Theoretical Advances in Hexahedral Mesh Generation

Jeff Erickson
University of Illinois, Urbana-Champaign
I don't know why anyone would think I have any authority to comment on that.

I have approximate knowledge of many things.
Why meshing?

- Numerical solution of differential equations via finite element methods
  - Decompose simulation volume into elementary pieces — “elements”
  - Approximate solution within each element as a low-degree polynomial.

[blog.pointwise.com]
Typical finite elements

- Linear functions over tetrahedra
  
  \[ f(\lambda_0, \lambda_1, \lambda_2, \lambda_3) = \lambda_0 \cdot f(v_0) + \lambda_1 \cdot f(v_1) + \lambda_2 \cdot f(v_2) + \lambda_3 \cdot f(v_3) \]

- Higher-degree polynomials are also used
Typical finite elements

- Trilinear functions over hexahedra

\[
\begin{align*}
  f(\alpha_1, \alpha_2, \alpha_3) &= (1 - \alpha_1)(1 - \alpha_2)(1 - \alpha_3) \cdot f(v_{000}) \\
  &\quad + (1 - \alpha_1)(1 - \alpha_2)\alpha_3 \cdot f(v_{001}) \\
  &\quad + (1 - \alpha_1)\alpha_2(1 - \alpha_3) \cdot f(v_{010}) \\
  &\quad + (1 - \alpha_1)\alpha_2\alpha_3 \cdot f(v_{011}) \\
  &\quad + \alpha_1(1 - \alpha_2)(1 - \alpha_3) \cdot f(v_{100}) \\
  &\quad + \alpha_1(1 - \alpha_2)\alpha_3 \cdot f(v_{101}) \\
  &\quad + \alpha_1\alpha_2(1 - \alpha_3) \cdot f(v_{110}) \\
  &\quad + \alpha_1\alpha_2\alpha_3 \cdot f(v_{111})
\end{align*}
\]

- Higher-degree polynomials and other element shapes are also used
Typical finite elements

- Trilinear functions over hexahedra

- Higher-degree polynomials and other element shapes are also used
Tetrahedral meshing

- Well-developed theory and robust general-purpose software

[Tournier Wormser Alliez Desbrun 2009]
Hexahedral meshing

Huge variety of practical heuristics, engineering expertise, and software, but no automatic general methods. In other words, no *algorithms*. 

[Truegrid] [Ansys] [Ruiz-Gironès Roca Sarrate 2012]
heu·ris·tic /həˈrɪs-tik/ [Gr. εὑρίσκω = find, discover]
n. An algorithm that doesn’t work.
Why hex meshing?

- Hex meshes are better for some finite element methods, for some applications, and for some classes of geometry.

- But advantages are subtle!
Why hex meshing?

“Received wisdom” makes unfair comparisons:

- Unstructured tet meshes versus (locally) structured hex meshes
  - Fewer elements for a given number of nodes
    - For regular cubical grids, sure, but for arbitrary point sets?
    - For many finite-element methods, the size of the linear system size depends on the number of nodes, not the number of elements.
  - Exploit tensor product structure and/or anisotropy in solution
  - But structured meshes require special geometry

- Linear tet elements versus multilinear hex elements
  - Fewer elements for same accuracy
  - Avoid shear and volume locking
  - But higher-order tet elements avoid these problems! [Weingarten 94]
Defining our terms

I know some of these words
What’s a “hexahedron”?

- **Topological**: What (most) mathematicians mean
  - A *topological cube* is the image of an injective map $q : [0,1]^3 \hookrightarrow \mathbb{R}^3$
  - Topological ball with boundary subdivision: 8 vertices, 12 edges, 6 facets
What’s a “hexahedron”?

- **Polyhedral**: What computational geometers mean
  - A *hexahedron* is a convex polytope isomorphic to the cube \([0,1]^3\).
  - Edges are line segments; facets are *planar* convex polygons.
What’s a “hexahedron”? 

- **Polyhedral**: What computational geometers mean
  - A *hexahedron* is a convex polytope isomorphic to the cube $[0,1]^3$.
  - Edges are line segments; facets are *planar* convex polygons.

Not “approximately planar”
Not “planar up to engineering tolerances”
Not “planar up to floating-point error”
What’s a “hexahedron”?

- **Multilinear**: What engineers mean
  - A *hexahedral finite element* is the “multilinear hull” of 8 points in $\mathbb{R}^3$.
  - Edges are line segments; facets are *ruled surface patches*
Jacobian $J = \nabla q$

- $3\times3$ matrix of partial derivatives of multilinear map $q : [0,1]^3 \hookrightarrow \mathbb{R}^3$

- Most numerical methods require $\det J > 0$ everywhere

- Equivalently: *locally convex* at every vertex
What’s a “hex mesh”?

- **Standard cube complex**: A collection of cubes such that
  - Interiors are disjoint.
  - Union is the desired volume.
  - Intersection of any two cubes is a common facet, a common edge, a common vertex, or nothing.
What’s a “hex mesh”?

- **Standard cube complex:** A collection of cubes such that
  - Interiors are disjoint.
  - Union is the desired volume.
  - Intersection of any two cubes is a common facet, a common edge, a common vertex, or nothing.

- **WARNING! Some papers use more relaxed definitions!**
  - Two cubes can share multiple faces
  - Shared faces need not be connected
  - A single cube can be adjacent to itself
  - Some hexes can have zero or negative Jacobians
What is the hex meshing problem?

- **First attempt:** Given a 3d volume, subdivide it into a valid hex mesh.
What is the hex meshing problem?

- **First attempt:** Given a 3d volume, subdivide it into a valid hex mesh.
- **Solved!** Triangulate, then split each tetrahedron into four *hexahedra*!
What is the hex meshing problem?

- **Second attempt:** Given a 3d volume, subdivide it into a valid hex mesh, where every hex has some guaranteed quality.
What is the hex meshing problem?

- Second attempt: Given a 3d volume, subdivide it into a valid hex mesh, *where every hex has some guaranteed quality*.

- Solved! Compute a tet mesh with guaranteed quality, then split each tetrahedron into four hexahedra!
What is the hex meshing problem?

- **Third attempt:** Given a 3d volume, subdivide it into a *useful* hex mesh, where every hex has at least some guaranteed *practical* quality.
What is the hex meshing problem?

- **Third attempt:** Given a 3d volume, subdivide it into a *useful* hex mesh, where every hex has at least some guaranteed *practical* quality.

- *This is open!*
What is the hex meshing problem?

- **Third attempt:** Given a 3d volume, subdivide it into a **useful** hex mesh, where every hex has at least some guaranteed **practical** quality.

- *This is open!*

- But what do “useful” and “practical quality” actually mean?
  - Without crisp, unambiguous definitions, the problem is ill-defined.
  - Computational geometers like meshes with flat faces, but “skew” (variance from flatness) is only one of many possible quality measures, and probably not the most important one.
Practical techniques

- Decomposition: Break the model into pieces that are easier to mesh

[Tautges '01]
Practical techniques

- Advancing fronts: sweeping, paving, plastering, etc.

[Staten Kerr Owen Blacker Stupazzini Shimada ’11]

[Roca Sarrate Huerte 2004]

[Ruiz-Gironès Roca Sarrate 2012]
Practical techniques

- Grids and octtrees

[Ito Shih Soni ’08] [Maréchal ’09]
Practical techniques

- Mapping

[Gregson Sheffer Zhang ’11]
Practical techniques

- 3D parametrization / frame fields

[Li Liu Xu Wang Guo ’12]
Subdivision methods

- Default to tetrahedra when direct hex-meshing methods break down. Possibly include transitional pyramid and prism elements.

- Then refine mixed mesh into a hex mesh.

[Yamakawa and Shimada ’01]
Subdivision methods

- Default to tetrahedra when direct hex-meshing methods break down. Possibly include transitional pyramid and prism elements.

- Then refine mixed mesh into a hex mesh.

[Yamakawa and Shimada ’01]
First try at refinement
First try at refinement
First try at refinement
First try at refinement

[Schneiders ’95]
Mitchell’s Geode

- Transition layer between tets and hexes. [Mitchell ’98]
Yamakawa and Shimada’s HexHoop

- General templates for converting mixed meshes to hex meshes

[Yamakawa Shimada ’01]
HexHoop

- Split pyramid into tets; use standard template for tets
- Templates for cubes and prisms depend on neighboring cells.

[Yamakawa Shimada ’01]
88-hex Schneider’s pyramid

- Standard cube complex: Hexes meet properly face to face. ✓
- Positive Jacobians: No twisted, inverted, or degenerate hexes. ✓
- Internal faces are (slightly) warped.

[Yamakawa Shimada '10]
HexHoop

- Internal faces are (slightly) warped.

[Yamakawa Shimada ’01]
Internal faces are (slightly) warped.

[Yamakawa Shimada ’01]
No boundary refinement

- **New problem:** Subdivide a given volume into a valid hex mesh *whose boundary is equal to a given boundary quad mesh.*

  - After decomposition, need to mesh each component independently
    - Parts agree on common boundary meshes in advance

  - Some applications require greater accuracy at the boundary.
    - Separately generate a high-quality boundary mesh, then don’t change it.
No boundary refinement

- **Open** for polyhedral meshes, even for the simplest nontrivial inputs.

![Octagonal spindle](image1)
![Bicuboid](image2)
![Schneiders’ pyramid](image3)

- **Closed** for topological meshes!  
  
  *Thurston ’93, Mitchell ’96, Erickson ’13*
Necessary condition

- **Lemma:** Every hex mesh has an even number of boundary quads.

- **Proof:**
  - Every hex has six boundary quads. Six is even.
  - Gluing two hexes removes two boundary quads. Two is even.
Dual curves

- The dual $Q^*$ of any surface quad mesh $Q$ is an immersion of circles.
Dual surfaces

- The dual of any hex mesh is an immersion of surfaces.
  - dual arrangements of zonotopes [Fedorov 1885]
  - “derivative complex” [Jockusch ’93 (MacPherson, Stanley)]
  - “hyperplanes” [Sageev ’95]
  - “spatial twist continuum” [Murdoch Benzley ’95, Mitchell ’96]
  - “canonical surface” [Aitchison et al. ’97]
Surface immersion

- Every point in the volume has one of these neighborhoods:
Genus-zero meshes

- **Theorem:** A quad mesh of *the sphere* can be extended to a hex mesh of *the ball* if and only if the number of quads is even.  
  *[Thurston '93, Mitchell ’96]*

- **Proof:**
  - Extend dual curves on the sphere to surface immersion in the ball.  
    *[Csikós Szűcs ’95]*
  - Refine surface immersion into the dual of hex mesh.
Extending curves to surfaces

- Shrink the sphere inward; the dual curves sweep out surfaces.
- Simplify the curves with the following moves — regular homotopy.
- Simplification yields disjoint circles and figure 8s.
- Cap off circles, pair up figure 8s.

[Boy 1901, Whitney ’37]

[Francis ’71, Titus ’73]

[Hass Hughes ’92]
Bubble-wrapping

- Any surface immersion can be refined into the dual of a hex mesh.

[Babson Chan ’08]

- Mitchell gets this slightly wrong.
Bubble-wrapping

- Any surface immersion can be refined into the dual of a hex mesh.

- Mitchell gets this slightly wrong.

[Babson Chan ’08]
Bubble-wrapping

- Any surface immersion can be refined into the dual of a hex mesh.

- Mitchell gets this slightly wrong.
Bubble-wrapping

- Any surface immersion can be refined into the dual of a hex mesh.

[Babson Chan ’08]

- Mitchell gets this slightly wrong.
Bubble-wrapping

- Any surface immersion can be refined into the dual of a hex mesh.

- Mitchell gets this slightly wrong.

[Babson Chan ’08]
Bubble-wrapping

- Any surface immersion can be refined into the dual of a hex mesh.

[ Babson Chan ’08 ]

- Mitchell gets this slightly wrong.
Complexity

- $O(n^2)$ moves are always enough \cite{Francis'69, Nowik'08}
  $\Rightarrow O(n^2)$ hexes in the worst case

- $\Omega(n^2)$ triangle moves may be necessary \cite{Arnold'94, Nowik'08}
  $\Rightarrow \Omega(n^2)$ hexes in the worst case

\[\text{Fig. 21 \cite{Meister 1769}}\]
Linear complexity

- In fact, every quad mesh of the sphere with $2n$ quads is the boundary of a hex mesh of the ball with only $O(n)$ quads. [Eppstein ’99]

- **Eppstein’s proof:** See my talk this morning.

- **Proof #2:** If we also use saddle moves, we need only $O(n)$ moves. [Csikós Szűcs ’95]

- **Proof #3:** Use cycle separators. [Miller ’86]
Practical approaches

- **Whisker weaving** [Tautges Blacker Mitchell ’96, Folwell Mitchell ’99]
  - Contract dual curves inward to create dual surfaces, maintain double curves

- **Cycle elimination** [Müller-Hannemann ’01]
  - Eliminate simple dual cycles one at a time
  - In reverse: For each dual cycle, introduce a layer of hexes

- Both methods require dual curves **without self-intersections**.
Further extensions

- Eppstein’s algorithm actually works for *bipartite* surface meshes with *arbitrary topology*. [Eppstein ’99]

- Generalizes to higher-genus surface meshes satisfying a certain topological constraint. [Erickson ’13]

- Sufficient regular refinement gives us a mesh of *trilinear* hexes
  - ...with *terrible* quality, *especially* near the boundary. [Bern Eppstein ’01]

- Sufficient regular refinement gives us a mesh of *polyhedral* hexes if and only if every *bicuboid* has a polyhedral hex mesh. [Bern Eppstein ’01]
Extreme bubble-wrapping

- For any immersed surface $\Sigma$ in $\mathbb{R}^3$, there is a cubical 4-polytope whose dual 2-skeleton contains a subdivision of $\Sigma$. [Schwartz Ziegler ’04]

- Construction uses a generalization of the HexHoop template [Yamakawa Shimada ’01]

Boy’s surface
Cube flips

- Replace a connected subset of cube facets with its complement.

- Analog of “bistellar flips” for tetrahedral meshes. \[\text{[Alexander '30, Pachner '78]}\]

\[\text{[Bern Eppstein '01]}\]
Habegger’s problem

- Given two cube complexes with the same underlying space, can one always be transformed into the other by cube flips? [Problem 5.13 in Kirby’s *Problems in Low-Dimensional Topology, 1995*]
  - Trivially, no! Cube flips preserve parity.

- So *when* are two homeomorphic cube complexes connected by flips?

- Any two PL triangulations of the same PL manifold are connected by bistellar flips. [*Pachner ’78, ’90*]
Cubulations mod flips

- **Conjecture:** Two cubulations of the same manifold are connected by cube flips if and only if their dual surface immersions are cobordant. [Funar ’99].

- **Theorem:** Two quad meshes of the same 2-manifold are equivalent if and only if they have
  - the same number of quads mod 2,
  - and homologous dual curves. [Funar ’07]
Cubulations mod flips

- **Conjecture:** Two cubulations of the same manifold are connected by cube flips if and only their dual surface immersions are cobordant. [Funar ’99].

- **Theorem:** Two quad meshes of the same 2-manifold are equivalent if and only if they have
  - the same number of quads mod 2,
  - and homologous dual curves. [Funar ’07]

This was the key inspiration for my SOCG result.
Elementary moves

- **Lemma**: Two immersed curves with the same parity are homologous iff one can be transformed to the other using these elementary moves:
Bubble-wrapped moves

- **Lemma:** After suitable refinement, each elementary move can be executed by a finite sequence of cube flips. [Funar ’07, Bern Eppstein ’01]
Summary

- With boundary refinement:
  - Hex meshing is *no harder than tet meshing*.
  - **Good** hex meshing, on the other hand, is a black art.

- Without boundary refinement:
  - Topological hex meshing is *solved*...
    - ...except for tight bounds on worst-case mesh complexity.
  - Trilinear hex meshing is *“solved”*...
    - ...but **terrible** quality, especially near boundary; *no* bounds on mesh complexity
  - Polyhedral hex meshing reduces to bicuboids
  - **No** quality guarantees of any kind.
The open problem
The open problem

- Define *useful* but *general* constraints on the input geometry.
  - Polyhedra with all angles $\geq 90^\circ$?
  - Smooth surfaces?
The open problem

- Define *useful* but *general* constraints on the input geometry.
  - Polyhedra with all angles $\geq 90^\circ$?
  - Smooth surfaces?

- Define *useful* quality constraints for trilinear hexahedral meshes.
  - All (scaled) Jacobian determinants $\geq 0.1$?
  - All Jacobians *close* to (scaled) orthogonal? (What does *close* mean?)
  - In every hex, *similar* Jacobians at all 8 vertices? (What does *similar* mean?)
  - Number of hexes = $O(\int 1/lfs^3)$?
The open problem

- Define *useful* but *general* constraints on the input geometry.
  - Polyhedra with all angles $\geq 90^\circ$?
  - Smooth surfaces?

- Define *useful* quality constraints for trilinear hexahedral meshes.
  - All (scaled) Jacobian determinants $\geq 0.1$?
  - All Jacobians *close* to (scaled) orthogonal? (What does *close* mean?)
  - In every hex, *similar* Jacobians at all 8 vertices? (What does *similar* mean?)
  - Number of hexes $= O(\int 1/lfs^3)$?

- Given an *arbitrary* volume meeting the input constraints, *provably* compute a hex mesh that *provably* meets the output constraints.
The open problem

- Define *useful* but *general* constraints on the input geometry.
  - Polyhedra with all angles \( \geq 90^\circ \)?
  - Smooth surfaces?

- Define *useful* quality constraints for trilinear hexahedral meshes.
  - All (scaled) Jacobian determinants \( \geq 0.1 \)?
  - All Jacobians *close* to (scaled) orthogonal? (What does *close* mean?)
  - In every hex, *similar* Jacobians at all 8 vertices? (What does *similar* mean?)
  - Number of hexes = \( O(\int 1/ \text{lfs}^3) \)?

- Given an *arbitrary* volume meeting the input constraints, *provably* compute a hex mesh that *provably* meets the output constraints.

- ...and make it work in practice!
I hate meshes.
I cannot believe how hard this is.
Geometry is hard.

— David Baraff
Senior Research Scientist
Pixar Animation Studios