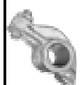
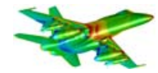
Meshing:
A (Biased) Crash Course

Mathieu Desbrun

The Big Picture

Meshing, an essential preprocessing step


- For surface representation
  - complex geometry
  - with a few basic geometric primitives
- For simulation of physical phenomena
  - realistic/accurate animation of fluid, deformable solids, electromagnetism
    - often modeled as PDE
  - domain discretization
    - Then FEM/FVM to integrate PDE in space (& time)

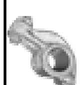
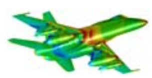



The Big Picture

Meshing, an essential preprocessing step



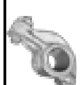
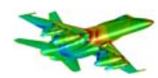
- For surface r
  - compl
  - with primitives
- For sim phenomena
  - realisti on of fluid, deformable solids,
    - often
  - domain d
    - Then F egrate PDE in space (& time)

The Big Picture

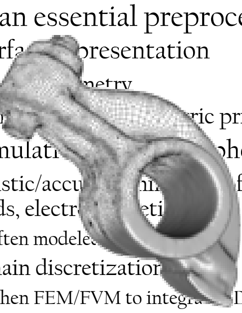
Meshing, an essential preprocessing step


- For surface representation
  - complex geometry
  - with a few basic geometric primitives
- For simulation of physical phenomena
  - realistic/accurate animation of fluid, deformable solids, electromagnetism
    - often modeled as PDE
  - domain discretization
    - Then FEM/FVM to integrate PDE in space (& time)

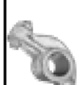
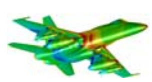



The Big Picture

Meshing, an essential preprocessing step

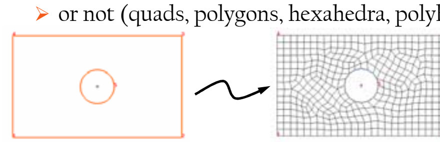

- For surfa presentation
  - com etry
  - with ric primitives
- For simulati phenomena
  - realistic/accu f fluid, deformable solids, electr
    - often modele
  - domain discretization
    - Then FEM/FVM to integr DE in space (& time)

What is Meshing?

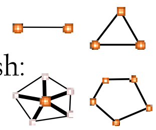
General Idea: "flat" 2D "flat" 3D

- breaking up a physical domain
  - 2D domain in 2D, or in 3D, or 3D domain,...
- into simpler subdomains—the *elements*
  - simplices (triangles, tetrahedra)
  - or not (quads, polygons, hexahedra, polyhedra)



Brief Glossary

Node, element:



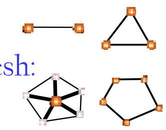
Structured/Unstructured Mesh:
- regular valence and degree

Isotropic/Anisotropic Mesh:
- without/with stretched elements

Graded Mesh:
- w/ elements varying in size (fct of position)

Brief Glossary

Node, element:

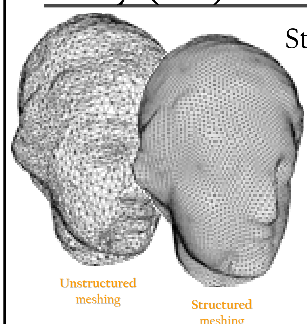


Structured/Unstructured Mesh:
- regular valence and degree

Isotropic/Anisotropic Mesh:
- without/with stretched elements

Graded Mesh:
- w/ elements varying in size (fct of position)

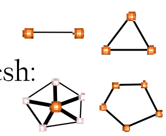
Why (Un)Structured?



Structure brings:
- simpler data structure
- better compression
- reuse of DSP algos
  - smoothing
  - wavelets

Brief Glossary

Node, element:

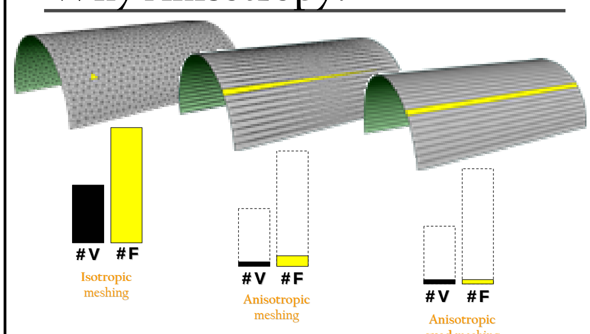


Structured/Unstructured Mesh:
- regular valence and degree

Isotropic/Anisotropic Mesh:
- without/with stretched elements

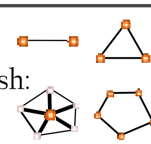
Graded Mesh:
- w/ elements varying in size (fct of position)

Why Anisotropy?



Brief Glossary

Node, element:

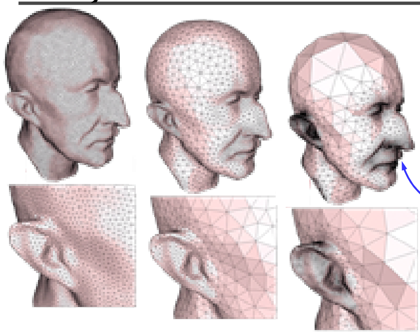


Structured/Unstructured Mesh:
- regular valence and degree

Isotropic/Anisotropic Mesh:
- without/with stretched elements

Graded Mesh:
- w/ elements varying in size (fct of position)

Why Gradation?



Goals in Mesh Generation

Control over shape of elements

- mass matrix conditioning
- shape often induced by PDE
  - good bet: **regular** shape

Control over sizing

- often dictated by simulation … & boundary

Control over total size

- low number of elements is preferred

How to Mesh a Domain?

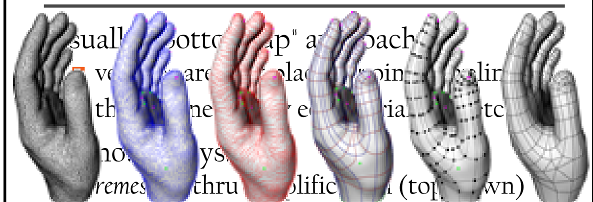
Usually, "bottom-up" approach

- vertices are first placed (point sampling)
- then connected by edges, triangles, etc…

But not always…

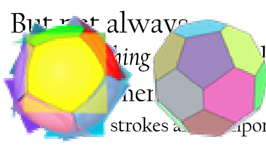
- *remeshing* thru simplification (top-down)
- placement of edges first [Alliez *et al.* 2003]
  - as strokes along important directions
- placement of elements first [Alliez *et al.* 2004]
  - to best approximate local shape, based on normal field

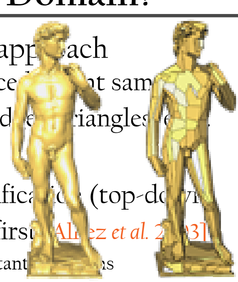
How to Mesh a Domain?



- placement of edges first [Alliez *et al.* 2003]
  - as strokes along important directions
- placement of elements first [Alliez *et al.* 2004]
  - to best approximate local shape, based on normal field

How to Mesh a Domain?





- placement of elements first [Alliez *et al.* 2004]
  - to best approximate local shape, based on normal field

What We Won't Mention [Much]

Meshing of curved manifolds

- often, can be flattened thru parameterization
  - see STAR report aim@shape, Alliez et al. 2004

Meshing of non manifold objects

Polygonal Meshing

Surface Approximation

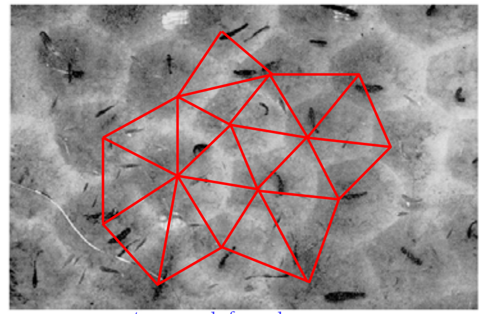
- vertices can be *off* the original surface
  - see [Nadler '86, Simpson '94, Heckbert-Garland '99]

So How do I Mesh a Domain?

Seemingly, vertex placement is crucial

- we must "sample" the domain appropriately
- well-spaced point sets?
  - spread nodes in the (flat or curved) domain
  - spread them evenly
    - using an attraction/repulsion simulation for instance
  - done, right?
    - *most papers in graphics use this approach*
  - good news: sufficient in *2D*

So How do I Mesh a Domain?

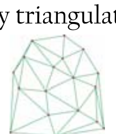


An example from the sea...

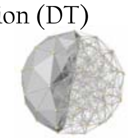
Introducing Delaunay Triang.

'*Canonical*' triangulation of a point set

- with numerous optimal properties
  - see: Eppstein, Meshing Roundtable 2001
    Shewchuck, his UC Berkeley's website
    Lots of robust, existing codes (Triangle, CGAL, ...)
- "territories" = Voronoi diagram
- "dual" = Delaunay triangulation (DT)
- Live demo



In 2D

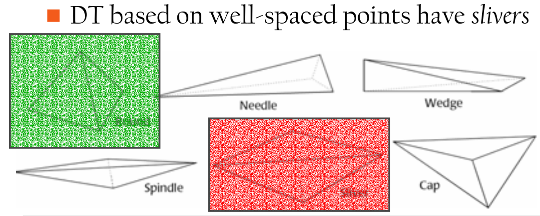


In 3D

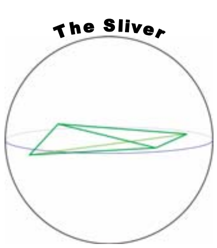
Moving on to 3D

Everything breaks...

- the regular tet does NOT tile space
  - perfectly shaped tets everywhere not attainable
- DT based on well-spaced points have *slivers*



Scrutinizing the Beast



Four (well-spaced) vertices near the equatorial plane of their circumsphere

Taming the Beast?

Recipes to get rid of slivers

- local jiggling of faulty vertices
  - through optimization of shape's "quality"
    - for example, penalize small dihedral angles
  - often, non-linear and slow
  - no guarantee of success—in fact, may be worse
- locally altering the Delaunay triangulation
  - called "sliver exudation"
  - no longer Delaunay, obviously
  - guarantees, but no *practical* bounds

Popular 3D Meshing Methods

Delaunay-based
- refinement
- sphere packing

specific subdivision
- octree
- regular lattice

advancing front

Global Node Placement

spring energy
- Laplacian *(bad!)*
- non-zero rest length

aspect / radius ratios

dihedral / solid angles

max-min/min-max
- volumes, lengths
- sphere radii

sliver exudation

Local Node Optimizations

An Interesting Alternative

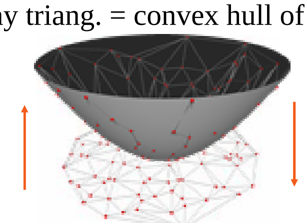
Emergence of variational approaches
- Centroidal Voronoi Tesselation
- Optimal Delaunay Triangulation

Idea:
- position of vertices also *optimal* in some way
- often based on **lifting transformation**

Lifting Transformation (nD)

Lift points ***(x, y)*** to paraboloid ***(x, y, x² + y²)***
- placement of origin irrelevant

Delaunay triang. = convex hull of 3D points



Lifting Transformation (nD)

Lift points ***(x, y)*** to paraboloid ***(x, y, x² + y²)***
- placement of origin irrelevant

Voronoi Diagram = proj. of upper enveloppe

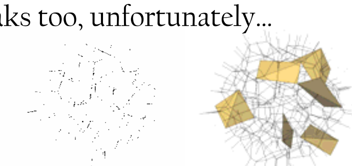
Pushing This Analogy Further

Idea: **move vertices to perfect the approximation of the paraboloid**
- roots in *function approximation*
  - Hessian($||x||^2$) = *Id*
- studied quite a bit in 2D
- *Lloyd algo:*
  - for given points, DT gives optimal L^1 approx
  - for given conn., find optimal vertex positions
    - Live Demo

volume btw approximant and paraboloid

Lloyd Algorithm in 3D

Breaks too, unfortunately...



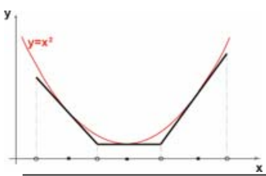
(after Lloyd relaxation)

- why? dual cells well-shaped
- but not sufficient for well-shaped tets!
- use of a dual approach, maybe?

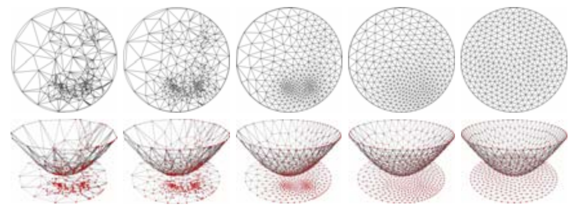
Under/Overlaid Approximant

CVT

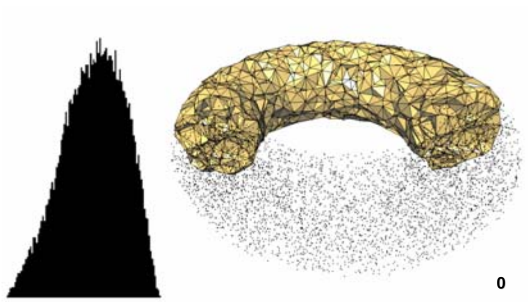
- approximant
- compact Voronoi cells
- isotropic sampling



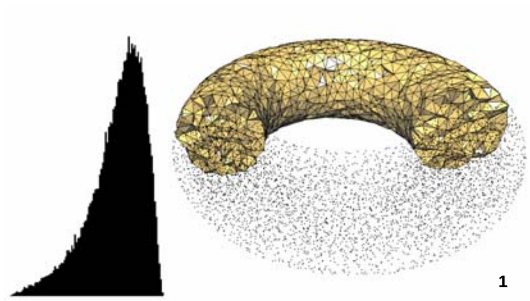
Results of Optimization in 2D



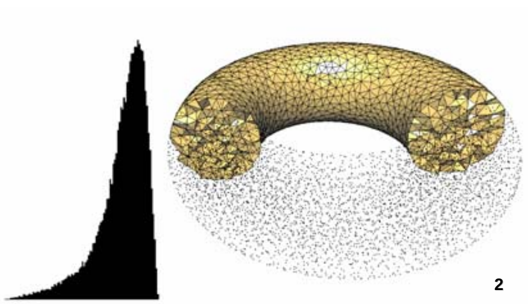
Optimization: init



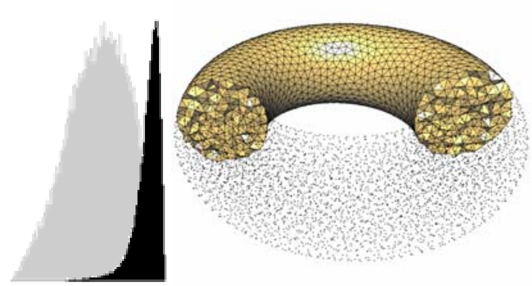
Optimization: step 1



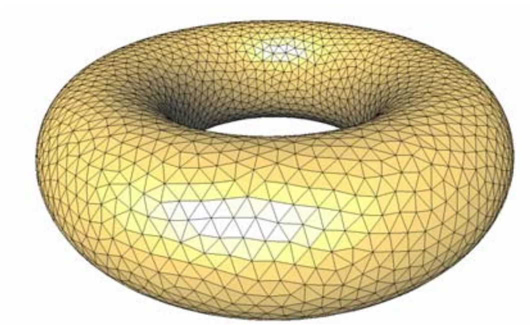
Optimization: step 2



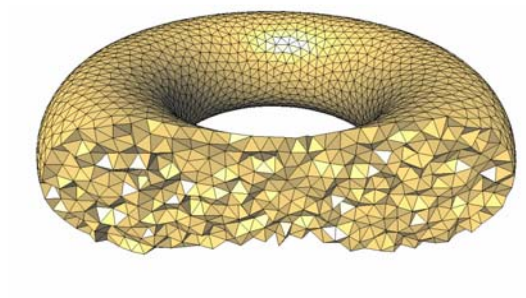
Optimization



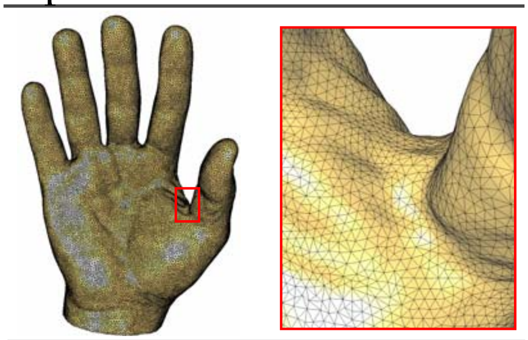
Optimization



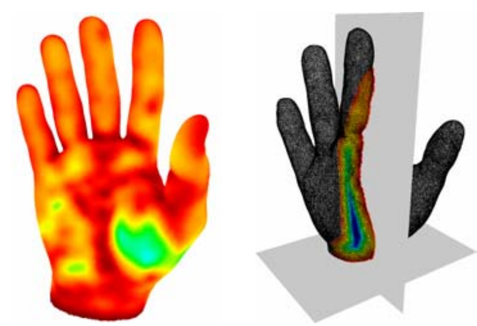
Optimization



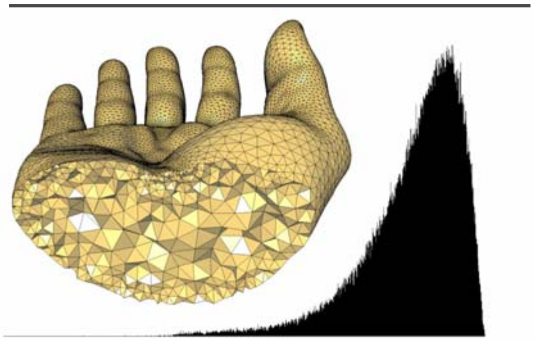
Input mesh



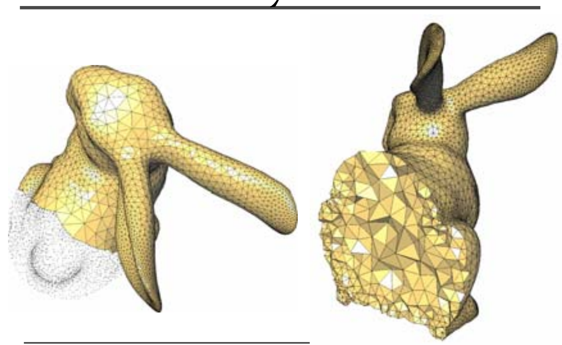
Hand: Sizing



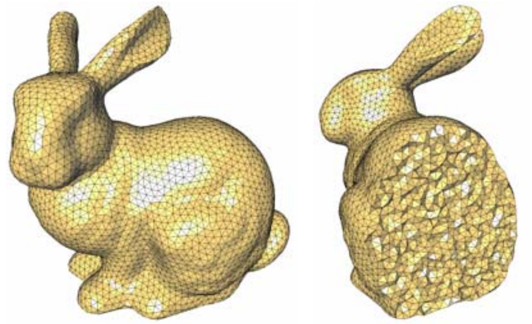
Hand Remeshed



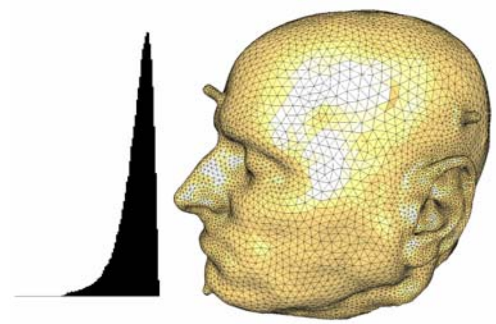
Stanford Bunny



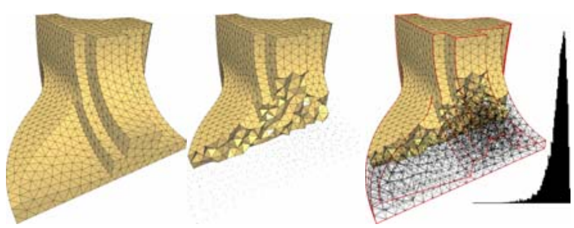
Stanford Bunny (uniform)



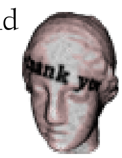
The Visible Human



Fandisk



Conclusions



Meshing: a fascinating research field
- needs theoretical guarantees
- needs practical results

Link to DEC
- improves Hodge star if mesh is nice

Please, consider using Delaunay
- not slow (unlike what you've heard)