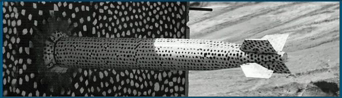
Adventures in statistical thermodynamics, hexahedral meshing, and mechanics tools



Michael Buche[1,*], Chad Hovey[1], Alvin Chen[1], Scott Grutzik[1], John Emery[1], Meredith Silberstein[2], Matthew Grasinger[3], Jason Mulderrig[3]

Air Force Research Laboratory – Wright-Patterson Air Force Base

[1]Sandia National Laboratories, [2]Cornell University, [3]AFRL

*mrbuche@sandia.gov

4/17/2026

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND NO. 2026-19924C

Abstract

Statistical thermodynamics

- Asymptotic theory for steep interaction potentials.
- So far, most applications in polymers, some in solid mechanics.
- Many ideas in progress, desire to break into computational physics.

Hexahedral meshing

- Rapid hexahedral mesh generation with adaptivity and conformity.
- Variety of other features for segmentation, meshes, and geometry.

Mechanics tools

- Modular constitutive model creation using theoretical rules.
- Simultaneous time integration of internal state variables.

Statistical thermodynamics

Introduction

Statistical mechanics:

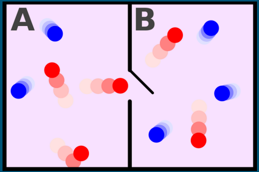
- Probabilistic interpretation of mechanics through $f(p, q, t)$.
- State variables are all atomic positions/momenta, time.

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \sum_{j=1}^{3N} \left(\frac{\partial f}{\partial q_j} \dot{q}_j + \frac{\partial f}{\partial p_j} \dot{p}_j \right) = 0$$

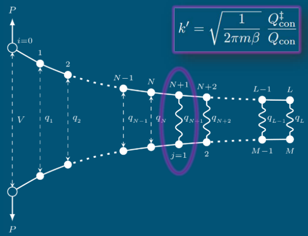
- Fundamentally correct, but can be extremely unwieldy.

Statistical thermodynamics:

- Statistical features do not evolve in time (equilibrium).
- Severely reduced number of state variables (ensemble).
- Macroscopic thermodynamics from constituent particles.
- Only a few axioms and equations, but a lot of examples.



Wikipedia



Statistical thermodynamics

Partition functions:

- Probability normalization for all calculations.
- Compute once, if possible, for all states.
- Configuration integral is typically impossible.
- Connection to thermodynamics by inference.
- Laplace transforms change the ensemble.

$$Q(N,V,T) = \frac{1}{N!h^{3N}} \int \cdots \int e^{-H(p,q)/kT}\, dp\, dq = \frac{1}{N!} \left(\frac{2\pi mkT}{h^2} \right)^{3N/2} Z(N,V,T)$$

$$Z(N,V,T) = \int \cdots \int e^{-U(q)/kT}\, dq$$

Difference from macroscopic thermodynamics:

- Non-state variables are averages and fluctuate.
- Ensemble-dependent results for small systems.
- Calculate averages of molecular variables.
- Things like temperature become nebulous.

$$A = -kT \ln Q(N,V,T)$$

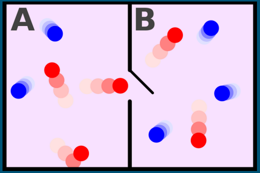
$$\langle x \rangle = \frac{1}{Q} \int \cdots \int x(q)\, e^{-U(q)/kT}\, dq$$

$$P = -\left. \frac{\partial A}{\partial V} \right|_{N,T}$$

Statistical thermodynamics

Fundamental axioms [1]:

- Principle of equal *a priori* probabilities.
- The entropy is maximized at equilibrium.
- The entropy takes a specific form.
- Gibbs' postulate, (in)distinguishability of particles.



Wikipedia

Two approximation techniques:

- If $U = U_0 + U_1$, where U_1 is weak ($U_1 \ll kT$) [2], i.e. derive van der Waals.

$$A \sim A_0 + \langle U_1 \rangle_0 - \frac{1}{2kT} \left[\langle U_1^2 \rangle_0 - \langle U_1 \rangle_0^2 \right] + \cdots$$

- If $U = U_0 + U_1$, where U_1 is steep ($U_1 \gg kT$ and narrow) [3], i.e. correct RRHO.

$$A \sim A_0 + U_1\big|_0 + kT \left[\left(\frac{A_0'}{kT} \right)^2 - \frac{A_0''}{kT} + \cdots \right]_0 + \cdots$$

Asymptotic theory

General partition function with a special set of degrees of freedom and related potential.

$$Z = \int d\Gamma_0 \int dX \, e^{-\beta H_0(\Gamma_0, X)} \, e^{-\beta U_1(X)}$$

Separate contributions in nondimensional form, where $\phi'(\hat{x}_i) = 0$ and $\phi''(\hat{x}_i) = 1$.

$$\beta U_1(X) = \sum_{i=1}^{N} \kappa_i \phi_i(x_i)$$

For steep ($\kappa_i \gg 1$) potentials, an asymptotic approximation upon the reference system [3].

$$Z \sim \left(\prod_{i=1}^{N} \sqrt{\frac{2\pi}{\kappa_i}} \right) \left[Z_0(\hat{X}) + \sum_{j=1}^{N} \frac{g_j(\hat{x}_j)}{\kappa_j} + \mathrm{ord}\left(\kappa_j^{-2}\right) \right]$$

Corrections for potential nonlinearity and configurational-vibrational coupling are included.

Extensible freely jointed chains

Freely jointed chains with extensible links [4, 5].

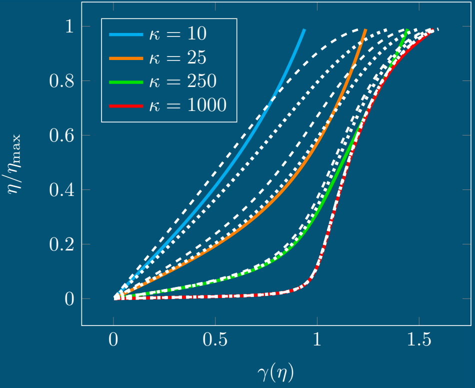
- Ensemble is links N_b, force f, temperature T.
- Resistance due to entropy and link stretching.
- Analytic relations using asymptotic approach.

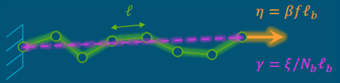
$$\gamma(\eta) \sim \mathcal{L}(\eta) + \frac{\eta}{\kappa}\left[\frac{1 - \mathcal{L}(\eta)\coth(\eta)}{c + (\eta/\kappa)\coth(\eta)}\right] + \Delta\lambda(\eta)$$

where $\kappa = \beta k_b \ell_b^2$ and $\Delta\lambda = \partial\beta u/\partial\eta$.

- Harmonic potential cases are correct to within transcendentally small terms.
- Anharmonic potential cases are correct to within $\mathrm{ord}(\kappa^{-2})$ as verified numerically.

Currently working on a systematic comparison and reformulation of another method.





Extensible freely rotating chains

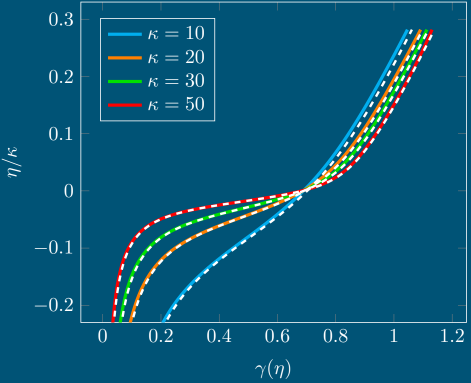
The freely rotating chain is analytically intractable, and link extensibility is even more difficult.

$$\frac{Z}{Z_0} \sim 1 + \frac{N_b}{\kappa}\left(1 + 2\eta\gamma_0 + \frac{\eta^2}{2N_b}\sum_{i=1}^{N_b}\left\langle\cos^2\theta_i\right\rangle_0\right)$$

The asymptotic then produces a general relation:

$$\gamma(\eta) \sim \gamma_0(\eta) + \kappa^{-1}h(\eta) + \mathrm{ord}\left(\kappa^{-2}\right)$$

- Exactly correct for freely jointed chains.
- Working on it for freely rotating chains.
- Iso-energy is extremely easy for Monte Carlo.
- Build upon numerics for any link stiffness.



The general asymptotic relation appears to be valid, but needs further study (N_b, θ_b, error).

Molecular stretching

Device effects in stretching experiments [6, 7].

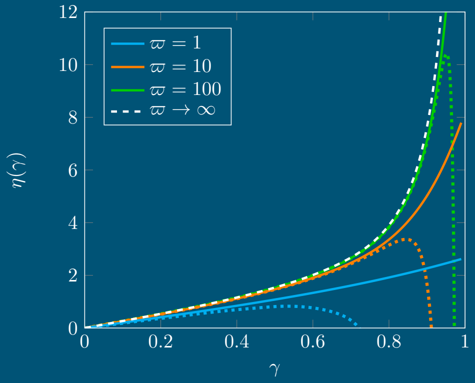
- No device can apply a constant force or extension due to finite stiffness and size.
- Either ensemble provides a zeroth order approximation in certain device limits.
- Weak & steep theories provide corrections.

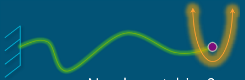
Nearly isometric ($\varpi \gg 1$) stretching device:

$$\eta(\gamma) \sim \eta_0(\gamma) - \frac{1}{N_b\varpi}\left[\eta_0(\gamma)\eta_0'(\gamma) - \frac{\eta_0''(\gamma)}{2N_b}\right]$$

$$\eta_0(\gamma) = \frac{1}{N_b\gamma} + \left(\frac{1}{2} - \frac{1}{N_b}\right)\frac{h(\gamma,3)}{h(\gamma,2)}$$

$$h(\gamma,n) \equiv \sum_{s=0}^{s_{\max}(\gamma)} (-1)^s \binom{N_b}{s} \left(\frac{1-\gamma}{2} - \frac{s}{N_b}\right)^{N_b-n}$$





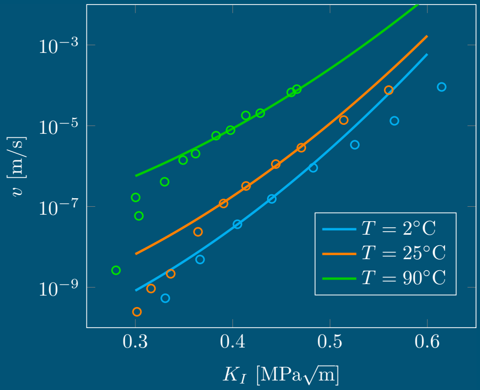
Subcritical crack growth

Statistical mechanical crack growth model [8, 9].

- Harmonic bending, Morse potential bonds.
- Applied displacement or force ensembles.
- Velocity from transition state theory rate.
- Analytic solutions using assumptions:
  - Big system, steep, small stretch.

$$\frac{v}{b} \sim \frac{\omega_0}{\pi} \exp\left(\frac{f\Delta x^\ddagger - \Delta u^\ddagger}{kT}\right)\left(\frac{Rb^2}{2kT}\right)$$

- Similar to before, but better parameters now.
- SLS glass in water; no water in model [10, 11].
  - Possible explanation for v overestimation.
  - Need environment or threshold effects.
  - Interesting to consider more dimensions.



Thermal fluctuations

Thermodynamic fluctuations in freely jointed chains under an applied force vector [12].

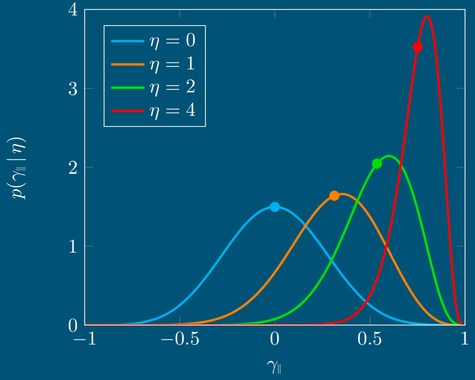
- Extension has many relevant components.
- All extension components fluctuate.
- Every link angle fluctuates independently.

Some immediate consequences:

- Using constant extensions can be problematic.
- Assuming link conformations is much worse.
- Effects can be included in network models [13].

Plenty of interesting work left to do:

- Adding link extensibility, other ensembles [14].
- Compounding effects of imperfect devices [6].
- Hopefully, other systems besides polymers.



$$\langle \gamma_\parallel \rangle = \mathcal{L}(\eta)$$

Summary

Adventures in statistical thermodynamics:

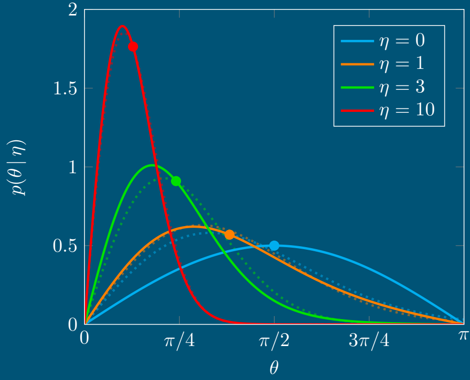
- An asymptotic theory for steep potentials [3].
- Applications in polymers [4, 6] and glass [8].
- Basics like fluctuations [12] are still vital.

Too many ideas, not enough time:

- Combined weak and steep approximations.
- Path-integral formulations [15], WLC, QFT.
- Revisit polymer network constitutive models.

For the asymptotic theory to gain respect:

- Large-scale computational physics calculations.
- Conductivity calculations in crystalline solids?
- Free energy calculations in large molecules?



$$p(\theta \mid \eta) = \frac{e^{\eta \cos\theta} \sin\theta}{2 \sinh\mathrm{c}\, \eta}$$

Hexahedral meshing

Introduction

Creating finite element meshes from image stacks is typically difficult.

- Difficult to recover a high-quality, physically-representative, tractable mesh.
- Often times one (or even two) of these three goals must be sacrificed.

Generating high-quality all-hexahedral finite element meshes is also difficult.

- Adaptivity reduces element count, but is challenging for hexahedral elements [16].
- Conformity accurately represents surfaces, but can create low-quality elements.

`automesh` [17, 18], open-source automatic mesh generation written in Rust (CLI+Python).

- Many robust and automatic features (hex meshing, surface reconstruction, etc.).
- Testing of completed features is showing significantly faster time-to-solution.
- Adaptivity is done, conformity is ongoing, and material interfaces are next.

Segmentation

Images become segmentations through categories.

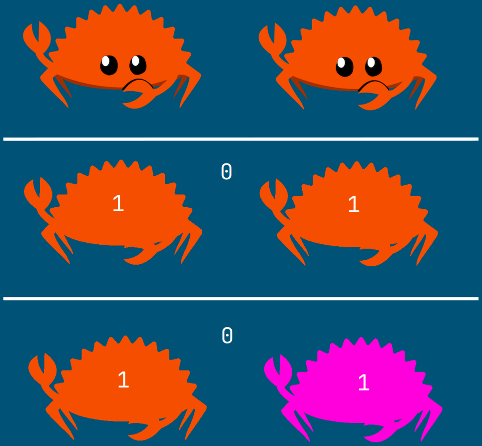
- Image stacks obtained from scans (e.g., CT).
- Each pixel in an image is assigned a class.
- Semantic segmentations aggregate objects.
- Instance segmentations differentiate objects.

Stacks of categorized pixels create a set of voxels.

- Simply an ordered list of unsigned integers.
- Storage cost is typically small (NPY, SPN).
- This is the starting point for `automesh` users.

Many applications for image-to-mesh workflows:

- Traumatic brain injury modeling (ONR).
- Defects, microstructures, surveillance, etc.



Meshing

Direct voxel-to-hex meshing:

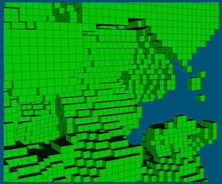
- Simple, robust, perfect quality [19].
- Often intractable element count.
- Poor representation of internal surfaces.
- Removal of void material blocks.

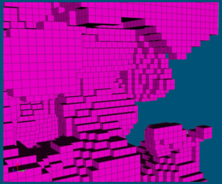
Internal surface reconstruction:

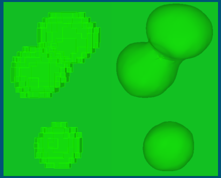
- Marching cubes, dual contouring, etc.
- Smooth, but preserve volume and manifold.

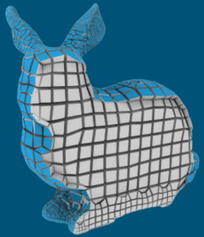
Adaptive [16] all-hexahedral meshing:

- Reconstructed surfaces or other geometry.
- Use small elements only where necessary.
- Non-trivial to conform to the surfaces.









Performance

Measure time for direct voxels-to-hexes meshing.

- Perfect cube of a single material type.
- Optionally[†] also an embedded sphere.

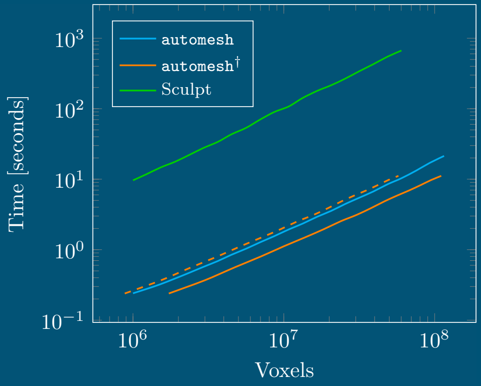
Ideal $O(N)$ scaling on one processor, rates are:

- `automesh` (5.7 million voxels/second)
- `automesh`[†] (9.9 million voxels/second)
- Sculpt (0.1 million voxels/second)

[†] Rate is approximately scaled by the fraction of retained voxels, i.e., it is about 89% of the original rate when including removed voxels.

Memory not studied as closely yet, but with 125 GB:

- `automesh` could do about 1 billion voxels.
- Sculpt could do about 100 million voxels.



Smoothing

Laplace smoothing:

- Moves nodes towards average of neighbors.

$$\Delta \mathbf{x}_a = \lambda \left(\frac{1}{n} \sum_{b=1}^{n} \mathbf{x}_b - \mathbf{x}_a \right)$$

- Drastically reduces volume (-16% in 10 steps).

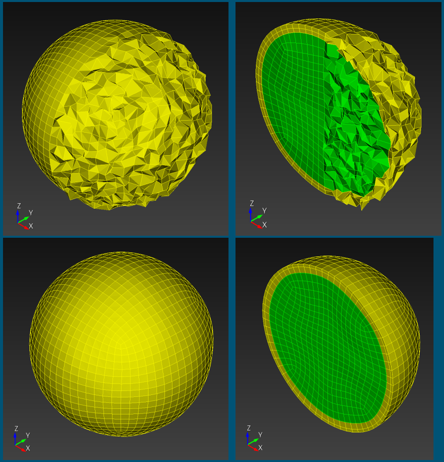
Taubin [20] smoothing:

- Alternates deflation and inflation ($\lambda < -\mu$).
- Nearly preserves volume (+1% in 200 steps).
- Acts like a low-pass filter, $k = \lambda^{-1} + \mu^{-1}$.

$$\lambda = 0.63,\ k = 0.1,\ \mu = \left(k - \lambda^{-1}\right)^{-1} = -0.67$$

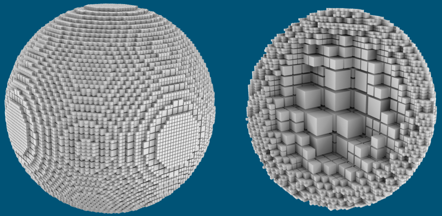
Optional [21] hierarchical control:

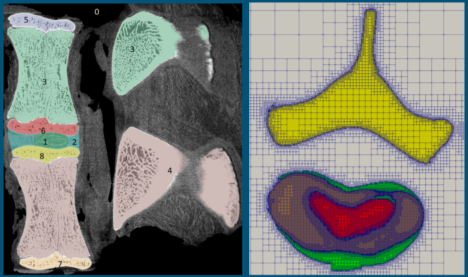
- "Separately" smooth surfaces and volumes.



Octree

- Efficient representation using an octree [22, 23].
  - Recursive subdivision of space into 8 octants.
  - Rules for subdivision based on materials.
  - Additional balancing and pairing rules.
- An octree as an intermediate workflow step.
  - Creates an "adaptive" segmentation.
  - Accelerates methods like defeaturing.
  - Enables adaptive finite element meshes.
- Micro-CT of a spinal unit from SwRI [24].
  - 1 billion voxels become 10 million cells.
  - 5 million cells are removable void.
  - 200x reduction, and in only 36 seconds.





Reconstruction

Smooth reconstruction of internal surfaces.

- More representative of physical features.
- Output (STL) facilitates volume meshing.

`automesh` simply facets the material boundaries.

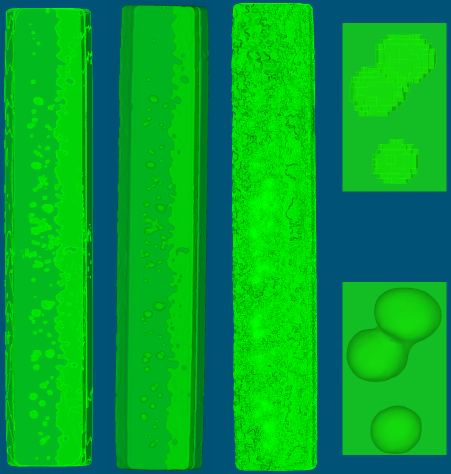
- Volume-preserving and free of holes.
- Rectify non-manifold edges and vertices.
- Defeaturing and Taubin smoothing are key.

Micro-CT scan of a laser weld section [25].

- Input: 2 materials, 6,748,800 voxels.
- Read, defeature[†], mesh[‡], smooth, write.
- Output: STL with 762,396 facets, 33 seconds.

[†] 13.5 seconds, compared to 36 minutes for Sculpt (160x slower).

[‡] 18.2 seconds; with defeaturing, is about 95% of the total time.



Hexahedral adaptivity

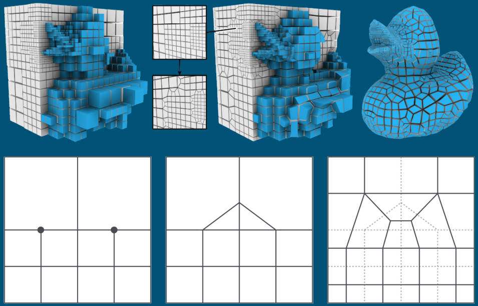
Interior mesh adaptivity via dualization [16].

- Strongly (or weakly) balanced octree.
- Cell subdivision to capture features.
- Connect hanging nodes for polyhedra.
- Centroids connect for guaranteed hexes.

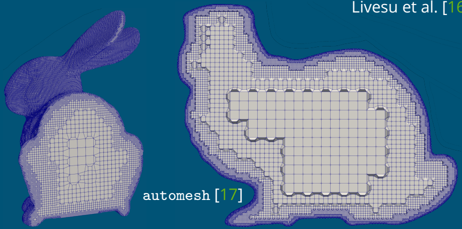
Many features make this the top choice.

- Dual schemes are unambiguously applied.
- Degrees of adaptivity are fully automatic.
- Optimal transitions for low edge valences.
- Some of the lowest element counts and fastest transitions possible [26].

Recently shown that element quality can be kept somewhat high through smoothing [27].



Livesu et al. [16]



automesh [17]

Hexahedral adaptivity

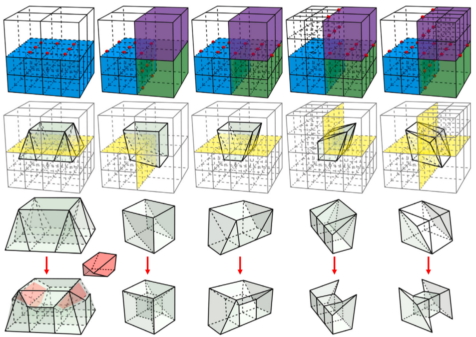
Dual schemes can be fully templated [17, 27].

- 1 face template, 4 edge templates [27].
- No need for intermediate polyhedra.
- Templates can be placed independently.
- Smoothing takes 0.258 MSJ to 0.5+ [27].

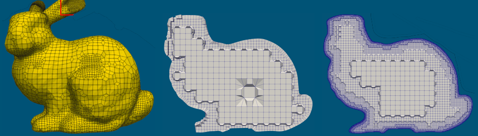
Templates can be placed wicked fast [17].

- Cell centers form indexable node map.
- Another clever map for transition nodes.
- Some number of explicit vertex templates.
- Uniform surface refinement for now.
- Eventually, weakly-balanced templates.

Difficult to make comparisons, but `automesh` could be hundreds (or thousands) times faster.



Tong et al. [27] 358 s (21.7k) | `automesh` [17] 182 ms (21.7k) | `automesh` [17] 805 s (54.7m)



Hexahedral conformity

Conformity to the input surface(s).

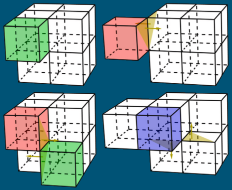
- Delete elements with nodes outside.
- Rules to clean up buffer region.
- Fill buffer with elements to surface.
- Laplacian smooth, element improvement.

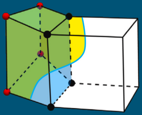
Different methods to fill buffer region.

- Strict duplication-then-projection [27].
- Multi-step process with pillowing [28].
- Closest-point projections typically.

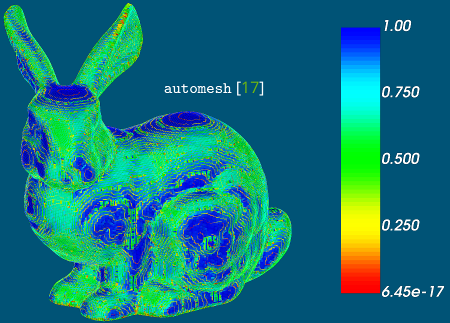
Other aspects to consider simultaneously.

- Size via local shape diameter or curvature.
- Interior surfaces (multi-material cases).
- Snap nodes to geometric corners/edges.





Tong et al. [27]



Hexahedral conformity

Several pieces efficiently implemented.

- Outside element and node removal.
- Removal within the buffer region.
- Closest-point (and other) projections.

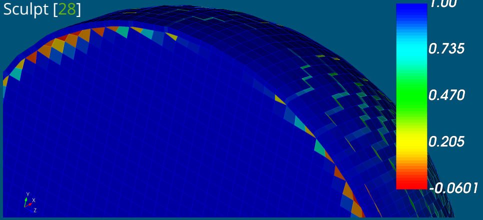
Current challenge is element improvement.

- Laplace surface & volume smoothing [28].
- General metric improvement [29].
- Jacobian [27], or just smart Laplace.
- Energetic (variational) smoothing [30].

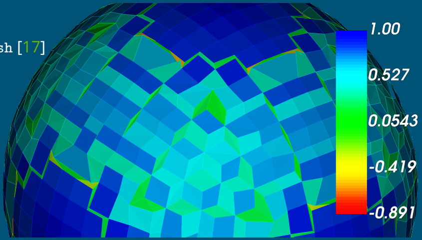
Can element quality be improved very rapidly?

- Fast/parallel treatment of limited stencils.
- Lazy enforcement of surface constraints.
- Critical thinking for new optimizations.

Sculpt [28]



automesh [17]



Summary

Major progress and current challenges.

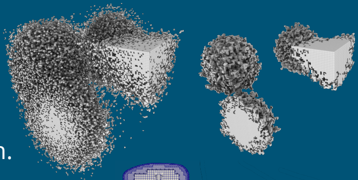
- Adaptivity is blazingly-fast and robust.
- Conformity will likely be conquered soon.
- Simple methods for surface adaptivity.
- Determine weakly-balanced templates.
- Conformity to interior surfaces and model geometry edges or corners.

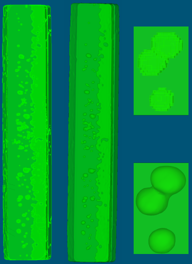
Numerous plug-and-play features.

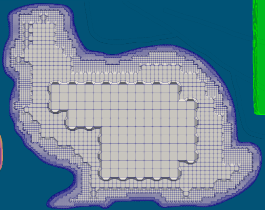
- Voxel defeaturing, surface reconstruction.
- Taubin [20] or (smart) Laplace smoothing.

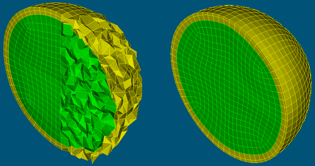
Outside-the-box ideas and opportunities.

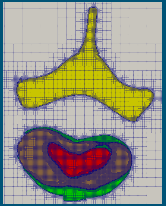
- Hex-dominant meshes with VEM [31].
- Solution adaptive remeshing with octree.











Mechanics tools

Modularity

Modular assembly of constitutive pieces.

- Algorithmic and flexible approach.
- Readily explore model combinations.
- Reduce number of implemented models.
- May lead to automatic model generation.

Different than current notions of modularity.

- Rules for sequential assembly rather than drop-in replacement.
- Specific combinations of model types create other specific model types.

Some conceptual problems to handle.

- Nested sets of constitutive variables.
- Compilation rules for objectivity, combos.

The constitutive relation for the stress

$$a = a(\mathbf{F}), \quad \boldsymbol{\sigma}(\mathbf{F}) = \frac{1}{J}\frac{\partial a}{\partial \mathbf{F}} \cdot \mathbf{F}^T$$

and that for the viscoplastic flow

$$\dot{\mathbf{F}}_{\mathrm{p}} = \mathbf{D}_{\mathrm{p}} \cdot \mathbf{F}_{\mathrm{p}}, \quad \mathbf{D}_{\mathrm{p}} = f\left(\frac{|\mathbf{M}'_{\mathrm{e}}|}{Y(S)}\right)\frac{\mathbf{M}'_{\mathrm{e}}}{|\mathbf{M}'_{\mathrm{e}}|}$$

are combined using a kinematic relation

$$\mathbf{F} = \mathbf{F}_{\mathrm{e}} \cdot \mathbf{F}_{\mathrm{p}}$$

and other derivable rules for the stresses

$$\mathbf{M}_e = J\mathbf{F}_{\mathrm{e}}^T \cdot \boldsymbol{\sigma}(F_{\mathrm{e}}) \cdot \mathbf{F}_{\mathrm{e}}^{-T}$$

into a hyperelastic-viscoplastic model.

Simultaneity

Simultaneous time integration and solving.

- Enables modular constitutive models.
- Model implementation is simpler.
- Extends solver capabilities to models.
- Could be more efficient and robust.

Considerable change to current paradigm.

- Popular iterative approach is a special case sort of like primal-dual solves.
- Existing work assumes time integrator and/or specific constitutive model [32].

Working so far, a few challenges are left.

- Inequality-constraints (plasticity).
- Softening, history, other physics.

Hyperelastic-viscoplastic constitutive model:

$$\boldsymbol{\sigma}(\mathbf{F}) = \frac{2\mu}{J}\,\mathbf{h}' + \frac{\kappa}{J}\,\mathrm{tr}(\mathbf{h})\mathbf{1}, \quad \mathbf{h} = \tfrac{1}{2}\ln(\mathbf{B})$$

$$\mathbf{F} = \mathbf{F}_\mathrm{e}\cdot\mathbf{F}_\mathrm{p}, \quad \dot{\mathbf{F}}_\mathrm{p} = \mathbf{D}_\mathrm{p}\cdot\mathbf{F}_\mathrm{p}, \quad \dot{\varepsilon}_\mathrm{p} = |\mathbf{D}_\mathrm{p}|$$

$$\Pi(\mathbf{F},\mathbf{F}_\mathrm{p},\varepsilon_\mathrm{p}) = a(\mathbf{F}_\mathrm{e}) - \boldsymbol{\lambda} : (\mathbf{F} - \mathbf{F}_0) - \mathbf{P}_0 : \mathbf{F}$$

Differential-algebraic equations:

$$x = \mathbf{F}, \quad y = \{\mathbf{F}_\mathrm{p},\ \varepsilon_\mathrm{p}\}$$

$$\dot{y} = f(x,y,t), \quad g(t,x,y) = 0$$

Finite elements (and more) are the same:

$$\mathbf{x}^a \text{ for } a \in [1,N], \quad y = \left\{\mathbf{F}_\mathrm{p}^g,\ \varepsilon_\mathrm{p}^g\right\} \text{ for } g \in [1,M]$$

Demonstration

Hencky hyperelastic models:

$$\boldsymbol{\sigma}(\mathbf{F}) = \frac{2\mu}{J}\,\mathbf{h}' + \frac{\kappa}{J}\,\mathrm{tr}(\mathbf{h})\mathbf{1}, \quad \mathbf{h} = \tfrac{1}{2}\ln(\mathbf{B})$$

Viscoplastic flow [33] models:

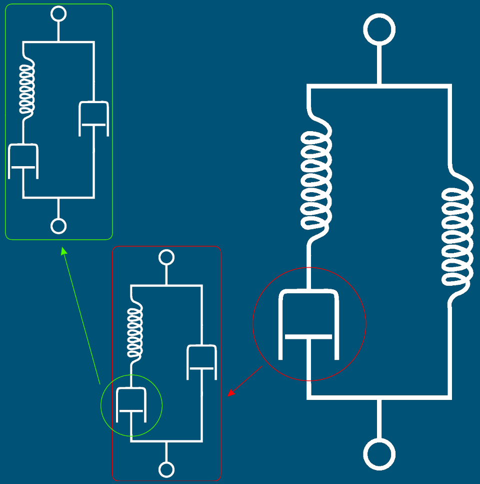
$$\dot{\mathbf{F}}_{\mathrm{p}} = \mathbf{D}_{\mathrm{p}} \cdot \mathbf{F}_{\mathrm{p}}, \quad \mathbf{D}_{\mathrm{p}} = d_0 \left(\frac{|\mathbf{M}'_{\mathrm{e}}|}{Y(S)}\right)^{\frac{1}{m}} \frac{\mathbf{M}'_{\mathrm{e}}}{|\mathbf{M}'_{\mathrm{e}}|}$$

$$\dot{\varepsilon}_{\mathrm{p}} = |\mathbf{D}_{\mathrm{p}}|, \quad Y = Y_0 + H\varepsilon_{\mathrm{p}}, \quad \mathbf{M}_e = J\mathbf{F}_{\mathrm{e}}^T \cdot \boldsymbol{\sigma} \cdot \mathbf{F}_{\mathrm{e}}^{-T}$$

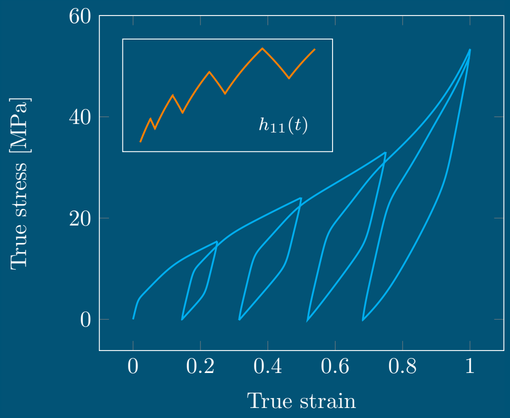
Arruda-Boyce [34] hyperelastic model:

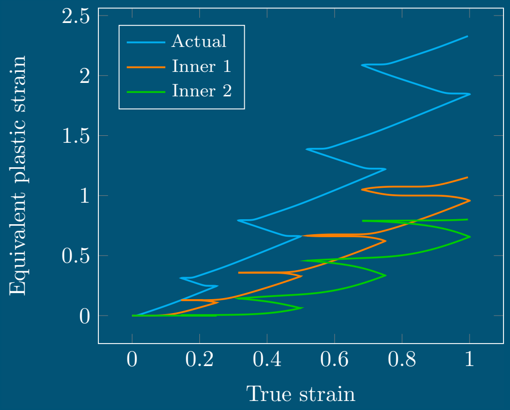
$$\boldsymbol{\sigma}(\mathbf{F}) = \frac{\mu\gamma_0\eta}{J\gamma\eta_0}\,\mathbf{B}^{*\prime} + \frac{\kappa}{2}\left(J - \frac{1}{J}\right)\mathbf{1}$$

$$\eta = \mathcal{L}^{-1}(\gamma), \quad \gamma = \sqrt{\mathrm{tr}(\mathbf{B}^*)/3N_b}$$



Demonstration





Summary

Modularity

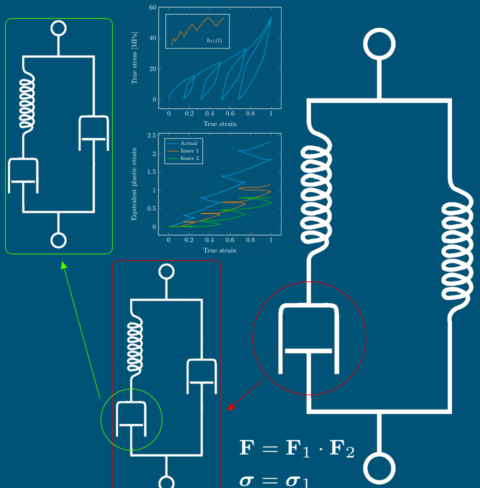
- Flexible and practical implementations.
- Support more models but develop less.
- Genetic algorithms or AI/ML assistance.

Simultaneity

- Enables modularity in constitutive models.
- More efficient and robust implementation.
- Same theme as constraints, multi-physics.
- Could this increase parallel performance?

Considerations

- Constitutive models integral type [35–37].
- Cohesive zone, damage, other models.
- Complexity for modern [38] or older tools.



$$\mathbf{F} = \mathbf{F}_1 \cdot \mathbf{F}_2$$
$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_1$$
$$\boldsymbol{\sigma}_2 = \mathbf{M}_1$$

Conclusion

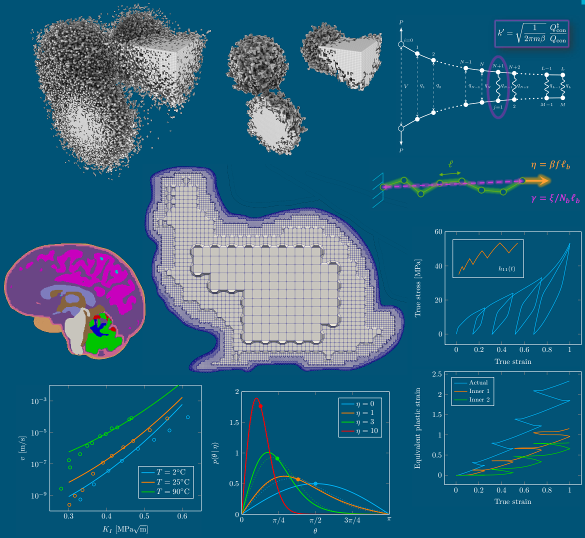
Highlighted recent work and ideas in:

- Statistical thermodynamics.
- Hexahedral meshing.
- Mechanics tools.

Other areas not discussed here:

- Polymer network models [13, 14, 37].
- Cohesive/localization elements [39].
- Polyhedral virtual elements [40].
- Sierra/Solid Mechanics support [41].
- Microstructure DNS for GPSR [42].
- Nuclear weapons program support.

Thank you so much!



References

[1] D. A. McQuarrie, *Statistical Mechanics*, (University Science Books, Mill Valley, CA, 2000).

[2] R. W. Zwanzig, The Journal of Chemical Physics **22**, 1420 (1954).

[3] M. R. Buche, Cornell University (2021).

[4] M. R. Buche, M. N. Silberstein, and S. J. Grutzik, Physical Review E **106**, 024502 (2022).

[5] M. R. Buche and S. J. Grutzik, uFJC v1.3.5 (2022).

[6] M. R. Buche and J. M. Rimsza, Physical Review E **108**, 064503 (2023).

[7] M. R. Buche, Polymers v0.3.7 (2023).

[8] M. R. Buche and S. J. Grutzik, Phys. Rev. E **109**, 015001 (2024).

[9] M. R. Buche and S. J. Grutzik, statMechCrack v0.5.0 (2022).

[10] S. Wiederhorn and L. Bolz, J. Am. Cer. Soc. **53**, 543 (1970).

[11] S. Grutzik, K. Strong, and J. Rimsza, J. Non-Crystal. Solids X **16**, 100134 (2022).

[12] M. R. Buche and A. Chen, arXiv **2604**, 11713 (2026).

[13] J. Mulderrig, M. Buche, and M. Grasinger, arXiv **2504**, 17239 (2025).

[14] M. R. Buche and M. N. Silberstein, Physical Review E **102**, 012501 (2020).

[15] M. R. Buche, Sandia National Laboratories **02140R** (2024).

[16] M. Livesu, L. Pitzalis, and G. Cherchi, ACM Transactions on Graphics **41**, 1 (2021).

References

[17] C. B. Hovey and M. R. Buche, automesh v0.3.9 (2026).

[18] C. B. Hovey and M. R. Buche, Journal of Open Source Software **10**, 8768 (2025).

[19] S. J. Owen, C. E. Ernst, and C. J. Stimpson, Sandia National Laboratories **6412** (2019).

[20] G. Taubin, Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 351 (1995).

[21] Y. Chen and M. Ostoja-Starzewski, Acta Mechanica **213**, 155 (2010).

[22] D. J. Meagher, Rensselaer Polytechnic Institute IPL-TR-80-111 (1980).

[23] M. Bracci, M. Tarini, N. Pietroni, M. Livesu, and P. Cignoni, Computer-Aided Design **110**, 24 (2019).

[24] D. P. Nicolella, S. K. Shaffer, L. L. Frazer, A. D. Pant, and V. B. Kote, I-PREDICT, MTEC (2025).

[25] A. T. Polonsky, J. D. Madison, M. Arnhart, H. Jin, K. N. Karlson, A. J. Skulborstad, J. W. Foulk, and S. G. Murawski, Tomography of Materials and Structures **2**, 100006 (2023).

[26] L. Pitzalis, M. Livesu, G. Cherchi, E. Gobbetti, and R. Scateni, ACM Transactions on Graphics **40**, 1 (2021).

[27] H. Tong, E. Halilaj, and Y. J. Zhang, Journal of Computational Science **78**, 102278 (2024).

[28] S. J. Owen, M. L. Staten, and M. C. Sorensen, Proceedings of the 20th International Meshing Roundtable, 161 (2011).

[29] M. L. Brewer, L. F. Diachin, P. M. Knupp, T. Leurent, and D. J. Melander, International Meshing Roundtable (2003).

[30] K. Garanger, A. Mota, J. W. Foulk III, and J. J. Rimoli, SSRN **6017300** (2026).

[31] B. B. Xu, L. B. da Veiga, Y. J. Zhang, and P. Wriggers, Computer Methods in Applied Mechanics and Engineering **448**, 118432 (2026).

References

[32] D. V. Kulkarni, D. A. Tortorelli, and M. Wallin, Computer Methods in Applied Mechanics and Engineering **196**, 1169 (2007).

[33] M. E. Gurtin, E. Fried, and L. Anand, The Mechanics and Thermodynamics of Continua (2010).

[34] E. M. Arruda and M. C. Boyce, Journal of the Mechanics and Physics of Solids **41**, 389 (1993).

[35] S. Paolucci, Continuum Mechanics and Thermodynamics of Matter (2016).

[36] R. Long, K. Mayumi, C. Creton, T. Narita, and C.-Y. Hui, Macromolecules **47**, 7243 (2014).

[37] M. R. Buche and M. N. Silberstein, Journal of the Mechanics and Physics of Solids **156**, 104593 (2021).

[38] M. R. Buche, conspire v0.6.0 (2026).

[39] A. Chen and M. R. Buche, Sandia National Laboratories **09923R** (2025).

[40] A. Chen and N. Sukumar, International Journal for Numerical Methods in Engineering **125**, e7384 (2024).

[41] Comp. Solid Mech. & Structural Dyn., Sandia National Laboratories (2023).

[42] D. Birky, J. Emery, C. Hamel, and J. Hochhalter, Computer Methods in Applied Mechanics and Engineering **436**, 117738 (2025).

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

DEPARTMENT OF THE NAVY ONR Science & Technology