

THE UDWADIA-KALABA FORMULATION: A REPORT ON ITS NUMERICAL EFFICIENCY IN MULTIBODY DYNAMICS SIMULATIONS AND ON ITS TEACHING EFFECTIVENESS

D. de Falco^{*}, E. Pennestrì[†], and L. Vita[†]

^{*}Dipartimento di Ingegneria Spaziale e Aeronautica
II Università di Napoli
Real Casa dell'Annunziata, Via Roma,29 81031 Aversa - Italy
e-mail: domenico.defalco@unina2.it

[†] Dipartimento di Ingegneria Meccanica
Università di Roma Tor Vergata
Facoltà Ingegneria, via del Politecnico, 1 00133 Roma, Italy
e-mails: pennestrì@mec.uniroma2.it, web page: www.ingegneriemeccanica.org
vita@ing.uniroma2.it

Keywords: Pseudoinverse matrix computation, Udwadia-Kalaba dynamic formulation, Multi-body dynamics, Least squares.

Abstract. *The formulation of the dynamic equations of motion proposed by Udwadia-Kalaba is discussed from the point of view of teaching effectiveness and numerical efficiency. Since this formulation requires the computation of a pseudoinverse matrix, it is investigated the influence of the method of pseudoinverse computation on the dynamic simulation of an overconstrained linkage.*

1 INTRODUCTION

In a recent series of papers (e.g. [1]) and in a textbook [2] Udwadia and Kalaba, starting from the Gauss Principle of Least Constraint [3, 5, 6], deduced a new formulation of the dynamics equations of motion for a system of constrained particles or rigid bodies.

The main features of this formulation are:

- the equations of motion can be reduced to a system of ordinary differential equations (ODE), even when a redundant set of coordinates is used;
- variations of degrees-of-freedom due to the change of topology or other causes are allowed and do not require special effort in computer programming;
- rheonomic and scleronomic constraints are treated in the same way;
- forward and inverse dynamics problems can be solved within the same tool;
- easy computer implementation, provided that a subroutine for computing the pseudoinverse matrix is available.

The above features make the formulation very attractive. In fact, one of the shortcomings associated with the use of a redundant set of coordinates is the integration of a mixed system of differential-algebraic equations (DAE). These systems are different from ordinary differential equations (ODE) and require specialised numerical methods for their solution. The matrices are of high order, but sparse. The computer programming effort required by redundant coordinates dynamic formulations is relatively low and within the capabilities of the average mechanical engineering student.

A fundamental step of the Udwadia-Kalaba (UK) formulation is the computation of the Moore-Penrose generalized inverse or pseudoinverse matrix.

Arabyan and Wu [7] investigated numerically the advantages of this formulation within the framework of multibody dynamics and compared the number of floating point operations (flops) required for computing the pseudoinverse with Gram-Schmidt (GS) orthogonalization and Singular Value Decomposition (SVD).

However, the authors of the present paper believe that the comparison of the methods for computing the pseudoinverse only on the basis of flops operations required, although important, is somewhat limited. For this reason we decided to compare different methods by observing other parameters such as the reliability, accuracy of results and speed of computation.

Thus, one of the purposes of this paper is to report about numerical tests, for accuracy and computational efficiency, of different pseudoinverse matrix calculation algorithms with explicit reference to multibody dynamics simulations. These tests do not seem to be readily available in literature.

The paper is organized in four parts. The first and second parts are mainly tutorial. In the first part the main steps of the following methods for computing the pseudoinverse matrix are summarized:

- Singular value decomposition (SVD) [22];
- Varga's algorithm [19] with QR decomposition computed by means of the Householder factorization [9];
- Greville's algorithm [20, 21];

- Least squares solution with modified Gram-Schmidt (GS) QR factorization [22];
- Least squares solution with Householder QR factorization. [22]

In the second part the formulation of Udwadia-Kalaba is concisely deduced. The third part describes the test problem and reports the numerical results. Finally, the fourth part of the paper discuss some issues related to the teaching of the UK formulation in a multibody dynamics course.

2 REVIEW OF METHODS FOR COMPUTING THE MOORE-PENROSE PSEUDOINVERSE MATRIX

This section, for completeness, summarizes the main properties of the Moore-Penrose pseudoinverse matrix and the steps of the different algorithms tested during the dynamic simulations.

2.1 Definitions

The main properties of the Moore-Penrose pseudoinverse matrix $[A]^+$ of a matrix $[A]$ are:

- $([A] [A]^+)^T = [A] [A]^+$
- $([A]^+ [A])^T = [A]^+ [A]$
- $[A] [A]^+ [A] = [A]$
- $[A]^+ [A] [A]^+ = [A]^+$

When $[A]$ is a square matrix with full rank, then its pseudoinverse coincide with the inverse.

The Moore-Penrose pseudoinverse matrix is associated with the least squares solution of the linear system of equations

$$[A] \{x\} = \{b\} , \quad (1)$$

where the number m of equations is not equal to the number n of unknowns and $[A]$ does not necessarily has full rank.

In particular, the following cases are distinguished:

Overdetermined system of equations ($m > n$)

By requiring that

$$h \equiv \|[A] \{x\} - \{b\}\|_2^2 \quad (2)$$

is a minimum, one obtains

$$[A]^T [A] \{x\} = [A]^T \{b\} . \quad (3)$$

Therefore, the solution of (1) can be stated as

$$\{x\} = [A]^+ \{b\} , \quad (4)$$

where

$$[A]^+ = \left([A]^T [A] \right)^{-1} [A]^T \quad (5)$$

is the *right pseudoinverse matrix*.

Undetermined system of equations ($m < n$)

The solution is obtained imposing the minimum of the Euclidean norm

$$g \equiv \|x\|_2^2, \quad (6)$$

with $\{x\}$ subjected to (1). Thus, introduced the new objective function,

$$g' \equiv g + \{\lambda\}^T ([A] \{x\} - \{b\}) \quad (7)$$

the solution is achieved solving the system

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{Bmatrix} x \\ \lambda \end{Bmatrix} = \begin{Bmatrix} 0 \\ b \end{Bmatrix}, \quad (8)$$

or

$$\{x\} = [A]^+ \{b\}, \quad (9)$$

where

$$[A]^+ = [A]^T ([A][A]^T)^{-1}, \quad (10)$$

is the *left pseudoinverse matrix*.

2.2 The least squares method

Since there is abundance of software procedures for computing the least squares solution of a system of algebraic equations

$$[A] \{x\} = \{b\}, \quad (11)$$

the computation of the pseudoinverse can be reduced to such solution. These procedures are often based on QR decomposition by means of Householder reflections or GS orthogonalization.

Let

$$\begin{aligned} \{b_1\} &= \{ 1 \ 0 \ 0 \ \dots \ 0 \}^T, \\ \{b_2\} &= \{ 0 \ 1 \ 0 \ \dots \ 0 \}^T, \\ \{b_3\} &= \{ 0 \ 0 \ 1 \ \dots \ 0 \}^T, \\ &\dots\dots\dots \\ \{b_m\} &= \{ 0 \ 0 \ 0 \ \dots \ 1 \}^T. \end{aligned}$$

The procedure differs according to the dimensions of $[A]$.

Case $m > n$

1. Solve m times¹ the system

$$[A]^T [A] \{x\} = [A]^T \{b\}. \quad (12)$$

2. From the m solutions $\{x_1\}, \{x_2\}, \dots, \{x_m\}$ one can form the pseudoinverse matrix

$$[A]^+ = [\{x_1\} \ \{x_2\} \ \{x_3\} \ \dots \ \{x_m\}] \quad (13)$$

¹Computationally, the system is solved only once. The factor matrices used for the first solution are saved and used also for the remaining ones.

Case $n \geq m$

1. Solve m times the system

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{Bmatrix} x \\ \lambda \end{Bmatrix} = \begin{Bmatrix} 0 \\ b \end{Bmatrix} \quad (14)$$

Also in this case the pseudoinverse is given by equation (13).

The square matrices in (12) and (14) are singular or ill conditioned. Thus their solution requires special care. As mentioned, an appropriate way to solve least squares problems is by means of GS orthogonalization or Householder QR factorization.

2.3 Singular Value Decomposition method

1. Apply the Singular Value Decomposition (SVD) to matrix $[A]$

$$[A] = [U] [\Sigma] [V]^T ,$$

where $[U] [U]^T = [I]_{m \times m}$, $[V] [V]^T = [I]_{n \times n}$

$$[\Sigma] = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}$$

is a diagonal matrix.

2. Compute

$$[A]^+ = [V] [\Sigma]^+ [U]^T , \quad (15)$$

where

$$[\Sigma]^+ = \begin{bmatrix} \sigma_1^+ & & 0 \\ & \ddots & \\ 0 & & \sigma_n^+ \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}$$

and

$$\sigma_i^+ = \begin{cases} 0 & (\sigma_i = 0) \\ \sigma_i^{-1} & (\sigma_i \neq 0) \end{cases} \quad (16)$$

for $i = 1, 2, \dots, n$.

2.4 Varga's method

1. Apply the QR factorization to matrix $[A]$. Thus, one obtains:

$$[A] = [U][G] = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} G_1 \\ 0 \end{bmatrix}$$

with $[G_1]$ matrix with full row rank and $[U]^T [U] = [I_m]$.

2. Apply the QR factorization to $[G_1]^T$

$$[G_1]^T = [V]^T \begin{bmatrix} G_2^T \\ 0 \end{bmatrix}, \quad (17)$$

with $[G_2]^T$ invertible matrix and $[V][V]^T = [I_n]$.

3. Since

$$[A] = [U] \begin{bmatrix} G_2 & 0 \\ 0 & 0 \end{bmatrix} [V],$$

the pseudomatrix is readily obtained from

$$[A]^+ = [V]^T \begin{bmatrix} G_2^{-1} & 0 \\ 0 & 0 \end{bmatrix} [U]^T. \quad (18)$$

2.5 Greville's method

1. Decompose the matrix $[A]_{m \times n}$ into row vectors $\{a_i\}$, ($i = 1, 2, \dots, m$)

$$[A] = \begin{bmatrix} a_1^T & a_2^T & \dots & a_m^T \end{bmatrix}$$

2. Let matrix

$$[A_i]_{i \times n} = \begin{bmatrix} A_{i-1} \\ a_i \end{bmatrix} \quad (19)$$

with $[A_1]_{1 \times n} = \{a_1\}_{1 \times n}$.

3. For $i = 2, \dots, m$ compute the matrices $[A_i]^+$ as

$$[A_i]_{n \times i}^+ = \left[[A_{i-1}]^+ - \{b_i\}^T \{d_i\} \left| \{b_i\}^T \right. \right],$$

where

$$\begin{aligned} \{d_i\}_{1 \times (i-1)} &= \{a_i\} [A_{i-1}]^+ \\ \{c_i\}_{1 \times n} &= \{a_i\} - \{d_i\} [A_{i-1}] \\ \{b_i\}_{1 \times n} &= \begin{cases} \frac{\{c_i\}}{\{c_i\}\{c_i\}^T} & (\|c_i\| \neq 0) \\ \frac{\{d_i\}[A_{i-1}]^+}{1 + \{d_i\}\{d_i\}^T} & (\|c_i\| = 0) \end{cases} \\ [A_1]^+ &= \begin{cases} \frac{\{a_1\}^T}{\{a_1\}\{a_1\}^T} & (\|a_1\| \neq 0) \\ \{a_1\}^T & (\|a_1\| = 0) \end{cases} \end{aligned}$$

4. After m repetitions $[A_m]^+$ gives the pseudoinverse $[A]_{n \times m}^+$ of matrix $[A]$.

3 THE FORMULATION OF UDWADIA-KALABA

In this section the formulation of Udwadia-Kalaba is deduced following the approach of Arabyan and Wu [7].

Let us denote with

- $\{F\}$ the vector of external generalized forces;
- $[M]$ the mass matrix;
- $[\Psi_q]$ the Jacobian of constraints equations;
- $\{q\}$ the vector of generalized coordinates;
- $\{\gamma\} = -([\Psi_q]\{\dot{q}\})_q\{\dot{q}\} - 2[\Psi_{qt}]\{\dot{q}\} - \{\Psi_{tt}\}$;
- the upperscript $+$ the operation of pseudoinverse of a matrix.

When a redundant set of coordinates is used, the following system of differential-algebraic system of equations (DAE) is obtained

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} F \\ \gamma \end{Bmatrix} \quad (20)$$

The square matrix at the left-hand can be inverted by partitioning [23]

$$\begin{bmatrix} M & \Psi_q^T \\ \Psi_q & 0 \end{bmatrix}^{-1} = \begin{bmatrix} M^{-1} - M^{-1}\Psi_q^T(\Psi_q M^{-1}\Psi_q^T)^{-1}\Psi_q M^{-1} & M^{-1}\Psi_q^T(\Psi_q M^{-1}\Psi_q^T)^{-1} \\ (\Psi_q M^{-1}\Psi_q^T)^{-1}\Psi_q M^{-1} & -(\Psi_q M^{-1}\Psi_q^T)^{-1} \end{bmatrix}. \quad (21)$$

Introduced the vector

$$\{\ddot{q}_f\} = [M]^{-1}\{F\}, \quad (22)$$

representing the acceleration vector of the unconstrained system, the solution of (20) follows

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-1}[\Psi_q]^T \left([\Psi_q][M]^{-1}[\Psi_q]^T\right)^{-1} (\{\gamma\} - [\Psi_q]\{\ddot{q}_f\}) \quad (23)$$

and

$$\{\lambda\} = \left([\Psi_q][M]^{-1}[\Psi_q]^T\right)^{-1} (\{\gamma\} - [\Psi_q]\{\ddot{q}_f\}). \quad (24)$$

If we let

$$[M]^{-1} = [M]^{-\frac{1}{2}}[M]^{-\frac{1}{2}}, \quad (25)$$

$$[D] = [\Psi_q][M]^{-\frac{1}{2}}, \quad (26)$$

then equation (23) can be rewritten in the form

$$\begin{aligned} \{\ddot{q}\} &= \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} \left([M]^{-\frac{1}{2}}[\Psi_q]^T\right) \left([\Psi_q][M]^{-\frac{1}{2}}[M]^{-\frac{1}{2}}[\Psi_q]^T\right)^{-1} (\{\gamma\} - [\Psi_q]\{\ddot{q}_f\}), \\ &= \{\ddot{q}_f\} + [M]^{-\frac{1}{2}}[D]^T \left([D][D]^T\right)^{-1} (\{\gamma\} - [\Psi_q]\{\ddot{q}_f\}) \end{aligned} \quad (27)$$

Taken into account the definition of right pseudoinverse matrix formerly stated, we can let

$$[D]^+ = [D]^T \left([D] [D]^T \right)^{-1} , \quad (28)$$

and the previous equation can be concisely expressed as follows [7]:

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ (\{\gamma\} - [\Psi_q] \{\ddot{q}_f\}) . \quad (29)$$

This is the dynamic formulation originally proposed by Udwadia and Kalaba [2] starting from Gauss Principle of Least Constraint.

If the Baumgarte stabilization is introduced, then (29) is modified as follows

$$\{\ddot{q}\} = \{\ddot{q}_f\} + [M]^{-\frac{1}{2}} [D]^+ \left(\{\gamma\} - 2\alpha \{\dot{\Psi}\} - \beta^2 \{\Psi\} - [\Psi_q] \{\ddot{q}_f\} \right) , \quad (30)$$

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

4 THE PROBLEM OF RANK COMPUTATION

A serious numerical problem that arises in the computation of the pseudoinverse is the accurate computation of the matrix rank. At the practical level, the user must establish a numerical threshold under which the numerical values are all considered to be zero. One may set this value equal to the hardware precision. However, since the inverse of these very small values is required by the algorithms, this inflates a perturbation in the numerical solution. Thus, a truncation or other means of regularization are compulsory to obtain acceptable results.

In all the examples discussed in the paper the numerical threshold necessary for the computation of the pseudoinverse is set equal to the error tolerance parameter TOL required by the following IMSL numerical integration subroutines

- DIVPAG, which solve an ODE initial-value problem using the Adams-Moulton or Gear's method.
- DIVPRK, which solve an ODE initial-value problem using the Runge-Kutta-Verner fifth and sixth order method.

During the integration process, both subroutines try to control the norm of the local error such that the global error is proportional to TOL.

It should be acknowledged that the adoption of different criterion may alter the results herein presented.

5 SUMMARY OF FORTRAN SUBROUTINES USED

Whenever possible the computation of the pseudoinverse has been executed by means of standard math libraries, such as IMSL. For instance, the Householder QR factorization has been carried out through the combined use of IMSL library subroutines DLQRRR and DLQERR. For other relevant matrix computations, the Fortran code available on the web site www.netlib.org, such as LAPACK, has been used.

All the Fortran subroutines described in this section and implemented by the authors are available upon request.

- **Pseudoinverse by means of the SVD method** The IMSL library [9] supplies the ready-to-use DLSGRR subroutine.

- **Pseudoinverse by means of the Varga's method** The two Householder QR factorizations required are executed by means of the combined use of the IMSL library subroutines DLQRRR and DLQERR. For the inversion of the triangular matrix G_2 the IMSL routine DLINRT is used. Computation of transpose and matrix multiplications are done with DTRNRR and DMRRRR, respectively.
- **Pseudoinverse by means of Greville's method** A Fortran subroutine has been implemented with matrix operations executed with LAPACK subroutines [4].
- **Pseudoinverse by means of Least squares solution and Householder QR factorization** For the least squares solution of a linear system of equations, it has been used the IMSL subroutines DLQRRR and DLQRSL. DLQRRR provides information on the QR factorization, whereas DLQRSL compute the least squares solution.
- **Pseudoinverse by means of Least squares solution and modified Gram-Schmidt QR factorization** For the least squares solution of a linear system of equations, it has been used the code written by R.H. Wapler [10].

6 NUMERICAL EXAMPLE

The example chosen to test the different algorithms of pseudoinverse computation is the overconstrained mechanism shown in Figure 1. The length, mass and moment of inertia of each of the three parallel links are $L=1$, $m=1$ and $I=0.1$, respectively. The length, mass and moment of inertia of the coupler are $L_2=2$, $m_2=2$, $I_2=0.2$. The kinematic constraints were modelled by means of the method described in Nikravesh textbook [18]. The position of each body is described by three coordinates, thus there are $n=12$ coordinates and $k=12$ algebraic constraints due to the presence of six revolute pairs. Due to the particular link dimension, Jacobian rank is $r = 11$. Thus the degree-of-freedom is $F = n - r = 1$. Gravity force is included and dissipative effects neglected.

The results were monitored for $t_f=20$ s of simulation time. Sometimes the execution was halted because of precision loss or other causes.

The results of the numerical tests are summarized in Tables 2-7.

The simulations were carried out with a PC equipped with AMD Athlon 2600+, 512 MB RAM, running under Microsoft XP Professional. The CPU time reported is the average of the CPU times measured executing four times the same simulation.

7 TEACHING EXPERIENCES

In this section are discussed some teaching issues related with the actual implementation at the classroom level of the Udwadia-Kalaba formulation. The current formulations in the syllabus of the course Computational Kinematics and Dynamics (*Cinematica e Dinamica Computazionale*), offered at University of Roma Tor Vergata, are:

- Coordinate partitioning (CP) [11, 17, 18, 12];
- Orthogonalization of constraints by means of QR [13] and SVD [14, 15];
- Udwadia-Kalaba (UK) [2].

At textbook level, a thoughtful discussion of the differences between coordinate partitioning method and formulations based on the orthogonalization of constraints is presented by Bayo and de Jalon [16].

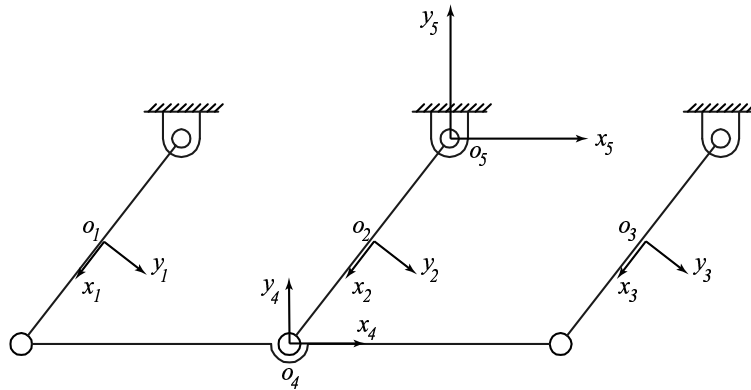


Figure 1: Five bar parallelogram linkage

Table 1: Comparison of dynamic formulations on the basis of students opinion

	Dynamic formulation			
	CP	SVD	QR	UK
Linear algebra background required	Average	Extra	Extra	Extra
Algebra manipulation effort	Medium	Medium	Medium	High
Computer programming effort	Medium	Medium	Medium	Low

From the didactic point of view, the coordinate partitioning method is very appealing. It is very intuitive and the students grasp it very quickly. In fact, from their very first year of engineering curriculum they became acquainted both with theory and computer programming of the Gaussian elimination method. This method can be applied to partition the set of coordinates.

The comprehension of the theoretical bases of the remaining mentioned methods (*i.e.* SVD, QR and Moore-Penrose inverse) requires an extra knowledge of linear algebra.

The amount algebraic manipulation work required to obtain the dynamic formulation based on coordinate partitioning is also lighter when compared to the others based on the orthogonalization of constraints.

In particular the algebraic manipulation necessary to deduce the equations of dynamics in the form proposed by Udwadia-Kalaba is not straightforward or obvious. The approach presented by Arabyan and Wu [7] (summarized in a previous section) because of its teaching effectiveness is recommended.

On the other side, when it comes to computer programming, the Udwadia-Kalaba formulation appears to be preferred by the students once a reliable subroutine for computing the pseudoinverse matrix is made available. The Table 1 summarizes the opinions expressed by students when comparing the different dynamic formulations.

Based on the Udwadia-Kalaba formulation a simple 2D multibody dynamics simulation program has been developed. The core of the software is written in Fortran. Preprocessing module has been developed in GTK+ embedded in the Ch scripting language [24]. Mechanism animation is obtained making use of the useful tools available in the Ch Mechanism Toolkit [25].

8 CONCLUSIONS

Computing pseudoinverse by means of the Gram-Schmidt decomposition is usually less expensive than other methods based on the use of the SVD decompositions, Greville algorithm, or Householder reflections. However, these last methods require about twice as much arithmetic, but are more reliable and accurate near the limits of residual reduction. The loss of accuracy usually connected with the Gram-Schmidt method of computing the pseudoinverse requires the integration subroutine a considerable higher number of computational steps in order to maintain the error lower than a fixed limit. The other two methods discussed are very accurate and require less function evaluation calls during numerical integration.

Table 2: $\text{TOL}=10^{-10}$. Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. No Baumgarte stabilization applied.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	$3.7 \cdot 10^{-7}$	-	-	-
Varga	Fails ²	$9.9 \cdot 10^{-2}$	-	-	-
Greville	Correct up to 17.5 s	$4.5 \cdot 10^{-5}$	2,961,955	2,633,811	0.93
LS-Householder	Correct up to 16.5 s	$2.3 \cdot 10^{-4}$	6,131,159	4,177,395	11.57
LS-Modified GS	Correct up to 17.0 s	$1.1 \cdot 10^{-4}$	3,749,766	3,015,354	36.85

¹ After 9.1 s of correct simulation, the integration subroutine is halted because of repeated error-test failures (see [9], p. 668).

² After 8.0 s of correct simulation, the integration subroutine is halted after failing to pass the error-test failures (see [9], p. 668).

Table 3: $\text{TOL}=10^{-8}$. Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. No Baumgarte stabilization applied.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	$3.4 \cdot 10^{-6}$	-	-	-
Varga	Fails ²	$9.9 \cdot 10^{-2}$	-	-	-
Greville	Correct up to 17.5 s	$1.86 \cdot 10^{-4}$	1,648,055	1,422,689	0.53
LS-Householder	Correct up to 18.0 s	$7.9 \cdot 10^{-4}$	1,641,751	1,422,297	1.28
LS-Modified GS	Correct up to 18.0 s	$5.2 \cdot 10^{-4}$	1,644,341	1,424,864	6.03

¹ After 12.07 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [9], p. 668).

² After 8.06 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.

Table 4: $\text{TOL}=10^{-6}$. Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. No Baumgarte stabilization applied.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Correct up to 17.5 s	$4.1 \cdot 10^{-4}$	1,033,381	847,294	0.28
Varga	Fails ¹	$9.9 \cdot 10^{-2}$	-	-	-
Greville	Correct up to 19.0 s	$9.7 \cdot 10^{-4}$	1,017,558	1,422,689	0.37
LS-Householder	Correct	$7.9 \cdot 10^{-4}$	1,641,751	1,422,297	0.76
LS-Modified GS	Correct	$6.9 \cdot 10^{-4}$	1,013,795	839,020	3.40

¹ After 7.3 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.

Table 5: $\text{TOL}=10^{-10}$. Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. Baumgarte stabilization applied, $\alpha = 1000$, $\beta = 100$.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	-	-	-	-
Varga	Correct	$4.5 \cdot 10^{-5}$	78,491,398	47,074,599	11.4
Greville	Correct	$2.1 \cdot 10^{-7}$	74,043,990	44,053,874	18.3
LS-Householder	Correct	$3.3 \cdot 10^{-7}$	73,707,588	43,948,609	51.5
LS-Modified GS	Correct s	$4.1 \cdot 10^{-7}$	73,910,268	44,515,071	241.0

¹ After 1.17 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [9], p. 668).

Table 6: $\text{TOL}=10^{-8}$. Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. Baumgarte stabilization applied, $\alpha = 1000$, $\beta = 100$.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	-	-	-	-
Varga	Fails ²	-	-	-	-
Greville	Correct	$4.90 \cdot 10^{-7}$	80,109,155	55,569,995	19.4
LS-Householder	Correct	$3.5 \cdot 10^{-7}$	80,197,458	54,468,468	55.0
LS-Modified GS	Correct	$4.9 \cdot 10^{-7}$	80,112,706	54,604,452	290

¹ After 0.17 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [9], p. 668).

² After 14.97 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.

Table 7: TOL= 10^{-6} . Integration method: Adams-Moulton. Initial time step, $\Delta t = 0.66 \cdot 10^{-2}$ s. Baumgarte stabilization applied, $\alpha = 1000$, $\beta = 100$.

Method	Result	$\sum_i \Psi_i^2$	NFUNC	NSTEPS	CPU
SVD	Fails ¹	-	-	-	-
Varga	Fails ²	-	-	-	-
Greville	Correct	$1.9 \cdot 10^{-6}$	80,483,163	57,743,095	19.5
LS-Householder	Correct	$1.14 \cdot 10^{-6}$	80,689,071	58,099,798	55
LS-Modified GS	Correct	$1.06 \cdot 10^{-6}$	80,636,983	58,010,479	291

¹ After 0.46 s of correct simulation, the integration subroutine is halted because a fatal error 1 from DIVPAG (see [9], p. 668).

² After 9.7 s of correct simulation, the integration subroutine is halted because the precision loss of constraints equations.

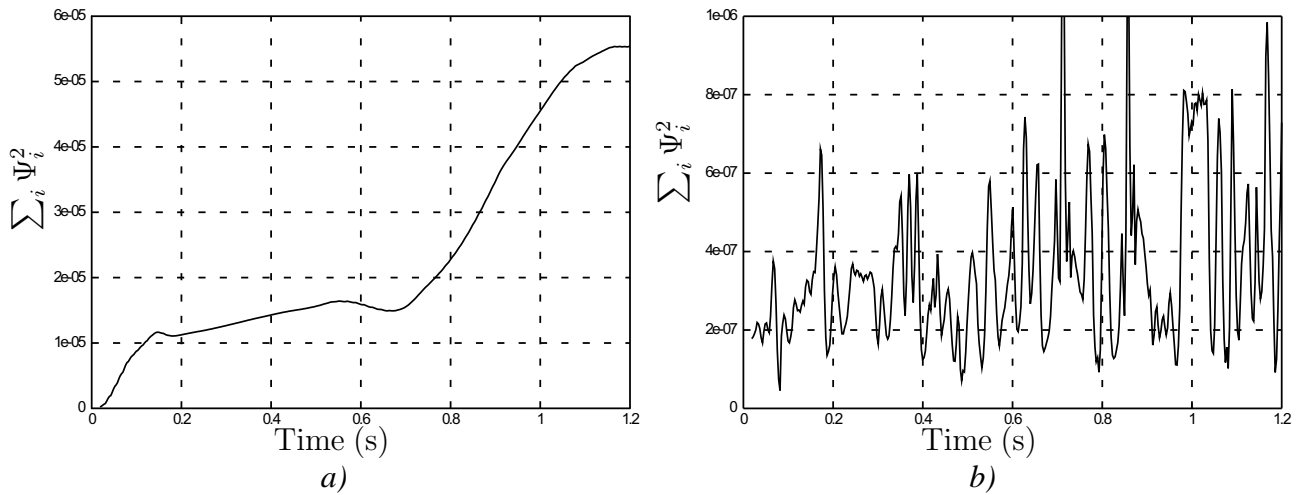


Figure 2: Violation of position constraints: a) Without stabilization b) With Baumgarte stabilization

REFERENCES

- [1] Udwadia, F.E., Kalaba, R.E., A new perspective on constrained motion, *Proceedings of the Royal Society of London*, A439, 1992, pp.407-410.
- [2] Udwadia, F. and Kalaba, R. *Analytical Dynamics a New Approach*. Cambridge University Press, Cambridge, 1996.
- [3] Gauss, C.F., Über ein neues allgemeines Grundgesetz der Mechanik, *Journal für Reine und Angewandte Mathematik*, vol.4, 1829, pp.232-235.
- [4] Anderson, E. *et al.* *LAPACK - Users's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, 1995.

-
- [5] Lanczos, C., *The Variational Principle of Mechanics*, Dover Publications, Inc., Fourth Edition, 1970, pp.106-110.
- [6] Chetaev, N., *Mécanique Rationnelle*, Éditions Mir, Moscou, 1993, pp.251-253.
- [7] Ara Arabyan, Fei Wu, An Improved Formulation for Constrained Mechanical Systems, *Multibody System Dynamics*, vol.2, 1998, pp.49-69.
- [8] Itiki, C., Neto, J.J., Complete automation of the generalized inverse method for constrained mechanical systems of particles, *Applied Mathematics and Computation*, 2004, pp.561-580.
- [9] IMSL Math library, vols.I, II, 1997.
- [10] Wampler, R.H., Modified Gram-Schmidt with iterative refinement, *ACM TOMS* 5, 1979, pp.494-499. Fortran code available at www.netlib.org
- [11] Wehage, R. and Haug, E. Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems. *ASME Journal of Mechanical Design*, 134:247–255, 1982.
- [12] Haug, E. and Yen, J. Generalized Coordinate Partitioning Methods for Numerical Integration of Differential-Algebraic Equations of Dynamics. In Haug, E. and Deyo, R., editors, *Real-Time Integration Methods for Mechanical System Simulation*, volume 69 of *NATO ASI Series*, pages 97–114. Springer-Verlag, 1991.
- [13] Kim, S. and Vanderploeg, M. QR Decomposition for State Space Representation of Constrained Mechanical Dynamic Systems. *ASME Journal of Mechanisms, Transmission, and Automation in Design*, 108:176–182, 1986.
- [14] Mani, N., Haug, E., and Atkinson, K. Singular Value Decomposition for Analysis of Mechanical System Dynamics. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 107:82–87, 1985.
- [15] Singh, R. and Likins, P. Singular Value Decomposition for Constrained Mechanical Systems. *ASME Journal of Applied Mechanics*, 52:943–948, 1985.
- [16] de Jalon, J. and Bayo, E., *Kinematic and Dynamic Simulation of Multibody Systems - The Real-Time Challenge*, Springer Verlag, New York (1994).
- [17] Haug, E., *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon (1989).
- [18] Nikravesh, P., *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall (1988).
- [19] Varga, A., On computing generalized inverse systems using matrix pencil methods, *Int. J. of Applied Mathematics and Computer Science*, vol.11, 2001, n.5.
- [20] Greville, T.N.E., Some Applications of the pseudoinverse matrix, *SIAM Rev.*, 1960, pp.15-22.

- [21] Udawadia, F.E., Kalaba, R.E., A Unified Approach for the Recursive Determination of Generalized Inverses, *Computers and Mathematics with Applications*, vol.37, 1999, pp.125-130.
- [22] Golub, G.H., Van Loan, C.F., *Matrix Computations*, 3rd Edition, The John Hopkins University Press, 1996.
- [23] Stephenson, G., *An Introduction to Matrices Sets and Groups for Science Students*, Dover Publications Inc., 1965, pp.48-51.
- [24] The Ch Language Environment, User's Guide, v.5.0, www.softintegration.com.
- [25] Ch Mechanism Toolkit, User's Guide, v.1.5, www.softintegration.com.