

COORDINATE REDUCTION STRATEGIES IN MULTIBODY DYNAMICS: A REVIEW

E. Pennestri¹ P.P. Valentini¹

¹Università di Roma Tor Vergata - Italy
Dipartimento di Ingegneria Meccanica

Conference on Multibody System Dynamics, Pitesti 2007

Presentation outline

- 1 Multibody dynamics formulation
The null space and the range space methods
- 2 Orthogonalization of constraints: Maggi like methods
- 3 Numerical methods
- 4 The coordinate partitioning method
- 5 The Udwadia-Kalaba formulation
- 6 Comparison of methods
- 7 Conclusions

The set of coordinates

- 1 Minimal set of coordinates (ODE)
- 2 Redundant set of coordinates (DAE)

The set of coordinates

- 1 Minimal set of coordinates (ODE)
- 2 Redundant set of coordinates (DAE)

Matrix method for computing the d.o.f.

Let

- n the number of coordinates q_k ($k = 1, \dots, n$) adopted for the definition of the mechanical system configuration;
- p the number of independent equations that can be established between the infinitesimal variations $(\delta q_1, \delta q_2, \dots, \delta q_n)$.

The degrees-of-freedom of the mechanical system follow from (Wittaker, 1917)

$$F = n - p . \quad (1)$$

It should be observed that p coincides with the rank r of the equations constraints **Jacobian matrix**.

Matrix method for computing the d.o.f.

Let

- n the number of coordinates q_k ($k = 1, \dots, n$) adopted for the definition of the mechanical system configuration;
- p the number of independent equations that can be established between the infinitesimal variations $(\delta q_1, \delta q_2, \dots, \delta q_n)$.

The degrees-of-freedom of the mechanical system follow from (Wittaker, 1917)

$$F = n - p. \quad (1)$$

It should be observed that p coincides with the rank r of the equations constraints **Jacobian matrix**.

Matrix method for computing the d.o.f.

Let

- n the number of coordinates q_k ($k = 1, \dots, n$) adopted for the definition of the mechanical system configuration;
- p the number of independent equations that can be established between the infinitesimal variations $(\delta q_1, \delta q_2, \dots, \delta q_n)$.

The degrees-of-freedom of the mechanical system follow from (Wittaker, 1917)

$$F = n - p. \quad (1)$$

It should be observed that p coincides with the rank r of the equations constraints **Jacobian matrix**.

The Equations of Dynamics

Differential-Algebraic Equations (DAE) system (differential index 3)

$$[M] \{\ddot{q}\} + [\Psi_q^T] \{\lambda\} = \{F_e\} \quad (2a)$$

$$\{\Psi\} = 0 \quad (2b)$$

The Equations of Dynamics

Differential-Algebraic Equations (DAE) system (differential index 1)

$$[M] \{\ddot{q}\} + [\Psi_q^T] \{\lambda\} = \{F_e\} \quad (3a)$$

$$[\Psi_q] \{\ddot{q}\} = \{\gamma\} \quad (3b)$$

Main steps of the simplest numerical integration procedure

- 1 Given the **initial conditions** at time t : $\{q(t)\}$, $\{\dot{q}(t)\}$.
- 2 Compute accelerations $\{\ddot{q}(t)\}$ and Lagrange multipliers $\{\lambda\}$

$$\begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^{-1} \begin{Bmatrix} F_e \\ \gamma \end{Bmatrix} \quad (4)$$

- 3 Estimate positions $\{q(t + \Delta t)\}$ and velocities $\{\dot{q}(t + \Delta t)\}$ at time $t + \Delta t$ by finite differences.

Main steps of the simplest numerical integration procedure

- 1 Given the **initial conditions** at time t : $\{q(t)\}$, $\{\dot{q}(t)\}$.
- 2 Compute accelerations $\{\ddot{q}(t)\}$ and Lagrange multipliers $\{\lambda\}$

$$\begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^{-1} \begin{Bmatrix} F_e \\ \gamma \end{Bmatrix} \quad (4)$$

- 3 Estimate positions $\{q(t + \Delta t)\}$ and velocities $\{\dot{q}(t + \Delta t)\}$ at time $t + \Delta t$ by finite differences.

Main steps of the simplest numerical integration procedure

- 1 Given the **initial conditions** at time t : $\{q(t)\}$, $\{\dot{q}(t)\}$.
- 2 Compute accelerations $\{\ddot{q}(t)\}$ and Lagrange multipliers $\{\lambda\}$

$$\begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^{-1} \begin{Bmatrix} F_e \\ \gamma \end{Bmatrix} \quad (4)$$

- 3 Estimate positions $\{q(t + \Delta t)\}$ and velocities $\{\dot{q}(t + \Delta t)\}$ at time $t + \Delta t$ by finite differences.

Range space method

For the purpose of block solution, the mass matrix is first decomposed such that $[M] = [M]^{1/2} [M]^{1/2}$, then (3) is rewritten as follows:

$$\begin{bmatrix} L & 0 \\ H^T & -L_1^T \end{bmatrix} \begin{bmatrix} L^T & H \\ 0 & L_1 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q \\ \gamma \end{Bmatrix} \quad (5)$$

where

$$[L] = [M]^{1/2}, \quad (6)$$

$$[H] = [L^{-1}]^T [\Psi_q]^T, \quad (7)$$

$$[L_1] = \text{chol} \left([H]^T [H] \right). \quad (8)$$

Range space method (cont.)

1 Compute Lagrange multipliers

$$\{\lambda\} = [L]^{-1} \{\lambda_1\} = [L_1]^{-1} \left([H]^T [L]^{-1} \{Q\} - \{\gamma\} \right) \quad (9)$$

2 Compute the generalized accelerations vector

$$\{\ddot{q}\} = [L^{-1}]^T \left([L]^{-1} \{Q\} - [H] \{\lambda\} \right) \quad (10)$$

Range space method (cont.)

1 Compute Lagrange multipliers

$$\{\lambda\} = [L]^{-1} \{\lambda_1\} = [L_1]^{-1} \left([H]^T [L]^{-1} \{Q\} - \{\gamma\} \right) \quad (9)$$

2 Compute the generalized accelerations vector

$$\{\ddot{q}\} = [L^{-1}]^T \left([L]^{-1} \{Q\} - [H] \{\lambda\} \right) \quad (10)$$

Range space method (cont.)

Since $[L_1]$ is an upper triangular matrix, the use of specialized procedures (*e.g.* DTRDI of LAPACK) is recommended for its inversion.

Null space method

The matrix

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}$$

is factored as follows

$$\begin{bmatrix} M_{11} & M_{12} & \Psi_1^T \\ M_{21} & M_{22} & \Psi_2^T \\ \Psi_1 & \Psi_2 & 0 \end{bmatrix} = \begin{bmatrix} I & 0 & M_{11}R_1^{-1} \\ R_2^T R_1^{-T} & I & M_{21}R_1^{-1} \\ 0 & 0 & L \end{bmatrix} \cdot \begin{bmatrix} 0 & \tilde{M}_{12}\tilde{L}^{-T} & R_1^T \\ 0 & \tilde{L} & 0 \\ R_1 & R_2\tilde{L}^T & 0 \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & \tilde{L}^T & 0 \\ 0 & 0 & L^T \end{bmatrix}$$

Comparison.

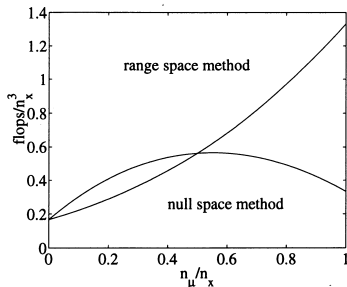


Figure: Floating point operations required by the two methods

- $n_\mu < \frac{n_x}{2}$ one should use the Range Space method
- $n_\mu > \frac{n_x}{2}$ one should use the Null Space method

Pitfalls

There are different problems associated with the approaches just described:

- 1 Positions and velocity constraints are not necessarily satisfied by the computed numerical solution (solution drift).
- 2 The constraints equations must be independent. (Thus the procedures do not work for overconstrained mechanical systems).

Pitfalls

There are different problems associated with the approaches just described:

- 1 Positions and velocity constraints are not necessarily satisfied by the computed numerical solution (solution drift).
- 2 The constraints equations must be independent. (Thus the procedures do not work for overconstrained mechanical systems).

Baumgarte's stabilization

$$\Psi \equiv \Psi(q, t) = 0$$

$$\dot{\Psi} \equiv \Psi_q \dot{q} - \{\Psi_t\} = 0 ,$$

$$\ddot{\Psi} \equiv \Psi_q \ddot{q} - \gamma = 0$$

The criterion of Baumgarte tries to damp oscillations of constraints violations

$$\ddot{\Psi} + 2\alpha\dot{\Psi} + \beta^2\Psi = 0 , \quad (11)$$

For critical damping $\alpha = \beta$.

Baumgarte's stabilization

$$\Psi \equiv \Psi(q, t) = 0$$

$$\dot{\Psi} \equiv \Psi_q \dot{q} - \{\Psi_t\} = 0 ,$$

$$\ddot{\Psi} \equiv \Psi_q \ddot{q} - \gamma = 0$$

The criterion of Baumgarte tries to damp oscillations of constraints violations

$$\ddot{\Psi} + 2\alpha\dot{\Psi} + \beta^2\Psi = 0 , \quad (11)$$

For critical damping $\alpha = \beta$.

Baumgarte's stabilization (cont.)

During the numerical integration of

$$\begin{aligned} [M] \{\ddot{q}\} + [\Psi_q^T] \{\lambda\} &= \{F_e\} \\ [\Psi_q] \{\ddot{q}\} &= \{\gamma\} \end{aligned}$$

position and velocity constraints fail to be satisfied, the right side of the acceleration constraint is altered as follows

$$\{\bar{\gamma}\} = \{\gamma\} - 2\alpha \{\Psi\} - \beta \{\dot{\Psi}\} \quad (12)$$

where α and β are the gain parameters usually chosen such that $\alpha = \beta$.

Baumgarte's stabilization (cont.)

The DAE system to be integrated is thus transformed to

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q \\ \bar{\gamma} \end{Bmatrix} \quad (13)$$

Other numerical integration procedures of multibody dynamics DAE systems

- 1 Maggi like methods. (Orthogonalization of constraints).
- 2 Coordinate partitioning method.
- 3 Udwadia-Kalaba dynamic formulation based on the least action Gauss' principle.

Other numerical integration procedures of multibody dynamics DAE systems

- 1 Maggi like methods. (Orthogonalization of constraints).
- 2 **Coordinate partitioning method.**
- 3 Udwadia-Kalaba dynamic formulation based on the least action Gauss' principle.

Other numerical integration procedures of multibody dynamics DAE systems

- 1 Maggi like methods. (Orthogonalization of constraints).
- 2 Coordinate partitioning method.
- 3 Udwadia-Kalaba dynamic formulation based on the least action Gauss' principle.

Coordinate
reduction

Pennestri,
Valentini

Multibody
dynamics
formulation

The null space and
the range space
methods

Orthogonalization
of constraints:
Maggi like
methods

Numerical
methods

The
coordinate
partitioning
method

The Udwadia-
Kalaba
formulation

Comparison
of methods

Conclusions

Maggi like methods - Orthogonalization of constraints

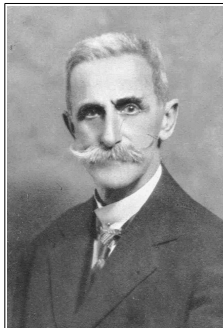


Figure: Gian Antonio Maggi, Milano (1856-1937)

$$[M] \{\ddot{q}\} + [\Psi_q]^T \{\lambda\} = \{Q\} \quad (14)$$

The idea behind the orthogonalization of constraints is the **elimination of reaction forces.**

For this purpose we need a matrix $[V]$ such that:

$$\{\dot{q}\} = [V] \{\dot{v}\}$$

$$[\Psi_q] [V] = [0]$$

$$[M] \{\ddot{q}\} + [\Psi_q]^T \{\lambda\} = \{Q\} \quad (14)$$

The idea behind the orthogonalization of constraints is the **elimination of reaction forces.**

For this purpose we need a matrix $[V]$ such that:

$$\{\dot{q}\} = [V] \{\dot{v}\}$$

$$[\Psi_q] [V] = [0]$$

Since

$$\{\ddot{q}\} = [V] \{\ddot{v}\} + [\dot{V}] \{\dot{v}\}$$

the equation of dynamics $[M] \{\ddot{q}\} + [\Psi_q]^T \{\lambda\} = \{Q\}$ are transformed into

$$[V]^T [M] [V] \{\ddot{v}\} = [V]^T \{Q\} + [V]^T [M] [\dot{V}] \{\dot{v}\} \quad (15)$$

Since

$$\{\ddot{q}\} = [V] \{\ddot{v}\} + [\dot{V}] \{\dot{v}\}$$

the equation of dynamics $[M] \{\ddot{q}\} + [\Psi_q]^T \{\lambda\} = \{Q\}$ are transformed into

$$[V]^T [M] [V] \{\ddot{v}\} = [V]^T \{Q\} + [V]^T [M] [\dot{V}] \{\dot{v}\} \quad (15)$$

- 1 The set $\{v\}$ of independent coordinates and the matrix $[V]$ is not unique.
- 2 There are different numerical methods for the computation of matrix $[V]$

An algebraic method for the computation of matrix $[V]$

Let us append to the constraint vector $\{\Psi\}$ the vector of conditions $\{\Phi\}$ that can be established between q (set of coordinates) and v (set of independent coordinates)

$$\{\Gamma\}_{(n) \times 1} = \begin{Bmatrix} \Psi(q) \\ \Phi(v, q) \end{Bmatrix} = 0 \quad (16)$$

The time derivative leads to

$$[\Gamma_v]_{(m+F) \times F} \{\dot{v}\} + [\Gamma_q]_{(n) \times n} \{\dot{q}\} = 0 \quad (17)$$

Assuming $[\Gamma_q]$ non singular and introduced the matrix

$$[V]_{n \times F} = -[\Gamma_q]^{-1} [\Gamma_v] \quad (18)$$

one obtains

$$\{\dot{q}\} = [V] \{\dot{v}\} \quad (19)$$

The time derivative leads to

$$[\Gamma_v]_{(m+F) \times F} \{\dot{v}\} + [\Gamma_q]_{(n) \times n} \{\dot{q}\} = 0 \quad (17)$$

Assuming $[\Gamma_q]$ non singular and introduced the matrix

$$[V]_{n \times F} = -[\Gamma_q]^{-1} [\Gamma_v] \quad (18)$$

one obtains

$$\{\dot{q}\} = [V] \{\dot{v}\} \quad (19)$$

Numerical methods for computing the matrix $[V]$

- 1 Zero Eigenvalue method (Walton and Steeves, 1969)
- 2 QR decomposition (Kim and Vanderploeg, 1986)
- 3 Singular value decomposition (SVD), (Mani and Haug, 1985), Likins and Singh (1985)
- 4 The PUTD decomposition (Amirouche et al., 1988)
- 5 The QTZ decomposition (Vita and Pennestrì, 2004)
- 6 The Schur decomposition (This investigation, 2007)

Zero Eigenvalue method - Walton and Steeves (1969), Kamman and Huston (1985)

Let us form the matrix

$$[H] = [\Psi_q]^T [\Psi_q] \quad (20)$$

Since $[H]_{n \times n}$ is symmetric, all the eigenvalues are positive and there exists the similarity transform

$$[T]^T [H] [T] = [\Lambda] \quad (21)$$

where $[T]$ is the orthogonal matrix of eigenvectors and $[\Lambda]$ the diagonal matrix of eigenvalues.

The matrix $[T]$ can be partitioned as follows

$$[T] = \begin{bmatrix} [T_1]_{n \times (n-r)} & [T_2]_{n \times r} \end{bmatrix} \quad (22)$$

where $[T_1]$ are the eigenvectors associated to zero eigenvalues.

The condition of orthogonality holds

$$[\Psi_q][T_1] = [0] \quad (23)$$

$$\therefore [V] = [T_1]$$

The condition of orthogonality holds

$$[\Psi_q][T_1] = [0] \quad (23)$$

$$\therefore [V] = [T_1]$$

QR decomposition - Kim and
Vanderploeg (1986)

$$[\Psi_q]^T = \begin{bmatrix} [Q_1]_{n \times (n-F)} & [Q_2]_{n \times F} \end{bmatrix} \begin{bmatrix} [R_1]_{r \times m} \\ [0]_{F \times m} \end{bmatrix} = [Q_1] [R_1] \quad (24)$$

where $[Q_1]$ and $[Q_2]$ simultaneously satisfy the conditions

$$[Q_2]^T [Q_1] = [0] \quad (25)$$

$$[Q_2]^T [Q_2] = [I] \quad (26)$$

QR decomposition (cont.)

$$[Q_2]^T [\Psi_q]^T = [Q_2]^T [Q_1] [R_1] = [0] \quad (27)$$

$$\therefore [V] = [Q_2]$$

SVD method - Singh and Likins (1985), Mani and Haug (1985)

$$\begin{aligned} [\Psi_q]^T &= \begin{bmatrix} [W_d]_{n \times r} & [W_i]_{n \times (n-r)} \end{bmatrix} \begin{bmatrix} [\Lambda_1]_{r \times r} \\ [0]_{(n-r) \times r} \end{bmatrix} [U]_{r \times r}^T \\ &= [W_d] [\Lambda_1] [U]^T \end{aligned} \quad (28)$$

where $[W_i]^T [W_d] = [0]$

SVD method (cont.)

$$[W_i]^T [\Psi_q]^T = [W_i]^T [W_d] [\Lambda_1] [U]^T = [0] \quad (29)$$

$$\therefore [V] = [W_i]^T$$

PUTD method - Amirouche *et al.* (1988)

The method's first step is to obtain the Householder transform matrix $[H]$ of $[\Psi_q]^T$.

Then, through the Gram-Schmidt process, it is identified a matrix $[D_2]$ such that

$$[D_2]_{(n-r) \times n} [H]_{n \times n}^T [\Psi_q]^T = [0] \quad (30)$$

The matrix $[D_2]$ has the following block structure

$$[D_2] = \begin{bmatrix} [0]_{(n-r) \times r} & [I]_{(n-r) \times (n-r)} \end{bmatrix} \quad (31)$$

$$\therefore [V] = [H] [D_2]^T$$

Note: $[D_2] [H]^T = [Q_2]^T$.

The matrix $[D_2]$ has the following block structure

$$[D_2] = \begin{bmatrix} [0]_{(n-r) \times r} & [I]_{(n-r) \times (n-r)} \end{bmatrix} \quad (31)$$

$$\therefore [V] = [H] [D_2]^T$$

Note: $[D_2] [H]^T = [Q_2]^T$.

Schur decomposition - This investigation

The Schur decomposition of the matrix $[H]$ has the form

$$[K]^T [H] [K] = [U] \quad (32)$$

where

- $[K]$ is a unitary matrix
- $[U]$ an upper triangular matrix.

The eigenvalues of $[H]$ are on the diagonal of $[U]$.

Schur decomposition (cont.)

The matrix $[K]$ can be partitioned as follows

$$[K] = \begin{bmatrix} [K_1]_{n \times (n-r)} & [K_2]_{n \times r} \end{bmatrix} \quad (33)$$

where $[K_1]$ are the generalized Schur vectors associated to zero eigenvalues.

Schur decomposition (cont.)

The condition of orthogonality holds

$$[\Psi_q] [K_1] = [0] \quad (34)$$

$$\therefore [V] = [K_1]$$

The coordinate partitioning method

In multibody dynamics the coordinate partitioning method is an historically important computational scheme.

The scheme received extensive application in multibody dynamics by Haug and his coworkers Wehage, Nikravesh and Shabana.

The coordinate partitioning method (cont.)

The set of coordinates is partitioned as follows:

$$\{q\} = \begin{Bmatrix} v \\ u \end{Bmatrix} \quad (35)$$

where

- $\{u\}$ = **dependent** coordinates
- $\{v\}$ = **independent** coordinates

$$\{u\} = \{h(v)\} \quad (36)$$

The coordinate partitioning method (cont.)

The set of coordinates is partitioned as follows:

$$\{q\} = \begin{Bmatrix} v \\ u \end{Bmatrix} \quad (35)$$

where

- $\{u\}$ = **dependent** coordinates
- $\{v\}$ = **independent** coordinates

$$\{u\} = \{h(v)\} \quad (36)$$

Coordinate
reduction

**Pennestri,
Valentini**

Multibody
dynamics
formulation

The null space and
the range space
methods

Orthogonalization
of constraints:
Maggi like
methods

Numerical
methods

**The
coordinate
partitioning
method**

The Udwadia-
Kalaba
formulation

Comparison
of methods

Conclusions

The coordinate partitioning method (cont.)

The velocity

$$[\Psi_q] \{\dot{q}\} = -\{\Psi_t\} \quad (37)$$

and acceleration

$$[\Psi_q] \{\ddot{q}\} = \{\gamma\} \quad (38)$$

constraints can respectively rearranged in the form

$$[\Psi_u] \{\dot{u}\} + [\Psi_v] \{\dot{v}\} = -\{\hat{\Psi}_t\} \quad (39)$$

$$[\Psi_u] \{\ddot{u}\} + [\Psi_v] \{\ddot{v}\} = \{\gamma\} \quad (40)$$

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q \\ \gamma \end{Bmatrix} \quad (41)$$

Equation of dynamics in partitioned form

$$[M^{vv}] \{\ddot{v}\} + [M^{vu}] \{\ddot{u}\} + [\Psi_v]^T \{\lambda\} = \{Q^v\} \quad (42)$$

$$[M^{uv}] \{\ddot{v}\} + [M^{uu}] \{\ddot{u}\} + [\Psi_u]^T \{\lambda\} = \{Q^u\} \quad (43)$$

$$[\Psi_u]_{m \times (n-F)} \{\ddot{u}\} + [\Psi_v]_{m \times F} \{\ddot{v}\} = \{\gamma\} \quad (44)$$

The coordinate partitioning
method (cont.)Eliminating λ and \ddot{u} one obtains

$$[\hat{M}] \{\ddot{v}\} = \{\hat{Q}\} \quad (45)$$

where

$$[\hat{M}] = [M^{vv}] - [M^{vu}] [\Psi_u]^{-1} [\Psi_v] \\ - [\Psi_v]^T \left([\Psi_u]^{-1}\right)^T \left([M^{uv}] - [M^{uu}] [\Psi_u]^{-1} [\Psi_v]\right) \quad (46)$$

and

$$\{\hat{Q}\} = \{Q^v\} - [M^{vu}] [\Psi_u]^{-1} \{\gamma\} \\ - [\Psi_v]^T \left([\Psi_u]^{-1}\right)^T \left(\{Q^u\} - [M^{uu}] [\Psi_u]^{-1} \{\gamma\}\right) \quad (47)$$

Steps of the coordinate partitioning method

- 1 Partition the vector $\{q\}$ of coordinates;
- 2 Given $\{v\}$ and $\{\dot{v}\}$ determine $\{u\}$ and $\{\dot{u}\}$ at time t
- 3 Compute \ddot{v} from

$$[\hat{M}] \{\ddot{v}\} = \{\hat{Q}\}$$

- 4 Compute \ddot{u} from

$$[\Psi_u] \{\ddot{u}\} = -[\Psi_v] \{\ddot{v}\} + \{\gamma\}$$

- 5 Integrate and compute $\{u\}$, $\{v\}$, $\{\dot{u}\}$, $\{\dot{v}\}$ at time $t + \Delta t$

Steps of the coordinate partitioning method

- 1 Partition the vector $\{q\}$ of coordinates;
- 2 Given $\{v\}$ and $\{\dot{v}\}$ determine $\{u\}$ and $\{\dot{u}\}$ at time t
- 3 Compute \ddot{v} from

$$[\hat{M}] \{\ddot{v}\} = \{\hat{Q}\}$$

- 4 Compute \ddot{u} from

$$[\Psi_u] \{\ddot{u}\} = -[\Psi_v] \{\ddot{v}\} + \{\gamma\}$$

- 5 Integrate and compute $\{u\}$, $\{v\}$, $\{\dot{u}\}$, $\{\dot{v}\}$ at time $t + \Delta t$

Steps of the coordinate partitioning method

- 1 Partition the vector $\{q\}$ of coordinates;
- 2 Given $\{v\}$ and $\{\dot{v}\}$ determine $\{u\}$ and $\{\dot{u}\}$ at time t
- 3 Compute \ddot{v} from

$$[\hat{M}] \{\ddot{v}\} = \{\hat{Q}\}$$

- 4 Compute \ddot{u} from

$$[\Psi_u] \{\ddot{u}\} = -[\Psi_v] \{\ddot{v}\} + \{\gamma\}$$

- 5 Integrate and compute $\{u\}$, $\{v\}$, $\{\dot{u}\}$, $\{\dot{v}\}$ at time $t + \Delta t$

Steps of the coordinate partitioning method

- 1 Partition the vector $\{q\}$ of coordinates;
- 2 Given $\{v\}$ and $\{\dot{v}\}$ determine $\{u\}$ and $\{\dot{u}\}$ at time t
- 3 Compute \ddot{v} from

$$[\hat{M}] \{\ddot{v}\} = \{\hat{Q}\}$$

- 4 Compute \ddot{u} from

$$[\Psi_u] \{\ddot{u}\} = -[\Psi_v] \{\ddot{v}\} + \{\gamma\}$$

- 5 Integrate and compute $\{u\}$, $\{v\}$, $\{\dot{u}\}$, $\{\dot{v}\}$ at time $t + \Delta t$

Steps of the coordinate partitioning method

- 1 Partition the vector $\{q\}$ of coordinates;
- 2 Given $\{v\}$ and $\{\dot{v}\}$ determine $\{u\}$ and $\{\dot{u}\}$ at time t
- 3 Compute \ddot{v} from

$$[\hat{M}] \{\ddot{v}\} = \{\hat{Q}\}$$

- 4 Compute \ddot{u} from

$$[\Psi_u] \{\ddot{u}\} = -[\Psi_v] \{\ddot{v}\} + \{\gamma\}$$

- 5 Integrate and compute $\{u\}$, $\{v\}$, $\{\dot{u}\}$, $\{\dot{v}\}$ at time $t + \Delta t$

Udwadia-Kalaba

If we let

$$\begin{aligned} [M]^{-1} &= [M]^{-\frac{1}{2}} [M]^{-\frac{1}{2}} , \\ [D] &= [\Psi_q] [M]^{-\frac{1}{2}} , \\ \{\ddot{q}_f\} &= [M]^{-1} \{Q\} , \end{aligned}$$

then the accelerations can be computed as follows

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) .$$

Numerical results

Table: Comparison of computational efficiency of the constraints orthogonalization methods.

Method	CPU Time (s)
Coordinate partitioning	11.45
QR	2.64
PUTD	4.83
SVD	2.73
Zero Eigenvalue	2.72
Schur	2.74
Udwadia-Kalaba	3.17

Conclusions

- 1 A review and comparison of methods for the orthogonalization of constraints has been presented.
- 2 The Schur decomposition has been herein proposed for the numerical computation of the null space of the Jacobian matrix.
- 3 There is a subtle difference between the Zero-Eigenvalue method and Schur decomposition. The similarity transformation may not exist, whereas Schur decomposition can be always obtained for any square matrix.
- 4 The numerical accuracy seems not influenced by the method used for the orthogonalization. However, this may not be necessarily true for large scale problems.
- 5 Computation of rank at each integration step may be avoided if the d.o.f. does not change.
- 6 Iterative algorithms for the algebra decompositions should be used.

Conclusions

- 1 A review and comparison of methods for the orthogonalization of constraints has been presented.
- 2 The Schur decomposition has been herein proposed for the numerical computation of the null space of the Jacobian matrix.
- 3 There is a subtle difference between the Zero-Eigenvalue method and Schur decomposition. The similarity transformation may not exist, whereas Schur decomposition can be always obtained for any square matrix.
- 4 The numerical accuracy seems not influenced by the method used for the orthogonalization. However, this may not be necessarily true for large scale problems.
- 5 Computation of rank at each integration step may be avoided if the d.o.f. does not change.
- 6 Iterative algorithms for the algebra decompositions should be used.

Conclusions

- 1 A review and comparison of methods for the orthogonalization of constraints has been presented.
- 2 The Schur decomposition has been herein proposed for the numerical computation of the null space of the Jacobian matrix.
- 3 There is a subtle difference between the Zero-Eigenvalue method and Schur decomposition. The similarity transformation may not exist, whereas Schur decomposition can be always obtained for any square matrix.
- 4 The numerical accuracy seems not influenced by the method used for the orthogonalization. However, this may not be necessarily true for large scale problems.
- 5 Computation of rank at each integration step may be avoided if the d.o.f. does not change.
- 6 Iterative algorithms for the algebra decompositions should be used.

Conclusions

- 1 A review and comparison of methods for the orthogonalization of constraints has been presented.
- 2 The Schur decomposition has been herein proposed for the numerical computation of the null space of the Jacobian matrix.
- 3 There is a subtle difference between the Zero-Eigenvalue method and Schur decomposition. The similarity transformation may not exist, whereas Schur decomposition can be always obtained for any square matrix.
- 4 The numerical accuracy seems not influenced by the method used for the orthogonalization. However, this may not be necessarily true for large scale problems.
- 5 Computation of rank at each integration step may be avoided if the d.o.f. does not change.
- 6 Iterative algorithms for the algebra decompositions should be used.

Conclusions

- 1 A review and comparison of methods for the orthogonalization of constraints has been presented.
- 2 The Schur decomposition has been herein proposed for the numerical computation of the null space of the Jacobian matrix.
- 3 There is a subtle difference between the Zero-Eigenvalue method and Schur decomposition. The similarity transformation may not exist, whereas Schur decomposition can be always obtained for any square matrix.
- 4 The numerical accuracy seems not influenced by the method used for the orthogonalization. However, this may not be necessarily true for large scale problems.
- 5 **Computation of rank at each integration step may be avoided if the d.o.f. does not change.**
- 6 Iterative algorithms for the algebra decompositions should be used.

Conclusions

- 1 A review and comparison of methods for the orthogonalization of constraints has been presented.
- 2 The Schur decomposition has been herein proposed for the numerical computation of the null space of the Jacobian matrix.
- 3 There is a subtle difference between the Zero-Eigenvalue method and Schur decomposition. The similarity transformation may not exist, whereas Schur decomposition can be always obtained for any square matrix.
- 4 The numerical accuracy seems not influenced by the method used for the orthogonalization. However, this may not be necessarily true for large scale problems.
- 5 Computation of rank at each integration step may be avoided if the d.o.f. does not change.
- 6 **Iterative algorithms for the algebra decompositions should be used.**