N89-24666

COMPUTATIONAL ASPECTS OF MULTIBODY DYNAMICS

K. C. Park
Lockheed Palo Alto Research Laboratory
Palo Alto, California

ABSTRACT

This paper addresses computational aspects impacting the requirements for developing a next-generation software system for flexible multibody dynamics simulation which include: criteria for selecting candidate formulations, pairing of formulations with appropriate solution procedures, need for concurrent algorithms to utilize computer hardware advances, and provisions for allowing open-ended yet modular analysis modules.

Computer Implementation Tasks for Multibody Dynamics Simulator

A successful next-generation multibody dynamics simulator requires a careful evaluation of existing formulations and computational procedures from which pairing of several candidate formulations/solution algorithms should evolve and, if necessary, need for new and/or improved formulations and solution algorithms must be identified. Concurrent with selecting formulations and solution algorithms, considerations must be given to software environment under which the next-generation simulator will be implemented. In addition, the associated hardware systems and their future trend must be incorporated from the outset of the computer implementation planning stage. These aspects are summarized.

Formulations
Solution Procedures
Software Environment
Hardware Systems

Review of Available Formulations

Formulations According to:

- Bodley/Frisch
- Fraeijs de Veubeke
- Hooker/Margoulis/Ho
- Kane/Likins
- Roberson/Wittenburg
- Russel/Jerkovsky

Criteria for Selecting Candidate Formulation

- Efficiency of the Resulting Software Rather Than Simplicity of the Resulting Equations of Motion
- Let Implementation Algorithm Select the Generalized Coordinates Rather Than Case-by-Case User Selection of Them

Review of Available Solution Algorithms

- Stiff Differential Integrator(Hindmarsh/Gear)
- Differential/Algebraic Solver(Petzold/Lötstedt)
- Partitioned Procedures(Park/Felippa)
- Semi-Implicit Runge-Kutta methods(Chipman/März)
- Impact-Contact Algorithms

Criteria for Selecting Candidate Algorithms

Reliability First, Then Efficiency
Rather Than
Efficiency First, Then Reliability

Minimum User Decision

Current Software and Hardware Environment

Most of the currently available computer programs for simulating multibody dynamics do not have any data base management. As such, the task of data handling remains time consuming and inflexible. In particular, an addition of enhanced capability can present varying difficulties. However, improved computational efficiency has been brought about by vectorization of part of the programs that require intensive computations to generate the discrete dynamical equations and then installing the resulting programs in CRAY-like supercomputers.

An Example of Challenging Deployment Task: 100-Meter Parabolic Truss

Deployed Diameter = 100 m
Stowed Diameter = 1.36 m
Core Strut Length = 4.8 m
Deployed Truss Depth = 4.0 m
Stowed Package Length = 4.79 m
Slenderness Ratio of Struts = 1069

Number of Nodes = 760
Number of Struts = 3234

Number of Control Links = 6468
Number of Slider Joints = 760
Number of Revolute Joints = 21,549

Desired Formulations for Next-Generation Simulator

1. System Topology Must Be Presented to the Computer by a General Graphic Theory with Efficient Search Algorithms.

2. Kinematic and Equilibrium Equations of Individual Elements Must Be Generated by Efficient Symbolic Manipulations.

3. Necessary Transformation Matrices for Assembling the System Equations Must be Flexible Enough And Yet Arranged in a Form That Requires a Minimum User Decision and Resulting Always in Nonsingular System.

4. Formulations Should Allow Assembly of System Equations Either With or Without Constraints as Primary Variables.

5. Most Important of All, Modeling of Element Flexibility Should Allow Either Generalized Coordinates or Finite-Element Physical Coordinates.

Outstanding Algorithmic Difficulties

1. Solution Matrix for Fully Implicit Algorithm Becomes Nonsymmetric.

2. Elimination of Constraint Forces Complicates Matrix Profiles. On the Other Hand, Preservation of Constraint Forces as Independent Variables Increases Equation Size.

3. Augmentation of Constraint Equations Introduces Algebraic Equations Which Can Lead to Numerical Drifting in the Solution: Stabilization Becomes Important.

4. Member Flexibility and Joint Friction Introduces High-Frequency Solution Components and Sometimes Severe Nonlinearities.

5. Systematic Selection of Independent Set of Generalized Coordinates Present Formidable Challenge.

6. Determination of Initial Conditions from a Known Partial Set of Initial Conditions Is Often a Difficult Task.

7. Finally, Matching a Particular Formulation with a Most Suitable Solution Algorithm Requires an In-depth Investigation of the Combined Characteristics of the System Equations and Numerical Algorithms.

Recommended Formulation

1. Dual Formulations:

   Newton/Euler for Rigid Bodies.
   Lagrange/Variational for Flexible Bodies.

2. Reference Frames:

   Both Kinematically and Dynamically Specified.

3. System Variables:

   Absolute Velocity for Dynamically Specified Sub-systems.
   Relative Velocity for Kinematically Specified Sub-systems.
   Generalized Momentum for Some Complex Sub-systems.
   Lagrange Multipliers for Closed Loops and Kinematic Constraints.

4. System Topology:

   Index Lists, Depth-First and Width-First Search Algorithms.

5. Treatment Constraints:

   Consistency Conditions for Kinematics and Closed Loops.
   Partitioning Algorithms for Parallel Computations.

6. Equation Generation:

   Numercial/Symbolic Calculations.

Recommended Computational Algorithms

1. Integrators:

   Semi-Explicit Methods for Rigid Bodies.
   Semi-Implicit Methods for Flexible Bodies.

2. Rotation Update:

   4-Parameter Euler Transformation.
   Euler-Rodrigue Rotation Matrix.

3. For Systems with Constraint Index ≥ 2:

   Special Equation Augmentation.
   Constraint Stabilization.

4. Provisions for Penalty Methods for Handling Constraint Equations.

4. Concurrent Computations:

   Partitioning Strategies.
   Software Considerations.
   Minimal Communications Algorithms.