Introduction to the SimTK Core toolkit and Simbody

Michael Sherman (Sherm)
Xulu Entertainment, Inc.
(was: Simbios chief software architect)

SimTK 1.5 Workshop, Sept. 25, 2008

NIH Roadmap
grant U54 GM072970
Why a simulation toolkit?

• Focused, self-contained applications
  – Short-term, low hanging fruit; exploit earlier work
  – Useful, likely to generate science
  – Substantial effort; little leverage

• General open source toolkit
  – Long lead time; technically difficult
  – Development, not research
  – Requires adoption by programmers before science contribution
  – Major impact; highly leveraged

“Eat for a day”

“Eat for a lifetime”
Some notable open source toolkits

- **VTK**: visualization
- **ITK**: medical imaging
- **Lapack/BLAS**: fast, accurate linear algebra
- … many others

- Each is an enabling technology for modelers & application programmers

We want to add:

- **SimTK**: physics-based simulation of biological structures
Reliable toolkit recipe

- Find the right abstractions
- Provide needed tools
- Prescribe a discipline
- Write a book
- Bake for a decade or so
Focusing tactics

• Look at “Driving Biological Problems” for common themes
• Limit hardware/language support
• Adopt/adapt existing open source software
• Exploit available expertise
• One step at a time (depth first)
What do these Driving Biological Problems have in common?

- Biological structures
- Dynamics well described by classical physics — that is, $F=ma$
- They’re “chunky”
Similar models across multiple scales

Model of human


Model of RNA (Tetrahymena group I intron)

Zheng, et al. PNAS 98(7), 2001
So, SimTK Core toolkit Phase I: “Multibody Biology”

- Biological systems that can be treated as interconnected rigid bodies
  - E.g. biomechanics, biopolymer simulation
  - Primarily ODE/DAEs; dense matrices

- Not systems best treated with continuum methods
  - E.g. fluid/tissue coupling; cardiovascular DBP
  - Primarily PDEs; sparse matrices
  - Defer to Phase II
Two multibody biology application areas for 1.5

1. Neuromuscular biomechanical simulations
   - Already advanced users of multibody dynamics
   - SimTK Core and Simbody already in use in OpenSim
   - Open source toolkit provides technical and practical advances

2. Internal coordinate and coarse grained biomolecular simulation
   - Little use of multibody dynamics to date (except NMR)
   - Promising early results; but research hampered by lack of available software
   - Open source toolkit provides new research opportunities
SimTK Core Programming Team

Staff currently full time on SimTK Core

Jack Middleton  Chris Bruns  Peter Eastman

Other SimTK Core software contributors:

Mark Friedrichs  Paul Mitiguy  Ajay Seth  Sam Flores  Radu Serban  Randy Radmer  Yours Truly
SimTK Core layers

- **Domain independent**
  - Biology
- **Hardware platform**
  - Application Support
    - Application Support
  - Modeling Support
    - Modeling Support
  - Physics
    - $F=ma$
  - Mathematics
    - Numerical methods
  - Software platform
    - Methodology; fundamental objects
  - Hardware platform
    - Speed in GFLOPS

Simbody lives here
A quick look at each layer ...

1. Hardware platform
   speed in GFLOPS

2. Software platform
   methodology; fundamental objects

3. MATHEMATICS
   numerical methods

4. PHYSICS
   \( F=ma \)

5. MODELING SUPPORT

6. APPLICATION SUPPORT
## Layer 1: Hardware exploitation

<table>
<thead>
<tr>
<th>Level</th>
<th>Section</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hardware platform</td>
<td><em>speed in GFLOPS</em></td>
</tr>
<tr>
<td>2</td>
<td>Software platform</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Mathematics</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Physics</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Modeling support</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Application support</td>
<td></td>
</tr>
</tbody>
</table>

- Basic linear algebra (BLAS)
- Molecular mechanics inner loop
- Exploit cache/multi-core/GPU
Numerical Recipes vs. SimTK Lapack

What does this leave on the table?

• Numerical Recipes is 20X too slow!
• 95% of hardware is wasted.
SimTK Lapack (& Blas)

• Full use of hardware for linear algebra
  – Single, dual, multi-core CPUs
  – Exploits cache & vector instructions
• Binaries available
  – Windows, Mac, Linux
• Download & link in
• Use the other 95% of your computer!
• And … it’s already on your machine.
Layer 2: SimTK abstraction layer

- System (model)
- Subsystems
- State
  - Discrete/continuous
  - Serialization
  - Caching, stale reference prevention
- Study
  - Dynamics, minimization, etc.
- Vectors & matrices (Simmatrix)
- C++ framework
  - Basic types & containers
  - OS/compiler independence
  - Binary compatibility
Terminology: modeling creates a “System”

- A “System” is a computational embodiment of a mathematical model
Properties of a System

- **Defines** its parametrization
- But ... it is *stateless.*
- Given a *State,* performs useful computations
Studying a system

- System + State + Study $\rightarrow$ Simulation
Systems are composed from subsystems

- Interlocking computations
- System provides the “edge pieces”
Layer 3: Simmath

- Linear algebra
  - Eigenvalues (normal modes), least squares, SVD, etc.

- Optimizer
  - Constrained, unconstrained

- Integrator
  - Stiff/nonstiff
  - Constraint projection
  - Event isolation
  - CPODES collaboration (LLNL)

- Miscellaneous
  - Root finders
  - Random numbers
  - Differentiator
  - Spline fitter
Simmath

- Matlab-like capability in C++
- Specialized for multibody biology use
- Constrained numerical optimization comparable to FSQP but free
- Custom stiff/nonstiff, error controlled, coordinate projection integrators
- Hybrid discrete/continuous simulation with event handling

\[
\begin{align*}
\dot{q} &= q(t) \\
g(q) &= 0
\end{align*}
\]
Layer 4: Simbody

- Rigid bodies
- Joints
- Constraints
- Generalized coordinates
- “Hooks” for forces
- Solve Newton’s 2nd law in $O(n)$ time

- We’ll come back to this
Layer 5: Modeling support

- Basic force subsystems
  - Contact, gravity, point charge MD
- Basic studies
  - Initial condition analyses
  - Forward dynamics
  - Optimization
- Molecule modeler (Chris Bruns)
  - Proteins & RNA
Layer 6: Application support

- VTK
- Pre-packaged binaries
- Documentation
- Examples
- Training (duh)
- Support
Simbody™

a SimTK Core toolset for multibody mechanics
What is a multibody system?
Matter

• Mass
• Spatial distribution
• Motion
Abstract matter

• The rigid body
• What is a rigid body?

• Mass distribution: 10 *constants*
• Decorate w/geom & other props
• Ground is a (heavy) rigid body
Joints

• Defines relative mobility between 2 rigid bodies

• Examples

  - Pin
  - Slider
  - Ball

• Joints may permit motion, or restrict it, or both
Mobilizers

• A new rigid body has no mobility
• Mobilizers precisely define the allowable mobility relative to parent
• Unlike joints, mobility is always increased by mobilizer
• These define the generalized coordinates $q$
Multibody system (1)

- Tree of bodies interconnected by mobilizers

\[ M \ddot{q} = f \]
Constraints

• But, that’s a little too floppy …
• Constraints introduce constraint equations (1 or more)
  – E.g., ball constraint adds 3 equations, -3 dofs
• Algebraic invariant relating q’s: g(q)=0
  – or qdot’s
• Restricts allowable motion – like negative mobility
• But … may not be independent.
Multibody system (2)

- Constraints permit loops

\[ M\ddot{\mathbf{q}} = f - f_c \]
\[ g(q) = 0 \]

- Joints can be mobilizers, constraints or both
Multibody systems

- Rigid parts …
- … freed by Mobilizers
- … influenced by Forces
- … and restricted by Constraints.
- Key feature: motion is localized.
What’s in a multibody system?

- Matter and forces
- Also:
  - Geometry (analytic & decorative)
  - Mass property calculation
  - Other properties, e.g. atom types
A molecular mechanics (MM) system is a kind of multibody system

- Has matter and a molecular mechanics force subsystem
- Helpful to have a “modeler” for molecules of interest, to coordinate the matter & forces (tomorrow)
Large systems + long time scales not *inherently* hard

- Must choose right representation
  - Matter, Space, Motion

F.C. Anderson, M. Pandy
Big molecules have rigid parts

... can we model them accordingly?
Simbody enables the experiment

• Removes daunting startup impediments
  – technical, time, $$$

• Mobility only where desired

• Performance is $O(n)$ in mobility, not atom count
Something like this …

- RNA with rigid duplexes
- 50 bodies, 150 internal dofs
- 31 constraints
- Gravity & a spring??
- If you can imagine it, you should be able to try it
- Runs in a few minutes
Thank you.

sherm@xulu.com