Applications of DEC:
Fluid Mechanics and Meshing

Mathieu Desbrun
Applied Geometry Lab

Overview

Putting DEC to good use

Fluids, fluids, fluids

- geometric interpretation of classical models
- discrete geometric interpretation
  - new geometry-based integration technique

Quadrangle Meshing

- discrete notion of harmonicity
- practical method to directly create quads

Part I

Computational Fluids with DEC

based on work with
Sharif Elcott, Yiying Tong, Eva Kanso, Peter Schröder

Fluid Models (I)

Euler Equations

$\rho = \text{const}$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\nabla p + f$$

$$\nabla \cdot u = 0$$

pressure | momentum eq. | velocity | body forces

- inviscid fluids (not viscous)
- incompressible
- non-linear PDE, with linear constraint

Fluid Models (I)

Euler Equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\nabla p + f \quad \text{momentum eq.}$$

$$\nabla \cdot u = 0 \quad \text{mass eq.}$$

- inviscid fluids (not viscous)
- incompressible
- non-linear PDE, with linear constraint

Fluid Models (II)

Navier-Stokes Equations

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)\, u = -\nabla p + f - \nu\, \Delta u$$

$$\nabla \cdot u = 0$$

- only change: viscosity
  - coefficient ν
- loss of total energy during motion

"Geometry" of Fluids?

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

Euler equations seem clear
- advection + div-free projection ad infinitum
  - Stam's Stable Fluids do this wonderfully well
  - numerous follow-up work (Fedkiw *et al.*)
- but what does it mean, geometrically?
  - "total energy" is rather unintuitive
  - is there a notion of momentum preservation?

Yes
- but of course, we need to massage the PDE
- so as to reveal the geometric structure

Geometry Revealed

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

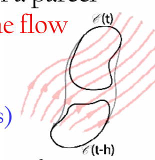
Pressure disappears when we take the curl:

$$\frac{\partial \omega}{\partial t} + \mathcal{L}_u \omega = 0 \qquad \omega = \nabla \times u \quad \text{(vorticity)}$$

$$\nabla \cdot u = 0 \qquad u \parallel \partial \mathcal{D}$$

- vorticity measures the "spin" of a parcel
- vorticity is "advected" along the flow
- the circulation around *any* *closed loop* is constant as it gets advected (by Stokes)
  - known as *Kelvin's theorem*
  - call it preserv. of angular momentum if you want

$\Gamma(t) = \oint_{\mathcal{C}(t)} u.dl$



Geometry Revealed

So we know:

Integral of vorticity constant on advected sheet

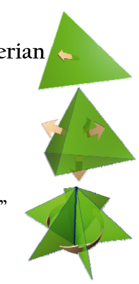
Additionally, ω *defines* u
- if we ignore complex topology for a moment
- $u = \nabla \times (\Delta^{-1}\omega)$ because u is divergence free!

Vorticity is the only real variable here and Kelvin's is *a defining property*

(Navier-Stokes: loss along the way)

Towards a Proper Discretization

Domain discretization = *simplicial complex*
- fluxes through faces for velocity
  - intrinsic (coordinate-free) and **eulerian**
    - » reminiscent of staggered grids…
- net flux for divergence
  - what comes in…must come out
- flux spin for vorticity
  - Torque created on a "paddle wheel"
- valid for any grid…

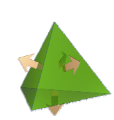


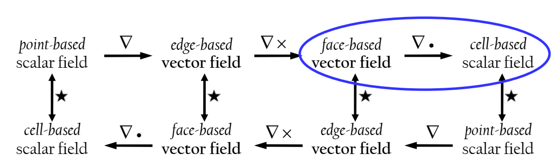
Enter Discrete Exterior Calculus

Need for proper link btw flux, vorticity, div
- hopefully matching differential counterparts
- to create a discrete differential structure
  - i.e., structure-preserving discretization
- Fortunately, that's DEC
  - we know how to do all that, right??
  - flux = 2-form
  - div = exterior derivative of flux
  - curl = $\star d \star$ of flux

Divergence Operator

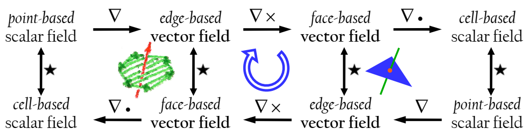
Simply **d** of 2-form
- summing face values of tets
- returning values in tets





Curl Operator

Curl requires going to the dual
- from faces to dual edges first
- then **d** (sum of dual edge values)
- then back onto primal edges



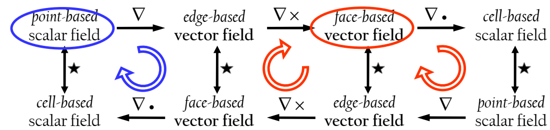
Laplacian Operator

$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\nabla p + \Delta u$

Fro Navier-Stokes, Laplacian needed
- from faces to faces

Try it for 0-forms at home: you'll get the **cot formula**...

$\Delta = \mathbf{d}\star\mathbf{d}\star + \star\mathbf{d}\star\mathbf{d}$



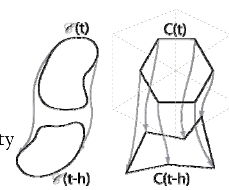
Integrating Equations of Motion

We have all the computational set-up
- But how do we integrate the motion?

Through preserving important structures?
- Circulation/vorticity preservation
- Crucial for visual impact
  - volutes in smoke
  - vortices in liquids

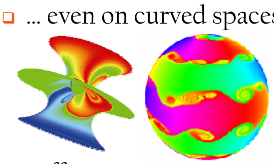
Discrete Kelvin's Theorem

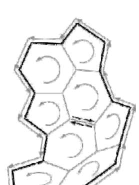
Simple way to integrate Euler equations:
- For each 1-simplex
  - backtrack local loop in current velocity field
  - deduce new circulation
    - i.e., new discrete vorticity
- Find new velocity field
  - simple Poisson equation
    - $u = \nabla \times (\Delta^{-1}\omega)$



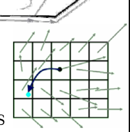
Discrete Kelvin's Theorem

Guarantees circulation preservation...
for any discrete loop!
- big loop = union of small ones
- ... even on curved spaces





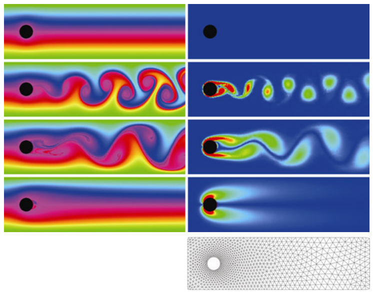
- Difference with Stable Fluids?
  - trace back integrals, not point values



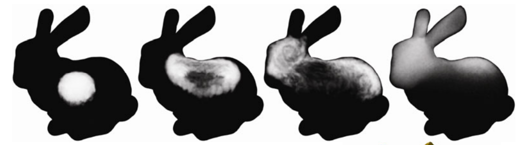
Results

New method
- exact discrete vorticity preservation
- arbitrary simplicial meshes
  - see also [Feldman *et al.* '05, Klingner *et al* '06]
- **everything is intrinsic**
- basic operators very simple (super parse)
- great flows for small meshes!
  - computationally efficient even on coarse mesh
  - no need for millions of vortex particles

Channel

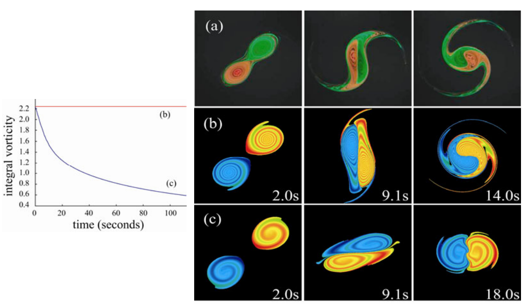


Smoking Bunny

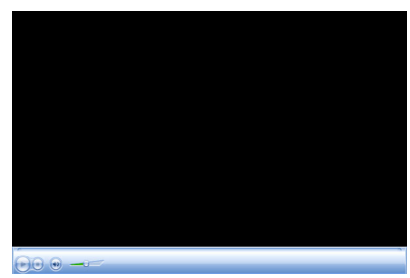


7k vertices, 32k tets; 0.45s per frame on PIV (3GHz)

Merging Vortices



Movie



Part II

Quad Meshing with DEC

based on work with
Yiying Tong, Pierre Alliez, David Cohen-Steiner

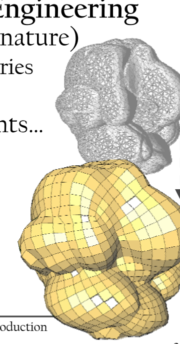
Quadrangulations

Needed in CAGD, Reverse Engineering

- Ubiquitous (tensor-product nature)
  - Modeling anisotropy/symmetries
  - FEM, texture atlasing
- But global topology constraints…

A Variety of Requirements:

- Isotropy vs anisotropy
- Orthogonality, Alignment
- Regularity, Sizing



Quad Meshes: *Reverse Engineering*

For a **local** patch of quadrangulation

- induce natural (*u*,*v*) parametrization
- **Edges: integer-valued isocurves of u/v**
- "Nice" mesh ≈ square mesh in certain metric

  $\langle \nabla u, \nabla v \rangle = 0 \qquad \langle \nabla u, \nabla u \rangle = \langle \nabla v, \nabla v \rangle.$

  - Cauchy-Riemann equations
  - using language of differential form, this is

    one-forms → $du = \star dv$
- Thus, *u* and *v* are both *harmonic* (Laplacian=0)
  - *du* and *dv* too! Cool, DEC seems perfect for that

Methods for Quadrangulations

Among many:

- clustering/Morse [Boier-Martin et al. 03, Carr et al. 06]
- curvature lines [Alliez et al. 03, Marinov/Kobbelt 05]
- isocontours [Dong et al. 04]
  - two continuous potentials
  - (much) more robust than streamlines
- periodic global param (PGP) [Ray et al. 06]
  - Pbs: PGP non linear + no real control

What about using discrete forms?

- global **conformal** param [Gu/Yau 03]

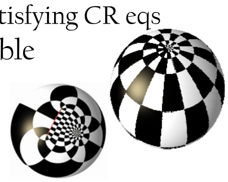
Problems, Problems…

Solving for two continuous potentials (u,v)

- with ***gradients fields*** satisfying CR eqs

Alas, *singularities* unavoidable

- either poles
- or line singularity
  - T-junctions…



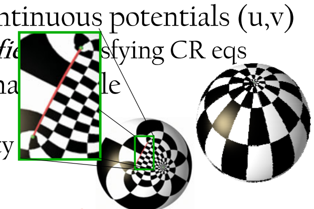
Problems, Problems…

Solving for two continuous potentials (u,v)

- with ***gradients fields*** satisfying CR eqs

Alas, *singularities* unavoidable

- either poles
- or line singularity
  - T-junctions…

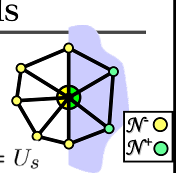


Can we find a better way?

- Only requirement: **continuity of 1-forms**
- so we can actually use discontinuous (u,v)!
  - [Yiying Tong et al. 2006]

Discontinuous Potentials

"Tweaked" Laplacian



- continuity of 1-form induces:

$$du^- = du^+$$

$$\Rightarrow u^- = u^+ + U_s \;\; i.e., \;\; u^- - u^+ = U_s$$

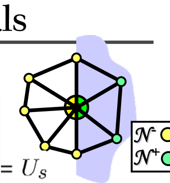
Similarly, $v^- - v^+ = V_s$

$$\sum_{j \in \mathcal{N}(i)} (\cot\alpha_{ij} + \cot\beta_{ij}) \begin{pmatrix} u_i - u_j \\ v_i - v_j \end{pmatrix} = \sum_{j \in \mathcal{N}^+(i)} (\cot\alpha_{ij} + \cot\beta_{ij}) \begin{pmatrix} U_s \\ V_s \end{pmatrix}$$

generate smooth fields *modulo* the jump!

Discontinuous Potentials

"Tweaked" Laplacian



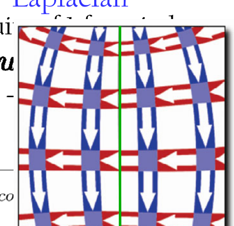
- continui…

du…

$\Rightarrow u^- = u^+$ … $= U_s$

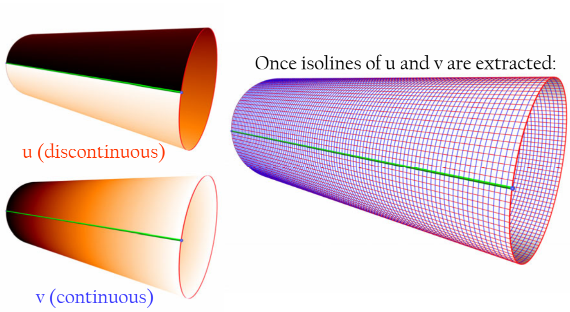
… $= V_s$

$\sum_{j \in \mathcal{N}(i)} (\cot\alpha_{ij} + \text{co}$ … $+ \cot\beta_{ij}) \begin{pmatrix} U_s \\ V_s \end{pmatrix}$



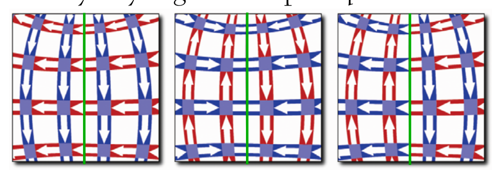
generate smooth fields *modulo* the jump!

Simple Example of Tweaked Δ



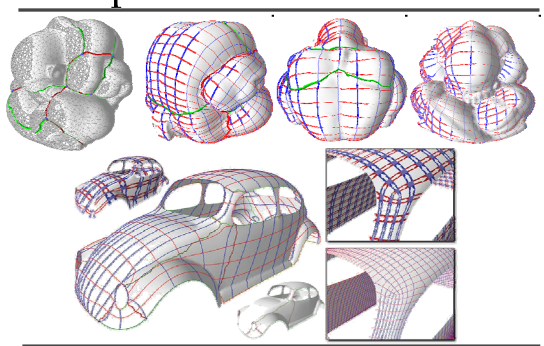
Possible Discontinuities

Only three different cases:
- only way to guarantee pure quads

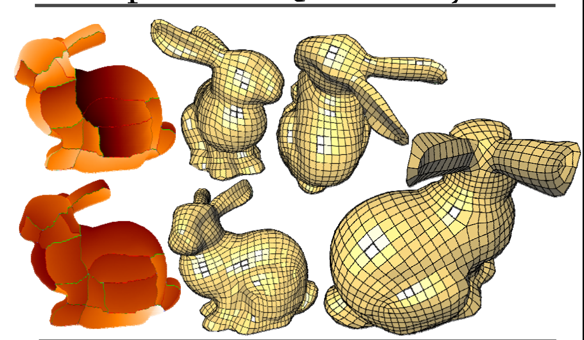


- in the 3 cases, just a tweak of the Laplacian
  - still only a linear system to solve!

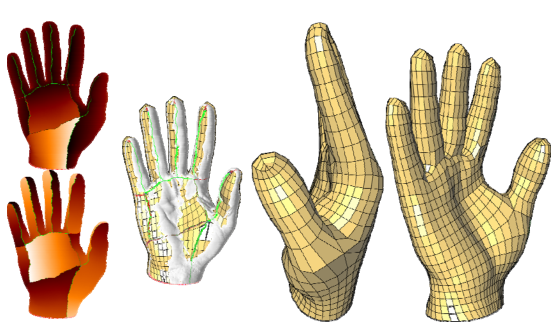
Examples



Example: Pure-Quad Bunny

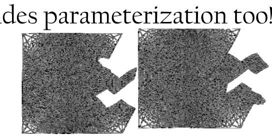


More Results



Advantages of DEC

Foundations of discrete forms powerful
- good grasp on vector field singularities
  - control of “irregular valences”
  - link with “cone singularities” [Kharevych et al. 06]
- still just a linear system to solve
- no need for well-tailored cuts
- provides parameterization too!



Take-Home Message

Don't Arbitrarily Discretize!

- discretize geometric structures
  - PDEs often hide these structures
- uncover the nature of the variables involved
  - usually, natural locations on mesh
- turn the crank with some DEC tools….

Next

- circles may be the right discrete geometry!
  - conformal geometry