonic shape functions

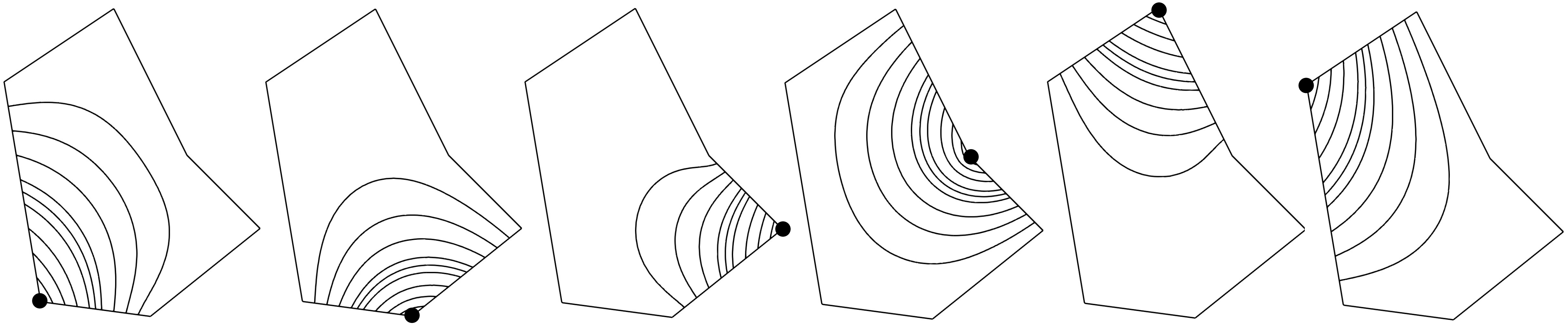
$$\nabla^2 \phi_I = 0$$







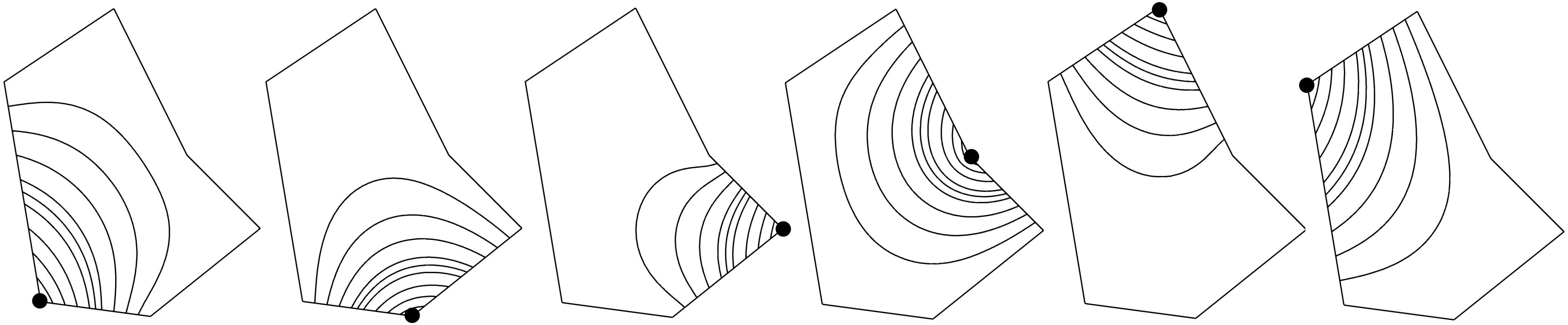
Harmonic shape functions



$$\sum_I \phi_I(\mathbf{x}) = 1$$ partition of unity

$$\sum_I \mathbf{x}_I \, \phi_I(\mathbf{x}) = \mathbf{x}$$ linear reproducibility

Quadrature weights



$\frac{w_1}{A} = 0.201$ $\frac{w_2}{A} = 0.147$ $\frac{w_3}{A} = 0.110$ $\frac{w_4}{A} = 0.196$ $\frac{w_5}{A} = 0.138$ $\frac{w_6}{A} = 0.209$

A = area

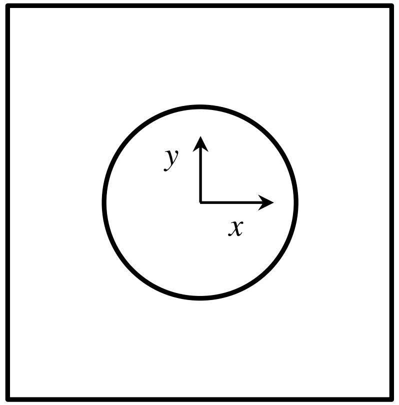
quadrature weights $\sum_K w_K = A$

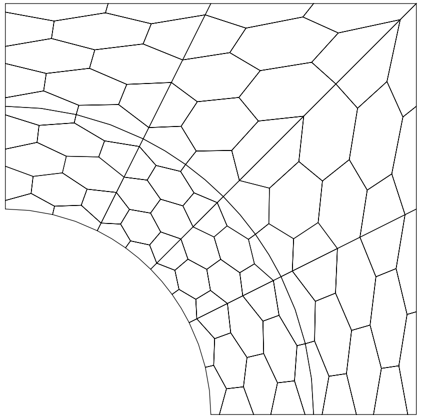
$$\boxed{w_K = \int_{\Omega_e} \phi_K(\mathbf{x})\, d\Omega}$$

Quadrature

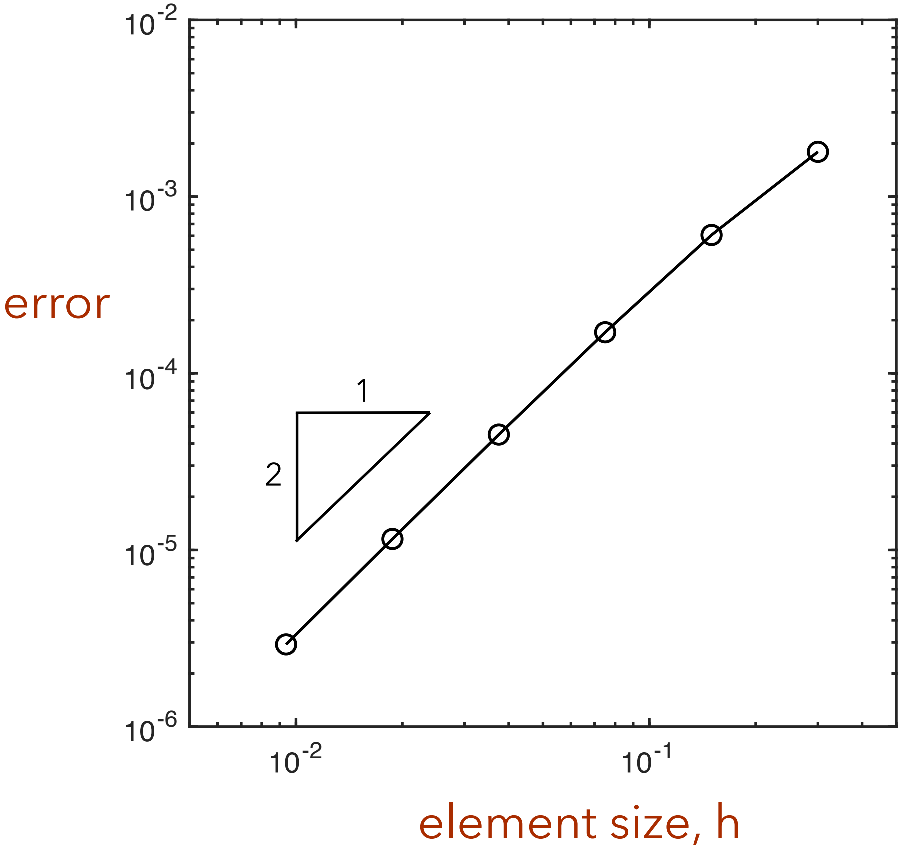
$$\text{error} = \left| \int f - \sum_i w_i f_i \right|$$

$$f(x,y) = \left[1 - \left(\frac{2x}{L_x}\right)^2\right]\left[1 - \left(\frac{2y}{L_y}\right)^2\right]$$

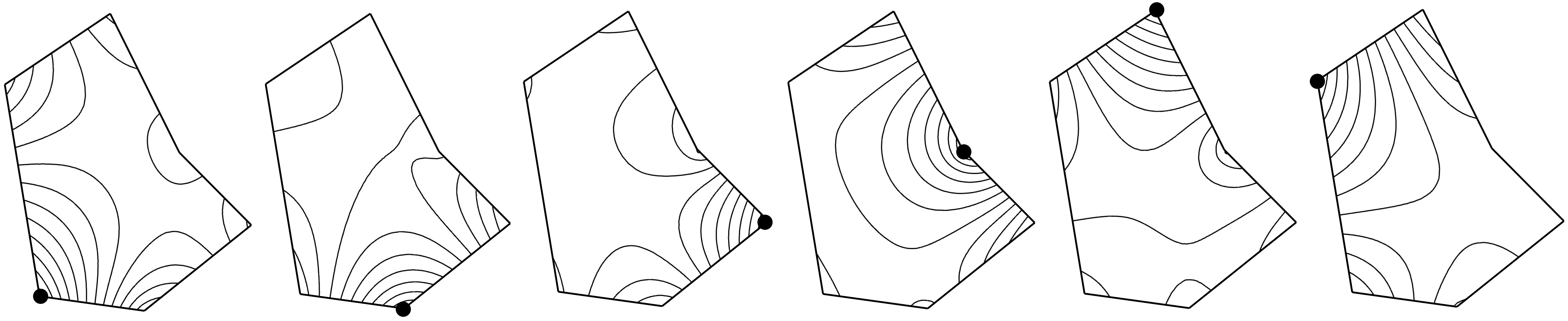




quadrature error



Conjugate functions



$(\Phi_I, \Phi^J) = \delta_I^J$ *bi-orthogonal*

Consistency of discrete form (integration)

- For convergence of discrete approximation, need to ensure consistency of discrete and continuous bilinear forms.
- Requires polynomial consistency of shape-function gradients (including quadrature).
- To obtain quadrature consistency, project the DoF shape function gradients to the subspace of quadrature shape functions.
- Only performed once in a pre-processing step.

$\{\phi_I, I = 1, \ldots, N\}$ *DoF basis (shape functions)*

$\{\Phi_K, K = 1, \ldots, M\}$ *Quadrature basis (shape functions)*

$$\overline{\nabla}\phi_I := \arg\min \int_\Omega \left(\nabla\phi_I - \sum_{K=1}^{M} a^K \Phi_K \right)^2 d\Omega \quad \textit{(}L_2\textit{ projection)}$$

The projection can be written in terms of the dual or conjugate basis $\{\Phi^J\}$

$$(\Phi_K, \Phi^J) = \delta_K^J \quad \textit{bi-orthogonal}$$

$$\overline{\nabla}\phi_I = \sum_K \underbrace{(\nabla\phi_I, \Phi_K)}_{\textit{covariant components}}\Phi^K = \sum_K \underbrace{(\nabla\phi_I, \Phi^K)}_{\textit{contravariant components}}\Phi_K$$

Can prove polynomial consistency up to the order of the precision of $\{\Phi_K\}$

$$\boxed{\text{Theorem:} \quad \int_\Omega \mathbf{p}\,\overline{\nabla}\phi_I\,d\Omega = \int_\Omega \mathbf{p}\,\nabla\phi_I\,d\Omega \quad \text{for all} \quad \mathbf{p} \in \mathbb{P}_k(\Omega)}$$

This ensures satisfaction of the patch test.

Replace the original bilinear form $a(u,v) = \int_{\Omega} \nabla u : \mathbb{C}\, \nabla v \, d\Omega$

with this modified bilinear form $\bar{a}(u,v) = \int_{\Omega} \overline{\nabla} u : \mathbb{C}\, \overline{\nabla} v \, d\Omega$

Note: This modified bilinear form is still symmetric (Bubnov-Galerkin).

$$\bar{a}(u,v) = \int_{\Omega} \left[\sum_{I} (\nabla u, \Phi_I) \Phi^I \right] \mathbb{C} \left[\sum_{J} (\nabla v, \Phi_J) \Phi^J \right] d\Omega$$

$$\bar{a}(u,v) = \sum_{I,J} (\nabla u, \Phi_I)\, \mathbb{C}\, (\nabla v, \Phi_J) \underbrace{\int_{\Omega_{\mathrm{e}}} \Phi^I \Phi^J \, d\Omega}_{G^{IJ}}$$

Can show that $G^{IJ} = (G_{IJ})^{-1}$

where $G_{IJ} = \int_{\Omega_e} \Phi_I \Phi_J \, d\Omega$ is the Gram matrix for the basis $\{\Phi_K\}$

Can show that $\Phi^I = G^{IJ}\Phi_J$ and $\Phi_I = G_{IJ}\Phi^J$

$$\boxed{\bar{a}(u,v) = \sum_{I,J} G^{IJ}(\nabla u, \Phi_I)\,\mathbb{C}\,(\nabla v, \Phi_J) = \underbrace{\sum_K (\nabla u, \Phi^K)\,\mathbb{C}\,(\nabla v, \Phi_K)}}$$

Looks like a sum over quadrature points.

Replace G_{IJ} with row-sum lumped version: $G^L_{IJ} := \sum_J G_{IJ} = \mathrm{diag}\{\mathrm{w}_K\}$

where recall $\mathrm{w}_K = \int_\Omega \phi_K(\mathbf{x})\, d\Omega$

Then $\overline{a}(u,v) \to \overline{a}^L(u,v) = \sum_K \frac{1}{\mathrm{w}_K} (\nabla u, \Phi_K)\, \mathbb{C}\, (\nabla v, \Phi_K)$ where $(G^L_{IJ})^{-1} = \mathrm{diag}\left\{\frac{1}{\mathrm{w}_K}\right\}$

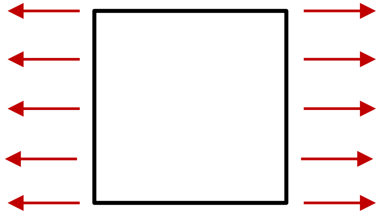
Can write $\overline{a}^L(u,v)$ as $\boxed{\overline{a}^L(u,v) = \sum_K \mathrm{w}_K\, (\overline{\nabla} u)_K : \mathbb{C}\, (\overline{\nabla} v)_K}$

where $\boxed{(\overline{\nabla} u)_K := \frac{1}{\mathrm{w}_K} \int_\Omega (\nabla u)\, \Phi_K\, d\Omega}$ which has the form of a discrete derivative at a quadrature point K.

Our discrete bilinear form is now “sparse.”

Verification: elasticity patch test

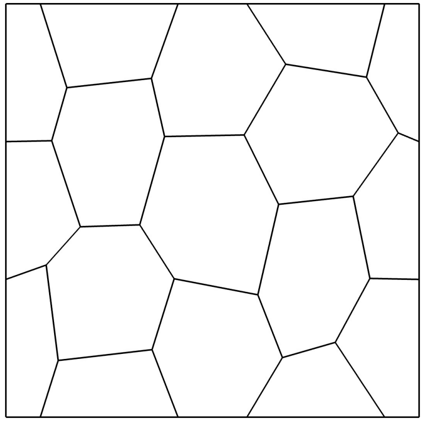
uniaxial tension



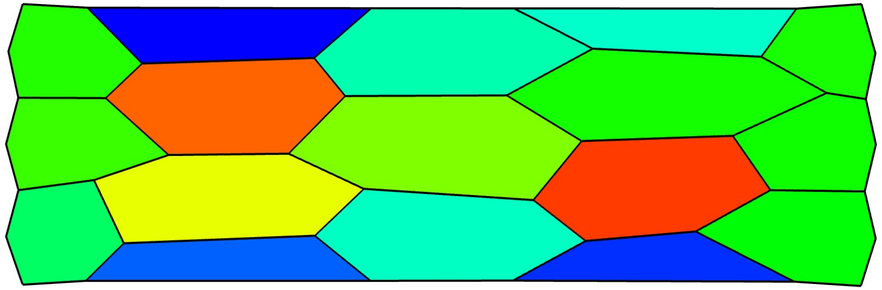
$E = 1.0$

$\nu = 0.3$

hexagon mesh

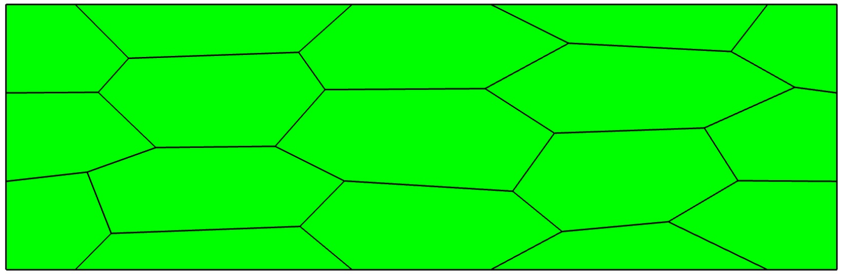


subtriangle quadrature



max stress error = 2%

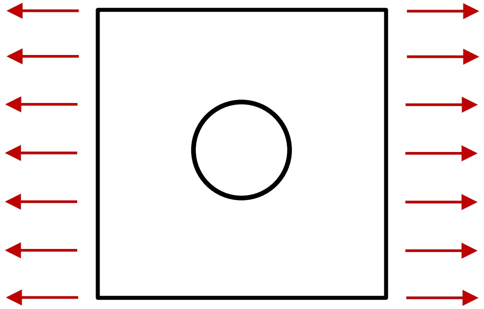
projection based quadrature



max stress error = 3 x 10^{-15}

Verification: elasticity, hole-in-plate tension

uniaxial tension

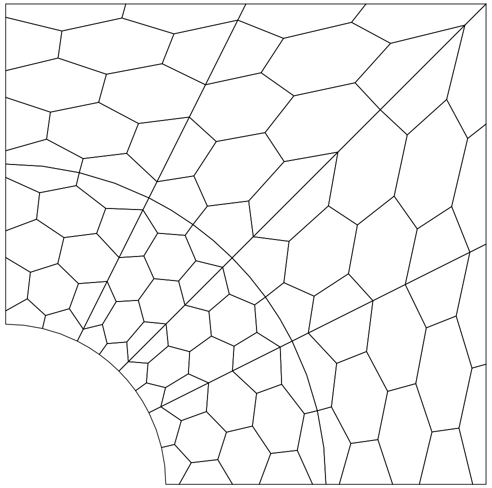


- exact tension prescribed corresponding to infinite plate
- plane strain
- quarter symmetry model used

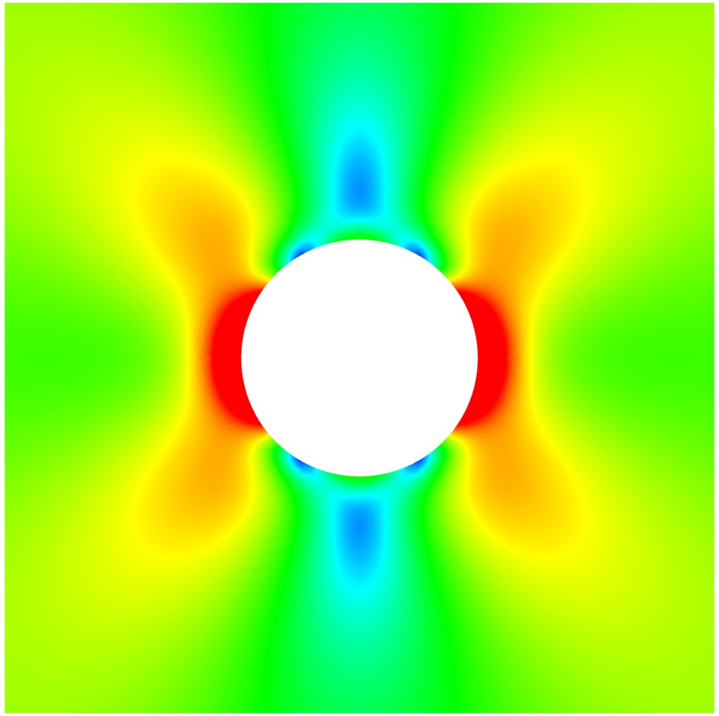
$$E = 1.0$$

$$\nu = 0.3$$

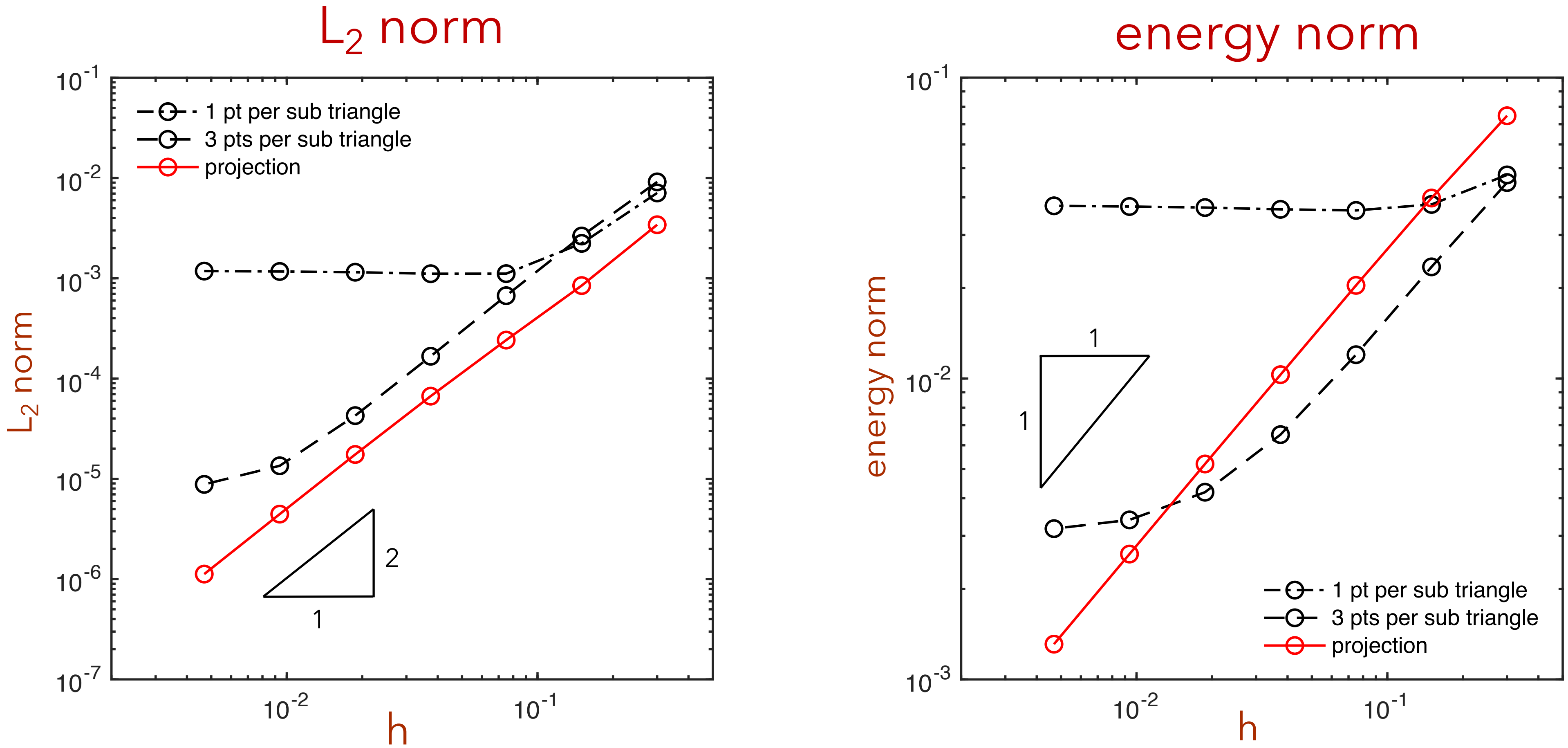
mapped hexagon mesh



von Mises stress invariant



Verification: elasticity, hole-in-plate tension



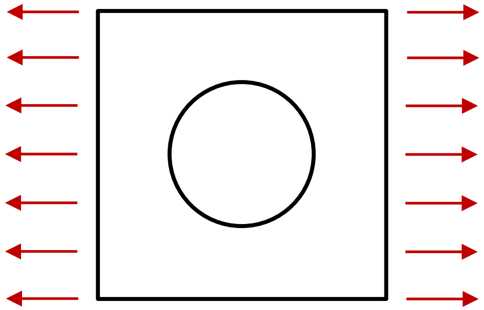
L_2 norm

energy norm

Optimal rates of convergence

Application example: hyperelastic, hole-in-plate

uniaxial extension



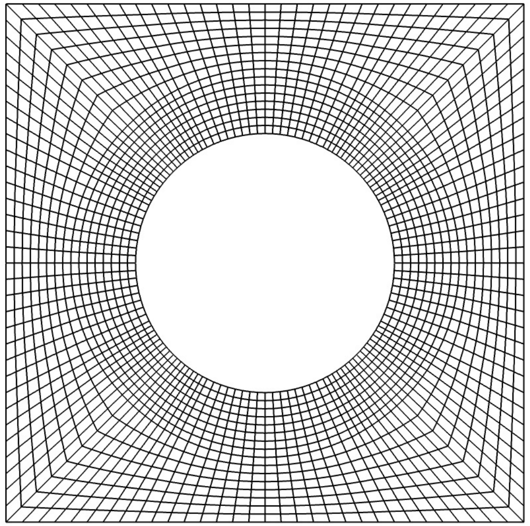
- plane strain
- quarter symmetry model used

compressible neo-Hookean material

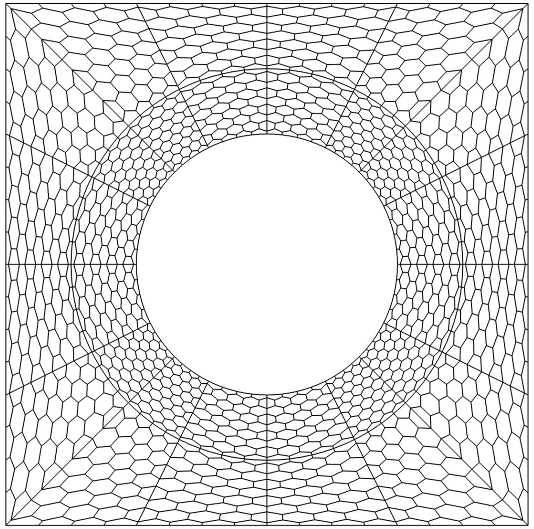
$$\boldsymbol{\sigma} = \frac{\mu}{J}(\mathbf{F}\mathbf{F}^T - \mathbf{I}) + \frac{\lambda}{\ln J}\mathbf{I}$$

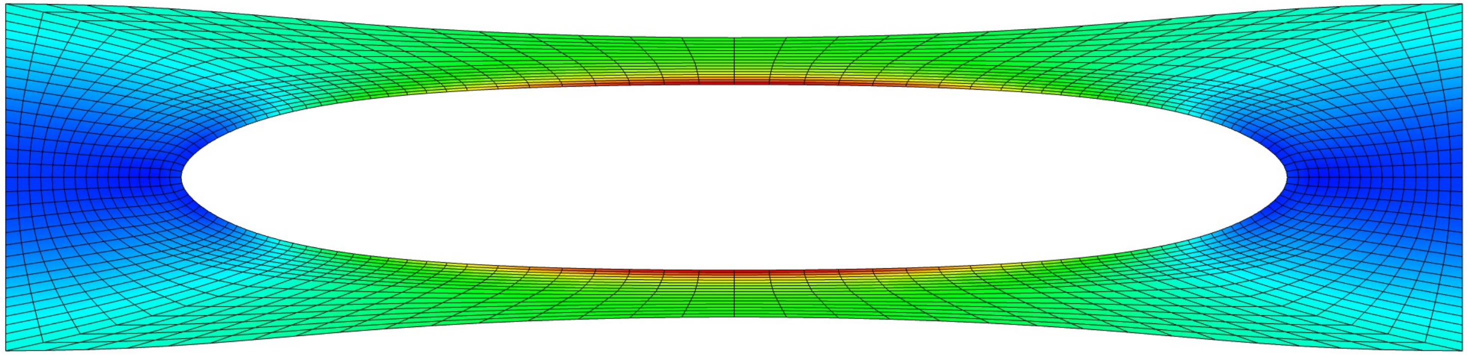
$$J = \det \mathbf{F} \qquad \mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$$

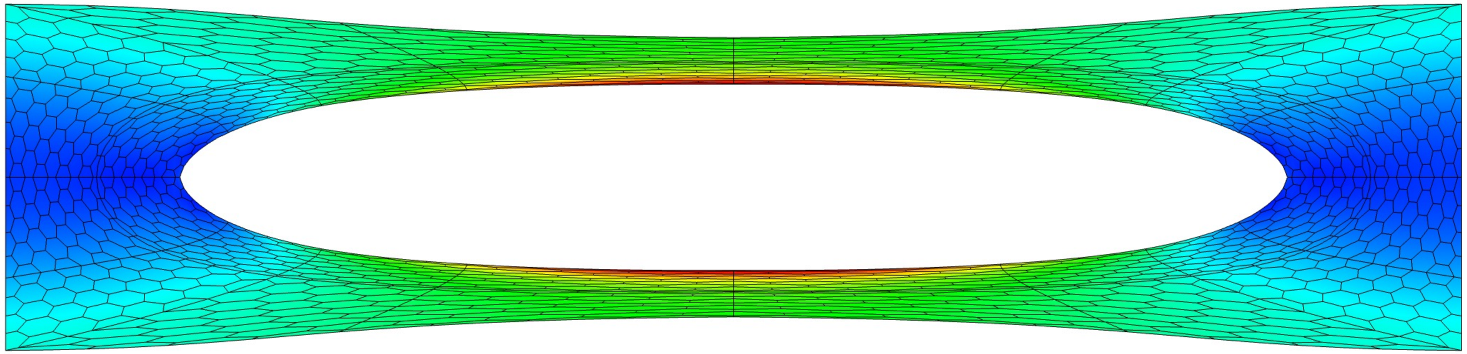
quad mesh

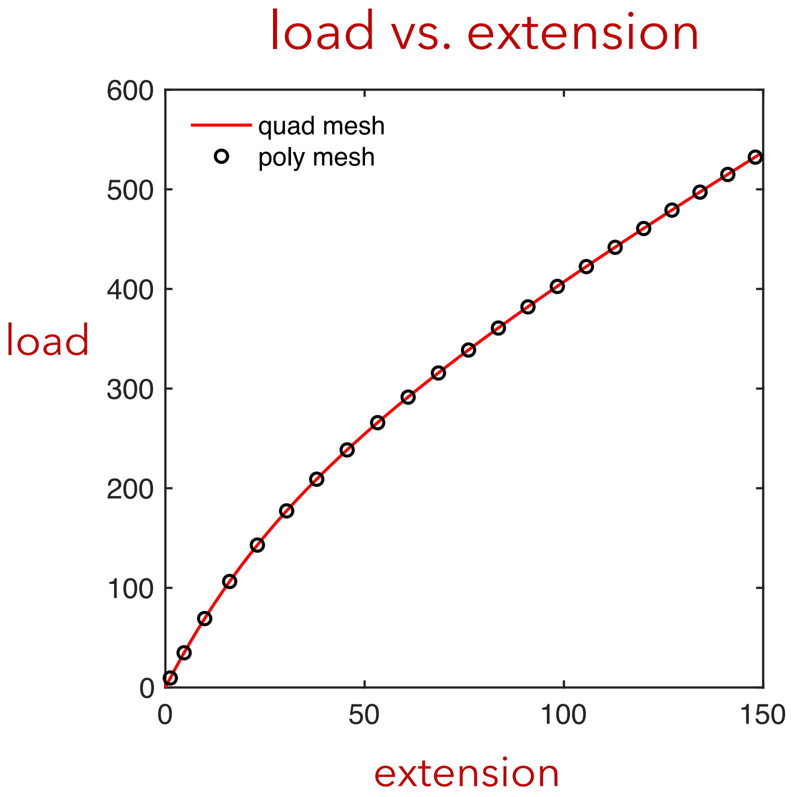


mapped hexagon mesh







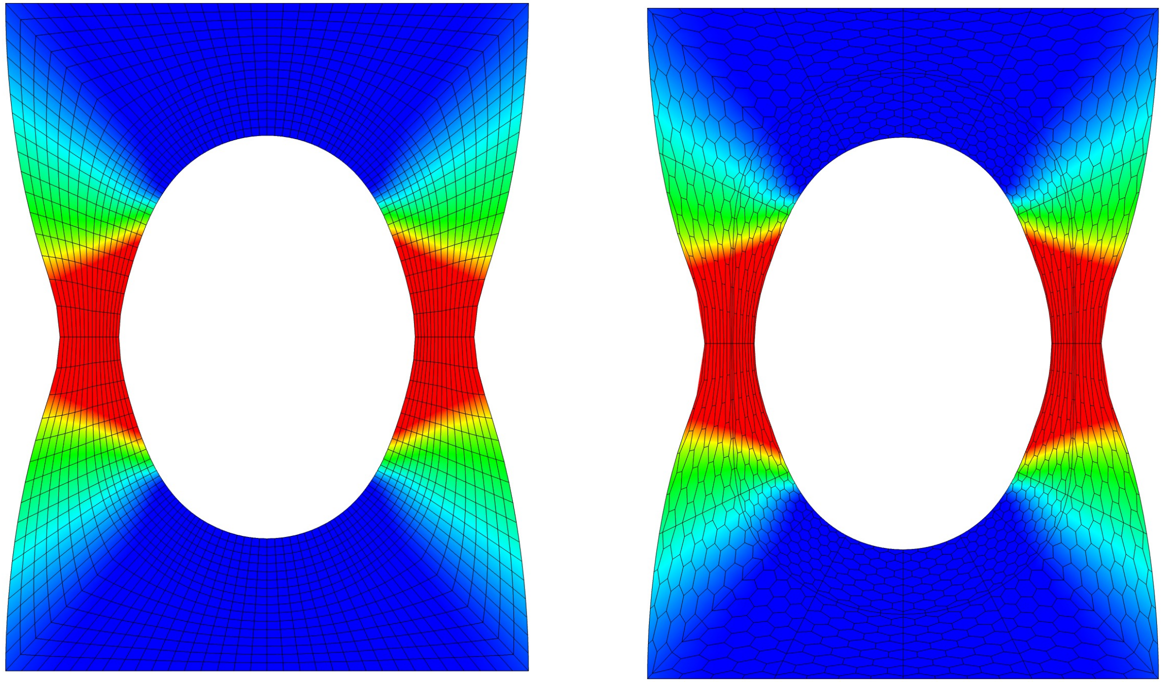

load vs. extension
quad mesh
poly mesh
load
extension
600
500
400
300
200
100
0
0
50
100
150

Application example: elastic-plastic, hole-in-plate

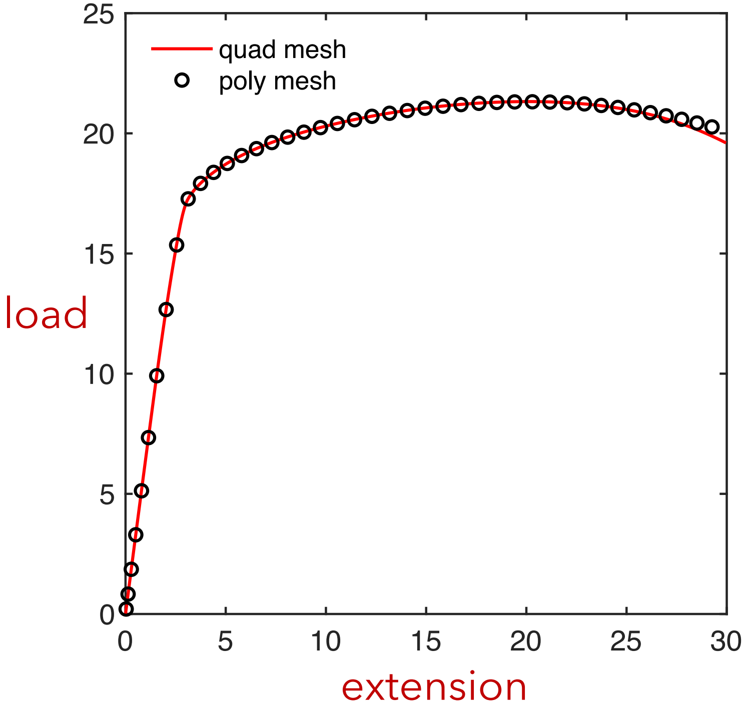
yield surface $f(\sigma, \bar{\varepsilon}^p) = \phi(\sigma) - \sigma_y(\bar{\varepsilon}^p) = 0$

$$\phi(\sigma) = \left\{ \frac{1}{2} \left(|\sigma_1 - \sigma_2|^2 + |\sigma_1 - \sigma_3|^2 + |\sigma_2 - \sigma_3|^2 \right) \right\}^{1/2}$$

plastic strain field



load vs. extension



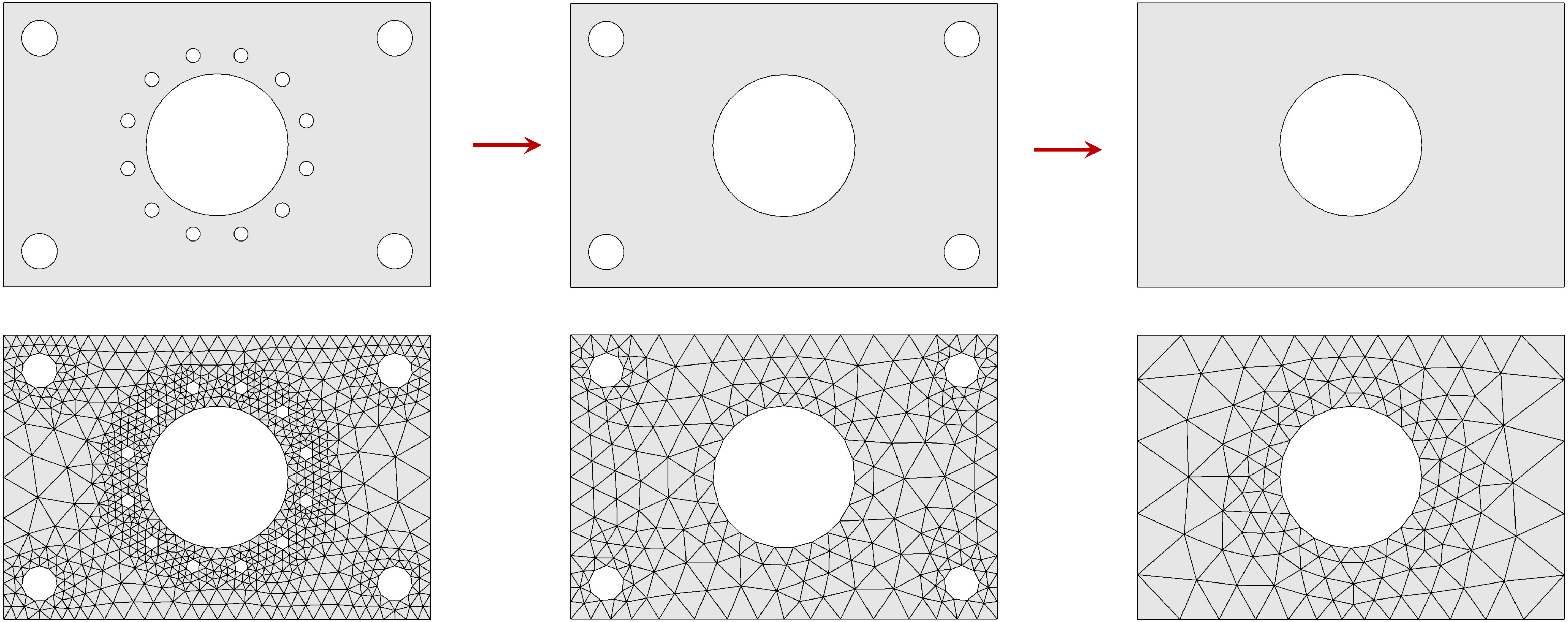
(Use F-bar methods for inf-sup stability.)

Calculation of $(\nabla\phi_I, \phi_K)$

- Currently solving for derivative projection using a sub-triangulation and FEA.
- Can also use Green's identities to calculate these if shape functions are harmonic.

Element-free discretizations

Motivation: Separate domain discretization from solution discretization



original domain

defeature

defeature

Impact of domain defeaturing? depends on goals of simulation

Motivation: Separate domain discretization from solution discretization

- Domain defeaturing is needed to control FEA discretization quality, size, and critical time step (explicit dynamics)
- Domain defeaturing typically requires human intervention (heuristics).
- For FEA, domain discretization and solution discretization are synonymous (isoparametric).
- Geometric features can require a fine local discretization while solution does not.
- Heuristics are often used in domain defeaturing and mesh design.
- Meshes are typically designed with goal in mind, thus making it cumbersome to reuse.
- Adaptivity requires going back to domain model (geometry).

A hybrid element-free approach

finite-element approach

- defeature domain geometry based on goals
- create a mesh based on goals
- mesh discretizes domain and solution
- quadrature of weak form is easy
- visualization of results using mesh
- adaptivity of mesh is hard

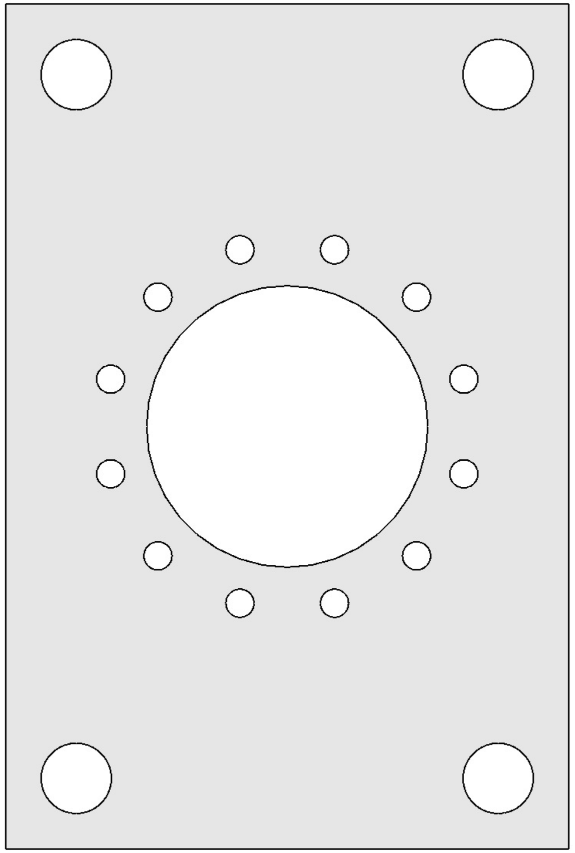
mesh-free approach

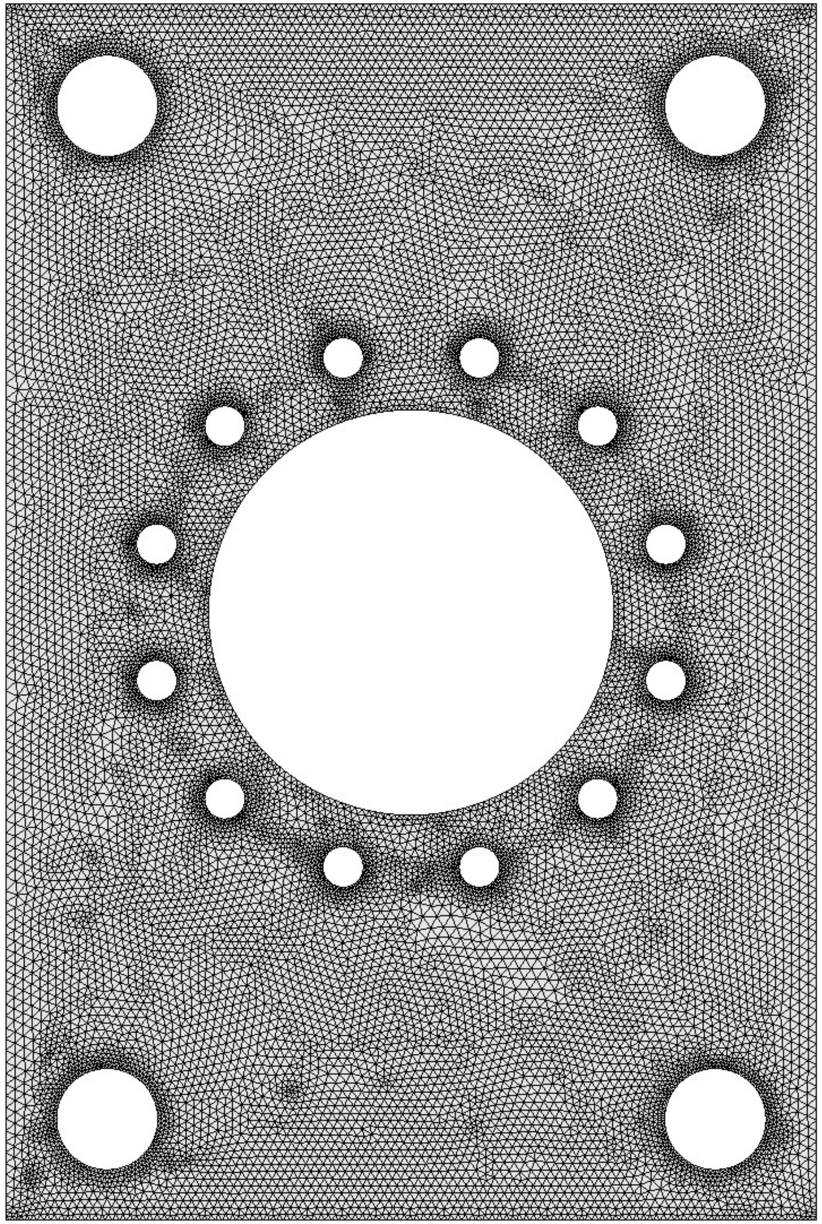
- no defeaturing of domain geometry
- no discretization of domain
- connectivity of domain is undefined (need computational geometry)
- quadrature of weak form is very hard
- visualization of results is cumbersome

Alternative hybrid approach: separate domain discretization and solution approximation using an element-free formulation.

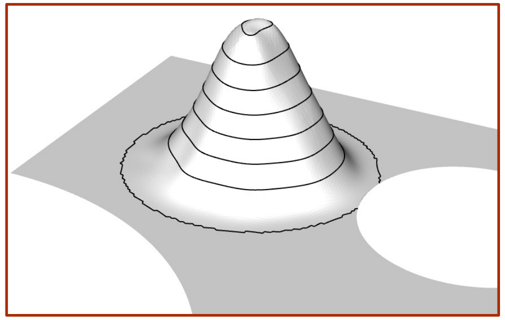
Hybrid approach: fine-scale triangulation

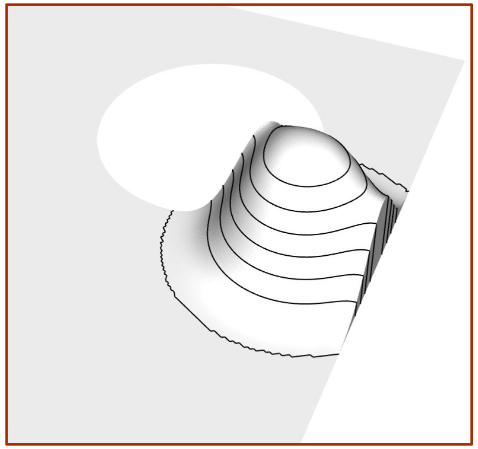
original domain





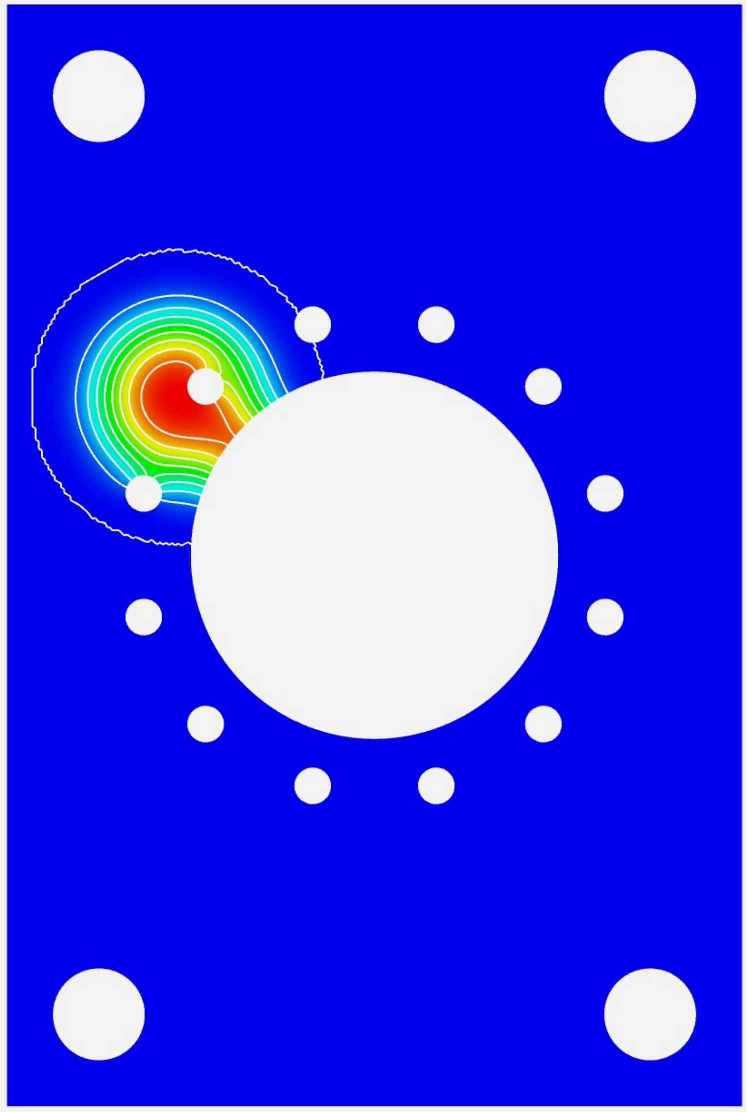
- Use a fine-scale triangulation to discretize domain.
- Define element-free basis using this triangulation.





Element-free basis functions

- Element-free basis functions automatically include geometric features at all scales.
- Solution discretization is separate from domain discretization.
- No need to defeature domain.



Hybrid element-free approach

- no defeaturing of domain
- discretize domain using fine-scale triangulation (a mesh, but poor quality is okay)
- use hp-cloud to define solution discretization (GBC, RK)
- use second hp-cloud to define quadrature and ensure coercivity
- projection of solution gradient to obtain polynomial consistency
- visualization of results using fine-scale mesh

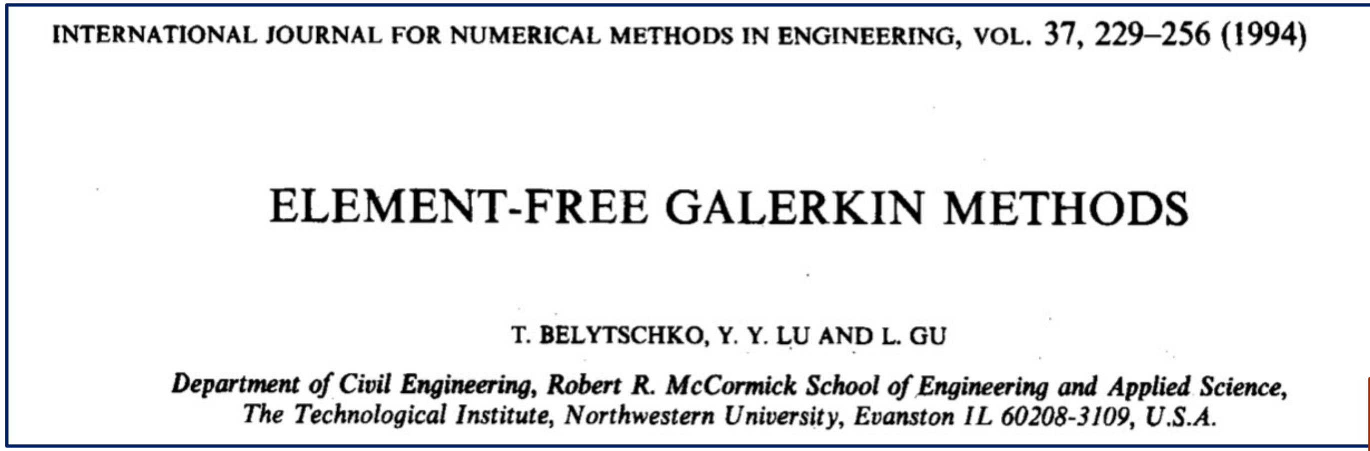
pros

- symmetric, Galerkin
- linear or nonlinear
- implicit or explicit dynamics
- can do higher order
- can do direct or mixed formulation
- adaptivity is seamless
- can use poor quality tet mesh
- adaptivity is facilitated
- should work for H(div) and H(curl) spaces
- reduced order modeling through coarse discretizations

cons

- constant material properties within a domain
- material interfaces: have to use weak enforcement such as mortar method
- less sparse

Hybrid element-free approach



INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 37, 229–256 (1994)

ELEMENT-FREE GALERKIN METHODS

T. BELYTSCHKO, Y. Y. LU AND L. GU

Department of Civil Engineering, Robert R. McCormick School of Engineering and Applied Science,
The Technological Institute, Northwestern University, Evanston IL 60208-3109, U.S.A.

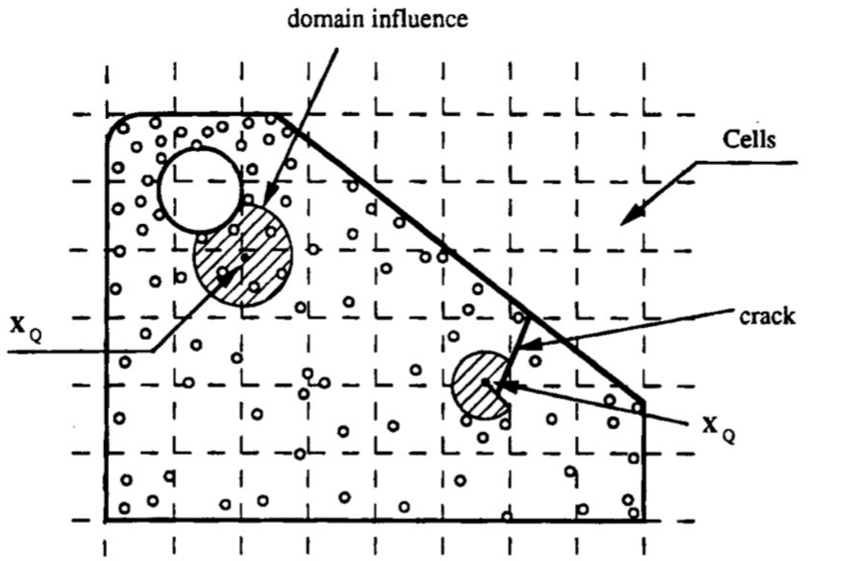


Figure 1. Cell structure for quadrature in EFGM and domains of quadrature point

Moving Least Squares (Reproducing Kernel)

The MLS shape functions $\phi_I(\mathbf{X})$ are defined as a spatial modulation of the nodal weight functions.

$$\phi_I(\mathbf{X}) = c_I(\mathbf{X}) w_I(\mathbf{X})$$

where the modulation function $c_I(\mathbf{X})$ is found through a least square minimization process resulting in

$$c_I(\mathbf{X}) = \mathbf{g}^{\mathrm{T}}(\mathbf{X})\mathbf{A}^{-1}(\mathbf{X})\mathbf{g}(\mathbf{X}_I)$$

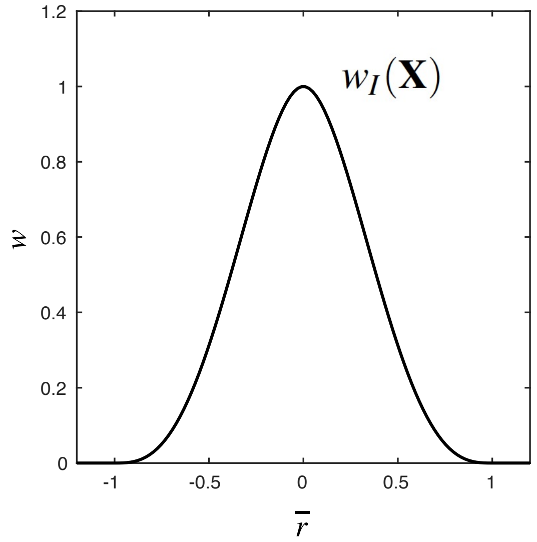
where

$$\mathbf{A}(\mathbf{X}) = \sum_{I \in \mathcal{N}} w_I(\mathbf{X})\mathbf{g}(\mathbf{X}_I)\mathbf{g}^{\mathrm{T}}(\mathbf{X}_I)$$ (sum over neighbors)

$$\mathbf{g}^{\mathrm{T}}(\mathbf{X}) = \{\, 1 \; X_1 \; X_2 \,\}$$ (linear reproducibility)

Note: shape function construction is algebraic.

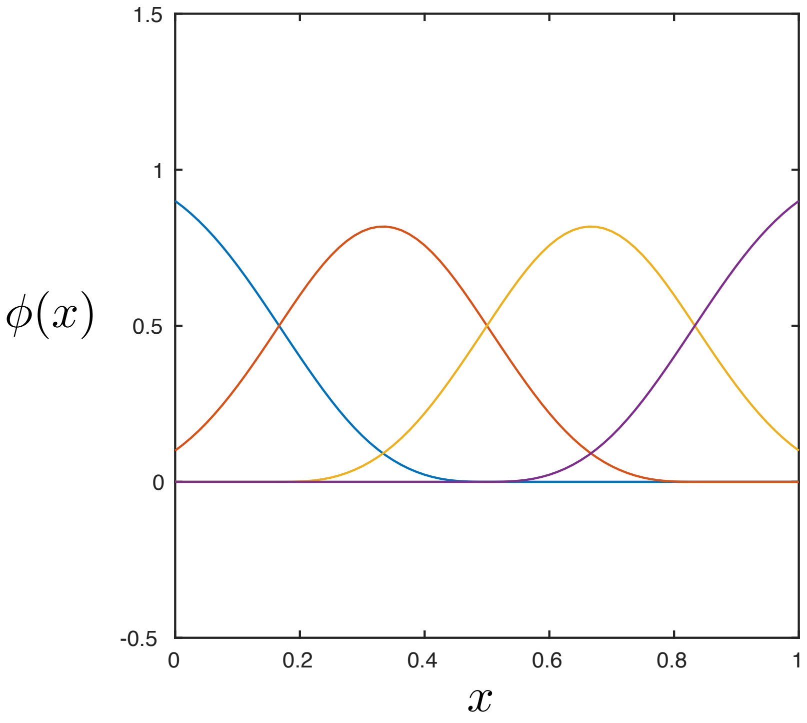
nodal weight function



circular or rectangular support

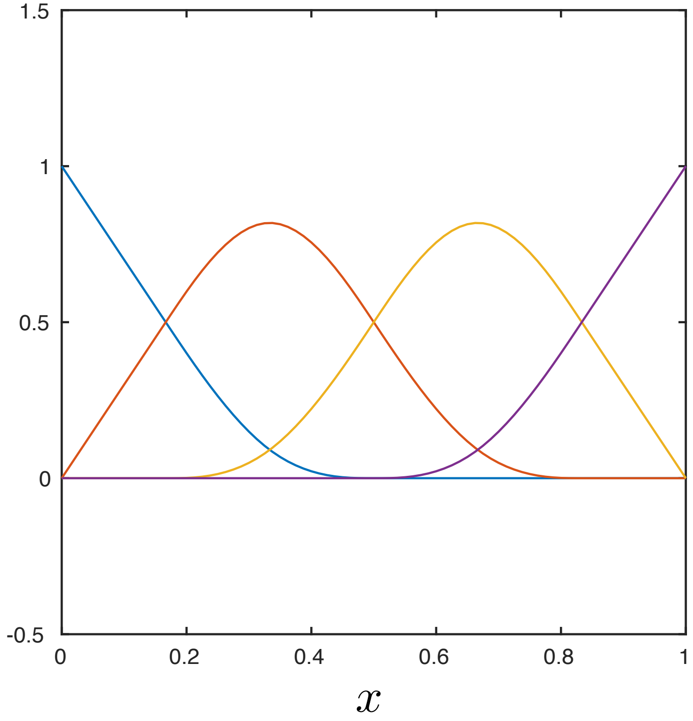
Moving Least Squares

$$\sum_K \phi_K(x) = 1$$



$$\sum_K \phi_K(x) = 1$$

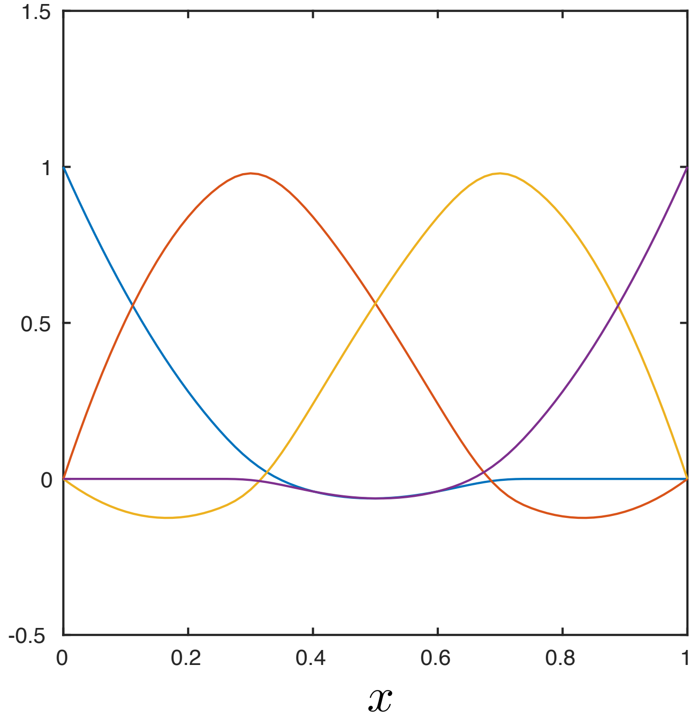
$$\sum_K x_K\,\phi_K(x) = x$$

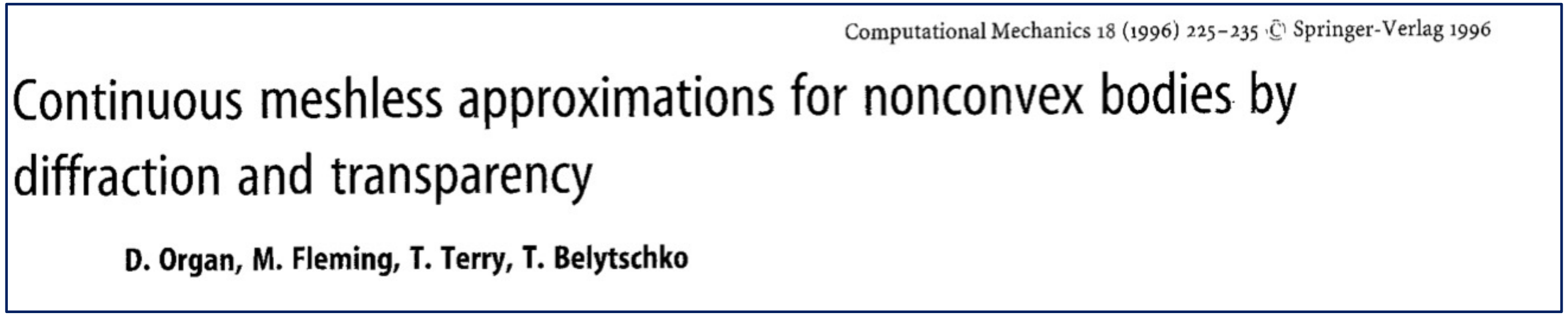


$$\sum_K \phi_K(x) = 1$$

$$\sum_K x_K\,\phi_K(x) = x$$

$$\sum_K x_K^2\,\phi_K(x) = x^2$$





Computational Mechanics 18 (1996) 225–235 © Springer-Verlag 1996

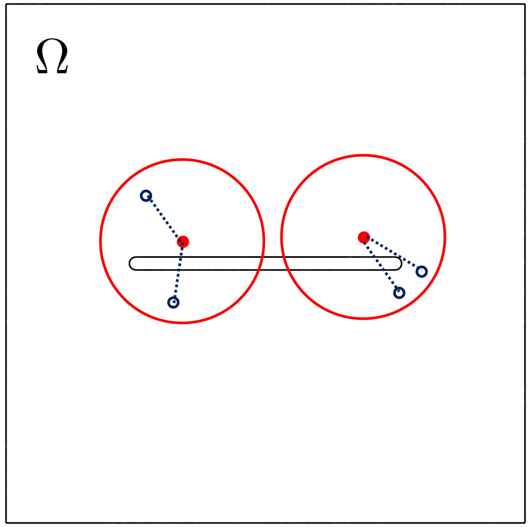
Continuous meshless approximations for nonconvex bodies by diffraction and transparency

D. Organ, M. Fleming, T. Terry, T. Belytschko

visibility criterion

visibility criterion

domain with slot



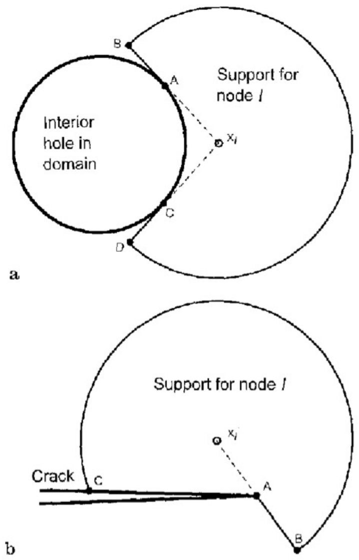


Fig. 1a,b. Domains of influence near nonconvex boundaries using the *visibility criterion* a Supportnear a hole, b Support near a crack tip

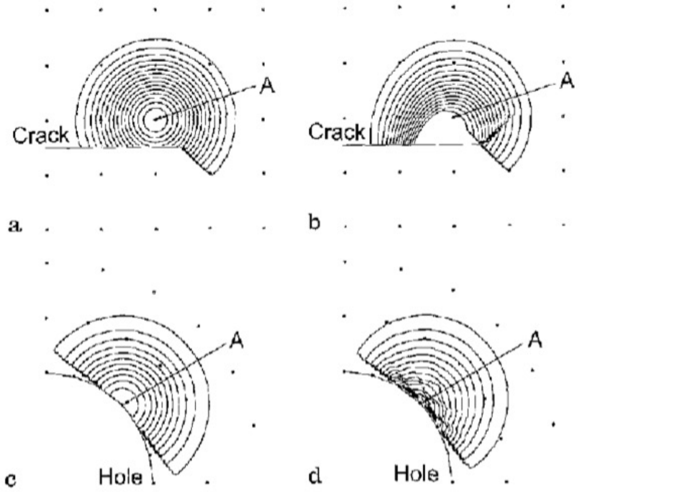
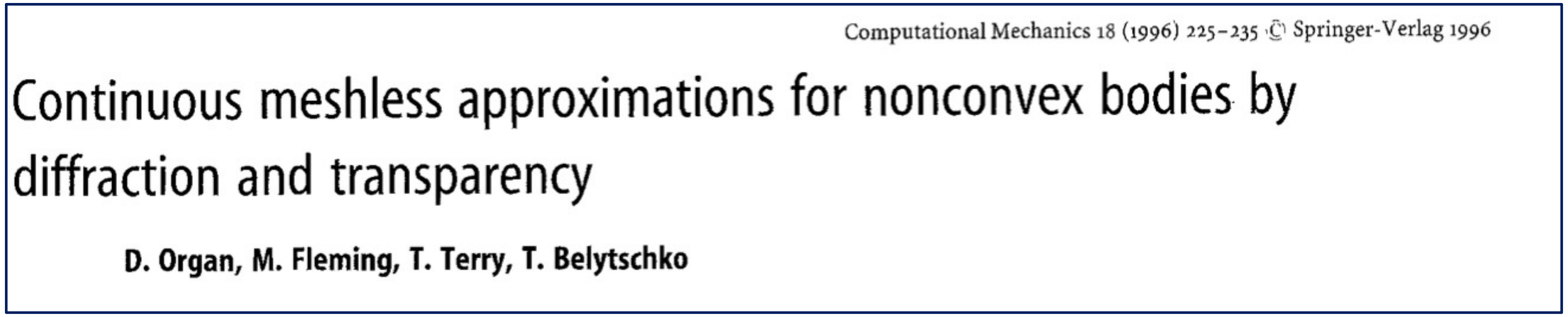


Fig. 2a–d. Contours for weight and shape functions associated with node A constructed using the *visibility criterion*. a Weight function near a crack tip, b Shape function near a crack tip, c Weight function near a hole, d Shape function near a hole



Computational Mechanics 18 (1996) 225–235 © Springer-Verlag 1996

Continuous meshless approximations for nonconvex bodies by diffraction and transparency

D. Organ, M. Fleming, T. Terry, T. Belytschko

diffraction method

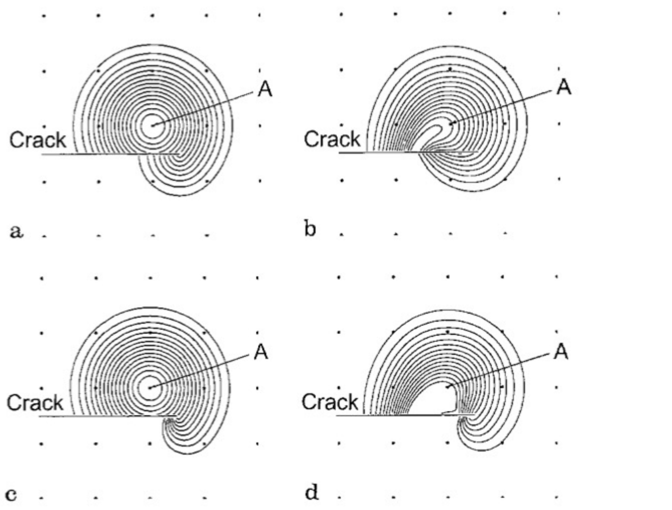


Fig. 5a–d. Contours for weight and shape functions associated with node A near a crack tip constructed using the *diffraction method*. The quartic weight function in (2.18b) was used with $d_{max} = 2.01$ **a** Weight function for $\lambda = 1$, **b** Shape function for $\lambda = 1$, **c** Weight function for $\lambda = 2$, **d** Shape function for $\lambda = 2$

transparancy method

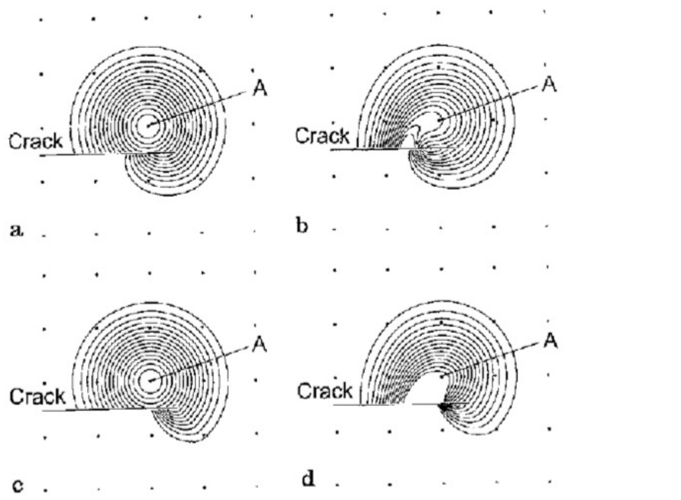
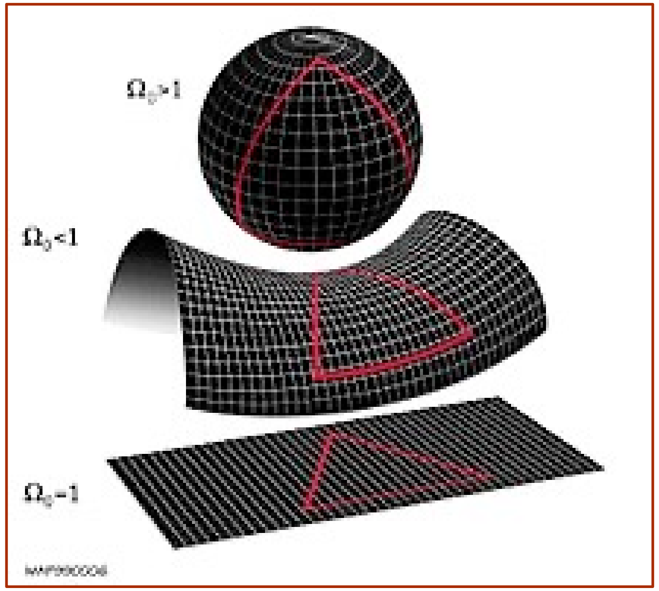


Fig. 8a–d. Contours for weight and shape functions associated with node A near a crack tip constructed using the *transparency method*. The quartic weight function in (2.18b) was used with $d_{max} = 2.01$. **a** Weight function for $\kappa = 1.0$, **b** Shape function for $\kappa = 1.0$, **c** Weight function for $\kappa = 0.5$, **d** Shape function for $\kappa = 0.5$

All these methods (visibility, transparency, diffraction) require use of computational geometry.

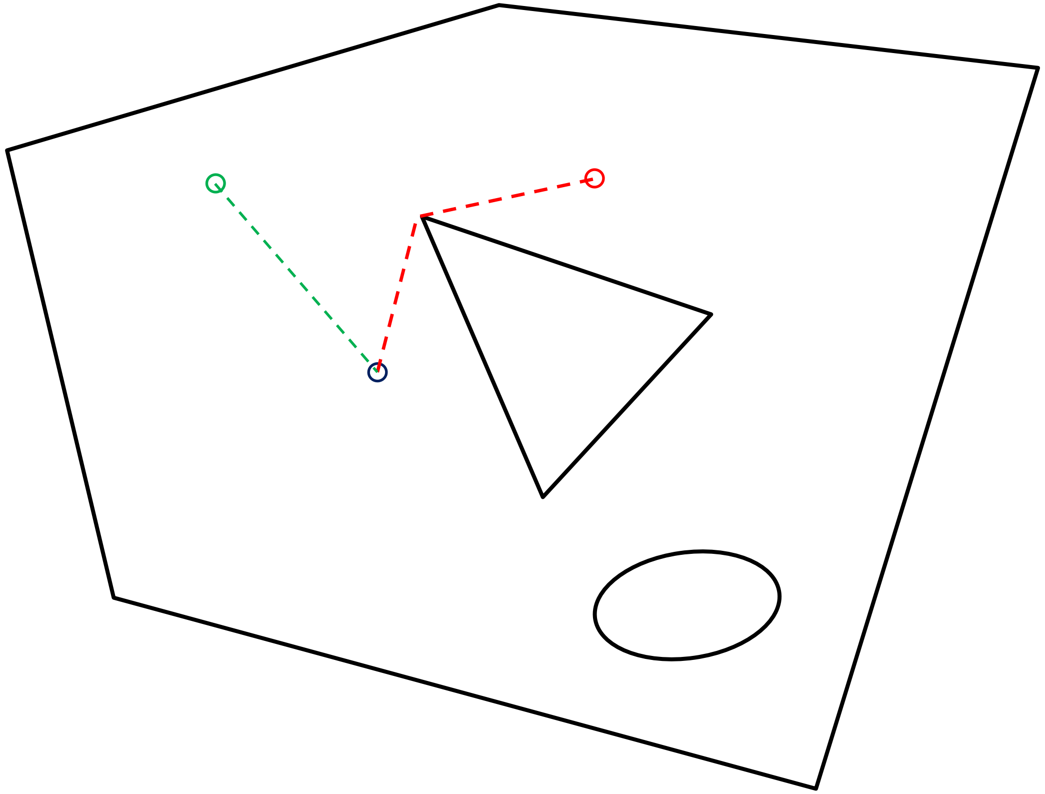
Manifold geodesic

Geodesic: path that provides the shortest distance along a manifold



https://en.wikipedia.org/wiki/Geodesic

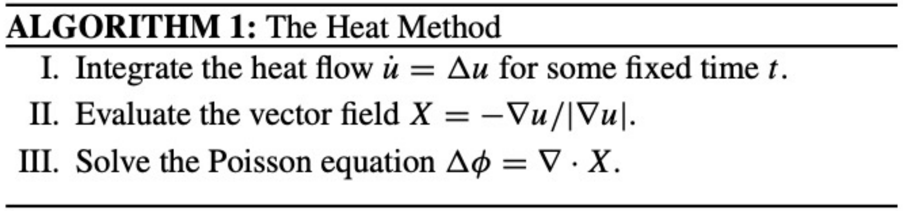
Euclidean manifold with boundary



Geodesics in Heat: A New Approach to Computing Distance Based on Heat Flow

KEENAN CRANE
Caltech
and
CLARISSE WEISCHEDEL and MAX WARDETZKY,
University of Göttingen

ACM Trans. Graph. 2013 Vol. 32 Issue 5 Pages Article 152



ALGORITHM 1: The Heat Method

```
  I. Integrate the heat flow $\dot{u} = \Delta u$ for some fixed time $t$.
 II. Evaluate the vector field $X = -\nabla u / |\nabla u|$.
III. Solve the Poisson equation $\Delta \phi = \nabla \cdot X$.
```

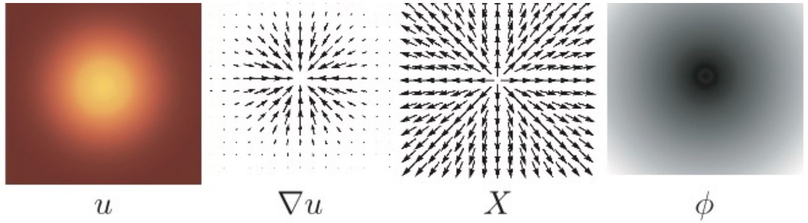


Fig. 5. Outline of the heat method. (I) Heat u is allowed to diffuse for a brief period of time *(left)*. (II) The temperature gradient ∇u *(center left)* is normalized and negated to get a unit vector field X *(center right)* pointing along geodesics. (III) A function ϕ whose gradient follows X recovers the final distance *(right)*.

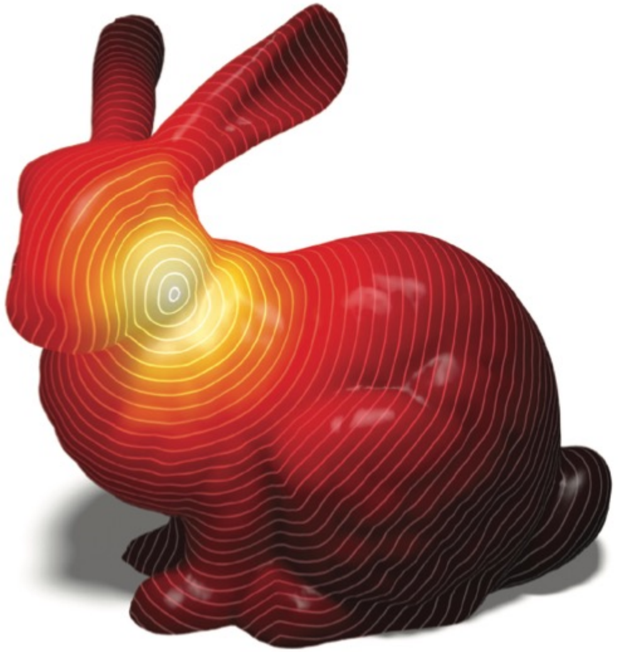
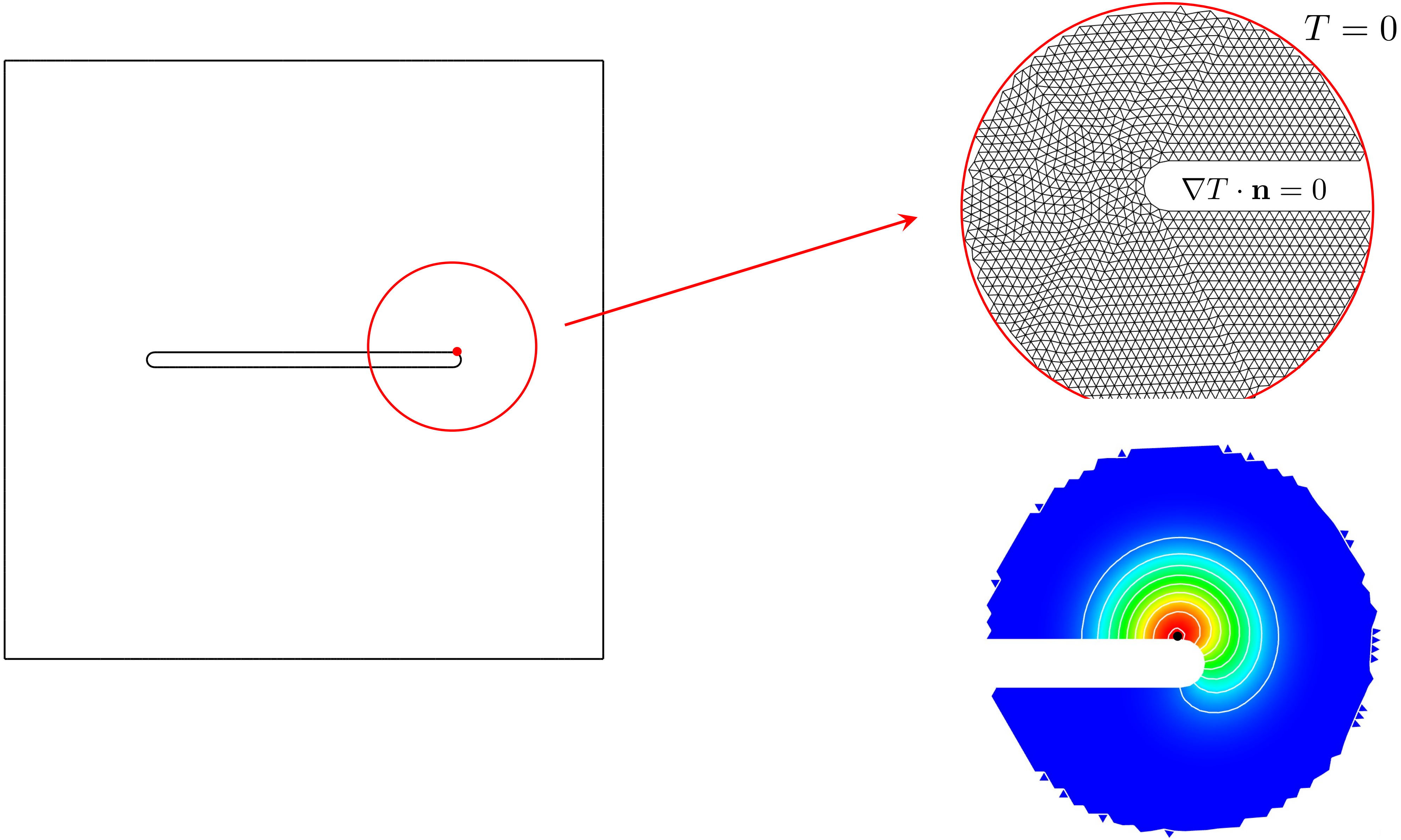
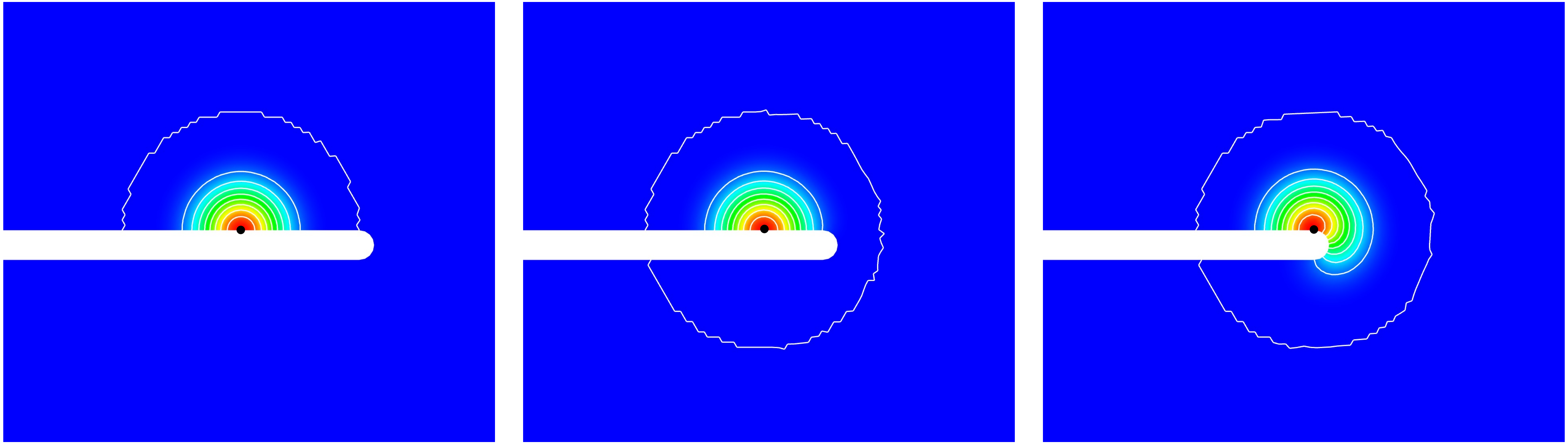


Fig. 1. Geodesic distance from a single point on a surface. The heat method allows distance to be rapidly updated for new source points or curves.

Weight functions using heat flow

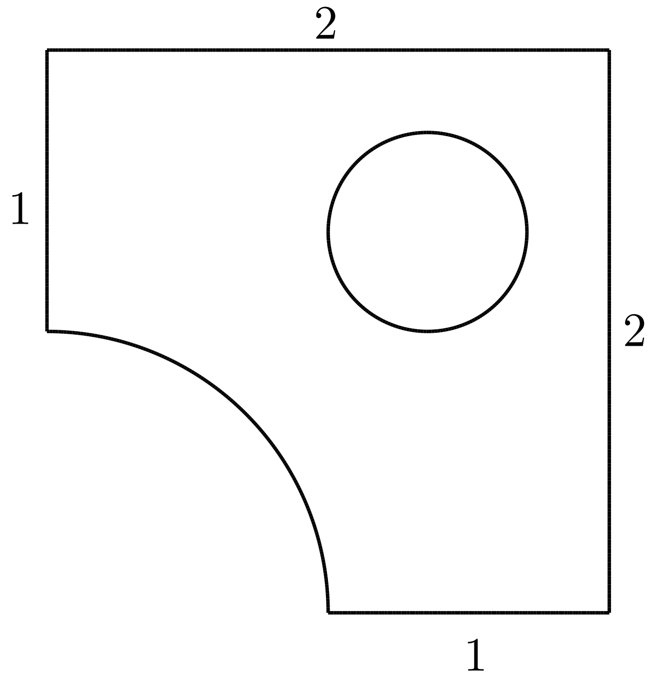


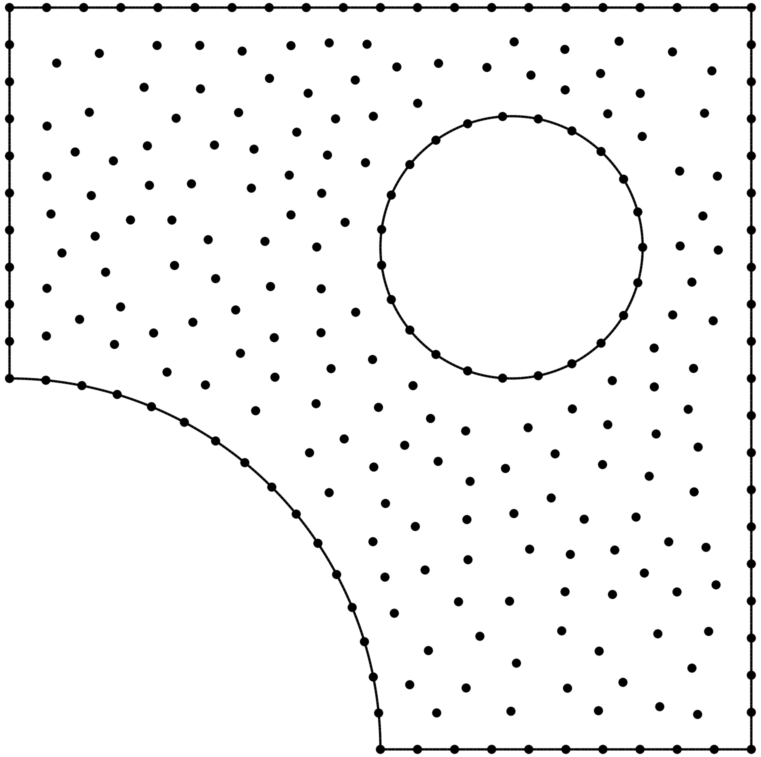
Weight functions using heat flow



"Point" placement

- uniform on boundary
- random close packing on interior (maximal Poisson sampling)



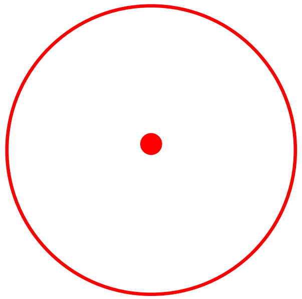


packing size:

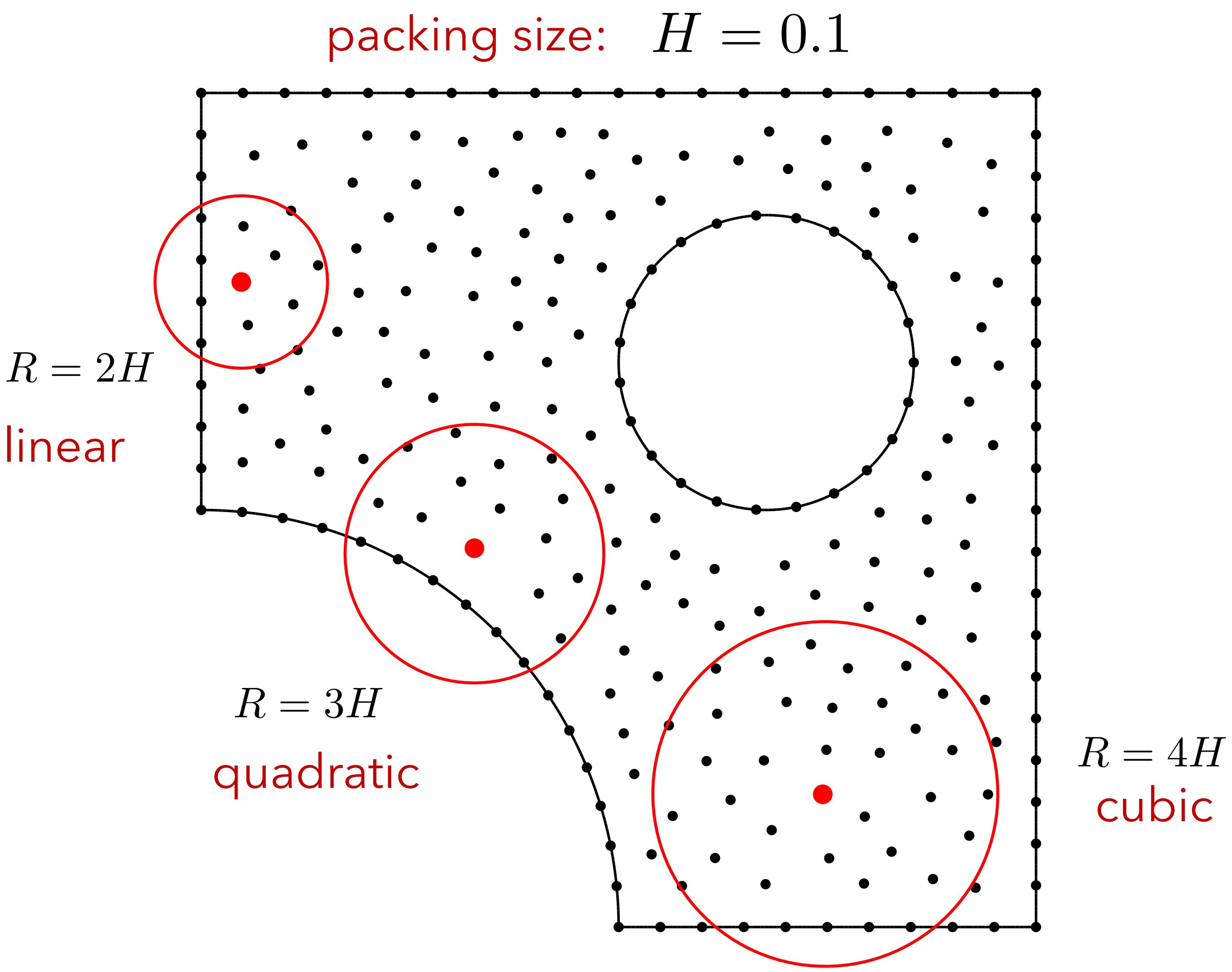
$H = 0.1$

Weight function support size

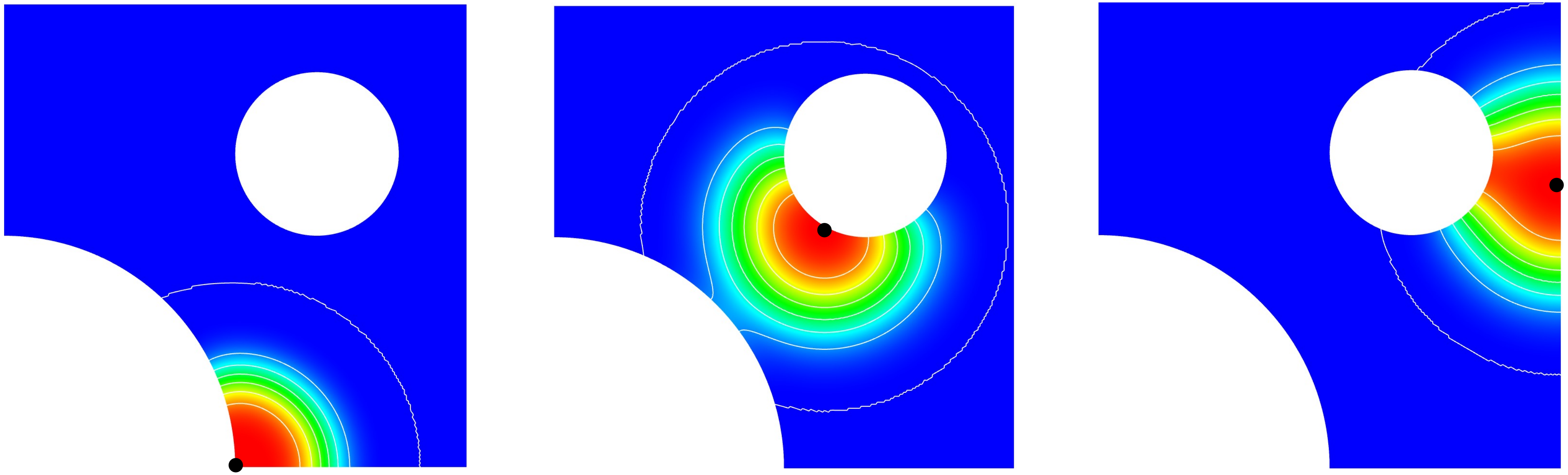
support size



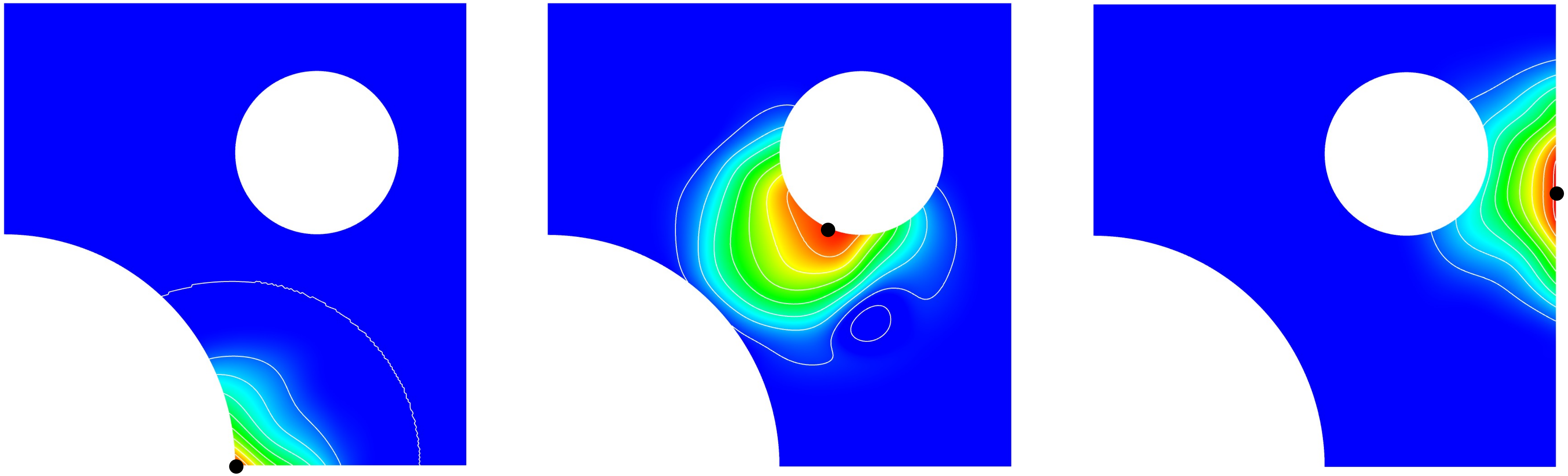
encloses underlying tri mesh



weight functions



shape functions (basis)

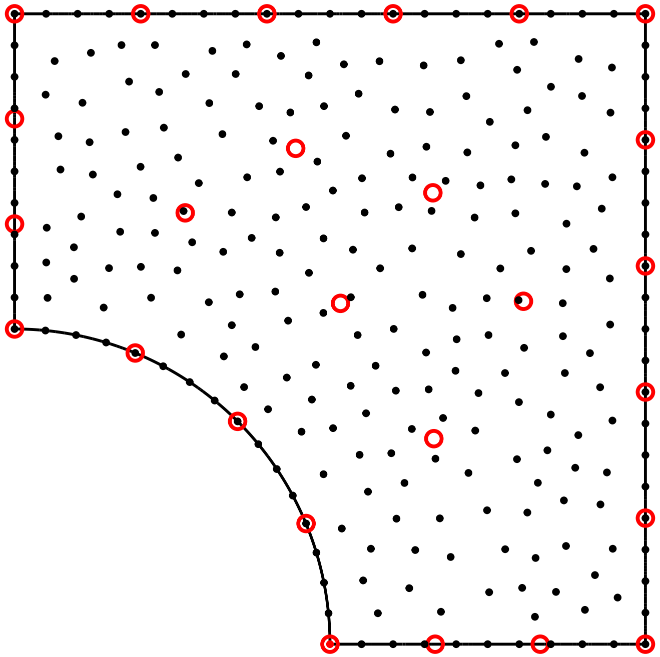


Element-free approach to solve BVPs

Use two meshfree clouds: one for solution discretization (DoF) and one for quadrature.

○ DoF node

● quadrature nodes

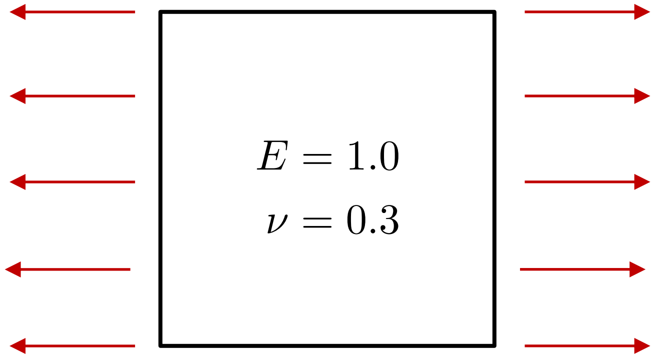


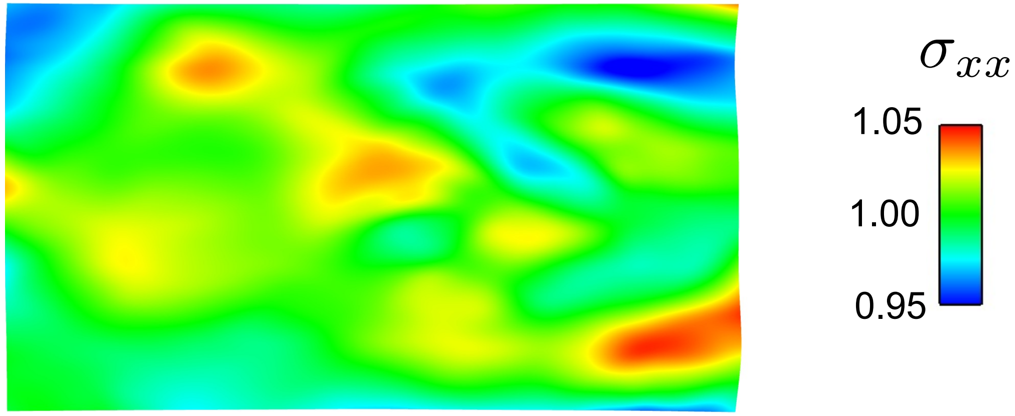
quad-to-dof ratio = 4^2

What ratio of quad nodes to dof nodes is needed for stability (coercivity of bilinear form)?

Patch test (linear consistency)

uniaxial tension





error > 5%

Consistency of discrete form (integration)

- For convergence of discrete approximation, need to ensure consistency of discrete and continuous bilinear forms.
- Requires polynomial consistency of shape-function gradients (including quadrature).
- To obtain quadrature consistency, project the DoF shape function gradients to the subspace of quadrature shape functions.
- Only performed once in a pre-processing step.

$\{\phi_I, I = 1, \ldots, N\}$ *DoF basis (shape functions)*

$\{\Phi_K, K = 1, \ldots, M\}$ *Quadrature basis (shape functions)*

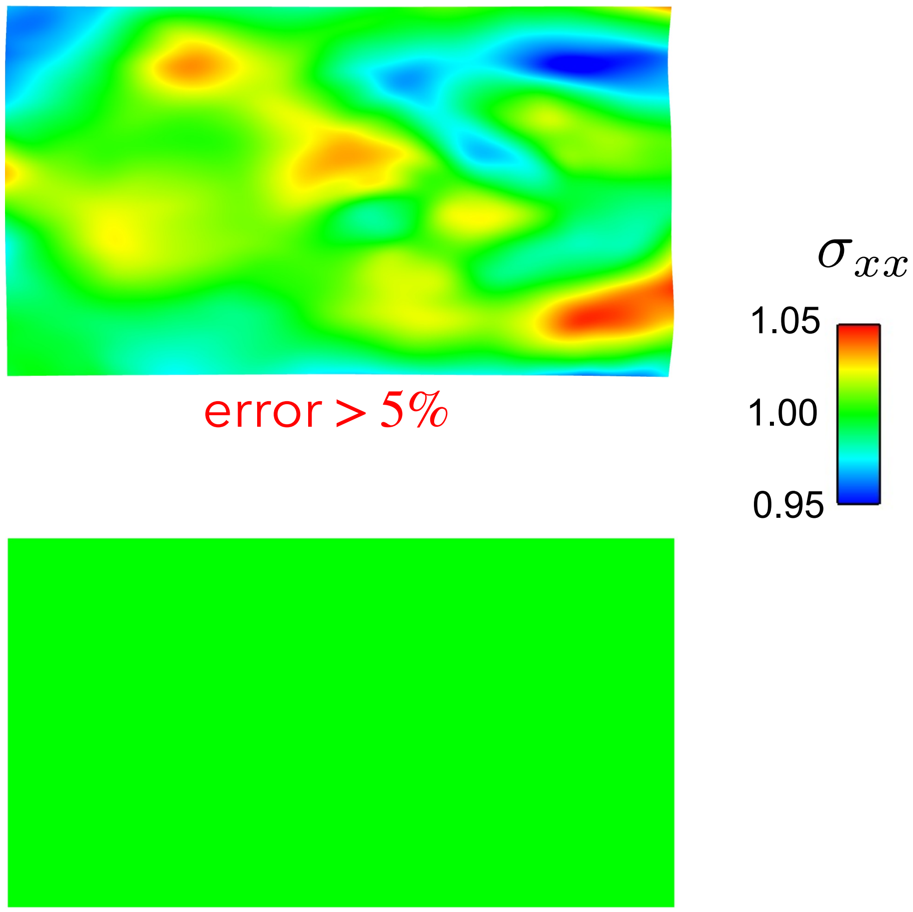
$$\overline{\nabla}\phi_I := \arg\min \int_\Omega \left(\nabla\phi_I - \sum_{K=1}^{M} a^K \Phi_K \right)^2 d\Omega$$ *(L_2 projection)*

Patch test (linear consistency)

uniaxial tension

no projection

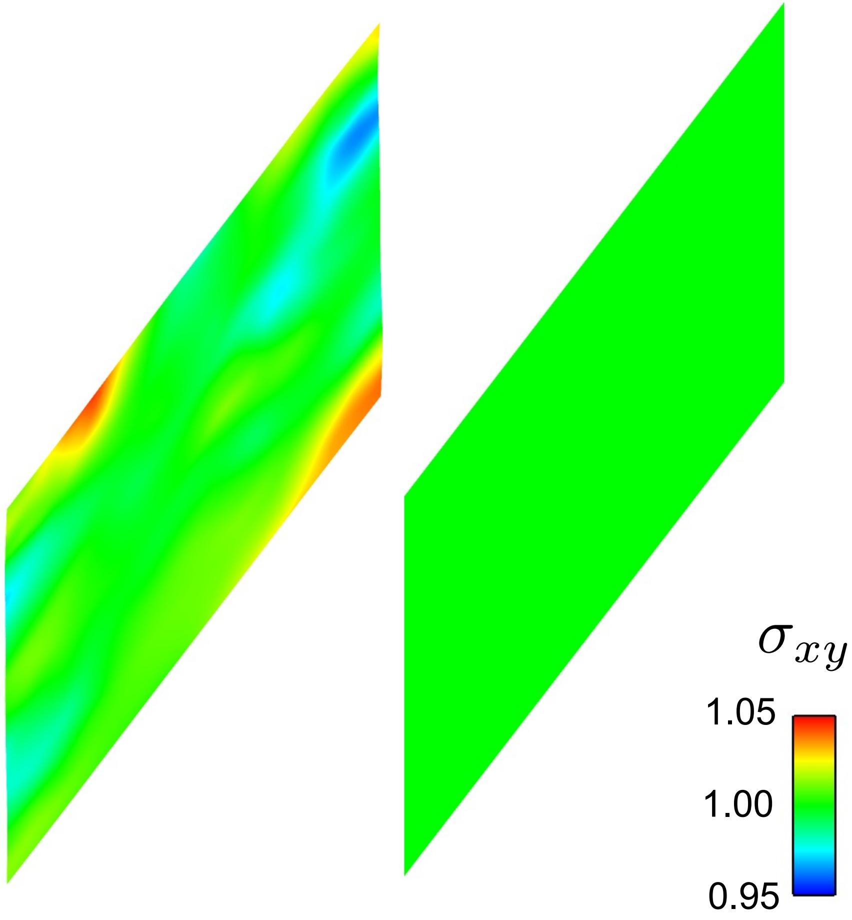
with projection



error > 5%

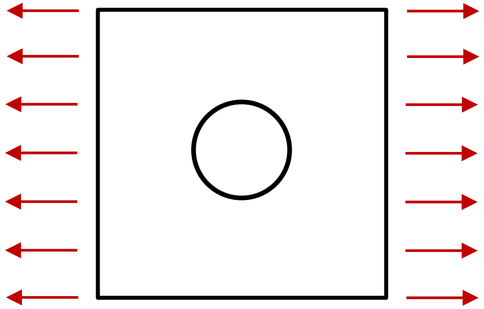
error < 10^{-13}

pure shear



Example: plate with hole

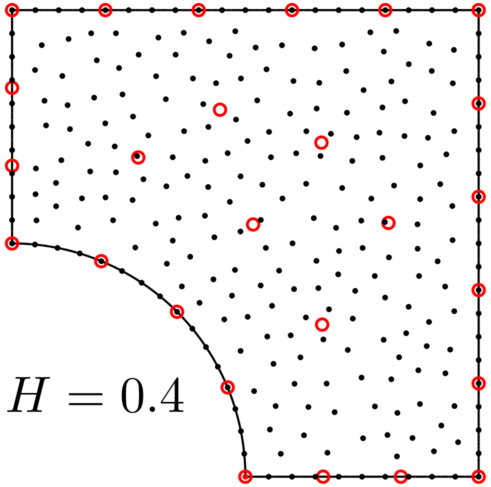
uniaxial tension



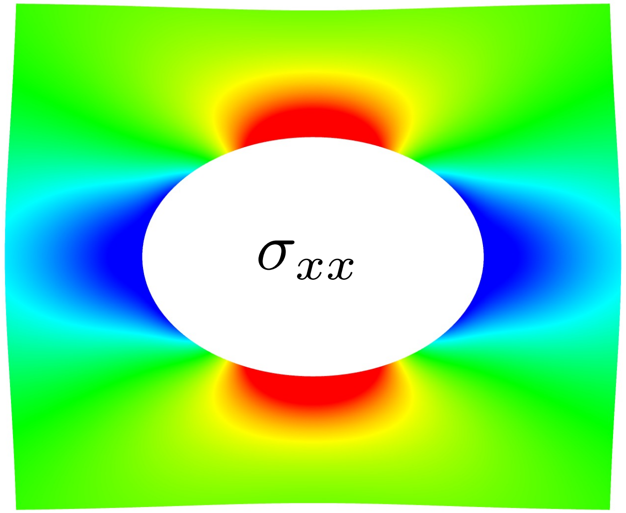
$E = 1.0$

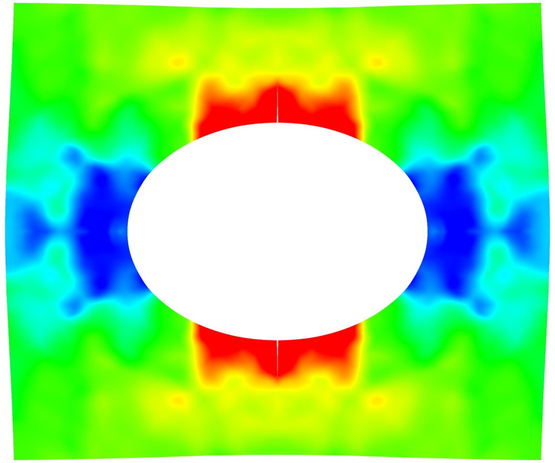
$\nu = 0.3$

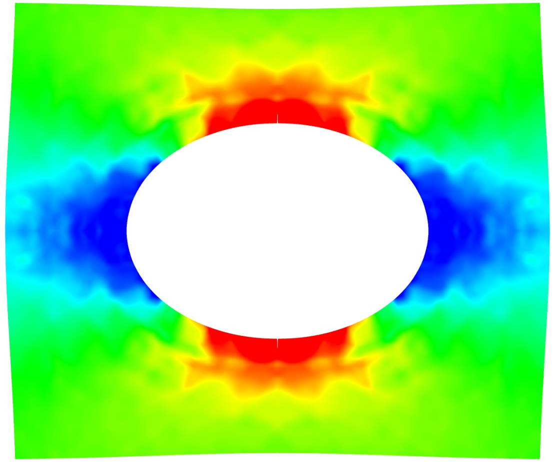
- plane strain
- quarter symmetry

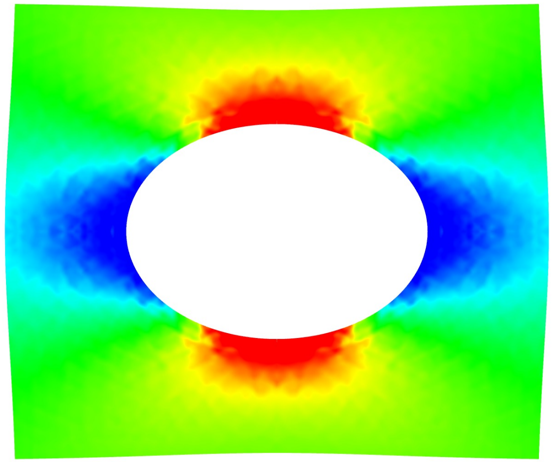


exact

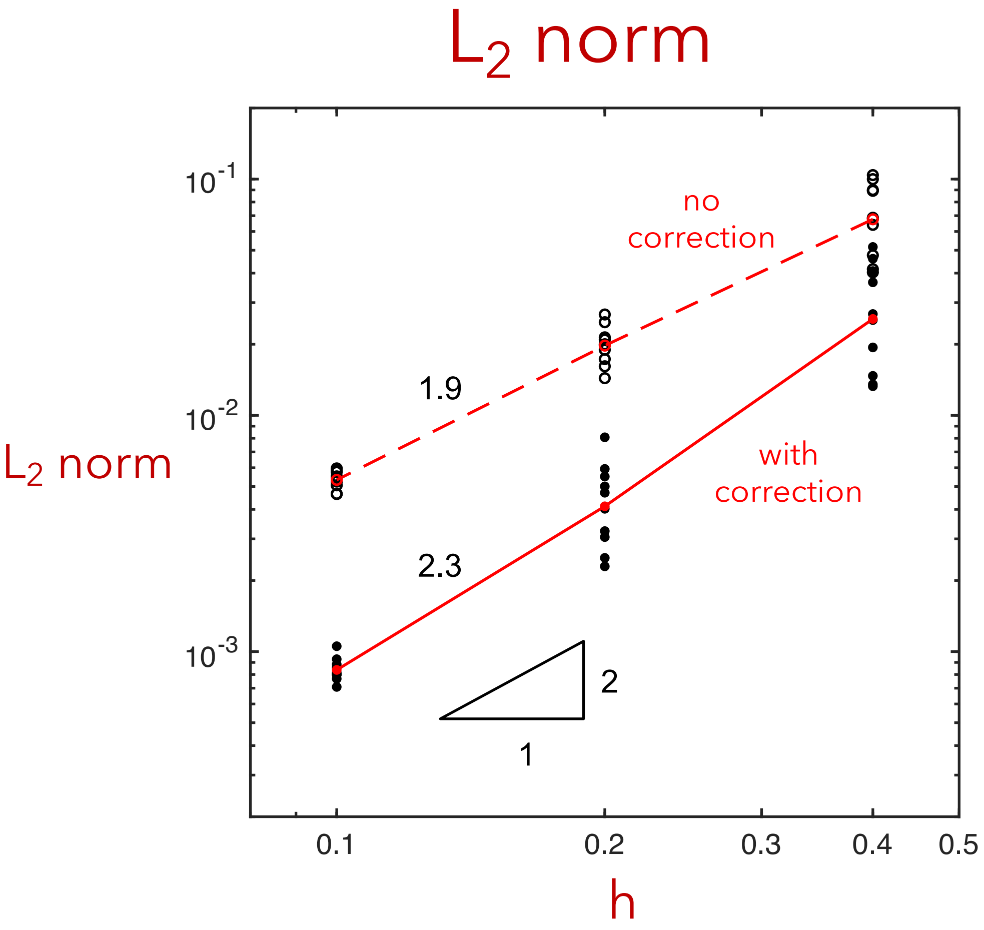


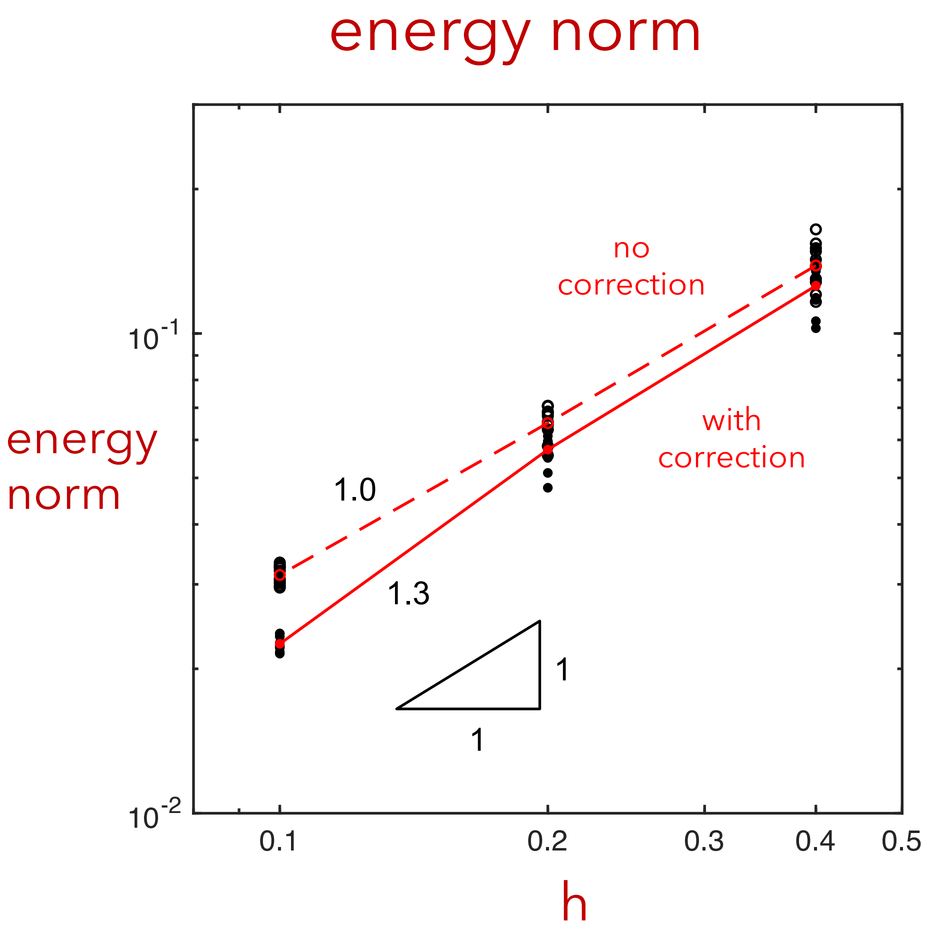




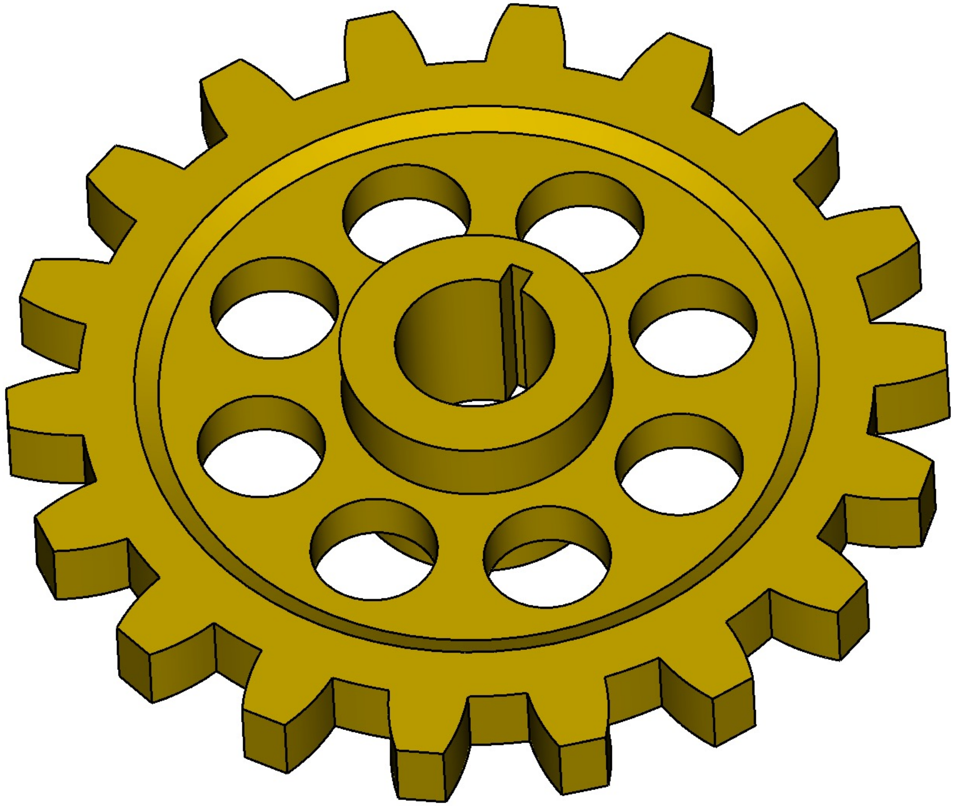


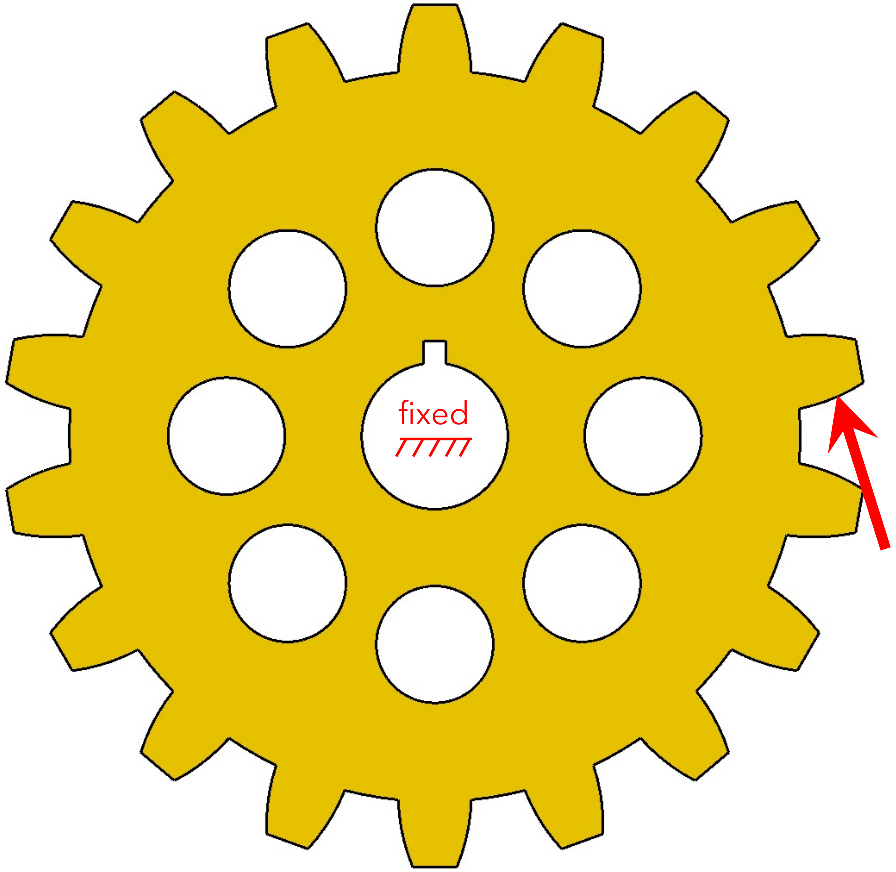
Example: plate with hole



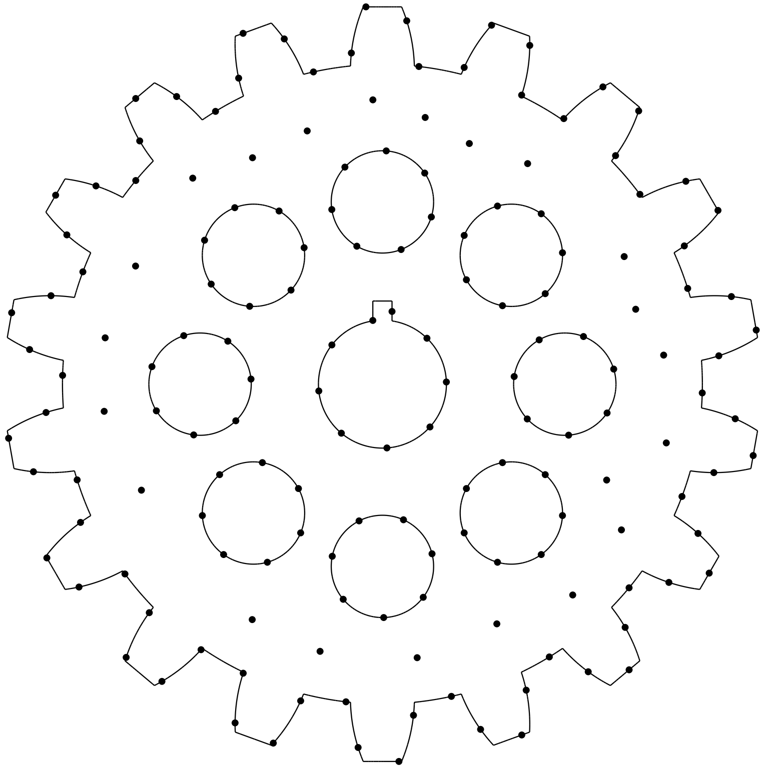


Example

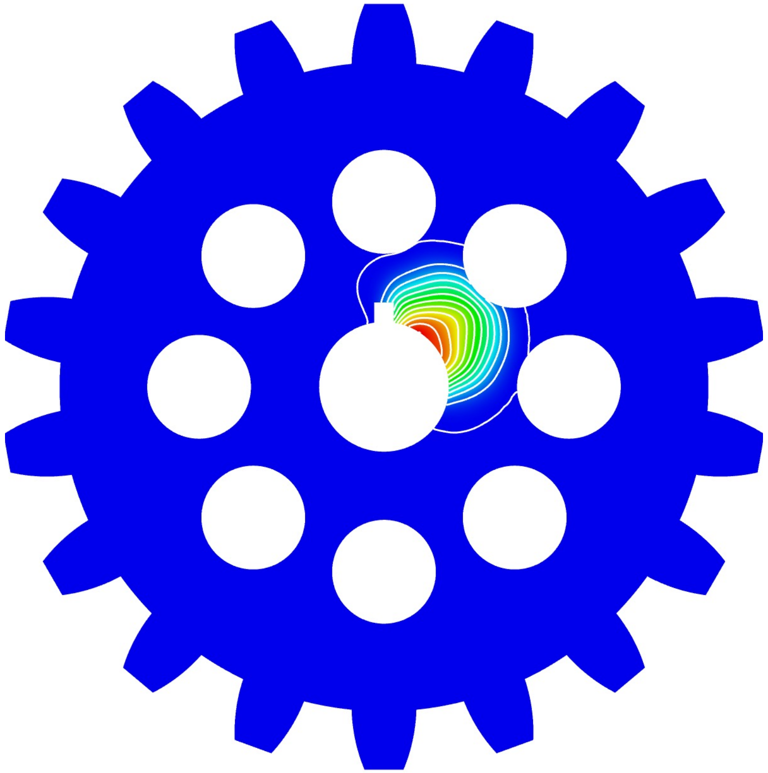




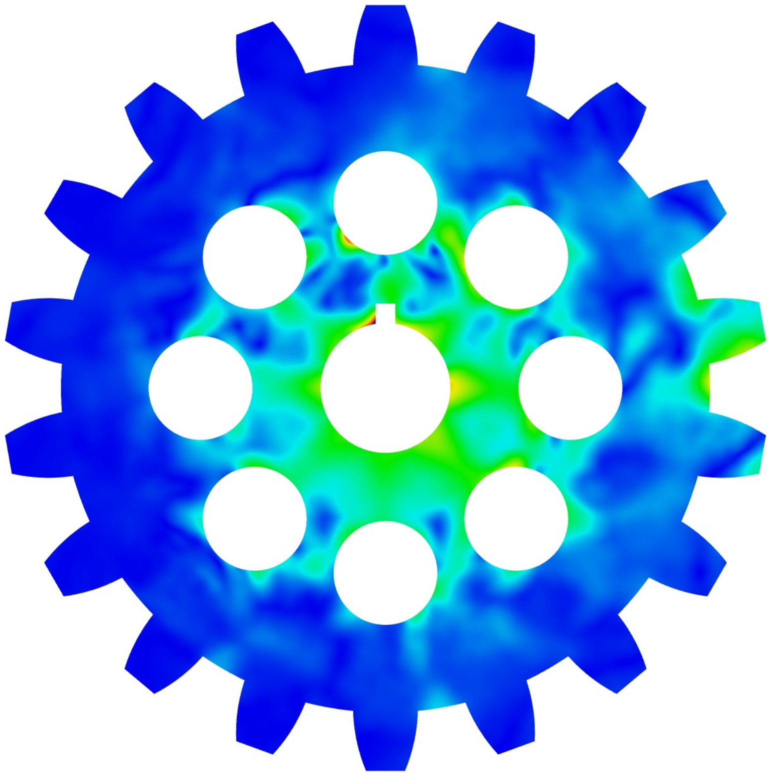
dof nodes

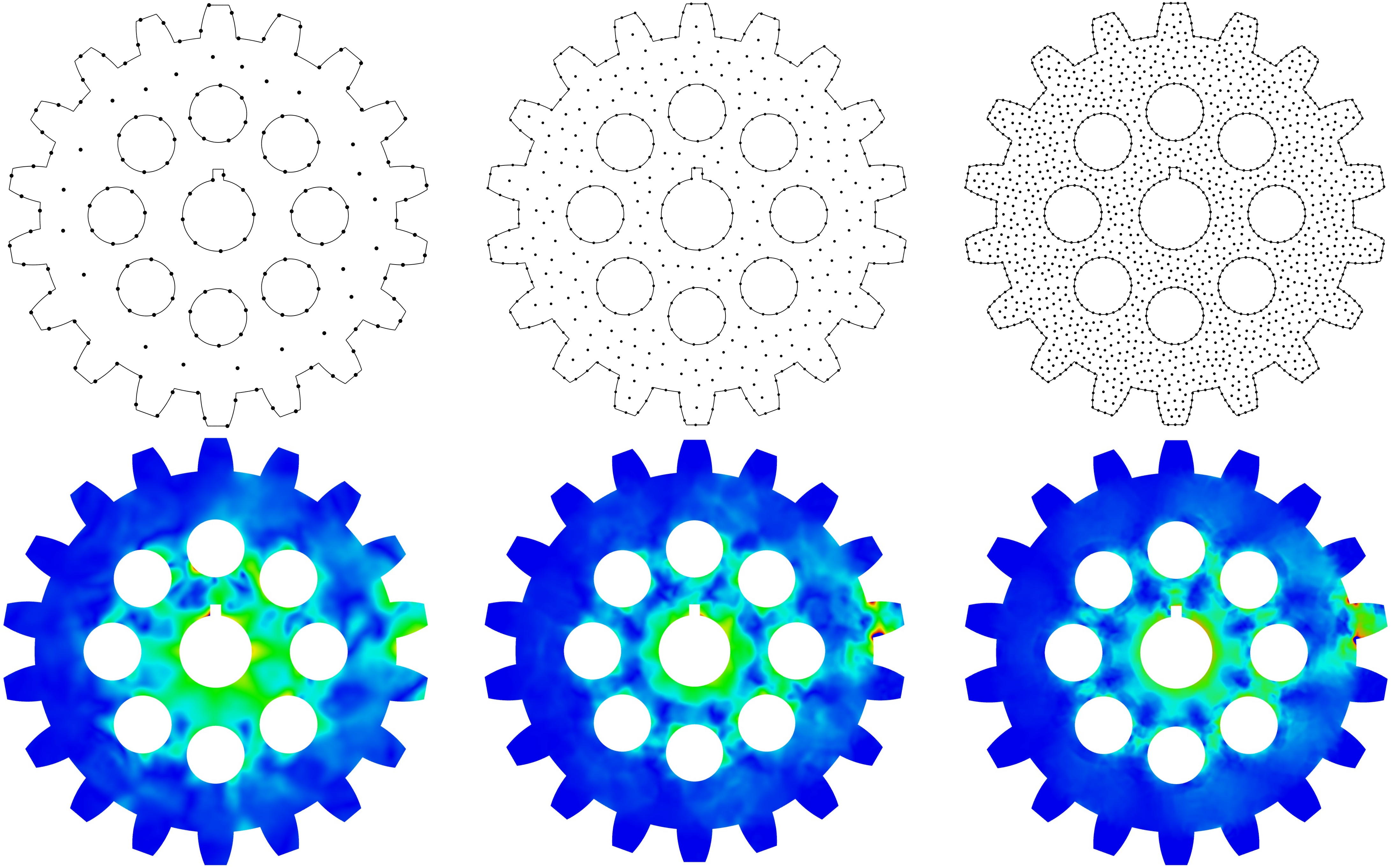


basis functions



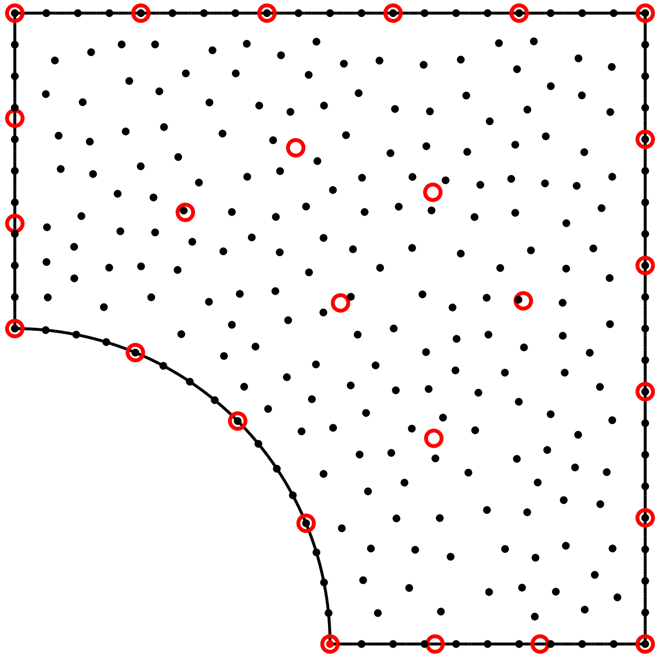
stress field (vm)





Nearly incompressible limit

- Can extend approach to handle nearly incompressible materials
- Use a “generalized” B-Bar/F-bar approach.
- Project dilatational portion of deformation gradient to smaller subspace, e.g. use original DOF points as quadrature basis.



○ DoF node and dilatational quadrature node

● deviatoric quadrature node

Summary

1. Showed applications of GBCs to both element-based and element-free PDE discretizations
2. GBCs enable formulation of a diverse set of polyhedral discretizations.
3. GBCs enable element-free discretization on complex disconnected domains without resource to computational geometry.
4. Element-free weight function used manifold geodesic (heat map)
5. GBCs also induce quadrature schemes for both element-based and element-free methods.
6. GBCs facilitate gradient projection schemes for PDE consistency.
7. Applications to multiresolution (wavelet) on complex shapes?