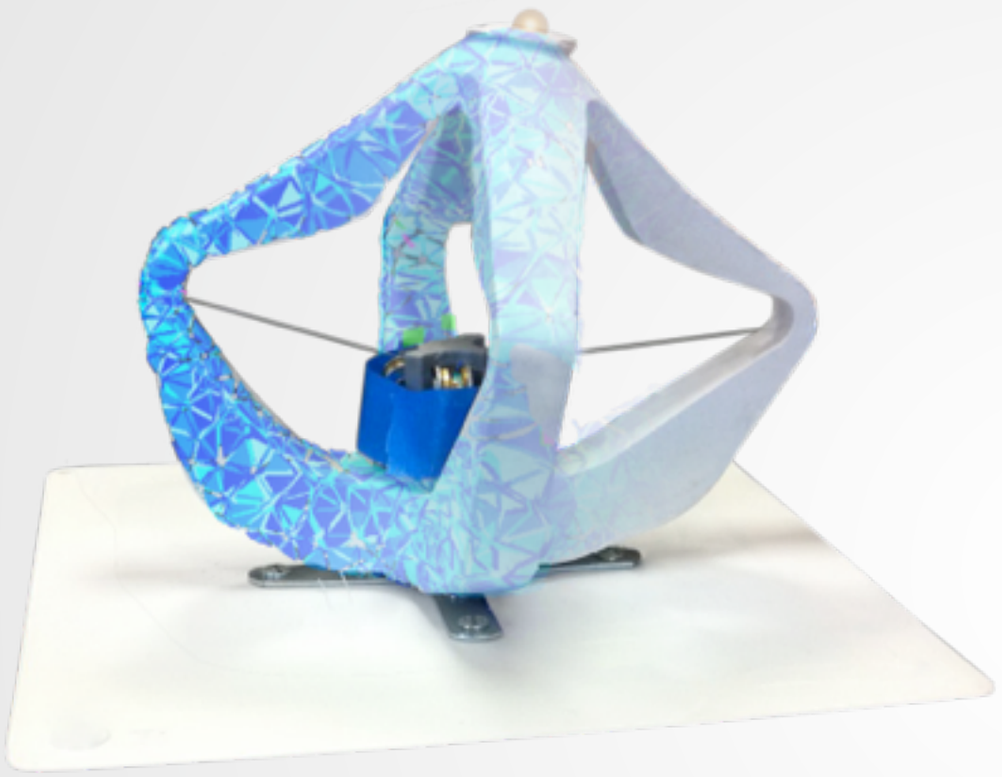
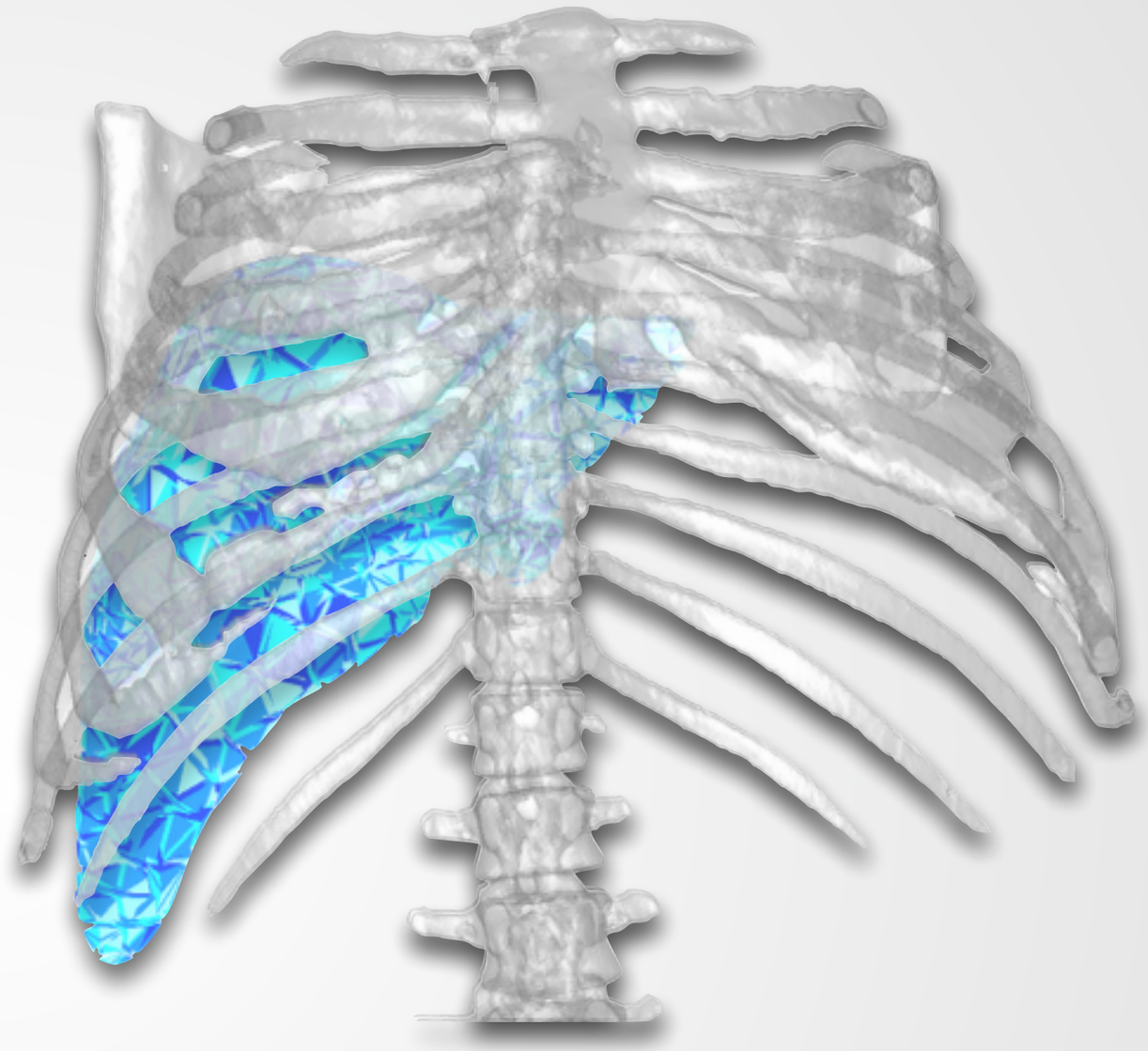
MODELING & SIMULATION

Christian Duriez
INRIA DEFROST-Team Lille





MECHANICAL DEFORMABLE MODELS FOR REAL-TIME COMPUTATION

- Mechanical models for real-time computation
- Constraint-based modeling of biomechanical interactions
- Haptic rendering and multithreading approaches
- Applications, ongoing research projects
- Perspective and Conclusion

MODELING OF INSTRUMENTS

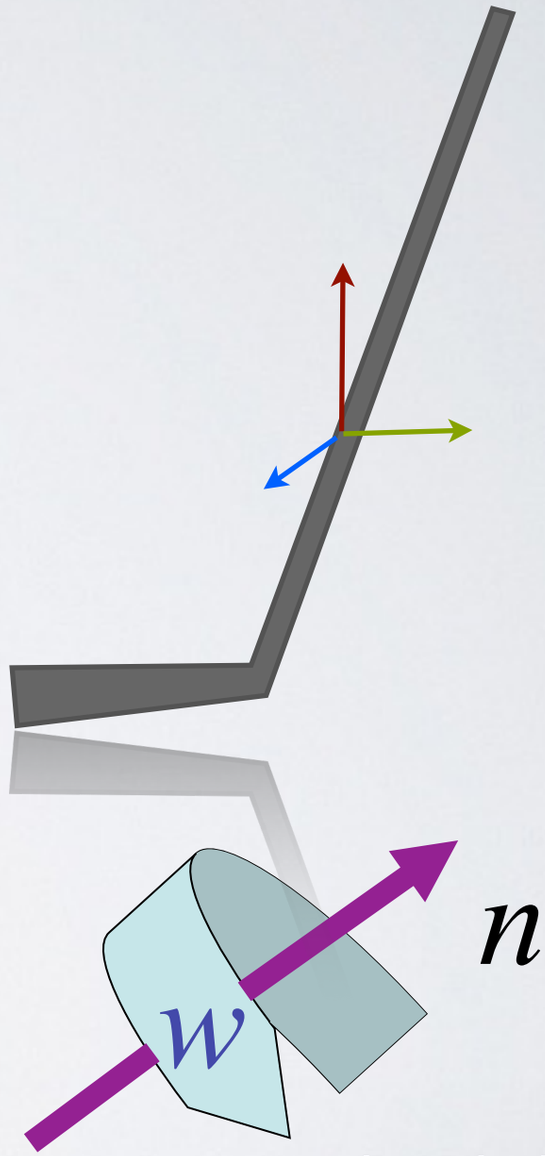
Rigid instruments

$$\mathbb{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}(\mathbf{q}, \mathbf{v})$$

Mass, Inertia

Gravity

centrifugal Forces



- 6 DOFs: 3 translation / 3 rotations => 6 equations
- Motion described at the center of Inertia

$$q_r = \begin{vmatrix} n.\sin(w/2) \\ \cos(w/2) \end{vmatrix}$$

- Mechanical models for real-time computation
- Constraint-based modeling of biomechanical interactions
- Haptic rendering and multithreading approaches
- Applications, ongoing research projects
- Perspective and Conclusion

MODELING OF INSTRUMENTS

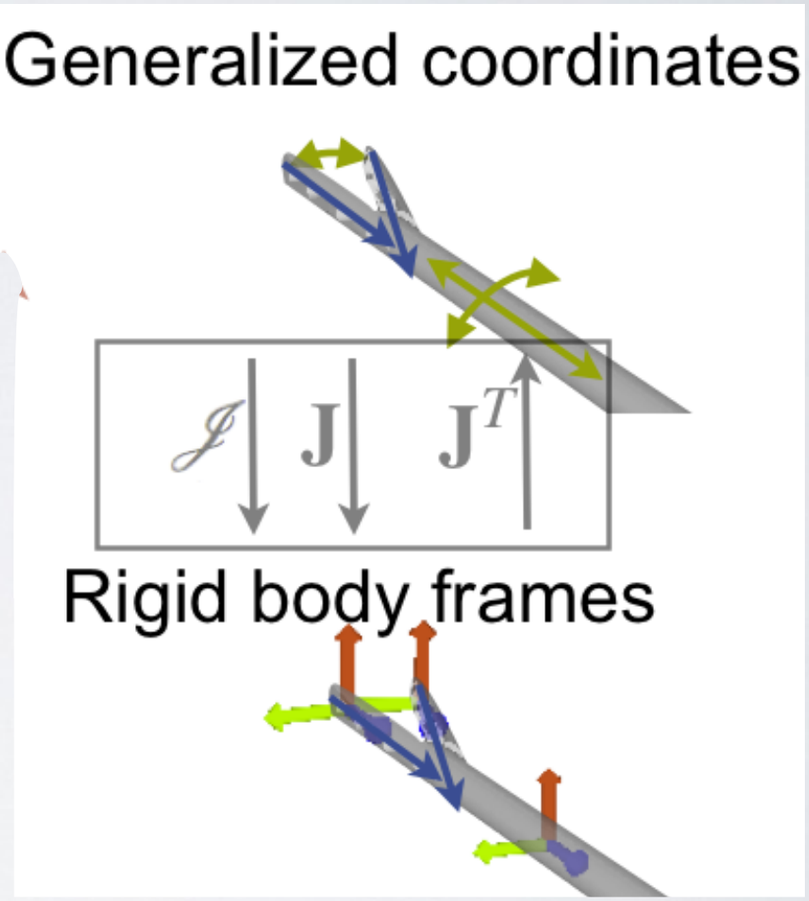
Articulated Models

$$\mathbb{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}(\mathbf{q}, \mathbf{v})$$

Mass, Inertia

Gravity

Coriolis and centrifugal Forces



- 6 DOFs + 1 DOF / articulation
- q,v => generalized coordinates (≠ absolute coordinates)

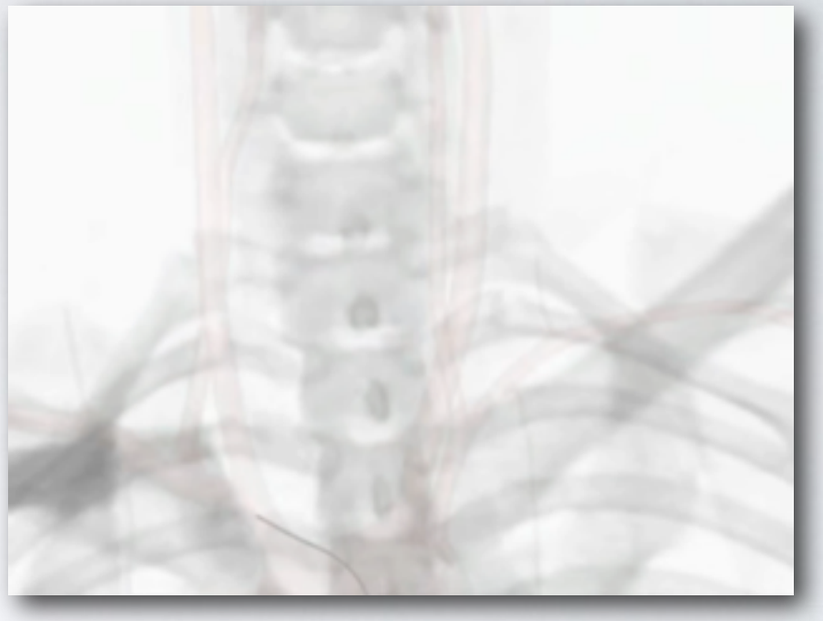
MODELING OF INSTRUMENTS

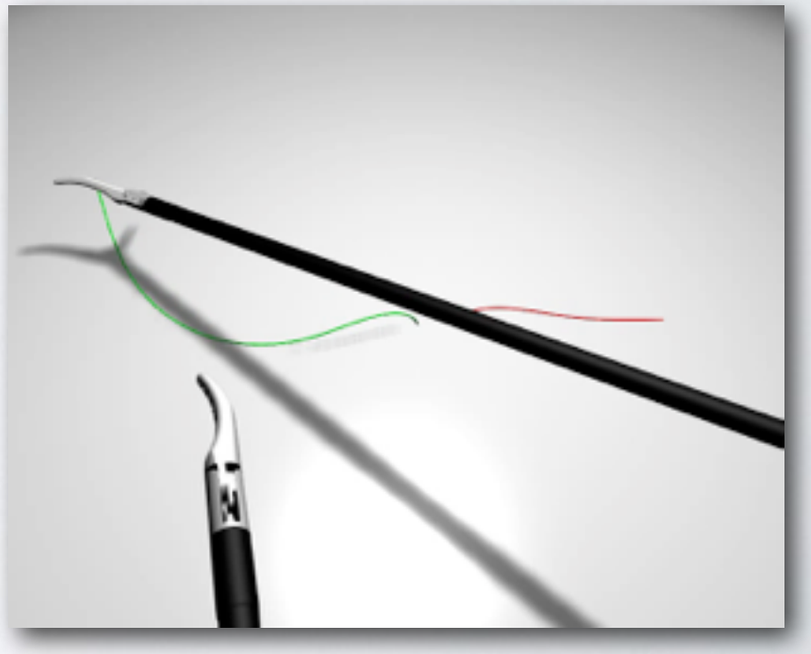
Wire-like instruments

- Several possible application
  - Interventional radiology instruments: catheters, guides, coils
  - Surgical instruments: flexible needle, suture thread..
- Several possible models
  - Beam Theory (6DoFs per nodes: absolute coords)
  - Cosserat Rods (reduced coordinates)
  - Spline-based models... (it depends...)

$$\mathbb{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}\left(\mathbf{q}, \mathbf{v}\right)$$

  - Could also be used for anatomical structures
    - Ligaments
    - Blood vessels





ANATOMICAL MODELING

Example of the Liver model

Patient specific

- Different kind of meshes
  - Visualisation (textures, colors...)
  - Volume mesh (FEM computation)
  - Blood vessel network
  - Collision mesh

The liver is not «alone»

- Respiratory system (diaphragm motion)
- Other surrounding deformable organs
- Ligaments, mechanical links

ANATOMICAL MODELING

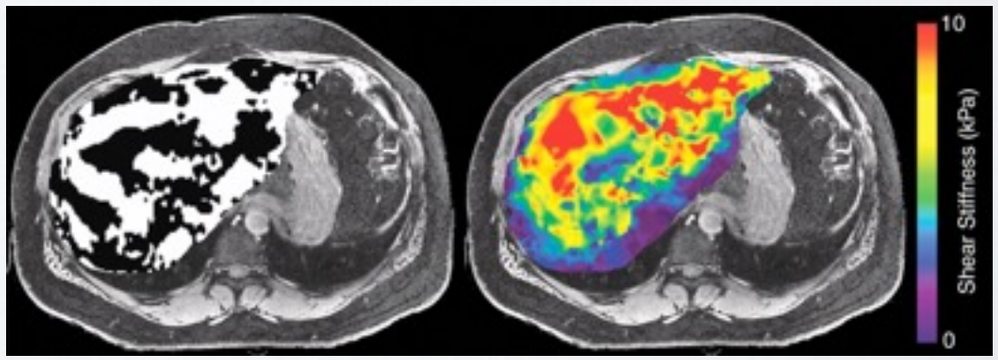
Anatomical models are key in medical simulation...

Various algorithms rely on meshes derived from anatomical model

- Post-processing is needed to generate derived meshes
- Integrate simulation-specific constraints within reconstruction process
- Need for more complete anatomical descriptions
  - embed tissue properties
  - other parameters such as patient-specific texture

... but need to be combined with patient-specific biomechanical data

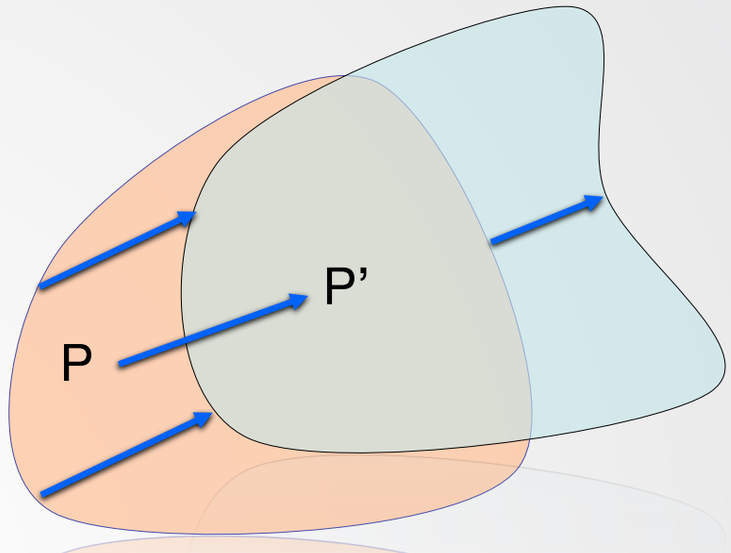
- Elastography (ultrasound, MRI, ...)
- Other (non-invasive) approaches



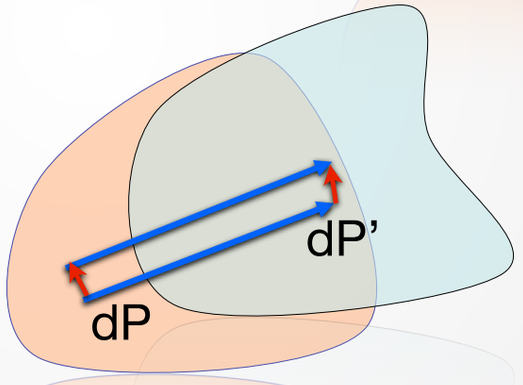
Elastographic image - Colors show different stiffness values

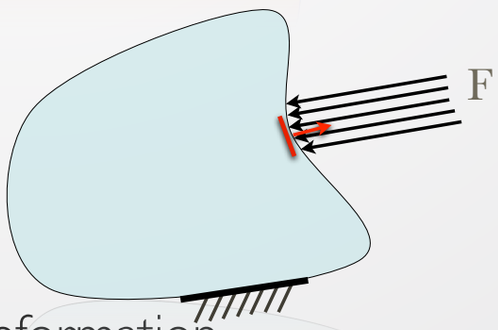
CONTINUUM MECHANICS

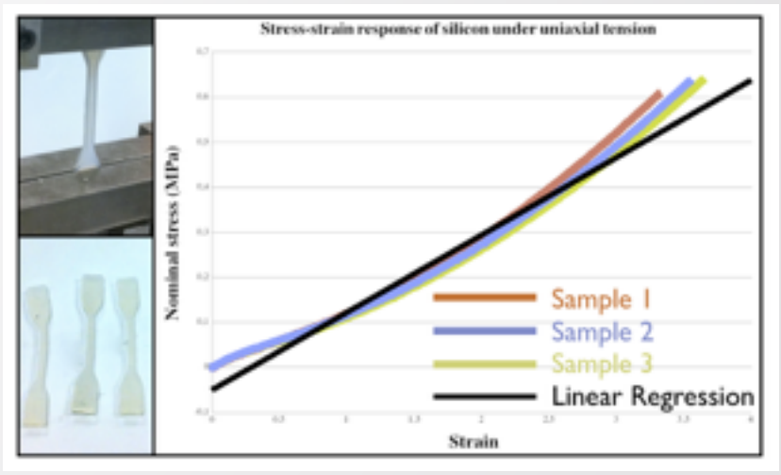
- Hypothesis: « Total Lagrangian » mechanics
  - « Continuity » of the material
  - Infinite number of points
  - A reference domain



- Use of partial derivatives (Mathematics)
- Strain tensor
  - ~ Geometrical or Kinematic Model







- Use measurements (Physics)
- Stress tensor
  - ~ Provide a notion of internal pressure due to deformation
  - Need to be in equilibrium with external forces

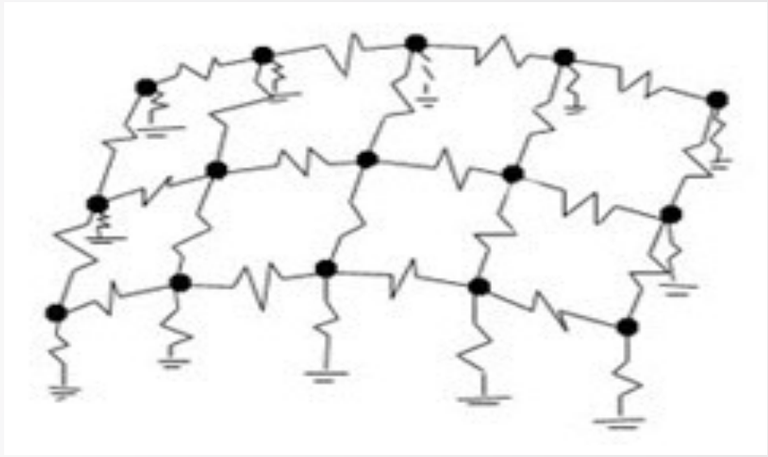
DISCRETIZATION

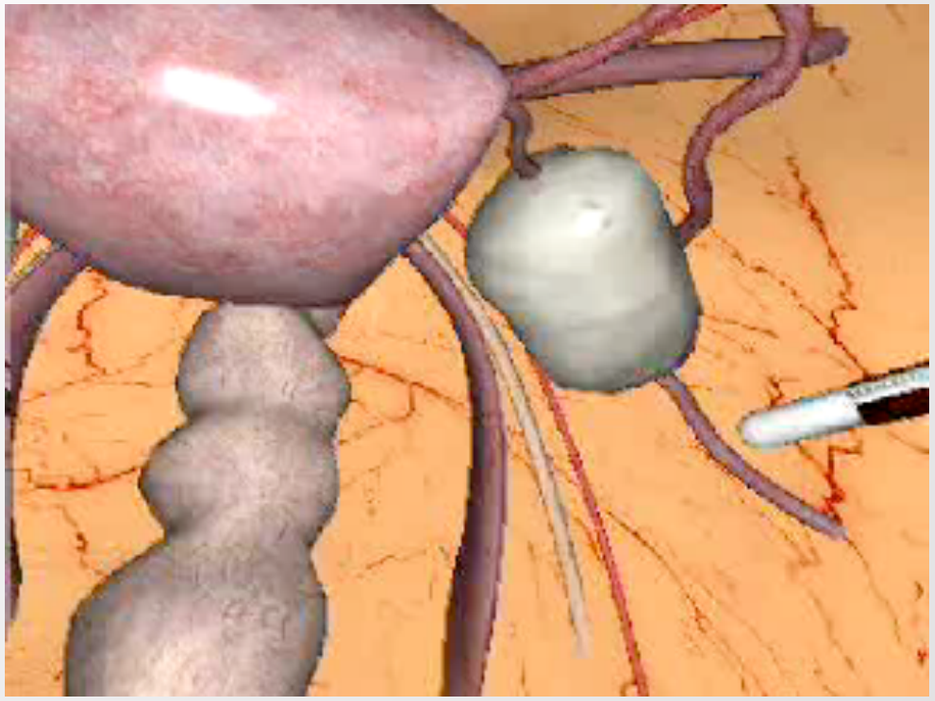
- Why do we need to discretize ?
- Definition of the Degrees Of Freedom (DOF)
- Several numerical methods
  - Finite Element
  - Finite Difference
  - ...

MODELING OF SOFT-TISSUES

$$\mathbb{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}\left(\mathbf{q}, \mathbf{v}\right)$$

- Mass-Spring & Discrete models
  - q,v = position and velocity of the nodes
  - Mass lumped at the nodes
  - visco-elastic links
- Positive features
  - Easy to programm !
  - Visual realism
- But not accurate
  - No direct identification & dependance on the mesh



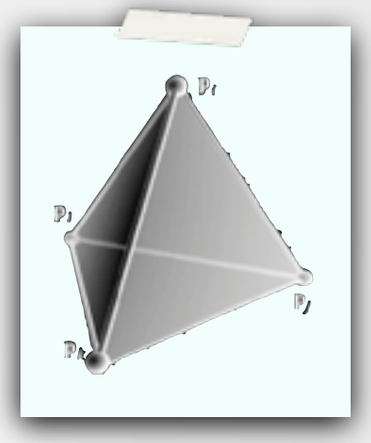


FEM DEFORMABLE MODEL

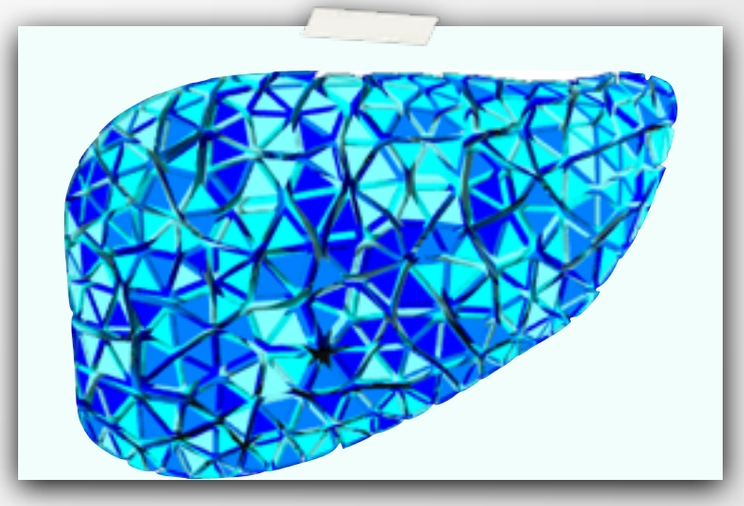
- no analytical solution… in the general case
- Use of Finite Element Method
- Advantages:
  - Numerical convergence to the solution
  - Directly based on the constitutive law
- Issues:
  - Computation is time-consuming
  - Non-linearities due to large displacements (large rotation)
- Real-time implementation:
  - Corotational approach [Felippa00] (static or dynamic)

$$\mathbf{f}(\mathbf{x_i}) \approx \mathbf{f}(\mathbf{x_{i-1}}) + \mathbf{K}(\mathbf{x_{i-1}})d\mathbf{x}$$

  - GPU implementation



1. Computation at the element level



2. Global discrete «stiffness» for the hole deformable model

MECHANICAL DEFORMABLE MODELS

- Newton's second law

$$\mathbb{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}\ (\mathbf{q}, \mathbf{v}) + \mathbf{H}^T\lambda$$

$\mathbf{q} \in \mathbb{R}^n$ Vector of generalized degrees of freedom (nodes of a deformable model)

$\mathbf{v} \in \mathbb{R}^n$ Vector of velocities

$\mathbb{M}(\mathbf{q}) : \mathbb{R}^n \mapsto \mathcal{M}^{n\times n}$ Inertia Matrix

$\mathbb{F}\ (\mathbf{q}, \mathbf{v})$ Internal forces (non-linear model)

$\mathbb{P}(t)$ External forces

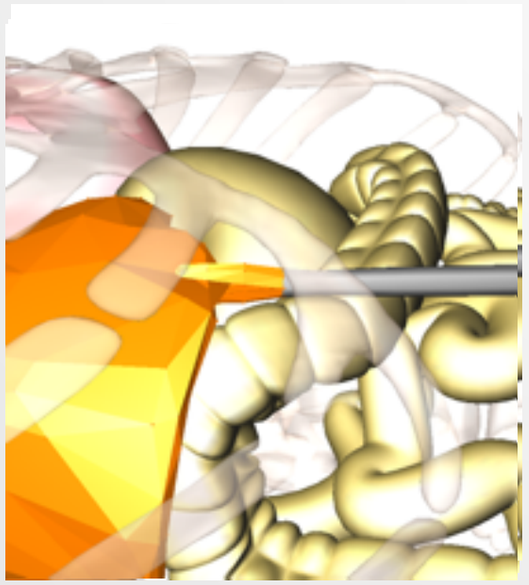
$\mathbf{H}^T\lambda \in \mathbb{R}^n$ Constraint force contribution

- Quasi-static case:

$$\frac{\delta\mathbb{F}}{\delta\mathbf{q}}\ d\mathbf{q} = \mathbb{P} - \mathbb{F}\ (\mathbf{q}) + \mathbf{H}^T\lambda$$

REAL-TIME INTEGRATION

- Interactive Simulation = the physician can modify the course of the simulation
- Time derivatives in model equation = integration scheme (notion of time step...)
- Between these two steps, the user do a certain motion in a REAL interval of time
- The time elapsed in the simulation must be EQUAL to the REAL interval of time



$h = dt$

$t + h$

$t + dt$

TIME INTEGRATION SCHEMES

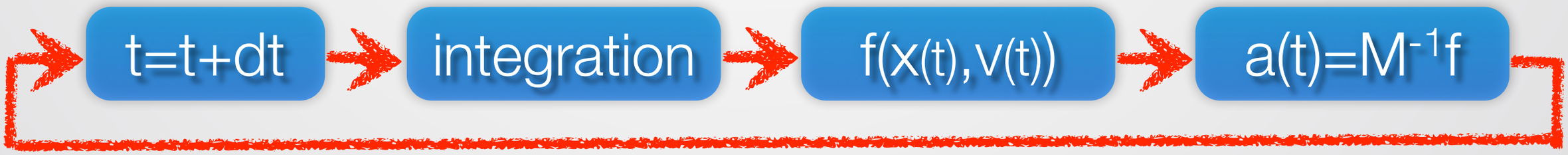
From Model to Algorithm...:

- Dynamic equation of 1 point: $f(x,v) = Ma$

- Euler explicit scheme:

$$v(t) = v(t-dt) + a(t-dt)\,dt$$
$$x(t) = x(t-dt) + v(t-dt)\,dt$$

- Simulation algorithm



TIME INTEGRATION SCHEMES

- Explicit Methods:

$$\mathbb{M}\,\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}\,(\mathbf{q}, \mathbf{v})$$

unknown — known from previous time steps

  - Conditionnally stable
  - High constraint on the time step used in the simulation
  - **h ≤ Le/c** (**h**: *time step*, **Le**: *Caracterstic lengh of smallest element*, **c**: *velocity of the deformation wave*)

DEFORMATION WAVE…

Propagation of the deformation wave… is very fast !!

TIME INTEGRATION SCHEMES

- Explicit Methods:

$$\mathbb{M}\,\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}\,(\mathbf{q}, \mathbf{v})$$

$\dot{\mathbf{v}}$: unknown

$(\mathbf{q}, \mathbf{v})$: known from previous time steps

  - Conditionnally stable
  - High constraint on the time step used in the simulation
  - **h** ≤ **Le/c** (**h**: *time step*, **Le**: *Caracterstic lengh of smallest element*, **c**: *velocity of the deformation wave)*

- Implicit Methods:

$$\mathbb{M}\,\dot{\mathbf{v}} = \mathbb{P}(t) - \mathbb{F}\,(\mathbf{q}, \mathbf{v})$$

$\dot{\mathbf{v}}$: unknown

$(\mathbf{q}, \mathbf{v})$: depends on $\dot{\mathbf{v}}$ and on previous time steps

  - Unconditionnally stable
  - Possible use of «large» time step **h** in the simulation
  - Needs the resolution of a large non-linear problem

TIME-STEPPING IMPLICIT INTEGRATION

- Implicit Euler Integration

  stability with quite large time-step and «non-smooth» events

$$\begin{aligned} \mathbf{M}(\mathbf{v}_f - \mathbf{v}_i) &= h\left(\mathbb{P}(t_f) - \mathbb{F}(\mathbf{q}_f, \mathbf{v}_f)\right) \\ \mathbf{q}_f &= \mathbf{q}_i + h\mathbf{v}_f \end{aligned}$$

- One linearization of the internal forces per time-step

  (Compromise between precision and computation time)

$$\mathbb{F}\left(\mathbf{q}_i + d\mathbf{q}, \mathbf{v}_i + d\mathbf{v}\right) = \mathbf{f}_i + \frac{\delta \mathbb{F}}{\delta \mathbf{q}} d\mathbf{q} + \frac{\delta \mathbb{F}}{\delta \mathbf{v}} d\mathbf{v}$$

- A (changing) linear system to be solved at each time step

$$\underbrace{\left(\mathbf{M} + h\frac{\delta \mathbb{F}}{\delta \mathbf{v}} + h^2 \frac{\delta \mathbb{F}}{\delta \mathbf{q}}\right)}_{\mathbf{A}} \underbrace{d\mathbf{v}}_{\mathbf{x}} = \underbrace{-h^2 \frac{\delta \mathbb{F}}{\delta \mathbf{q}} \mathbf{v}_i - h\left(\mathbf{f}_i + \mathbf{p}_f\right)}_{\mathbf{b}}$$

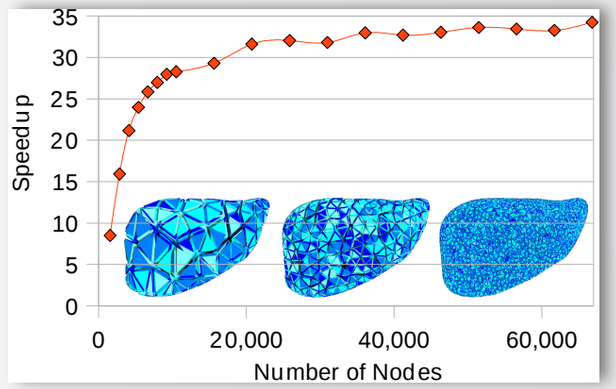
- Switch to quasi-static when computation is too slow

  (no notion of «time» in the simulation)

$$\underbrace{\frac{\delta \mathbb{F}}{\delta \mathbf{q}}}_{\mathbf{A}} \underbrace{d\mathbf{q}}_{\mathbf{x}} = \underbrace{\mathbf{P} - \mathbf{f}_{i-1}}_{\mathbf{b}}.$$

HOW TO MAKE IT REAL-TIME ?

- Parallel / GPU Computation
  - Easy for explicit integration / more complex for implicit (coupling btw equations)
  - Existing library (Cuda-Blas, Pardiso, ...)
- Specific Numerical methods
  - Numbering of nodes, Preconditionning, Multigrid ...
- Adaptive strategies
  - Put the degrees of freedom where (really) needed
  - Meshless approach / topological changes (also a challenge for cutting)
- Model reduction
  - Precomputation of a large number of deformation modes
  - Use of a reduce number but more «complex» degrees of freedom (Frames, deformation modes...)



HOW TO MAKE IT REAL-TIME ?

- One example of optimization: Asynchronous Preconditionner
  - Non-linear FEM: one linearization per time-step
  - Solve system A x = b (A, b known, x unknown)
    - Finding exact solution is « costly » → Iterative solvers
    - Problem of bad conditioning of matrix A
  - Use of a preconditionner:

$$\underbrace{P^{-1}A}_{\approx I} x = P^{-1}b$$

  - Motivations for asynchronous preconditionner:
    - Temporal Coherency of A
    - Factorization of an « old » value of A to find P
    - The cost of factorization of P is not supported by the real-time loop