Linear-Time Dynamics Using Lagrange Multipliers

David Baraff
CMU

Presentation by: Elif Tosun

Outline

- Introduction
- Motivation
- Lagrange Multiplier Formulation
- Approach
- Sparse Solution
- Auxiliary Constraints
- Results

Introduction

Problem: "Forward simulation with constraints"

Properties:

- Sparse constraints for articulated figures

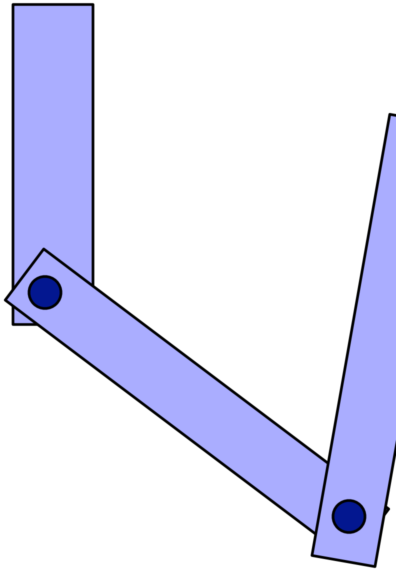
  n bodies, O(n) constraints
- Nearly or completely acyclic systems - robot arms, humans...

Methods:

7. *Reduced Coordinate*: Reduce the # of coordinates needed to describe system's state
8. *Constraint Forces*: Introduce additional forces into system to maintain constraints (*)

Very Basic Example

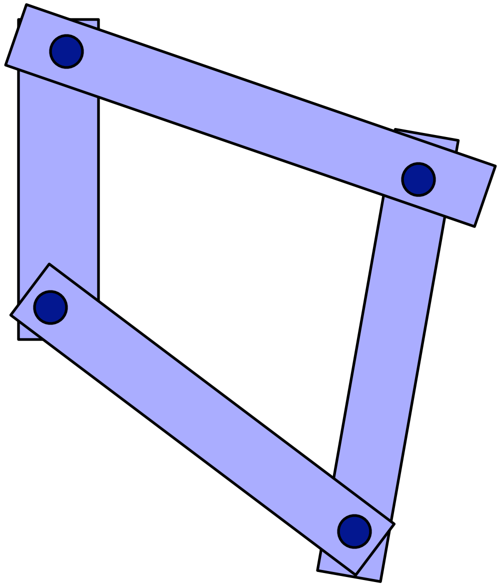
- Given: n bodies



- Primary Constraints
  - n-1
  - Each between 2 bodies
  - Acyclic
  - Can be non-holonomic (can be velocity dependent)
  - Must be equality

Example

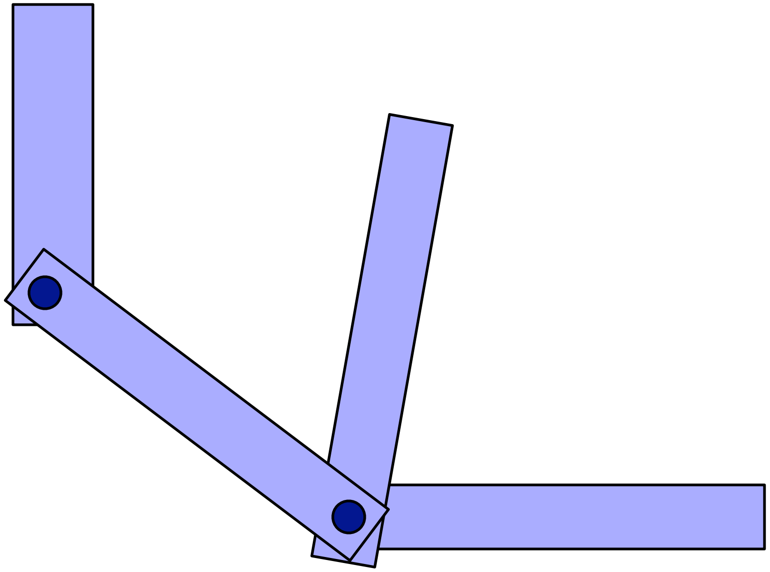
- Given: n bodies



- Auxiliary Constraints
  - Closed loops

Example

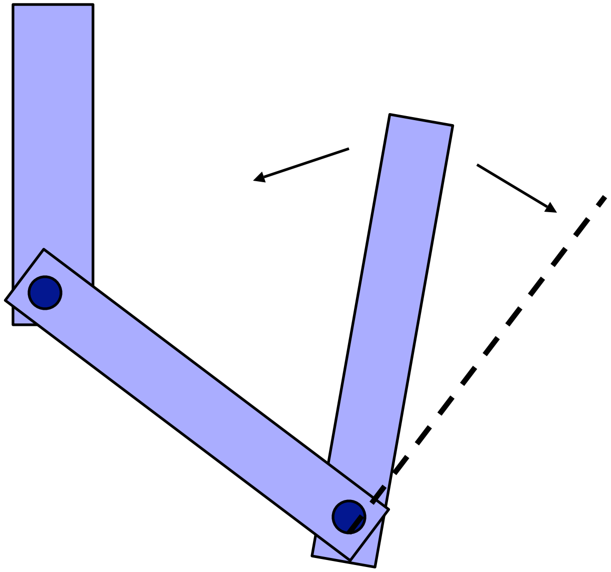
- Given: n bodies



- Auxiliary Constraints
  - Closed loops
  - Between 3+ bodies

Example

- Given: n bodies



- Auxiliary Constraints
  - Closed loops
  - Between 3+ bodies
  - Inequality Constraints

Overview of method

- Direct (non-iterative) method
- Constraints can be of various dimensions
- Bodies need not be rigid
- Uses a very simple sparse-matrix technique
- No steep learning curve
- Easy to implement
- Auxiliary method can be used with reduced coordinate methods too!

Motivation

- Given: a system with *m* d.o.f (called "*maximal coordinates*")
  + set of constraints that remove *c* d.o.f.

Reduced Coordinate Methods:

Method: Parameterize remaining $n=m-c$ d.o.f. using reduced set of *n* coordinates (called "*generalized coordinates*")

Cons:

- parameterization very hard
- if one can be found - need $O(n^3)$ time needed for acceleration computation.

Pros :

- loop-free articulated bodies - $O(n)$ time achievable
- Eliminates drifting problem in multiplier methods (need constraint stabilization)
- “may” run faster due to larger time steps taken by integrator

Motivation

Lagrange Multiplier Methods:

Method:

- use the maximal coordinates (*m* d.o.f.) + *new* constraint forces.
- Basis for constraint forces known apriori.
- Lagrange multipliers: vector of scalar coordinates for linear combination

Pros :

- Allow an arbitrary set of constraints to be combined.
- Allow/encourage highly modular knowledge/software design (bodies, constraints, geometry)
- Handle non-holonomic constraints(e.g. velocity-dependent)
- No need for parameterization

Cons:

- solving a O(n)x O(n) system.

BUT: This method takes linear time!

Lagrange Multiplier Formulation

- Bodies: $$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}$$
- Constraints: $$\mathbf{j}_{i1}\dot{\mathbf{v}}_1 + \cdots + \mathbf{j}_{ik}\dot{\mathbf{v}}_k + \cdots + \mathbf{j}_{in}\dot{\mathbf{v}}_n + \mathbf{c}_i = \mathbf{0}$$
  - Linear condition on the acceleration
  - For primary: Only 2 j_{ik} will be non-zero for constraint i.
  - All constraints: $$\mathbf{J}\dot{\mathbf{v}} + \mathbf{c} = \mathbf{0}$$
  - "Workless force"

$$\mathbf{F}_i^c = \begin{pmatrix} \mathbf{j}_{i1}^T \\ \vdots \\ \mathbf{j}_{in}^T \end{pmatrix} \lambda_i \quad \Rightarrow \quad \mathbf{F}^c = \mathbf{J}^T \mathbf{l}$$

Want to find l s.t. constraint force + ext force produce motion that also satisfies constraints

Lagrange Multiplier Formulation

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}$$

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}^{c} = \mathbf{J}^{T}\mathbf{l}$$

$$\dot{\mathbf{v}} = \underbrace{\mathbf{M}^{-1}\mathbf{J}^{T}\mathbf{l}}_{}$$

$$\mathbf{J}\dot{\mathbf{v}} + \mathbf{c} = 0 \quad \Longrightarrow \quad \boxed{\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T}\boldsymbol{\lambda} = \mathbf{c}}$$

Formulation

- Block-matrix formulation
- Block dimensions: based on body and constraint dimensions:
  - Body dim: # d.o.f. when unconstrained
  - Constraint dim: # of d.o.f. it removes
- Let p = largest dimension among bodies. Block operations take const time.

Approach

Newton's Law

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{J}^T\boldsymbol{\lambda} + \mathbf{F}^{\text{ext}}$$

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda} + \mathbf{M}^{-1}\mathbf{F}^{\text{ext}}$$

constraints

$$\mathbf{J}\dot{\mathbf{v}} + \mathbf{c} = \mathbf{0}$$

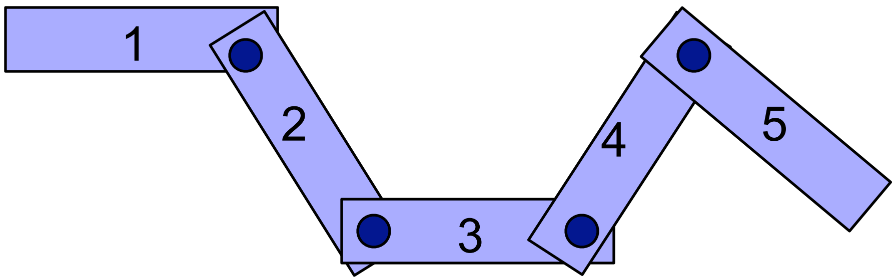
$$\mathbf{J}(\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda} + \mathbf{M}^{-1}\mathbf{F}^{\text{ext}}) + \mathbf{c} = \mathbf{0}$$

$$\mathbf{A} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T \quad \text{and} \quad \mathbf{b} = -(\mathbf{J}\mathbf{M}^{-1}\mathbf{F}^{\text{ext}} + \mathbf{c})$$

- Now all we need is to solve is: A l = b

A l = b

- if A not too large : Cholesky
- if Serial chain - Banded Cholesky – can take O(n)

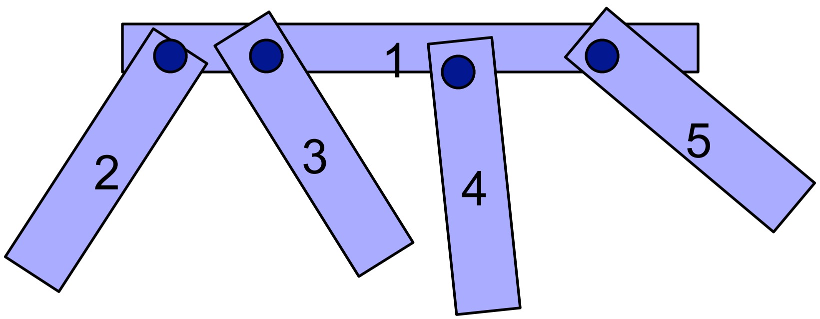


$$JM^{-1}J^{T} = \begin{bmatrix} A & A & 0 & 0 \\ A & A & A & 0 \\ 0 & A & A & A \\ 0 & 0 & A & A \end{bmatrix}$$

$$J = \begin{bmatrix} X & X & & & \\ & X & X & & \\ & & X & X & \\ & & & X & X \end{bmatrix} \qquad M = \begin{bmatrix} Y & & & & \\ & Y & & & \\ & & Y & & \\ & & & Y & \\ & & & & Y \end{bmatrix}$$

A l = b

- If branched – A is completely dense… $O(n^3)$



$JM^{-1}J^T =$

A	A	A	A
A	A	A	A
A	A	A	A
A	A	A	A

J =

X	X			
X		X		
X			X	
X				X

M =

Y				
	Y			
		Y		
			Y	
				Y

Need a sparse formulation!

Sparse Formulation

$$\underbrace{\begin{pmatrix} \mathbf{M} & -\mathbf{J}^T \\ -\mathbf{J} & \mathbf{0} \end{pmatrix}}_{H} \begin{pmatrix} \mathbf{y} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{b} \end{pmatrix}$$

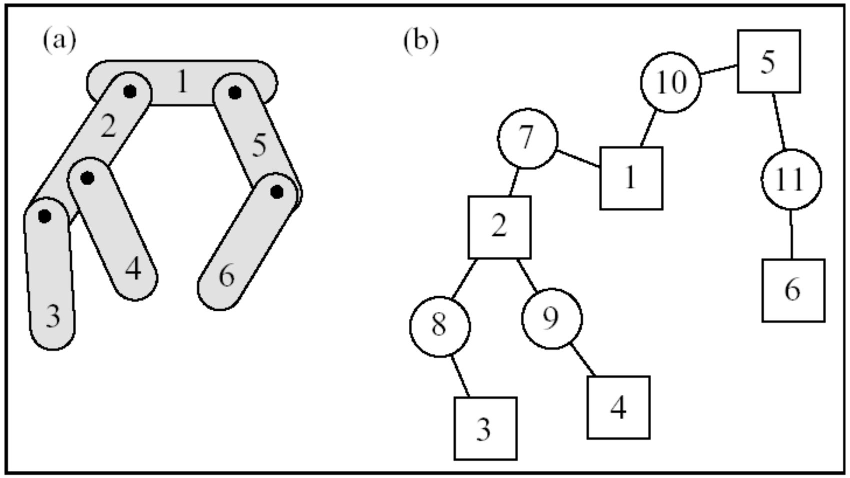
$My - J^T l = 0 \Rightarrow y = M^{-1} J^T l$

$-Jy = -b \Rightarrow JM^{-1}J^T l = b. \Rightarrow A\,l = b$

- Robotics/mech.eng literature
- H is <u>always</u> sparse
- Can be solved in O(n)

Non-Linear Solution

- Consider graph of H:

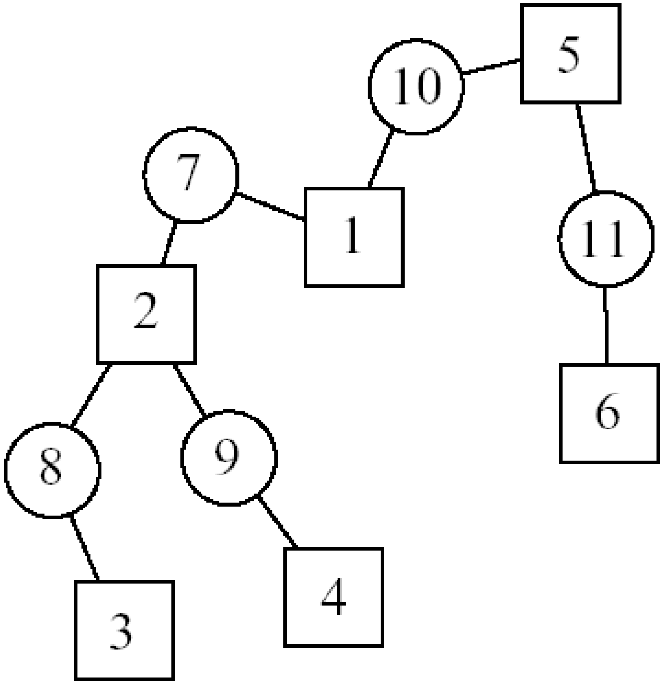


- Where the matrix looks like:

$$
\mathbf{H} = \begin{pmatrix}
\mathbf{M}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{11}^T & \mathbf{0} & \mathbf{0} & \mathbf{j}_{41}^T & \mathbf{0} \\
\mathbf{0} & \mathbf{M}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{12}^T & \mathbf{j}_{22}^T & \mathbf{j}_{32}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{M}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{23}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{34}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{45}^T & \mathbf{j}_{55}^T \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{56}^T \\
\mathbf{j}_{11} & \mathbf{j}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{j}_{22} & \mathbf{j}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{j}_{32} & \mathbf{0} & \mathbf{j}_{34} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{j}_{41} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{45} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{55} & \mathbf{j}_{56} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{pmatrix}
$$

- Factor $H=LDL^T$ $(O(n^3))$
- Then solve $(O(n^2))$
  $H=LDL^Tx=(0; -b)$

Linear Solution



- Sparse Matrix Theory:

"A matrix whose graph is acyclic possesses a **perfect elimination order**"

→ H can be reordered s.t. when factored the matrix L will be just as sparse as H.

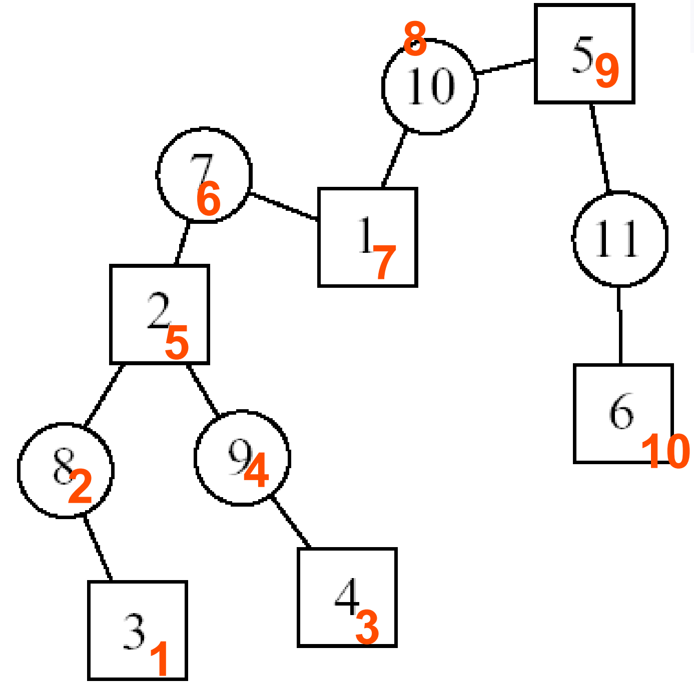
$$\mathbf{H} = \begin{pmatrix}
\mathbf{M}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{11}^T & \mathbf{0} & \mathbf{0} & \mathbf{j}_{41}^T & \mathbf{0} \\
\mathbf{0} & \mathbf{M}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{12}^T & \mathbf{j}_{22}^T & \mathbf{j}_{32}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{M}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{23}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{34}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{45}^T & \mathbf{j}_{55}^T \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{56}^T \\
\mathbf{j}_{11} & \mathbf{j}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{j}_{22} & \mathbf{j}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{j}_{32} & \mathbf{0} & \mathbf{j}_{34} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{j}_{41} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{45} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{55} & \mathbf{j}_{56} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{pmatrix}$$

Linear Solution

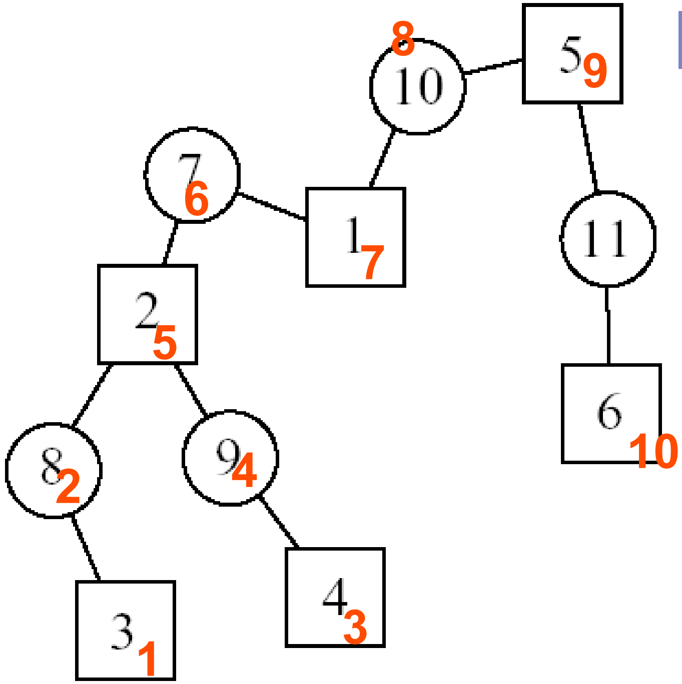
- Sparse Matrix Theory:

"A matrix whose graph is acyclic possesses a "perfect elimination order"

→ H can be reordered s.t. when factored the matrix L will be just as sparse as H.



$$
\mathbf{H} = \begin{pmatrix}
\mathbf{M}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{11}^T & \mathbf{0} & \mathbf{0} & \mathbf{j}_{41}^T & \mathbf{0} \\
\mathbf{0} & \mathbf{M}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{12}^T & \mathbf{j}_{22}^T & \mathbf{j}_{32}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{M}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{23}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{34}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_5 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{45}^T & \mathbf{j}_{55}^T \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{56}^T \\
\mathbf{j}_{11} & \mathbf{j}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{j}_{22} & \mathbf{j}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{j}_{32} & \mathbf{0} & \mathbf{j}_{34} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{j}_{41} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{45} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{55} & \mathbf{j}_{56} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
\mathbf{M}_3 & \mathbf{j}_{23}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{j}_{23} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{M}_4 & \mathbf{j}_{34}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{j}_{34} & \mathbf{0} & \mathbf{j}_{32} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{j}_{22}^T & \mathbf{0} & \mathbf{j}_{32}^T & \mathbf{M}_2 & \mathbf{j}_{12}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{12} & \mathbf{0} & \mathbf{j}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{11}^T & \mathbf{M}_1 & \mathbf{j}_{41}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{41} & \mathbf{0} & \mathbf{j}_{45} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{45}^T & \mathbf{M}_5 & \mathbf{0} & \mathbf{j}_{55}^T \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{M}_6 & \mathbf{j}_{56}^T \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{j}_{55} & \mathbf{j}_{56} & \mathbf{0}
\end{pmatrix}
$$



$$\begin{pmatrix}
\mathbf{M}_3 & \mathbf{j}_{23}^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{j}_{23} & 0 & 0 & 0 & \mathbf{j}_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{M}_4 & \mathbf{j}_{34}^T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{j}_{34} & 0 & \mathbf{j}_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \mathbf{j}_{22}^T & 0 & \mathbf{j}_{32}^T & \mathbf{M}_2 & \mathbf{j}_{12}^T & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mathbf{j}_{12} & 0 & \mathbf{j}_{11} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{j}_{11}^T & \mathbf{M}_1 & \mathbf{j}_{41}^T & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{j}_{41} & 0 & \mathbf{j}_{45} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{j}_{45}^T & \mathbf{M}_5 & 0 & \mathbf{j}_{55}^T \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{M}_6 & \mathbf{j}_{56}^T \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{j}_{55} & \mathbf{j}_{56} & 0
\end{pmatrix}$$

Ordering

- Graph is a rooted tree: parent child relationship between every edge.
- Every node's index greater than its children's indices. (DFS)

Bookkeeping (child/parent relations)

- In each row only one non-zero block every occurs to the right of the diagonal (at most 1 parent/node)
- Eliminates one inner for loop in each algorithm

L can be computed in O(n) time and LDLtx=(b) can be solved in O(n) time.

$$\dot{\mathbf{v}} = \mathbf{M}^{-1}(\mathbf{J}^T\boldsymbol{\lambda} + \mathbf{F}^{\mathrm{ext}})$$

Auxiliary Constraints

Idea:

- While computing multipliers for auxiliary constraints “anticipate” responses of primary constraints due to auxiliary forces.
- Once auxiliary constraints are computed, add primary constraints into system, which won't violate auxiliary constraints due to “anticipation”

Notation:

- For auxiliary constraints
  - K ~ J
  - m ~ l

Full Algorithm

1. Formulate sparse matrix H for primary constraints, and factor O(n)

2. Given F^{ext},

- Solve $\mathbf{A\, l_1 = -(JM^{-1}F^{ext}+c)}$ for $\mathbf{l_1}$
- Compute primary constraint force due to F^{ext} : $\mathbf{J^T l_1}$
- Compute system's acceleration without auxiliary constraints: $\dot{\mathbf{v}}^{\mathbf{aux}} = \mathbf{M^{-1}(J^T l_1 + F^{ext})}$.

O(n)

$$\mathbf{J^a\dot{v}+c^a = a}$$

$$J^a(M^{-1}(F^{resp}+k_i)m + \dot{v}^{aux}) + c^a = a.$$

3. For each of k auxiliary constraints

- Solve $\mathbf{Al_m = -(JM^{-1}k_m)}$
- Compute response force by primary to constraint k $_{\mathbf{m}}$

  $\mathbf{F^{resp} = J^T l_m}$
- Fill coefficient matrix

O(kn)

4. Solve for auxiliary multipliers m

$O(k^3)$

5. Given Km (the auxiliary constraint force)

Solve $\mathbf{A}\ \mathbf{l}_{\mathbf{final}} = -(\mathbf{JM}^{-1}(\mathbf{K}\ m + \mathbf{F}^{\mathbf{ext}}) + \mathbf{c})$ O(n)

Compute primary constraints response to $\mathbf{K}m + \mathbf{F}^{\mathbf{ext}}$

Total constraint force = $\mathbf{K}\ m + \mathbf{J}^{\mathbf{T}}\mathbf{l}_{\mathbf{final}}$

Total external forces = $\mathbf{F}^{\mathbf{ext}}$

NET FORCE acting = $\mathbf{K}\ m + \mathbf{J}^{\mathbf{T}}\mathbf{l}_{\mathbf{final}} + \mathbf{F}^{\mathbf{ext}}$

6.Solve for net acceleration of system and move to next step.

TOTAL RUNNING TIME: $O(n) + O(kn+k^3)$

Results

- Run on SGI indigo 250Mhz, R4400 processors

2D system :	54 primary constraints	7.75ms
	96 primary, 12 auxiliary	~25ms
3D system :	96 primary, 3 auxiliary	18 ms
	127 primary:	~45ms

- Comparisons:
  - Baraff (Lagrange, $O(n^3)$)
    - For smaller systems 2 fold improvement
    - For bigger systems as big as a factor of 40
  - Schröder (Reduced Coordinate, $O(n)$)
    - Competitive + no need for SVD which causes ill-conditioning.
  - Bramble (Lagrange, Iterative, $O(n^2)$)
    - For smaller matrices competitive, for larger clearly faster