Real-Time Soft Tissue Modeling

Stephane Cotin
cotin.stephane@mgh.harvard.edu
CIMIT Simulation Group / Harvard Medical School

Soft Tissue Modeling: a grand challenge for medical simulation

- Soft tissue is very complex
- We do not understand yet all the aspects of soft tissue behavior
- We need tools to investigate tissue properties
- Once we have a better understanding we need to design appropriate mathematical models
- These models need to be optimized to provide real-time interaction

Importance of Soft Tissue Modeling

- Most medical procedures involve the deformation (and tearing or cutting) of anatomical structures
- The ability to simulate that behavior is an important element of the learning process
- Applications for more accurate soft tissue models are not limited to medical simulation

Some references...

- In vitro property measurement

- In vivo property measurement

- Mathematical Modeling
  - In vitro measurements
  - In vivo measurements
  - Constitutive Laws
  - Continuum Models
  - Finite Element Method
  - Spring-Mass Models
  - Others

- Real-time Modeling

Validation, validation, validation
Simulation for Medical Training – MICCAI 2003

From Experimental Data to Predictive Models
- **First step**: build a database of experimental results
- **Second step**: define mathematical models that will fit the data and simulate tissue behavior across variable shapes and constraints

Soft-Tissue Constitutive Laws
Constitutive laws need to account for tissue complexity
- Non-linear stress-strain relationship
  - Forces are not linearly proportional to displacements
- Large deformations
  - Geometric non-linearities
  - Viscoelastic
    - Properties are function of time
- Non-homogeneous
  - Properties vary throughout tissue thickness
- Anisotropic
  - Properties vary with direction

Solving the Equations
- No close-form solution for the vast majority of constitutive laws
- Need to use numerical techniques that provide accurate results and account for boundary conditions and complex geometries
  - Continuum models
    - Approximated by Finite Element Models
    - More accurate than discrete models like spring-mass models
    - Other approaches exist that might be more appropriate for specific applications

Real-time Soft Tissue Modeling
- A typical FEM computation on a non-linear model can take several minutes on a fast computer...
- … while the required update rates for interactive simulation typically range from 25 Hz (visual) to 300 Hz (haptics)
- An acceleration factor of more than 10,000 is needed to permit interactive manipulation of accurately simulated soft-tissue
Real-time Soft Tissue Modeling

- How to improve speed?
  - Get a faster computer…
  - Optimize the algorithms
  - Simplify the models
    - Linear vs. non-linear
    - Surface vs. volume
    - Static vs. dynamic
  - Use ad-hoc / heuristic / discrete models
    - Spring-mass models
    - Long Elements, Chain Mail,…
    - Hybrid models
    - Neural networks

- Linear vs. non-linear
- Surface vs. volume
- Static vs. dynamic

- Use ad-hoc / heuristic / discrete models
  - Spring-mass models
  - Long Elements, Chain Mail,…
  - Hybrid models
  - Neural networks

Spring-Mass Models

- Particle System
  - Collection of unconnected mass points
  - Particle motion influenced by force fields

- Spring-Mass Model
  - Basically same as a particle system
  - Structured, rather than free form
  - Mass points part of model
  - Use spring forces to connect masses
  - Each force object knows which points it acts on

Spring-Mass Models

- 1-D elements linking a set of nodes distributed on the surface or in the volume of the “organ”
  - Surface springs
  - Volumetric springs

- Computation
  - Solve Newton’s second law of motion
    - Explicit methods (Euler, Runge Kutta, Midpoint,…)
    - Implicit methods (backward Euler,…)

Some references on spring-mass models for medical simulation


Eye Surgery Simulation

Courtesy of LIFL
Some remarks regarding mass-spring models

- Some good things…
  - Fast and easy to implement
  - Can handle geometric non-linearities
  - The use of 1D elements make it unrealistic to model volume (results depend on topology of the mesh…)
- … and not so good things
  - How to preserve volume?
  - Stability issues and jelly-like behavior
  - How to integrate soft-tissue properties into the model
  - Just another way of describing a FEM model for truss elements

The Finite Element Method

- Basic principles (continued)
  - Assemble the contribution of each element in the mesh to form (for instance) a linear system \( K \mathbf{u} = \mathbf{F} \)
  - The unknown vector \( \mathbf{u} \) contains the values of the approximate solution at the mesh points (i.e. displacements)
  - The matrix \( K \) is assembled from the stiffness coefficients of each element
  - And the right-hand side \( \mathbf{F} \) contains the external forces applied to the domain,

Some remarks regarding Finite Element models

- Advantages
  - This approach benefits from a solid background and established techniques, books and a vast literature.
  - It is a numerical technique for solving partial differential equations…
  - … so the results are less dependent on the initial mesh (as opposed to mass-spring models), and it is easier to integrate tissue properties into the model
- For real-time applications it is mainly this last step that needs to be accelerated
  - By applying new computation strategies (condensation, superposition, …)
  - By using multi-processing approaches

The Finite Element Method

- Basic principles (continued)
  - Compute the solution by using one of the numerous numerical techniques available for linear (or non-linear) systems of equations
  - Examples of numerical techniques
    - Conjugate Gradient for linear systems
    - Newton-Raphson method for non-linear systems
  - For real-time applications it is mainly this last step that needs to be accelerated
  - By applying new computation strategies (condensation, superposition, …)
  - By using multi-processing approaches
Some remarks regarding Finite Element models

- **Drawbacks**
  - It is slow (and very slow for non-linear models) when not combined with real-time strategies
  - Real-time computation calls for assumptions that are not always compatible with requirements for medical simulation
  - Not always easy to implement...

Example of real-time FEM model (linear elastic model simulated at 300Hz)

Multi-resolution Green’s Function Method (D. James & D. Pai)

Other approaches to real-time soft tissue deformation

- **Long (and radial) elements:**

- **Chainmail:**

Other approaches to real-time soft tissue deformation

- **Tensor-mass models:**

- **Adaptive sampling / mesh**

Where is the research going?

- Derive new models from Biology not from Computer Graphics
- Try to understand how things work in the real world before simulating it...
- ... then define models based on experimental (in vivo, in situ) data
Where is the research going?

- The more accurate the models will be, the more computational power we will need.
- Computation based on single processor approaches will soon reach their limits.
- Clusters of PCs might be a way of dealing with Medical Simulation in the future.

Cross-Validation of Real-Time models is mandatory

- Mathematical models (constitutive laws) are a tradeoff between accurately “translating” the experimental data and remaining applicable to other geometries and constraints.
- New data must be collected and compared with the results predicted by the model.
- Real-time models usually require additional tradeoffs to provide fast computation; therefore validation is even more important.

Validate, validate, validate...

- There is no ideal modeling technique for all simulations, only better, more stable, more accurate ways of doing things.
- In any case, it is key to validate the results of the simulation by comparing them to the real world.

Conclusion

- Medical simulation has become a reality.
- The medical community is expecting a lot from medical simulation.
- Negative training must be avoided by developing realistic training systems based on real-world data.
- Do not forget to validate!