AFEM for Geometric Biomembranes and Fluid-Membrane Interaction

R.H. Nochetto (1,2)
Joint with A. Bonito (3) and M. S. Pauletti (1)

Department of Mathematics (1) and Institute for Physical Science and Technology (2)
University of Maryland
Department of Mathematics, Texas A&M (3)

New Directions in Computational PDEs

Outline

Basic Models

Refs: Helfrich (73), Jenkins (77), Seifert (97)

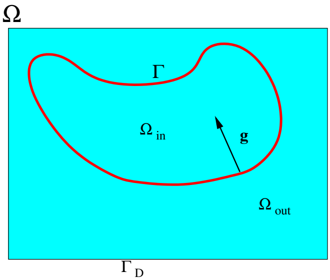
Assumptions

Membrane:

- Layer of Incompressible Fluid
- Bending Rigidity
- Non Permeable

Surrounding Fluid:

- Newtonian, Incompressible.



Basic Models

Refs: Helfrich (73), Jenkins (77), Seifert (97)

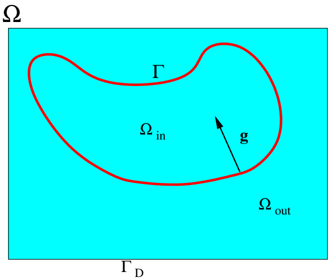
Assumptions

Membrane:

- Layer of Incompressible Fluid
- Bending Rigidity
- Non Permeable

Surrounding Fluid:

- Newtonian, Incompressible.



Gradient Flow

- Decreasing Energy
- Area and Volume Constraint
- Non Physical Dynamics

Fluid-Membrane Interaction

- Fluid Equations of Motion
- Immerse and Exerts Forces
- Membrane moves with Fluid

Basic Models

Refs: Helfrich (73), Jenkins (77), Seifert (97)

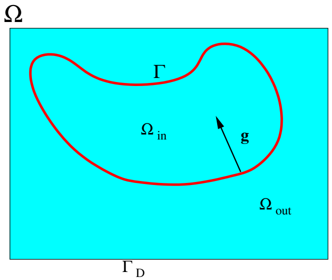
Assumptions

Membrane:

- Layer of Incompressible Fluid
- Bending Rigidity
- Non Permeable

Surrounding Fluid:

- Newtonian, Incompressible.



Gradient Flow

- Decreasing Energy
- Area and Volume Constraint
- Non Physical Dynamics

Fluid-Membrane Interaction

- Fluid Equations of Motion
- Immerse and Exerts Forces
- Membrane moves with Fluid

Basic Models

Refs: Helfrich (73), Jenkins (77), Seifert (97)

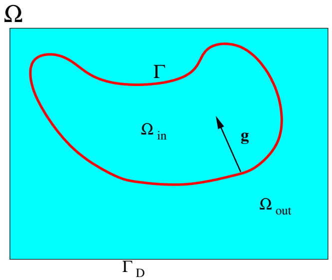
Assumptions

Membrane:

- Layer of Incompressible Fluid
- Bending Rigidity
- Non Permeable

Surrounding Fluid:

- Newtonian, Incompressible.



Gradient Flow

- Decreasing Energy
- Area and Volume Constraint
- Non Physical Dynamics

Fluid-Membrane Interaction

- Fluid Equations of Motion
- Immerse and Exerts Forces
- Membrane moves with Fluid

Basic Models

Refs: Helfrich (73), Jenkins (77), Seifert (97)

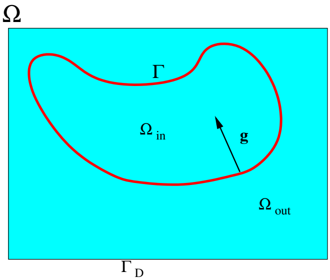
Assumptions

Membrane:

- Layer of Incompressible Fluid
- Bending Rigidity
- Non Permeable

Surrounding Fluid:

- Newtonian, Incompressible.



Gradient Flow

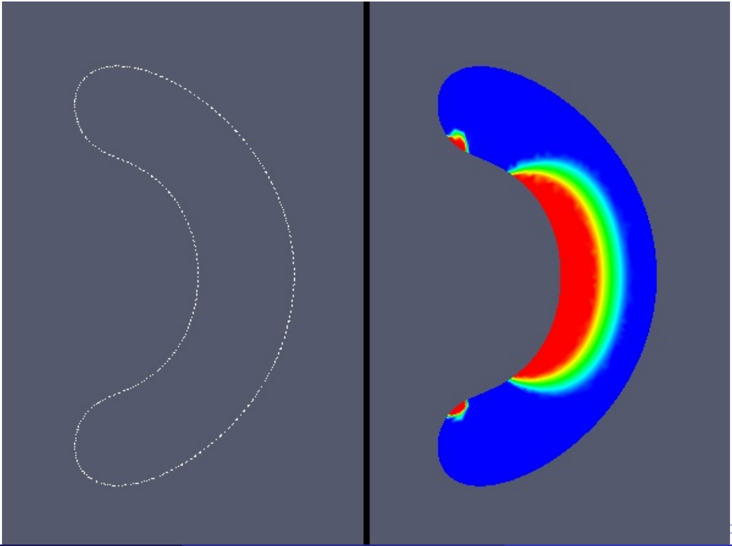
- Decreasing Energy
- Area and Volume Constraint
- Non Physical Dynamics

Fluid-Membrane Interaction

- Fluid Equations of Motion
- Immerse and Exerts Forces
- Membrane moves with Fluid

Geometric Model vs. Fluid-Membrane Model

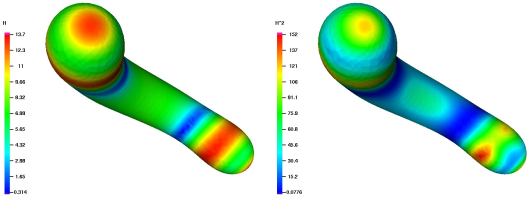
play



Bending energy (Refs: Kuwert, Schätzle, Simonett, Willmore)

$$W(\Gamma) = \int_\Gamma h^2$$

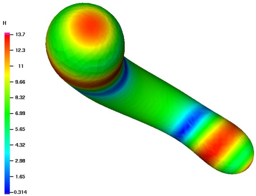
$$\delta W = \left(\Delta_\Gamma h + \frac{1}{2} h^3 - 2kh \right) \nu$$

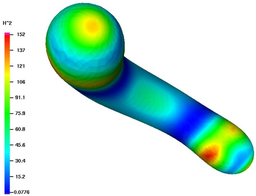


Bending energy (Refs: Kuwert, Schätzle, Simonett, Willmore)

$$J(\Gamma) = \int_\Gamma h^2 + \lambda \int_\Gamma 1 + p \int_\Gamma \mathrm{Id} \cdot \nu$$

$$\delta J = \left(\Delta_\Gamma h + \frac{1}{2} h^3 - 2kh \right) \nu + \lambda \mathbf{h} + p\nu$$





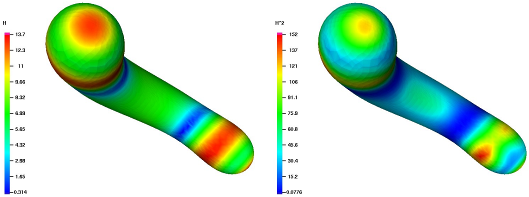
Constraints

Augmented Energy J, Lagrange Multipliers λ (area) and p (volume)

Bending energy (Refs: Kuwert, Schätzle, Simonett, Willmore)

$$J(\Gamma) = \int_\Gamma h^2 + \lambda \int_\Gamma 1 + p \int_\Gamma \mathrm{Id} \cdot \nu$$

$$\delta J = \left(\Delta_\Gamma h + \frac{1}{2} h^3 - 2kh \right) \nu + \lambda \mathbf{h} + p\nu$$



Constraints

Augmented Energy J, Lagrange Multipliers λ (area) and p (volume)

Gradient Flow for Biomembrane Modeling

Trajectory

$$G_T := \{(\mathbf{x}, t) : \mathbf{x} \in \Gamma(t), t \in [0, T]\}$$

Gradient Flow for Biomembrane Modeling

Trajectory

$$G_T := \{(\mathbf{x}, t) : \mathbf{x} \in \Gamma(t), t \in [0, T]\}$$

Weak Helfrich flow

Given Γ^0 and $T > 0$, find $\mathrm{rx} : G_T \to \mathbb{R}^{d+1}$, $\lambda : [0, T] \to \mathbb{R}$ and $p : [0, T] \to \mathbb{R}$ such that $\Gamma(0) = \Gamma^0$ and for all $t \in (0, T]$

$$\int_{\Gamma(t)} \dot{\mathrm{x}} \cdot \phi = \; - \delta J(\Gamma(t); \phi)$$

for all smooth $\phi : \Gamma(t) \to \mathbb{R}^{d+1}$ and

$$A(\Gamma(t)) = A(\Gamma(0)), \quad V(\Gamma(t)) = V(\Gamma(0)).$$

Gradient Flow for Biomembrane Modeling

Trajectory

$$G_T := \{(\mathbf{x}, t) : \mathbf{x} \in \Gamma(t), t \in [0, T]\}$$

Weak Helfrich flow

Given Γ^0 and $T > 0$, find $\mathrm{rx} : G_T \to \mathbb{R}^{d+1}$, $\lambda : [0, T] \to \mathbb{R}$ and $p : [0, T] \to \mathbb{R}$ such that $\Gamma(0) = \Gamma^0$ and for all $t \in (0, T]$

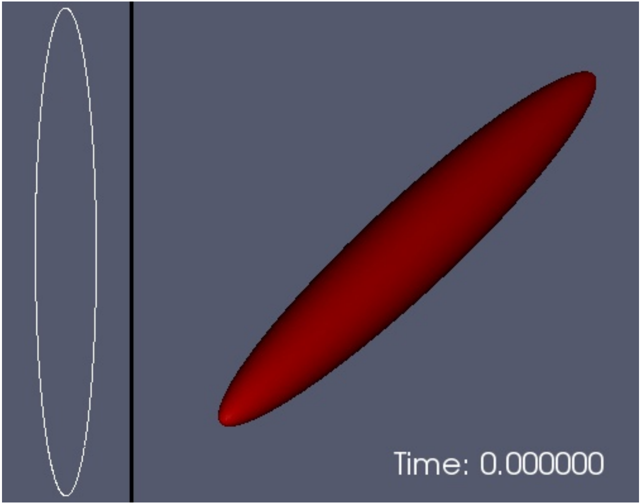
$$\int_{\Gamma(t)} \dot{\mathrm{x}} \cdot \phi = -\delta W(\Gamma(t); \phi) - \lambda\, \delta A(\Gamma(t); \phi) - p\, \delta V(\Gamma(t); \phi)$$

for all smooth $\phi : \Gamma(t) \to \mathbb{R}^{d+1}$ and

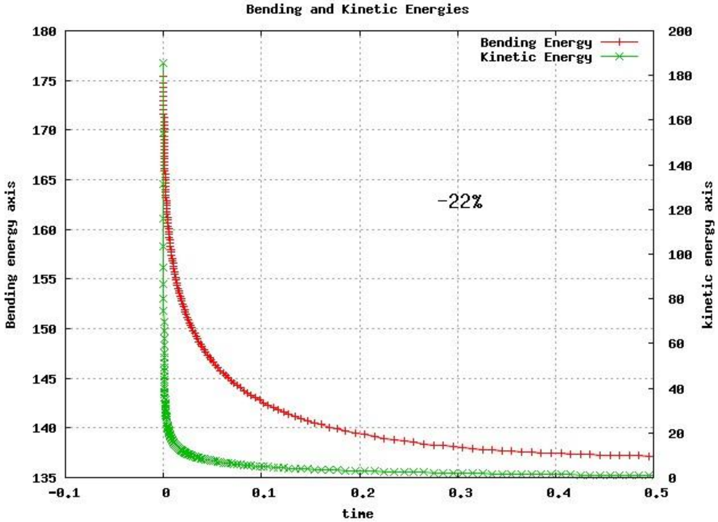
$$A(\Gamma(t)) = A(\Gamma(0)), \quad V(\Gamma(t)) = V(\Gamma(0)).$$

Dumbbell Bar Shaped Family

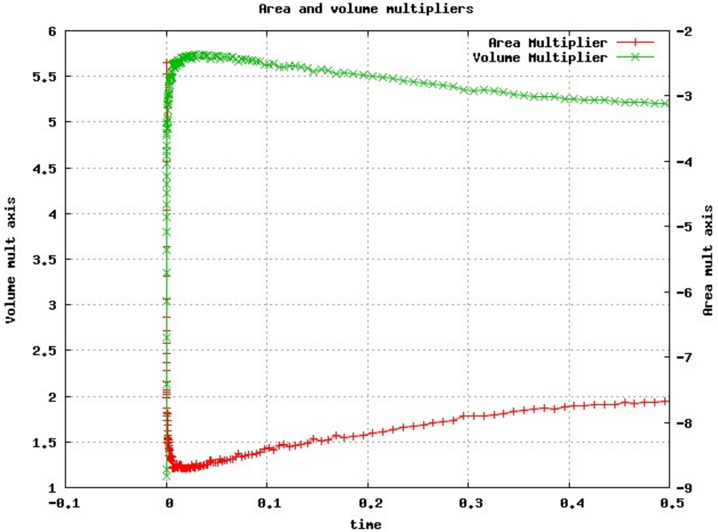
Ellipsoid 8x1x1



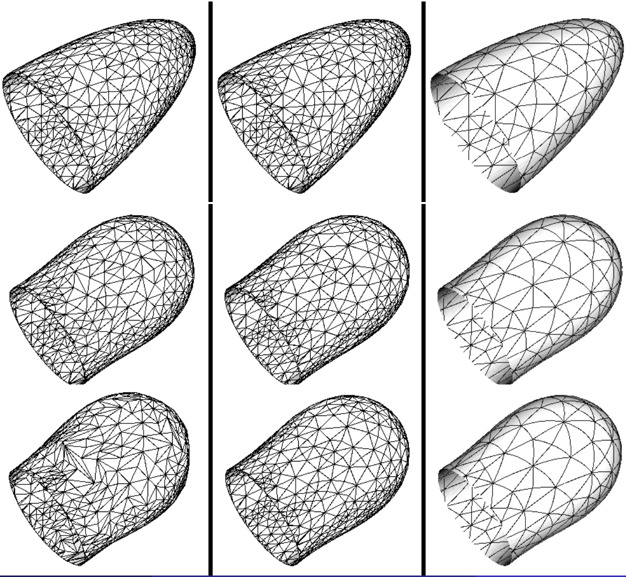
Dumbbell Bar Shaped Family - Ellipsoid 8x1x1: Energies



Dumbbell Bar Shaped Family - Ellipsoid 8x1x1: Multipliers

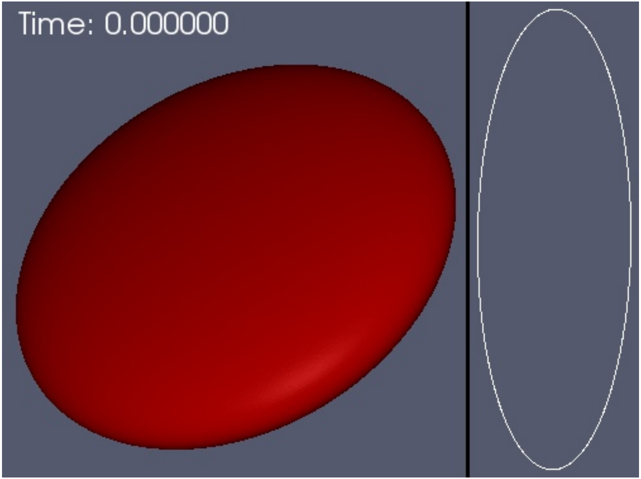


Mesh Smoothing: Comparison between P^1 and P^2



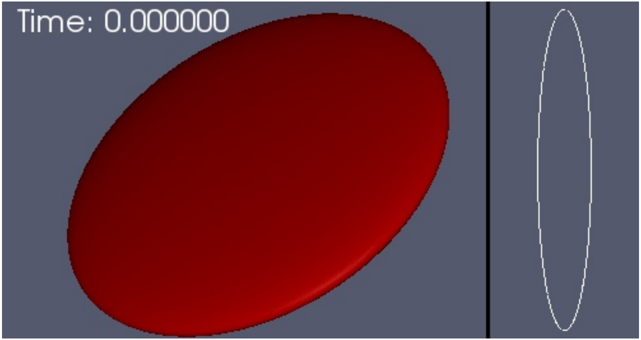
Red Bood Cell Family

Ellipsoid 3x3x1



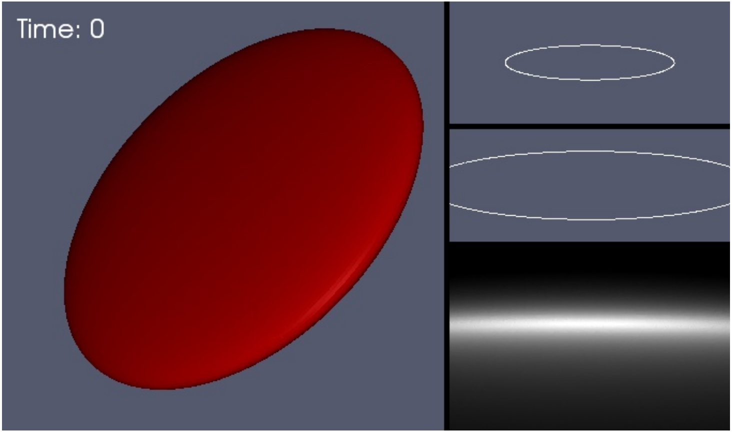
Red Bood Cell Family (Continued)

Ellipsoid 6x6x1



Red Bood Cell Family (Continued)

Ellipsoid 5x5x1



Shape Derivative of Willmore Energy δW: Vector Form

Refs: Rusu (2005), Dziuk (2008), Barrett-Garcke-Nürnberg (2008)

Shape Calculus: $\boldsymbol{\nu}'(\Gamma, \boldsymbol{\phi}) = -\nabla_\Gamma \phi, \quad h'(\Gamma, \boldsymbol{\phi}) = -\Delta_\Gamma \phi, \quad \partial_\nu h = -|\nabla_\Gamma \boldsymbol{\nu}|^2$

$$\delta W(\Gamma, \phi) = \int_\Gamma \nabla_\Gamma \phi \cdot \nabla_\Gamma h - \int_\Gamma h |\nabla_\Gamma \boldsymbol{\nu}|^2 \phi + \frac{1}{2} \int_\Gamma h^3 \phi; \qquad \phi = \boldsymbol{\phi} \cdot \boldsymbol{\nu}$$

Shape Derivative of Willmore Energy δW: Vector Form

Refs: Rusu (2005), Dziuk (2008), Barrett-Garcke-Nürnberg (2008)

Shape Calculus: $\boldsymbol{\nu}'(\Gamma, \boldsymbol{\phi}) = -\nabla_\Gamma \phi, \quad h'(\Gamma, \boldsymbol{\phi}) = -\Delta_\Gamma \phi, \quad \partial_\nu h = -|\nabla_\Gamma \boldsymbol{\nu}|^2$

$$\delta W(\Gamma, \phi) = \int_\Gamma \nabla_\Gamma \phi \cdot \nabla_\Gamma h - \int_\Gamma h |\nabla_\Gamma \boldsymbol{\nu}|^2 \phi + \frac{1}{2} \int_\Gamma h^3 \phi; \qquad \phi = \boldsymbol{\phi} \cdot \boldsymbol{\nu}$$

$$\int_\Gamma \nabla_\Gamma \phi \cdot \nabla_\Gamma h = \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \mathbf{h} - \int_\Gamma (\nabla_\Gamma \mathbf{x} + \nabla_\Gamma \mathbf{x}^T) \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \mathbf{h} - \int_\Gamma h \Delta_\Gamma \boldsymbol{\nu} \cdot \boldsymbol{\phi}$$

$$-\int_\Gamma h \Delta_\Gamma \boldsymbol{\nu} \cdot \boldsymbol{\phi} = \int_\Gamma h |\nabla_\Gamma \boldsymbol{\nu}|^2 - \frac{1}{2} \int_\Gamma \nabla_\Gamma h^2 \cdot \boldsymbol{\phi}$$

Shape Derivative of Willmore Energy δW: Vector Form

Refs: Rusu (2005), Dziuk (2008), Barrett-Garcke-Nürnberg (2008)

Shape Calculus: $\boldsymbol{\nu}'(\Gamma,\boldsymbol{\phi}) = -\nabla_\Gamma \phi$, $\quad h'(\Gamma,\boldsymbol{\phi}) = -\Delta_\Gamma \phi$, $\quad \partial_\nu h = -|\nabla_\Gamma \boldsymbol{\nu}|^2$

$$\delta W(\Gamma,\phi) = \int_\Gamma \nabla_\Gamma \phi \cdot \nabla_\Gamma h - \int_\Gamma h|\nabla_\Gamma \boldsymbol{\nu}|^2 \phi + \frac{1}{2}\int_\Gamma h^3 \phi; \qquad \phi = \boldsymbol{\phi}\cdot\boldsymbol{\nu}$$

$$\int_\Gamma \nabla_\Gamma \phi \cdot \nabla_\Gamma h = \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \mathbf{h} - \int_\Gamma (\nabla_\Gamma \mathbf{x} + \nabla_\Gamma \mathbf{x}^T)\nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \mathbf{h} - \int_\Gamma h \Delta_\Gamma \boldsymbol{\nu}\cdot\boldsymbol{\phi}$$

$$-\int_\Gamma h\Delta_\Gamma \boldsymbol{\nu}\cdot\boldsymbol{\phi} = \int_\Gamma h|\nabla_\Gamma \boldsymbol{\nu}|^2 - \frac{1}{2}\int_\Gamma \nabla_\Gamma h^2 \cdot \boldsymbol{\phi}$$

$$dW(\Gamma;\phi) = \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \mathbf{h} - \int_\Gamma (\nabla_\Gamma \mathbf{x} + \nabla_\Gamma \mathbf{x}^T)\nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \mathbf{h} + \frac{1}{2}\int_\Gamma \nabla_\Gamma \cdot \mathbf{h} \nabla_\Gamma \cdot \boldsymbol{\phi}$$

Discrete Geometric Scheme: Ingredients

- Curvature: $\mathbf{h} = -\Delta_\Gamma \mathbf{x}, \quad \mathbf{x} = \text{identity on } \Gamma$ (Dziuk' 91)
- Semi-implicit Time Discretization $(t_n \to t_{n+1})$: explicit geometry $(\Gamma = \Gamma_n, \quad \nabla_\Gamma = \nabla_{\Gamma^n}, \quad \boldsymbol{\nu} = \boldsymbol{\nu}^n)$ (Dziuk' 91)

$$\int_{\Gamma^n} \mathbf{h}^{n+1} \cdot \boldsymbol{\psi} = \int_{\Gamma^n} \nabla_{\Gamma^n} \mathbf{x}^{n+1} : \nabla_{\Gamma^n} \boldsymbol{\psi}, \qquad \mathbf{x}^{n+1} = \mathbf{x}^n + \tau^n \mathbf{v}^{n+1}$$

$$\Rightarrow \quad \int_{\Gamma^n} \mathbf{h}^{n+1} \cdot \boldsymbol{\psi} - \tau^n \int_{\Gamma^n} \nabla_{\Gamma^n} \mathbf{v}^{n+1} : \nabla_{\Gamma^n} \boldsymbol{\psi} = \int_{\Gamma^n} \nabla_{\Gamma^n} \mathbf{x}^n : \nabla_{\Gamma^n} \boldsymbol{\psi}$$

- Mixed Method: operator splitting
  1. Velocity: $\int_{\Gamma^n} \mathbf{v}^{n+1} \cdot \boldsymbol{\phi} = -\delta J^{n+1}(\Gamma^n; \boldsymbol{\phi})$
  2. Curvature: $\int_{\Gamma^n} \mathbf{h}^{n+1} \cdot \boldsymbol{\psi} - \tau^n \int_{\Gamma^n} \nabla_{\Gamma^n} \mathbf{v}^{n+1} : \nabla_{\Gamma^n} \boldsymbol{\psi} = \int_{\Gamma^n} \nabla_{\Gamma^n} \mathbf{x}^n : \nabla_{\Gamma^n} \boldsymbol{\psi}$
- Space discretization: linears or quadratics

Fully Discrete Geometric Scheme

$$\int_{\Gamma_h^n} \mathbf{V}^{n+1} \cdot \mathbf{\Phi} = -dW_h^{n+1}(\Gamma_h^n; \mathbf{\Phi}) - \lambda^{n+1} dA_h^n(\Gamma_h^n; \mathbf{\Phi}) - p^{n+1} dV_h^n(\Gamma_h^n; \mathbf{\Phi})$$

$$\int_{\Gamma_h^n} \mathbf{H}^{n+1} \cdot \mathbf{\Psi} - \tau_n \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \mathbf{V}^{n+1} : \nabla_{\Gamma_h^n} \mathbf{\Psi} = \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \mathbf{x}^n : \nabla_{\Gamma_h^n} \mathbf{\Psi},$$

for all $\mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{S}_h^{n+1}$,

Fully Discrete Geometric Scheme

$$\int_{\Gamma_h^n} \mathbf{V}^{n+1} \cdot \mathbf{\Phi} = -dW_h^{n+1}(\Gamma_h^n; \mathbf{\Phi}) - \lambda^{n+1} dA_h^n(\Gamma_h^n; \mathbf{\Phi}) - p^{n+1} dV_h^n(\Gamma_h^n; \mathbf{\Phi})$$

$$\int_{\Gamma_h^n} \mathbf{H}^{n+1} \cdot \mathbf{\Psi} - \tau_n \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \mathbf{V}^{n+1} : \nabla_{\Gamma_h^n} \mathbf{\Psi} = \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \mathbf{x}^n : \nabla_{\Gamma_h^n} \mathbf{\Psi},$$

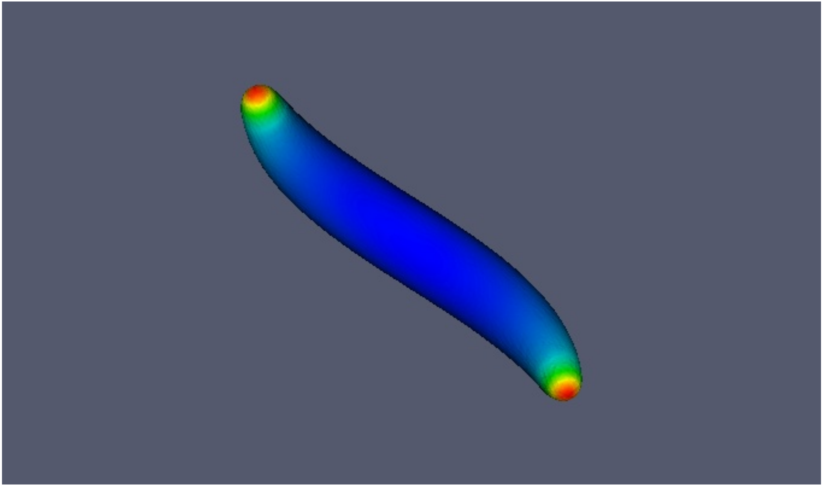
for all $\mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{S}_h^{n+1}$, where

$$dW_h^{n+1}(\Gamma_h^n; \mathbf{\Phi}) = \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \mathbf{H}^{n+1} : \nabla_{\Gamma_h^n} \mathbf{\Phi} + \frac{1}{2} \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \cdot \mathbf{H}^{n+1} \nabla_{\Gamma_h^n} \cdot \mathbf{\Phi}$$
$$- \int_{\Gamma_h^n} (\nabla_{\Gamma_h^n} \mathbf{x}^n + \nabla_{\Gamma_h^n} \mathbf{x}^{n\mathsf{T}}) \nabla_{\Gamma_h^n} \mathbf{\Phi} : \nabla_{\Gamma_h^n} \mathbf{H}^{n+1},$$

$$dA_h^n(\Gamma_h^n; \mathbf{\Phi}) = \int_{\Gamma_h^n} \mathbf{H}^n \cdot \mathbf{\Phi}, \qquad dV_h^n(\Gamma_h^n; \mathbf{\Phi}) = \int_{\Gamma_h^n} \mathbf{\Phi} \cdot \nu.$$

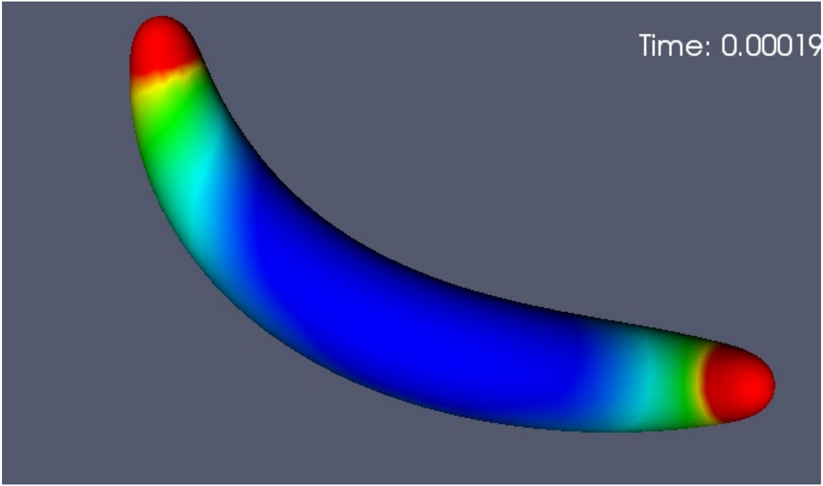
Boomerang Shape

Twisted Banana



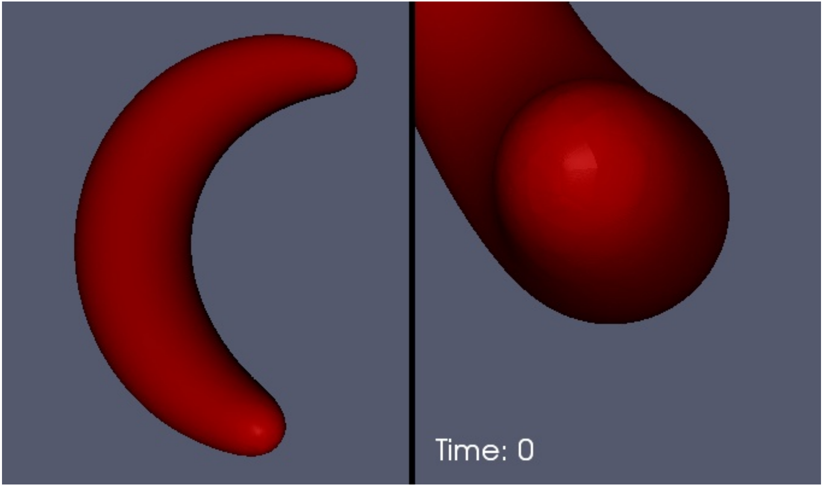
Boomerang Shape: Full Simulation

Full Simulation



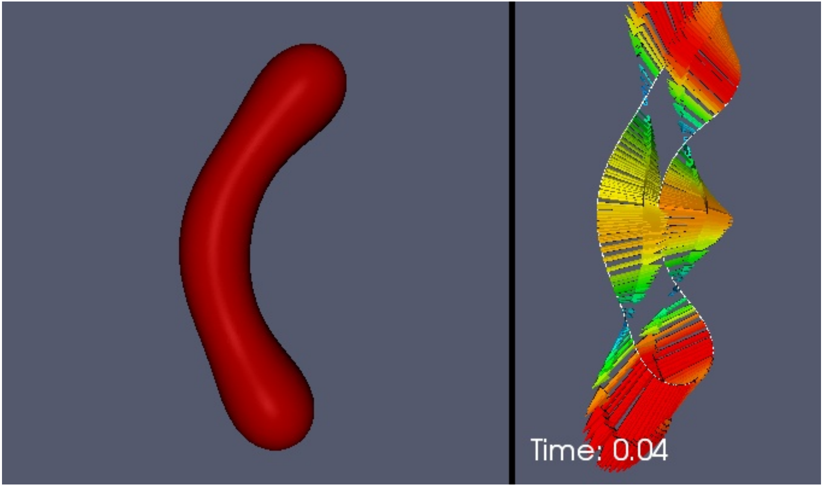
Boomerang Shape: Fast Time Scale

Fast time scale

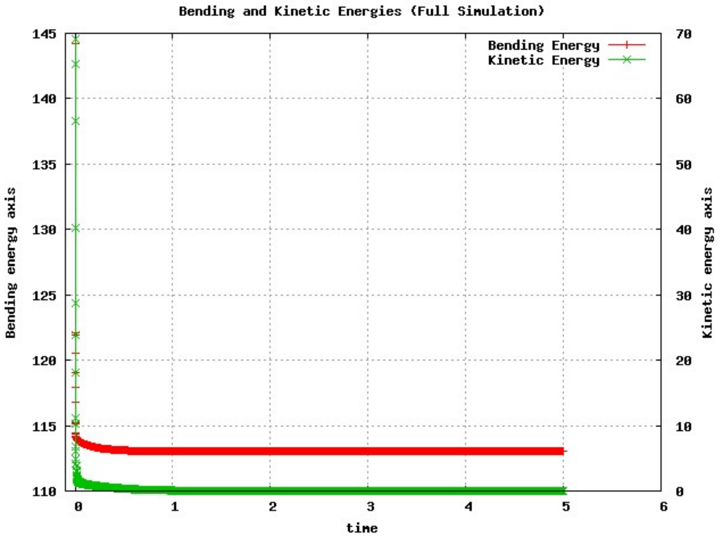


Boomerang Shape: Slow Time Scale

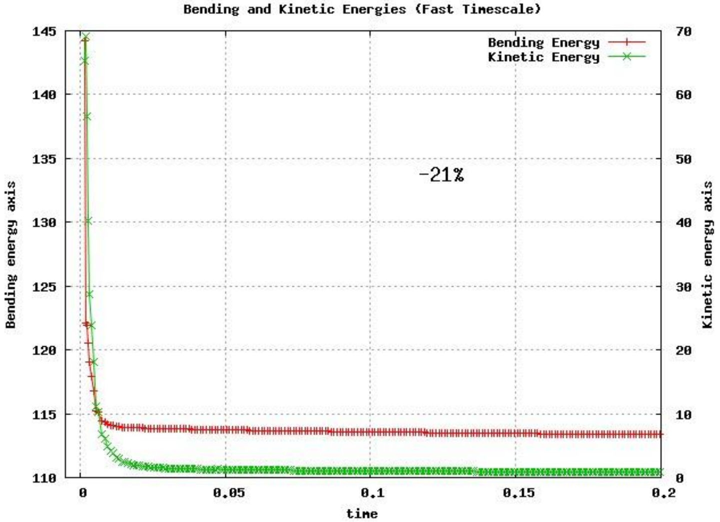
Slow time scale



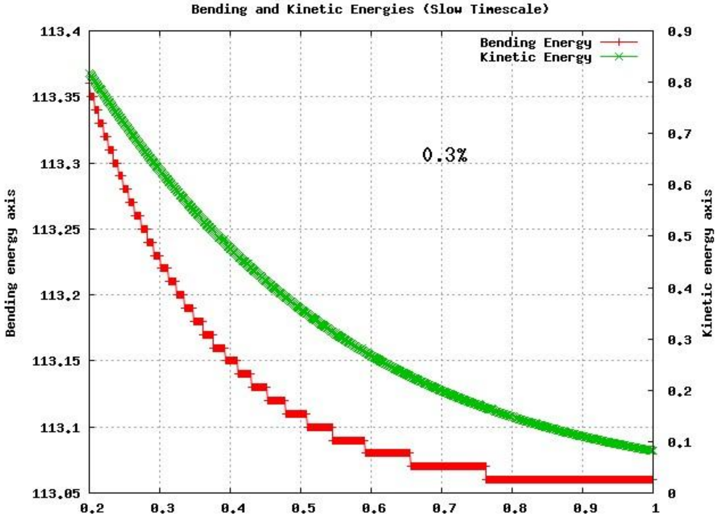
Boomerang Shape - Energy Graphs: Full



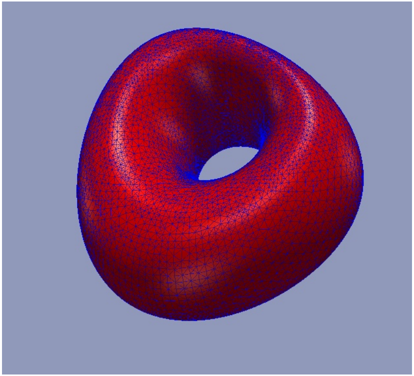
Boomerang Shape - Energy Graphs: Fast Time

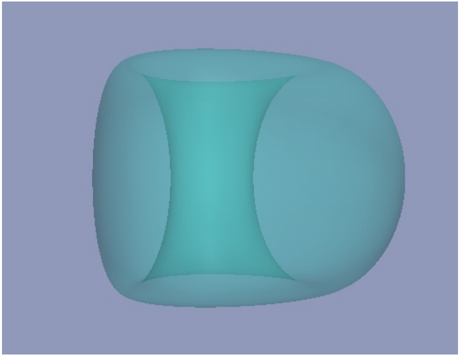


Boomerang Shape - Energy Graphs: Slow Time



Non-axisymmetric Torus





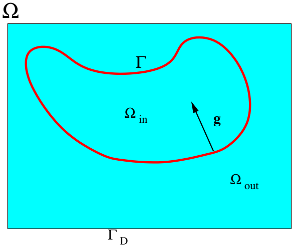
Coupled Fluid-Membrane Model

Refs: Coutand-Shkoller

(Incompressible) Navier-Stokes Equations

$$\rho\dot{\mathbf{v}} - \nabla\cdot\underbrace{(-p\mathbf{I} + \mu D(\mathbf{v}))}_{\Sigma} = \mathbf{b} \qquad \text{in } \Omega_t,$$

$$\nabla\cdot\mathbf{v} = 0 \qquad \text{in } \Omega_t,$$



Coupled Fluid-Membrane Model

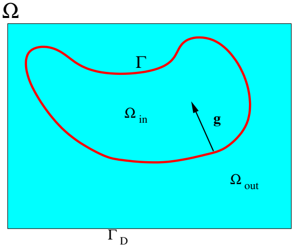
Refs: Coutand-Shkoller

(Incompressible) Navier-Stokes Equations

$$\rho\dot{\mathbf{v}} - \nabla\cdot(\underbrace{-p\mathbf{I} + \mu D(\mathbf{v})}_{\Sigma}) = \mathbf{b} \qquad \text{in } \Omega_t,$$

$$\nabla\cdot\mathbf{v} = 0 \qquad \text{in } \Omega_t,$$

$$[\boldsymbol{\Sigma}]\nu = \delta J \qquad \text{on } \Gamma_t,$$



Membrane Force: Bending

$$\delta J = \left(\Delta_\Gamma h + \frac{1}{2}h^3 - 2kh\right)\boldsymbol{\nu} + \lambda\mathbf{h}.$$

Coupled Fluid-Membrane Model

Refs: Coutand-Shkoller

(Incompressible) Navier-Stokes Equations

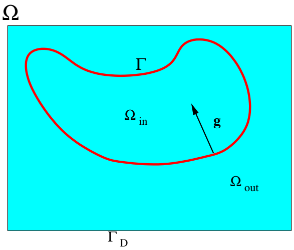
$$\rho\dot{\mathbf{v}} - \nabla\cdot\underbrace{(-p\mathbf{I} + \mu D(\mathbf{v}))}_{\Sigma} = \mathbf{b} \qquad \text{in } \Omega_t,$$

$$\nabla\cdot\mathbf{v} = 0 \qquad \text{in } \Omega_t,$$

$$[\boldsymbol{\Sigma}]\nu = \delta J \qquad \text{on } \Gamma_t,$$

$$\mathbf{v} = \vartheta \qquad \text{on } \Gamma_t,$$

$$\mathbf{v}(\cdot,0) = \mathbf{v}_0 \qquad \text{in } \Omega_0.$$

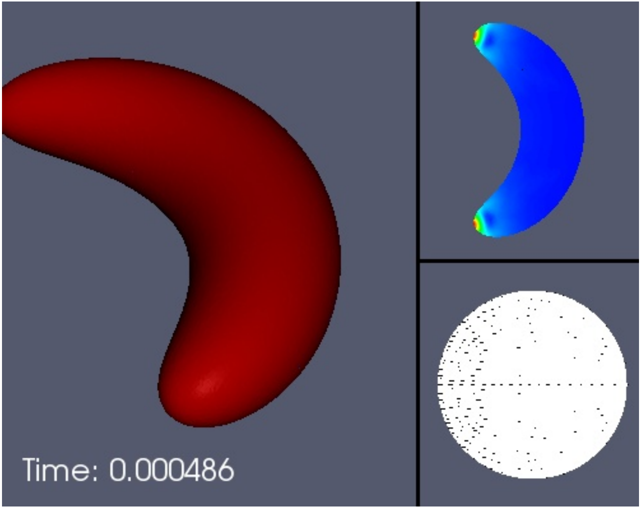


Membrane Force: Bending

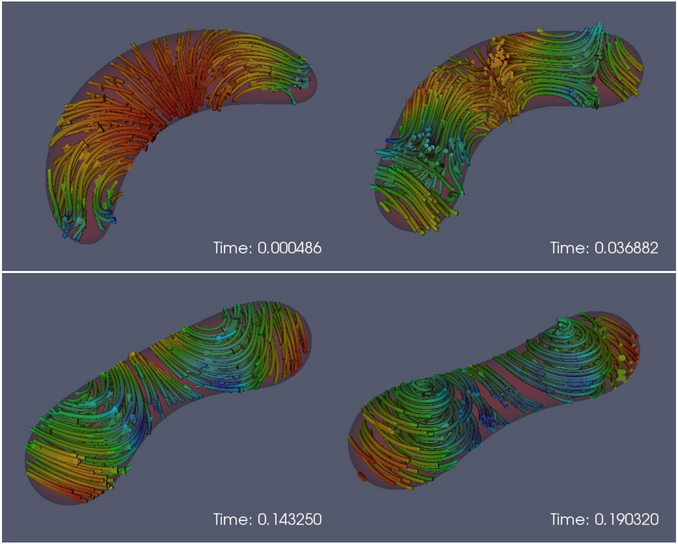
$$\delta J = \left(\Delta_\Gamma h + \frac{1}{2}h^3 - 2kh\right)\boldsymbol{\nu} + \lambda\mathbf{h}.$$

Fluid Biomembrane: 3D Boomerang

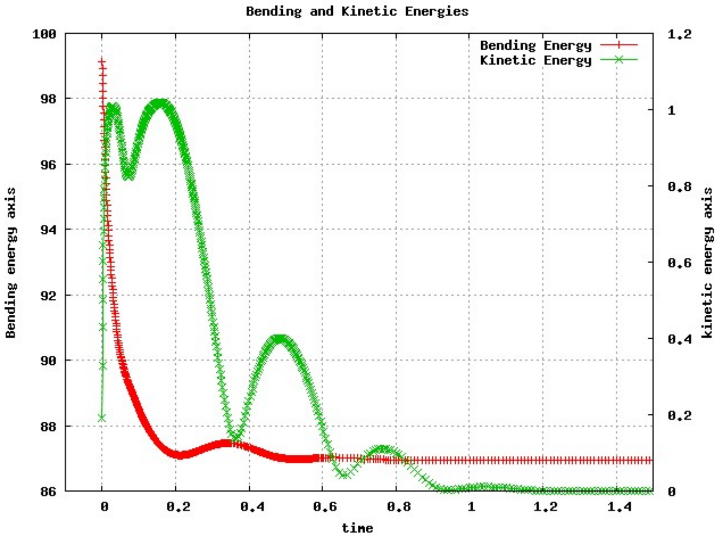
3D Banana



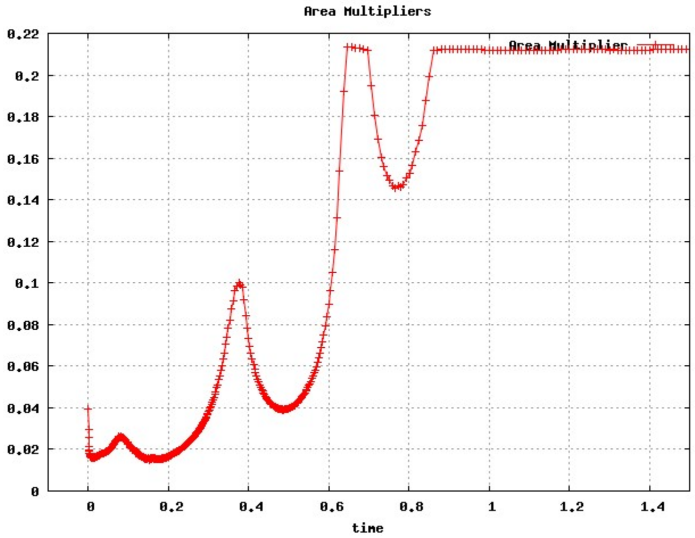
Fluid Biomembrane: 3D Boomerang Streamlines



Fluid Biomembrane: 3D Boomerang Energies

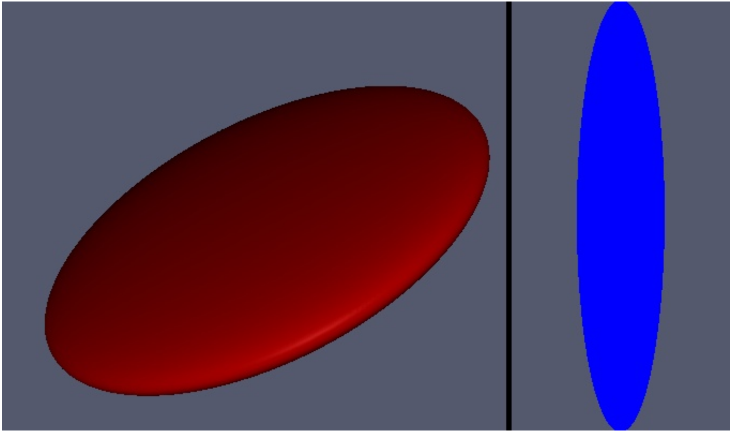


Fluid Biomembrane: 3D Boomerang Multiplier

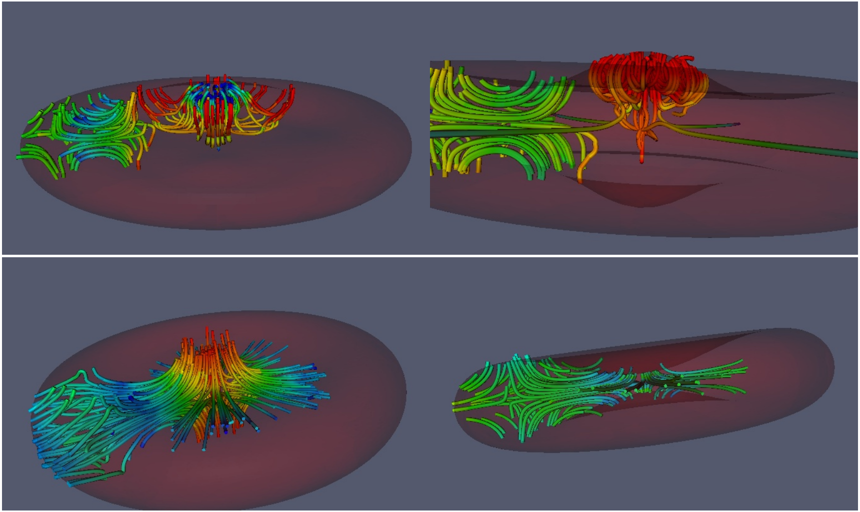


Fluid Red Blood Cell: Ellipsoid 5x5x1

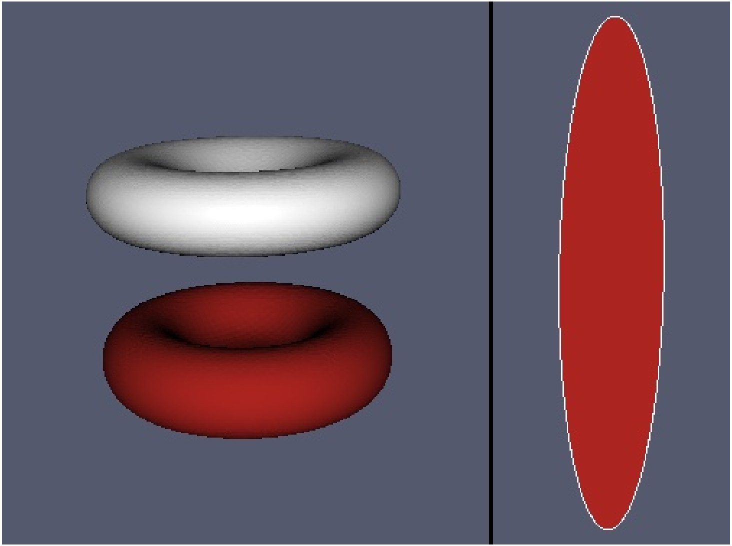
Ellipsoid 5x5x1



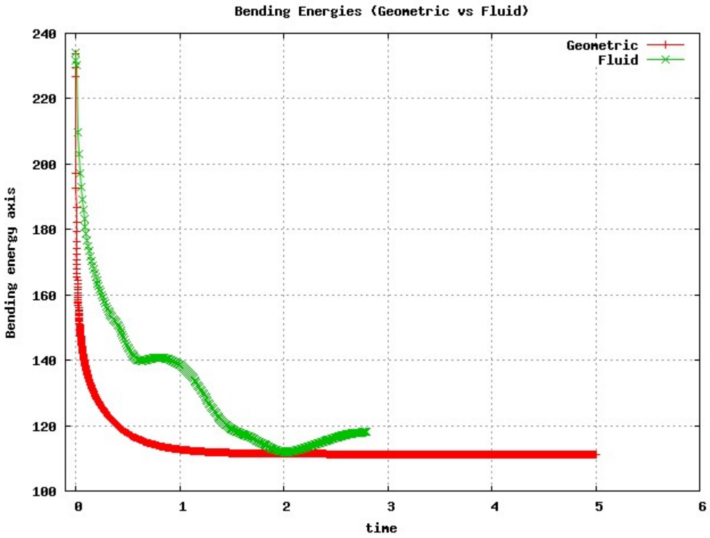
Fluid Red Blood Cell: Ellipsoid 5x5x1 Streamlines



Geometric vs Fluid Red Cell: 5x5x1 Ellipsoid

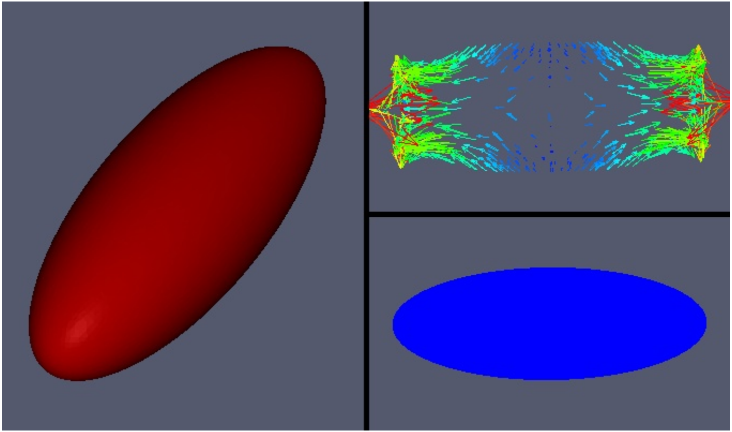


Geometric vs Fluid Red Cell - 5x5x1 Ellipsoid: Energies

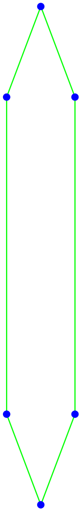


Fluid Biomembrane: Ellipsoid 4x1x1

Ellipsoid 4x1x1



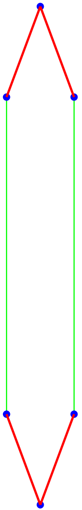
Refinement Consistent Refinement



Problem

- Free boundary problem
- Γ is major unknown
- Increase local resolution

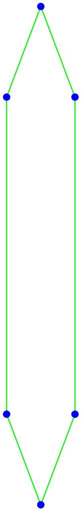
Refinement Consistent Refinement



Problem

- Free boundary problem
- Γ is major unknown
- Increase local resolution

Refinement Consistent Refinement



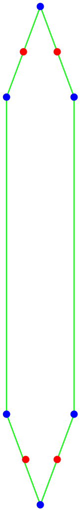
Problem

- Free boundary problem
- Γ is major unknown
- Increase local resolution

Question

How to add local resolution with incomplete geometric information of Γ?

Refinement Consistent Refinement



Problem

- Free boundary problem
- Γ is major unknown
- Increase local resolution

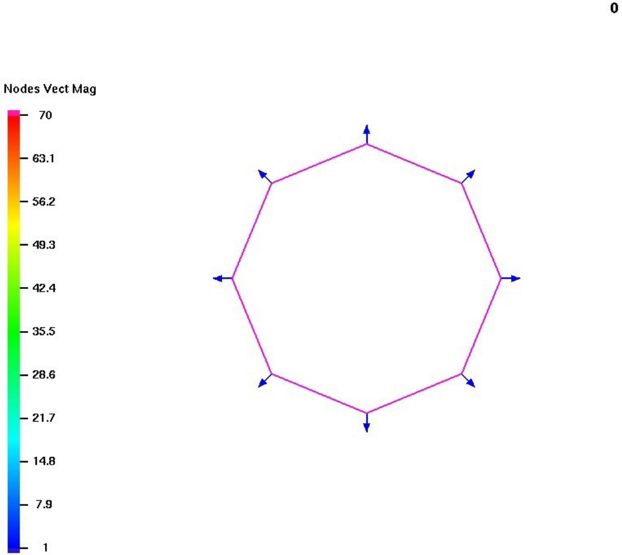
Question

How to add local resolution with incomplete geometric information of Γ?

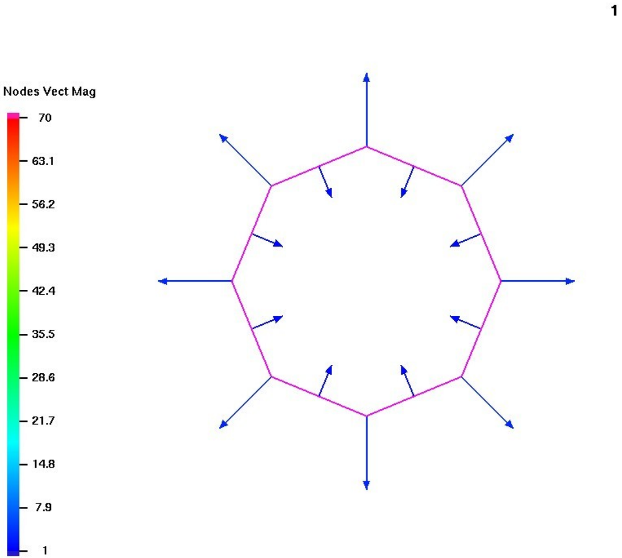
Answer 1

Linear interpolation

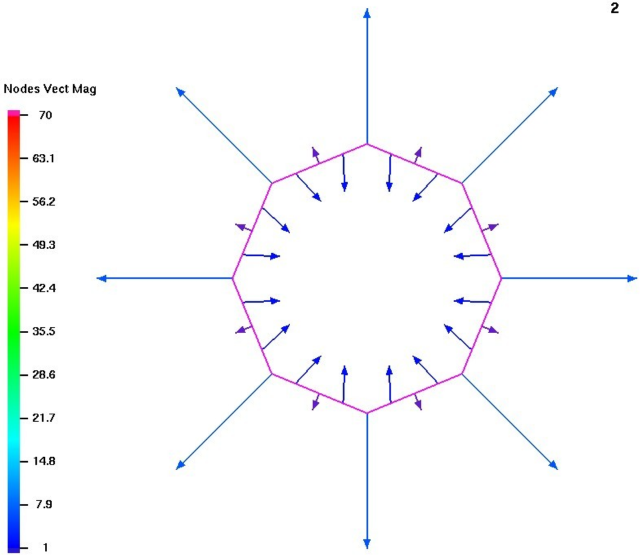
Counterxample to Answer 1



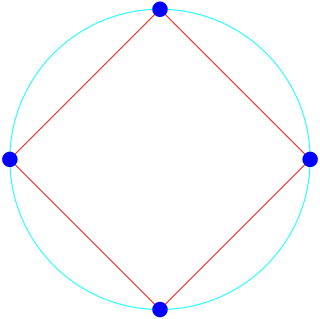
Counterxample to Answer 1



Counterxample to Answer 1



Understanding the Counterexample



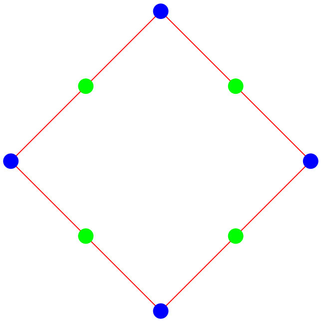
Perform linear interpolation:

- Smooth curve γ and polygonal approximation Γ
- Refine by linear interpolation
- Pass a smooth curve $\tilde{\gamma}$ through all interpolation points

For 1D curves the FEM theory in flat domain extends:

$$\int_\Gamma \mathbf{H}\cdot\phi = \int_\Gamma \partial_s \mathbf{X}\cdot\partial_s\phi = -\int_\Gamma \partial_s^2\tilde{\mathbf{x}}\cdot\phi \quad \forall\phi\in\mathbb{S}_h \quad \Rightarrow \quad \mathbf{H} = P_h(\partial_s^2\tilde{\mathbf{x}})$$

Understanding the Counterexample



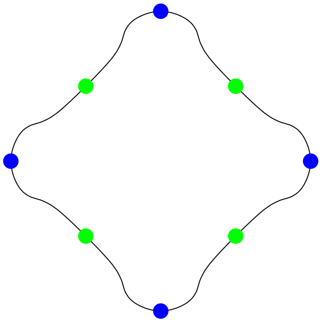
Perform linear interpolation:

- Smooth curve γ and polygonal approximation Γ
- Refine by linear interpolation
- Pass a smooth curve $\tilde{\gamma}$ through all interpolation points

For 1D curves the FEM theory in flat domain extends:

$$\int_\Gamma \mathbf{H} \cdot \phi = \int_\Gamma \partial_s \mathbf{X} \cdot \partial_s \phi = -\int_\Gamma \partial_s^2 \tilde{\mathbf{x}} \cdot \phi \quad \forall \phi \in \mathbb{S}_h \quad \Rightarrow \quad \mathbf{H} = P_h(\partial_s^2 \tilde{\mathbf{x}})$$

Understanding the Counterexample



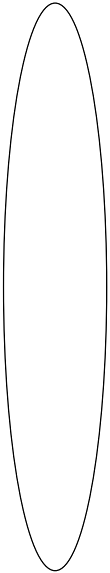
Perform linear interpolation:

- Smooth curve γ and polygonal approximation Γ
- Refine by linear interpolation
- Pass a smooth curve $\tilde{\gamma}$ through all interpolation points

For 1D curves the FEM theory in flat domain extends:

$$\int_\Gamma \mathbf{H} \cdot \phi = \int_\Gamma \partial_s \mathbf{X} \cdot \partial_s \phi = -\int_\Gamma \partial_s^2 \tilde{\mathbf{x}} \cdot \phi \quad \forall \phi \in \mathbb{S}_h \quad \Rightarrow \quad \mathbf{H} = P_h(\partial_s^2 \tilde{\mathbf{x}})$$

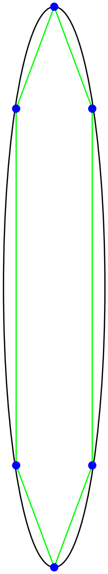
Exact Interpolation in 2D



Assume we know γ and interpolate it exactly

- γ smooth curve (surface)
- Γ polyhedral approximation of γ
- Refine locally γ
- Refine Γ by bisection
- Project new node to γ

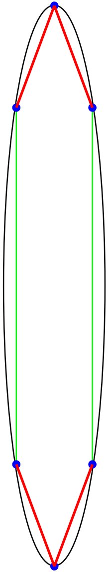
Exact Interpolation in 2D



Assume we know γ and interpolate it exactly

- γ smooth curve (surface)
- Γ polyhedral approximation of γ
- Refine locally γ
- Refine Γ by bisection
- Project new node to γ

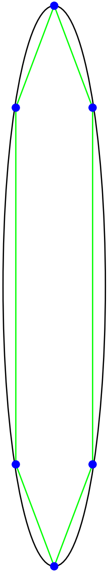
Exact Interpolation in 2D



Assume we know γ and interpolate it exactly

- γ smooth curve (surface)
- Γ polyhedral approximation of γ
- Refine locally γ
- Refine Γ by bisection
- Project new node to γ

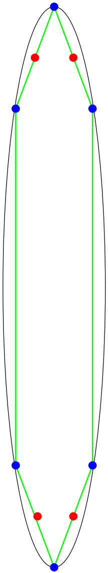
Exact Interpolation in 2D



Assume we know γ and interpolate it exactly

- γ smooth curve (surface)
- Γ polyhedral approximation of γ
- Refine locally γ
- Refine Γ by bisection
- Project new node to γ

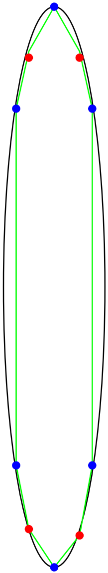
Exact Interpolation in 2D



Assume we know γ and interpolate it exactly

- γ smooth curve (surface)
- Γ polyhedral approximation of γ
- Refine locally γ
- Refine Γ by bisection
- Project new node to γ

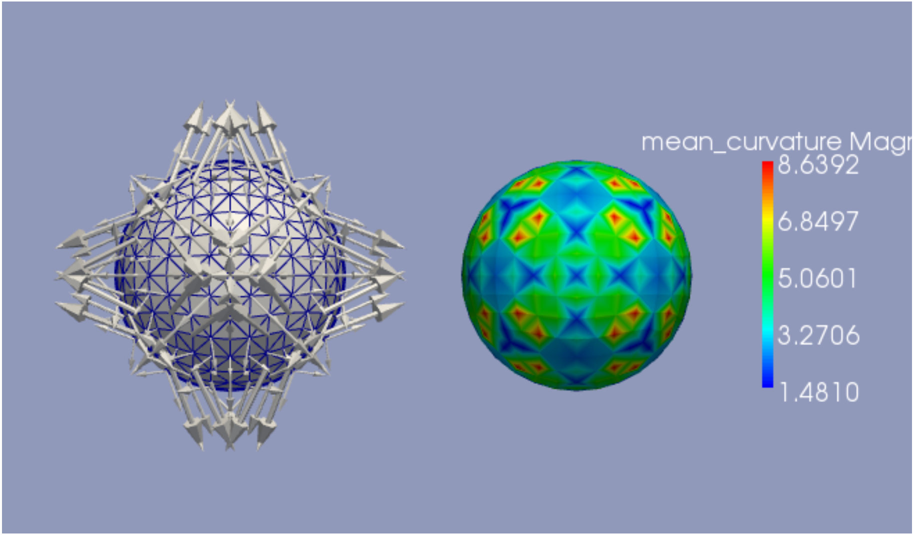
Exact Interpolation in 2D



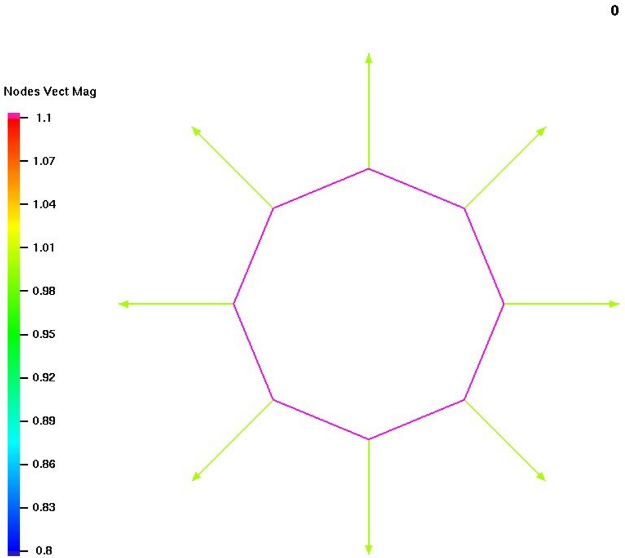
Assume we know γ and interpolate it exactly

- γ smooth curve (surface)
- Γ polyhedral approximation of γ
- Refine locally γ
- Refine Γ by bisection
- Project new node to γ

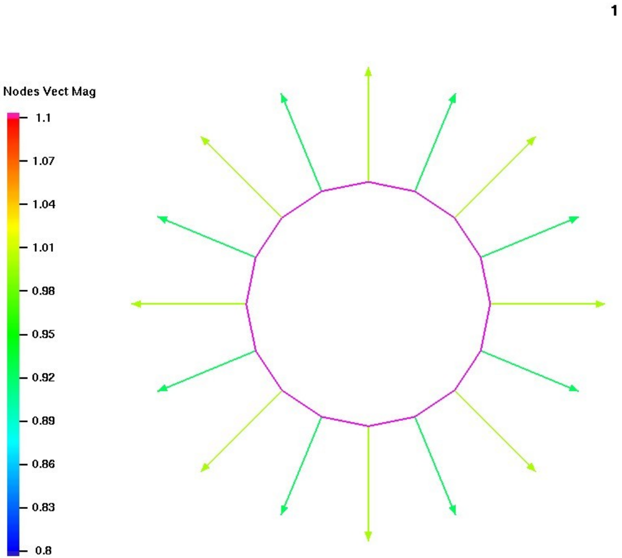
Exact Interpolation in 3D



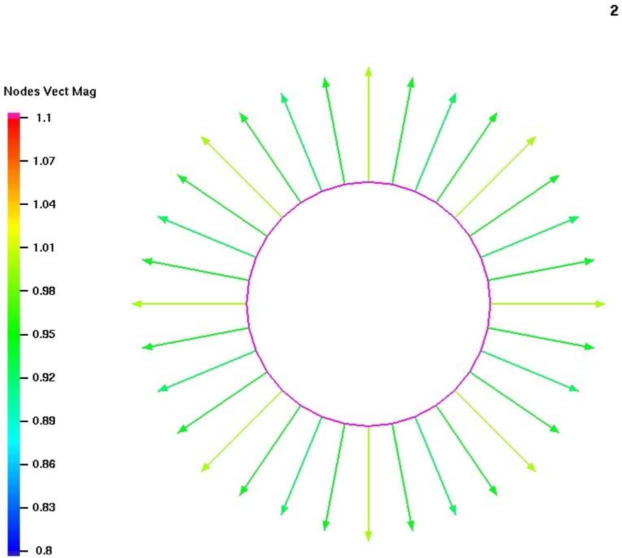
Geometrically Consistent Algorithm



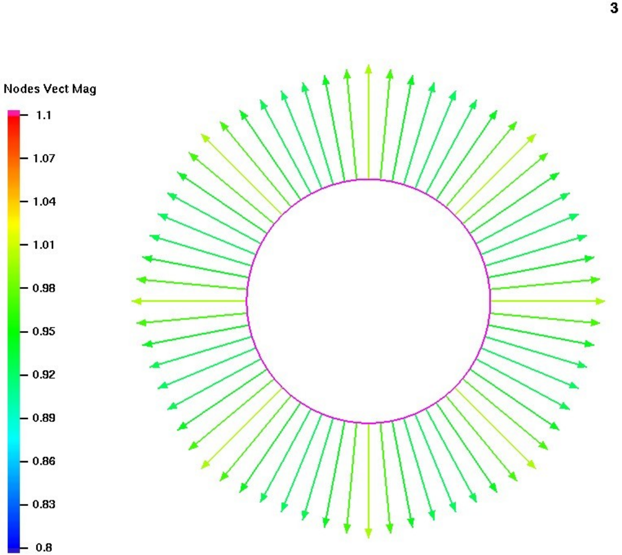
Geometrically Consistent Algorithm



Geometrically Consistent Algorithm



Geometrically Consistent Algorithm



Geometric Consistency

- Geometric identity $\mathbf{h} = -\Delta_\gamma \mathbf{x}$
- Discrete geometric identity $\mathbf{H} = -\Delta_\Gamma \mathbf{X}$
- Assume $\Gamma, \mathbf{X}, \mathbf{H}$ approximate $\gamma, \mathbf{x}, \mathbf{h}$
- *It may be impossible to satisfy the discrete geometric identity*,
- Geometric inconsistency
- Numerical artifacts

Geometric Consistency

- Geometric identity $\mathbf{h} = -\Delta_\gamma \mathbf{x}$
- Discrete geometric identity $\mathbf{H} = -\Delta_\Gamma \mathbf{X}$
- Assume $\Gamma, \mathbf{X}, \mathbf{H}$ approximate $\gamma, \mathbf{x}, \mathbf{h}$
- *It may be impossible to satisfy the discrete geometric identity,*
- Geometric inconsistency
- Numerical artifacts

Geometric Consistency

A finite element triple $(\Gamma, \mathbf{X}, \mathbf{H})$ is *GC* if

$$\mathbf{X}, \mathbf{H} \in \mathbb{V} : \qquad \int_\Gamma \mathbf{H} \cdot \Phi = \int_\Gamma \nabla_\Gamma \mathbf{X} : \nabla_\Gamma \Phi, \qquad \forall \Phi \in \mathbb{V},$$

and it is an approximation of the exact triplet $(\gamma, \mathbf{x}, \mathbf{h})$

Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

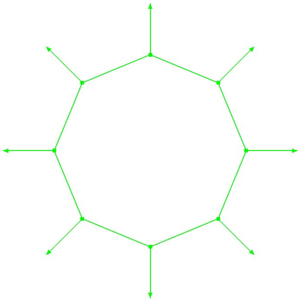
1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

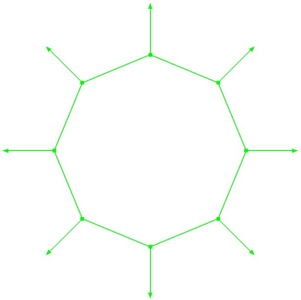


Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

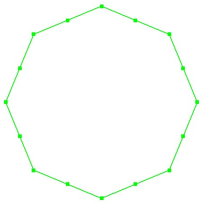


Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

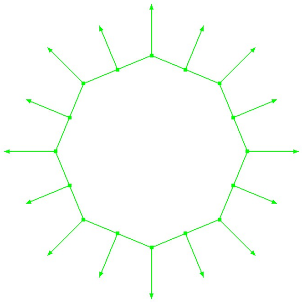


Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

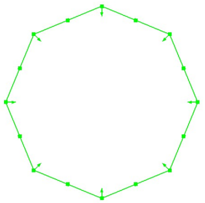


Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

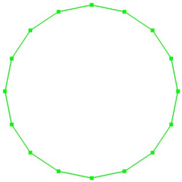


Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

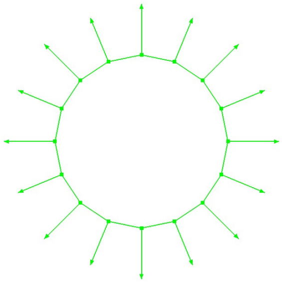


Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$

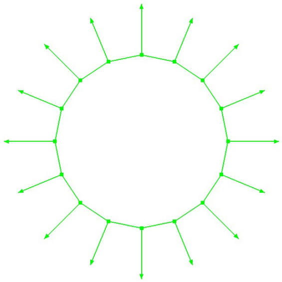


Geometrically Consistency Refinement

Refinement Algorithm

$(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*) = \text{Surf Ref}(\Gamma, \mathbf{H}, \mathbf{X}, \mathcal{M})$

1. $\Gamma^* = \text{Isoparametric Refinement}(\Gamma)$
2. $\mathbf{H}^* = \text{Interpolation}(\mathbf{H})$
3. $\mathbf{X}^* = \text{Inverse Laplace}(\mathbf{H})$



Remarks

- Procedure independent of polynomial degree and dimension
- Refinement can be replaced by coarsening and mesh smoothing

Mathematical Statement

In heuristic terms

1. This refinement guarantees that the errors for position and mean curvature are of the same order as they were before.
2. Unstable numerical differentiation is replaced by stable interpolation plus inversion of $-\Delta_\Gamma$.

Mathematical Statement

In heuristic terms

1. This refinement guarantees that the errors for position and mean curvature are of the same order as they were before.
2. Unstable numerical differentiation is replaced by stable interpolation plus inversion of $-\Delta_\Gamma$.

Theorem (Geometrically Consistent Refinement)

If the triple $(\Gamma, \mathbf{X}, \mathbf{H})$ *is GC and* $\mathcal{E}$ *is a Strang-type upper bound for the error* $|\mathbf{x} - \mathbf{X}|_{H^1(\Gamma)}$, *then the following statements are valid*

1. $\|\mathbf{h} - \mathbf{H}^*\|_{L^2(\Gamma)} = \|\mathbf{h} - \mathbf{H}\|_{L^2(\Gamma)}$;
2. $|\mathbf{x} - \mathbf{X}^*|_{H^1(\Gamma)} \leq \mathcal{E}$;
3. *the triple* $(\Gamma^*, \mathbf{X}^*, \mathbf{H}^*)$ *is GC.*

Conclusions

- Spherical caps: for shapes with distinctive ends, spherical caps seem to be most effective to reduce the bending energy
- Red cells: for disk-like shapes, there is a thickening of the outer edge and depression in the center. The fluid membrane dynamics is quite different from the gradient flow.
- Kinetic energy: is decays exponentially for gradient flows (with a nonobvious dependence of the equilibrium shape), but it oscillates for fluid membranes due to inertia.
- Geometric consistency: this is important for refinement, coarsening and mesh smoothing to avoid numerical artifacts.
- Mesh smoothing: control of mesh distortion due large domain deformations in a Lagrangian approach.
- Time-step control: this accounts for geometry and highly varying time scales.

Large Deformation: Willmore Flow of Helix

Large Simulation