A Rosenbrock-Nystrom State Space Implicit Approach for the Dynamic Analysis of Mechanical Systems: II – The Method and Numerical Examples *

Dan Negruţ†
MSCsoftware,
2300 Traverwood Drv., Ann Arbor, MI 48105.
Email: Dan.Negrut@mscsoftware.com.

Adrian Sandu
Department of Computer Science,
Michigan Technological University, Houghton, MI 49931.
Email: asandu@mtu.edu.

Edward J. Haug
Department of Mechanical Engineering,
The University of Iowa, Iowa City, IA 52242.
Email: Edward.Haug@uiowa.edu.

Florian A. Potra
Department of Mathematics and Statistics,
University of Maryland, Baltimore County, Baltimore, MD 21250.
Email: potra@math.umbc.edu.

Corina Sandu
Department of Mechanical Engineering–Engineering Mechanics,
Michigan Technological University, Houghton, MI 49931.
Email: csandu@mtu.edu.

June 2, 2003

*This research was supported in part by the US Army Tank-Automotive Research, Development, and Engineering Center (DoD contract number DAAE07-94-R094), a multi-university Center led by the University of Michigan
†Corresponding Author.

List of symbols

$\mathbf{q}$	vector of generalized coordinates
$\mathbf{\Phi}$	array of position kinematic constraints
$\mathbf{M}$	system mass matrix
λ	vector of Lagrange multipliers
$\mathbf{Q}^A$	generalized external forces
τ	kinematic acceleration equation right hand side
$ndof$	number of degrees of freedom in the mechanical system
y	generic variable used in the definition of the Initial Value Problem
y_n	solution of Rosenbrock method at t_n
$\hat{y}_n$	numerical solution of the embedded method at t_n
γ	diagonal element of the Rosenbrock formula
$(\alpha)_{ij}, (\gamma)_{ij}, (\delta)_{ij}, (\theta)_{ij}$	Rosenbrock-Nystrom method coefficients
$(a)_{ij}, (c)_{ij}, (b)_i, (\hat{b})_i, (m)_i, (\hat{m})_i$	Rosenbrock-Nystrom method coefficients
$(a)_i, (\gamma)_i, (\beta)_{ij}, (\beta')_i, (\mu)_i, (\hat{\mu})_i$	coefficients used in the derivation of the Rosenbrock-Nystrom formula
h	integration step-size
k_i	stage vector of Rosenbrock method
ℓ_i	stage vector of Rosenbrock-Nystrom method
J	Rosenbrock Jacobian w.r.t. y
J_1	Rosenbrock-Nystrom Jacobian w.r.t. y
J_2	Rosenbrock-Nystrom Jacobian w.r.t. y'
sc_i	integration composite tolerance for variable i
$Atol_i$	user prescribed absolute tolerance for variable i
$Rtol_i$	user prescribed relative tolerance for variable i
err	integration local error
p	integration order for the Rosenbrock-Nystrom method
$\hat{p}$	integration order for the embedded method
$fac, facmin, facmax$	step-size selection safety factors
$\mathtt{m}_1, \mathtt{m}_2$	mass of pendulums
$\mathtt{L}_1, \mathtt{L}_2$	length of pendulums
$\mathtt{k}_1, \mathtt{k}_2$	spring stiffnesses
$\mathtt{c}_1, \mathtt{c}_2$	damping coefficients
$x_1, y_1, \theta_1, \dot{x}_1, \dot{y}_1, \dot{\theta}_1$	initial conditions for first pendulum
$x_2, y_2, \theta_2, \dot{x}_2, \dot{y}_2, \dot{\theta}_2$	initial conditions for second pendulum
α_1^0, α_2^0	zero tension angles for the rotational springs
n	number of integration steps
N	number of integration steps taken during reference simulation
e	generic variable considered for error analysis
E	values obtained for the generic variable during the reference simulation
$\mathbf{r}(i)$	integration grid points
Δ_i	error at time step i
$\Delta^{(k)}, \overline{\Delta}^{(k)}$	maximum and average trajectory errors
E^*	value of E at the time step where the integration error is maximum
$RelErr$	integration relative error

Abstract

When performing dynamic analysis of a constrained mechanical system, a set of index 3 Differential-Algebraic Equations (DAE) describes the time evolution of the system [3, 2]. In the companion paper [4] a state-space based method for the numerical solution of the resulting DAE is developed. The numerical method uses a linearly-implicit time stepping formula of Rosenbrock type, which is suitable for medium accuracy integration of stiff systems. This paper discusses choices of method coefficients and presents numerical results. For stiff mechanical systems, the proposed algorithm is shown to significantly reduce simulation times when compared to state of the art existent algorithms. The better efficiency is due to the use of an L-stable integrator [2], and a rigorous and general approach to providing analytical derivatives required by it.

Keywords: Multibody dynamics, differential-algebraic equations, state space form, Rosenbrock methods.

1 Introduction

For the dynamic analysis of a mechanical system, this paper presents a method that uses a state-space implicit Rosenbrock-type integrator. The generalized coordinates $\mathbf{q}$ considered are Cartesian coordinates for position, and Euler parameters for orientation of body centroidal reference frames. Without loss of generality and in order to simplify the presentation, an assumption is made that the constraints are holonomic and scleronomic. The kinematic constraints are then formulated as algebraic expressions involving generalized coordinates,

$$\mathbf{\Phi}(\mathbf{q}) = \begin{bmatrix} \Phi_1(\mathbf{q}) & \dots & \Phi_m(\mathbf{q}) \end{bmatrix}^{\mathbf{T}} = \mathbf{0} \tag{1a}$$

where m is the total number of independent constraint equations that must be satisfied by the generalized coordinates throughout the simulation. Differentiating Eq.(1a) with respect to time leads to the velocity kinematic constraint equation

$$\mathbf{\Phi}_{\mathbf{q}}(\mathbf{q})\,\dot{\mathbf{q}} = 0\ , \tag{1b}$$

where the over dot denotes differentiation with respect to time and the subscript denotes partial differentiation. The kinematic acceleration equation is obtained by taking the time derivative of the velocity constraint equations to obtain

$$\mathbf{\Phi_q}(\mathbf{q})\,\ddot{\mathbf{q}} = \tau(\mathbf{q}, \dot{\mathbf{q}}) \ , \tag{1c}$$

The time evolution of the system is governed by the Lagrange multiplier form of the constrained equations of motion [3],

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{\Phi_q}(\mathbf{q})^{\mathbf{T}}\lambda = \mathbf{Q^A}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{t}) \tag{1d}$$

Equations (1a)–(1d) comprise a system of differential-algebraic equations (DAE). The companion paper [4] introduced a method based on the partitioning of the coordinates in dependent and independent; the integration of the resulting state-space ordinary differential equation was done using a Rosenbrock-Nystrom linearly implicit method. Rosenbrock methods are generally efficient for medium accuracy simulations. They do not require an iteration procedure and have optimal linear stability properties for stiff systems.

This paper introduces an actual Rosenbrock-Nystrom algorithm based on a fourth order L-stable Rosenbrock method, and discusses a second order method that can accommodate inexact Jacobians. Numerical experiments for two problems indicate that the Rosenbrock-Nystrom algorithm is reliable and efficient for medium accuracy integration of mechanical systems that lead to stiff state-space ODE (SSODE).

2 The Proposed Algorithm

For the Initial Value Problem $y' = f(t, y)$, an s-stage Rosenbrock method is defined as [2]

$$y_{n+1} = y_n + \textstyle\sum_{i=1}^{s} b_i k_i \ , \tag{2a}$$

$$k_i = hf\left(t_n + \alpha_i h, y_n + \textstyle\sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + \gamma_i h^2 \frac{\partial f}{\partial t}\,(t_n, y_n) + hJ \textstyle\sum_{j=1}^{i} \gamma_{ij} k_j \ , \tag{2b}$$

where the coefficients α, γ and b are chosen to obtain the desired accuracy and stability properties.

For the purpose of error control in the generic Rosenbrock method a second approximation of the solution at the current time step is used to produce an estimate of the local error. This second approximation $\hat{y}_{n+1}$ is usually of lower order and it uses the same stage values k_i with a different set of coefficients $\hat{b}_i$,

$$\hat{y}_{n+1} = y_n + \sum_{i=1}^{s} \hat{b}_i k_i \tag{3}$$

The approximation $|y_{n+1} - \hat{y}_{n+1}|$ of the local error depends on the size of the integration step-size, and the latter is increased or decreased to keep the local error below a user prescribed absolute and/or relative tolerance. In the multidimensional case as for the solution of the SSODE, $\mathbf{y} \in \mathbf{R}^{ndof}$, and at time step $n+1$ the goal is to keep the error in component i smaller than a composite error tolerance sc_i

$$|y^i_{n+1} - \hat{y}^i_{n+1}| < sc_i, \quad sc_i = Atol_i + \max(|y^i_n|, |y^i_{n+1}|) \cdot Rtol_i \tag{4}$$

where $Atol_i$ and $Rtol_i$ are the user prescribed absolute and relative integration tolerances for component i, $1 \leq i \leq ndof$. The value

$$err = \left(\frac{1}{ndof} \sum_{i=1}^{ndof} \frac{(y^i_{n+1} - \hat{y}^i_{n+1})^2}{sc_i^2} \right)^{1/2} \tag{5}$$

is considered as a measure of local error. If the order of the proper and embedded formulas used is p and $\hat{p}$ respectively, asymptotically $err \approx Ch^{q+1}$, where C is a constant depending on the choice of formulas and $q = \min(p, \hat{p})$. Optimally, $err = 1$ and therefore $1 \approx Ch_{opt}^{q+1}$. The optimal step-size is computed then as

$$h_{opt} = h \left(\frac{1}{err} \right)^{\frac{1}{q+1}} \tag{6}$$

A safety factor fac multiplies h_{opt} to decrease the chance of a costly rejected step-size, which happens whenever $err > 1$. Further, the step-size is not allowed to increase or decrease too fast. This is achieved by two control parameters $facmin$ and $facmax$,

$$h_{new} = h \cdot \min \left(facmax, \max \left(facmin, fac \cdot (1/err)^{1/(q+1)} \right) \right) \tag{7}$$

For most engineering applications, efficient simulation requires expeditious low to medium accuracy methods with very good stability properties. Integration formulas with few function and Jacobian evaluations are favored, since these operations for mechanical system simulation are typically costly. Based on these considerations, the integrator of choice is a 4 stages L-stable order 4 Rosenbrock-Nystrom method, provided with order 3 embedded formula for step-size control. The L-stability is a desirable attribute that allows for the integration of very stiff problems, which translates in efficient simulation of models with bushing elements and flexible components. Following an idea in [2], the number of function evaluations for the 4 stage method is kept to 3; i.e., one function evaluation is saved. This makes the proposed Rosenbrock-Nystrom method competitive with the trapezoidal method, whenever the latter requires 3 or more iterations for convergence. However, the trapezoidal method is of order 2 and only weakly A-stable.

The notation introduced in [4] is going to be used in the derivation of the method coefficients. Thus, $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$, $\beta'_i = \sum_{j=1}^{i-1} \beta_{ij}$, $\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}$, and $\gamma_i = \sum_{j=1}^{i} \gamma_{ij}$. For reasons of computational efficiency the coefficients γ_{ii} are identical for all stages; i.e., $\gamma_{ii} = \gamma$ for all $i = 1, \ldots, s$. Note that formally $\alpha_{ii} = 0,\ 1 \leq i \leq s$.

With this, the defining coefficients α_{ij}, γ_{ij}, and b_i of an order 4 Rosenbrock method of Eqs.(2a– 2b) are subject to the following order conditions [2]:

$$b_1 + b_2 + b_3 + b_4 = 1 \tag{8a}$$

$$b_2\beta'_2 + b_3\beta'_3 + b_4\beta'_4 = 1/2 - \gamma \tag{8b}$$

$$b_2\alpha_2^2 + b_3\alpha_3^2 + b_4\alpha_4^2 = 1/3 \tag{8c}$$

$$b_3\beta_{32}\beta'_2 + b_4\left(\beta_{42}\beta'_2 + \beta_{43}\beta'_3\right) = 1/6 - \gamma + \gamma^2 \tag{8d}$$

$$b_2\alpha_2^3 + b_3\alpha_3^3 + b_4\alpha_4^3 = 1/4 \tag{8e}$$

$$b3\alpha_3\alpha_{32}\beta'_2 + b_4\alpha_4\left(\alpha_{42}\beta'_2 + \alpha_{43}\beta'_3\right) = 1/8 - \gamma/3 \tag{8f}$$

$$b_3\beta_{32}\alpha_2^2 + b_4\left(\beta_{42}\alpha_2^2 + \beta_{43}\alpha_3^2\right) = 1/12 - \gamma/3 \tag{8g}$$

$$b_4\beta 43\beta_{32}\beta'_2 = 1/24 - \gamma/2 + 1.5\gamma^2 - \gamma^3 \tag{8h}$$

For the purpose of automatic step-size control, the stage values k_i are reused to provide an embedded formula of order 3 of the form $\hat{y}_1 = y_0 + \sum_{i=1}^{s} \hat{b}_i k_i$. The order conditions for the order 3 algorithm are as indicated as Eqs.(8a– 8d), and they lead to the system

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & \beta'_2 & \beta'_3 & \beta'_4 \\ 0 & \alpha_2^2 & \alpha_3^2 & \alpha_4^2 \\ 0 & 0 & \beta_{32}\beta'_2 & \beta_{42}\beta'_2 + \beta_{43}\beta'_3 \end{bmatrix} \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{b}_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 - \gamma \\ 1/3 \\ 1/6 - \gamma + \gamma^2 \end{bmatrix} \tag{9}$$

If the coefficient matrix in Eq.(9) is non-singular, uniqueness of the solution of this linear system implies $b_1 = \hat{b}_i$. To prevent this, one additional condition is considered to obtain a distinct order 3 embedded formula. It requires the coefficient matrix in Eq.(9) to be singular, which results in

$$\beta_{32}\beta'_2(\beta'_2\alpha_4^2 - \beta'_4\alpha_2^2) = (\beta'_2\alpha_3^2 - \beta'_3\alpha_2^2)(\beta_{42}\beta'_2 + \beta_{43}\beta'_3) \tag{10}$$

The number of coefficients that must be determined is 17; the diagonal coefficient γ, six coefficients γ_{ij}, six coefficients α_{ij}, and four weights b_i. The number of conditions that these coefficients have to satisfy is nine. There are eight degrees of freedom in the choice of coefficients and some of these are used to construct a method with one less function evaluation. Thus, if

$$\alpha_{41} = \alpha_{31} \ , \quad \alpha_{42} = \alpha_{32} \ , \quad \alpha_{43} = 0 \ , \tag{11}$$

stage 4 of the algorithm saves one function evaluation. Finally, the free parameters can be determined such that several order 5 conditions of the otherwise order 4 formula are satisfied. When the conditions of Eq.(11) hold, one of the nine order 5 conditions associated with a Rosenbrock type formula leads to

$$\alpha_3 = \frac{1/5 - \alpha_2/4}{1/4 - \alpha_2/3} \tag{12}$$

A second order 5 condition is satisfied by imposing the condition

$$b_4\beta_{43}\alpha_3^2(\alpha_3 - \alpha_2) = 1/20 - \gamma/4 - \alpha_2\,(1/12 - \gamma/3) \tag{13}$$

Next, two conditions are chosen as

$$b_3 = 0 \ , \quad \alpha_2 = 2\gamma \ , \tag{14}$$

to make the task of finding the defining coefficients α_{ij}, γ_{ij}, and b_i more tractable. Finally, the last condition regards the choice of the diagonal element γ. The value of this parameter determines the stability properties of the Rosenbrock method. In this context, the diagonal entry of the Rosenbrock formula is suggested in [2] as $\gamma = 0.57281606$, which is adopted for the proposed algorithm. With this, there is a set of 17 equations, some of them non-linear, in 17 unknowns. The solution of this system is provided in Table 1, along with the coefficients $\hat{b}_i$ of the order 3 embedded formula.

Once the coefficients of the underlying Rosenbrock formula are available, the coefficients of the Rosenbrock-Nystrom formula defined in the companion paper [4] are easily computed. The full set of coefficients for the order 4, L-stable formula is provided in Table 2.

It should be recalled that any Rosenbrock type formula requires an exact Jacobian for the numerical solution to maintain its stability and accuracy properties. Sometimes this might be a very challenging requirement. Consider for example the situation when complex tire models are present in a model, or for

$\gamma = 0.57281606$					
α_{21}	=	1.14563212	γ_{21}	=	2.341993127112013949170520
α_{31}	=	0.52092078913062902932851 6	γ_{31}	=	-0.027333746543489836196505
α_{32}	=	0.13429418684250480014923 2	γ_{32}	=	0.213811650836699689867472
α_{41}	=	0.52092078913062902932851 6	γ_{41}	=	-0.259083837785510222112641
α_{42}	=	0.13429418684250480014923 2	γ_{42}	=	-0.190595807732311751616358
α_{43}	=	0.0	γ_{43}	=	-0.228031035973133829477744
b_1	=	0.32453470789173451347419 6	$\hat{b}_1$	=	0.52092078913062902932851 6
b_2	=	0.04908654478752330868463 3	$\hat{b}_2$	=	0.14454971466536459958468 1
b_3	=	0.0	$\hat{b}_3$	=	0.12455968641470204977489 7
b_4	=	0.62637874732074217784117 1	$\hat{b}_4$	=	0.20996980978930432131190 6

Table 1: Coefficients for the Rosenbrock method.

a general purpose solver the case when user defined external routines are employed for the computation of active forces such as aerodynamic forces. Verwer et al. 1997, proposed a second order W-method [1], which is a Rosenbrock type method in the sense that it does not necessitate the solution of a non-linear system, but which does not require an exact Jacobian. The defining coefficients for this method are provided in Table 3, and within the proposed implicit SSODE integration framework they can be used for the numerical solution of the index 3 DAE of multibody dynamics.

3 Numerical Experiments

This paper discusses the implementation of a Rosenbrock-Nystrom method based on **Algorithm 1** of [4] and the 4 stage, order 4 L-stable Rosenbrock formula introduced in the previous Section. Note that a W-method can be similarly implemented by replacing the corresponding coefficients of the Rosenbrock-Nystrom formula with the appropriate W-coefficients of Table 3. A set of numerical experiments is first carried out to validate the proposed Rosenbrock-Nystrom method. Then a comparison with an explicit integrator is performed to assess the efficiency of the proposed algorithm for numerical integration of a more complex mechanical

θ_{21}	=	1.14563212	a_{21}	=	0.20000000000000000000
θ_{31}	=	0.78950916281563862962698 0	a_{31}	=	1.86794814949823713234476
θ_{32}	=	0.13429418684250480014923 2	a_{32}	=	0.23444556851723885002322
θ_{41}	=	0.78950916281563862962698 0	a_{41}	=	1.86794814949823713234476
θ_{42}	=	0.13429418684250480014923 2	a_{42}	=	0.23444556851723885002322
θ_{43}	=	0.0	a_{43}	=	0.0
c_{21}	=	-7.13764994334997983036926 0	δ_{21}	=	-1.19636100711201394917052 0
c_{31}	=	2.58092366650965771448805 0	δ_{31}	=	1.47028025440978071463387 0
c_{32}	=	0.65162988730203202338741 7	δ_{32}	=	0.34810583767920449001670 4
c_{41}	=	-2.13711526650661911680637 0	δ_{41}	=	0.00376509435555616579897 4
c_{42}	=	-0.32146953133995107076924 1	δ_{42}	=	-0.10976248675810325567539 8
c_{43}	=	-0.69496604928244522515732 9	δ_{43}	=	-0.22803103597313382947774 4
m_1	=	2.25556622860456524372884 0	$\hat{m}_1$	=	2.06839916052758373425867 0
m_2	=	0.28705506319415760766263 0	$\hat{m}_2$	=	0.23868135206753279795649 3
m_3	=	0.43531196337998321340270 7	$\hat{m}_3$	=	0.36337334543539170826174 7
m_4	=	1.09350765640324780321482 0	$\hat{m}_4$	=	0.36655712793615514430916 3
μ_1	=	1.59275081940958534207490 0	$\hat{\mu}_1$	=	1.43490397184820947262710 0
μ_2	=	0.19593826631025060969332 9	$\hat{\mu}_2$	=	0.22297867258869836904515 3
μ_3	=	0.0	$\hat{\mu}_3$	=	0.12455968641470204977489 7
μ_4	=	0.62637874732074217784117 1	$\hat{\mu}_4$	=	0.20996980978930432131190 6
γ_2	=	-1.76917706711201394917052 0	a_2	=	1.145632120
γ_3	=	0.75929396429320985367096 7	a_3	=	0.65521497597313382947774 8
γ_4	=	-0.10489462149095580320674 3	a_4	=	0.65521497597313382947774 8

Table 2: Coefficients for the Rosenbrock-Nystrom method.

$\gamma = 1.70710678118650$					
α_1	=	0.00000000000000	γ_1	=	1.70710678118650
α_2	=	1.00000000000000	γ_2	=	-1.70710678118650
a_{21}	=	0.58578643762690	c_{21}	=	-1.17157287525380
δ_{11}	=	1.70710678118650	δ_{22}	=	1.70710678118650
δ_{21}	=	-2.41421356237310			
θ_{21}	=	1.00000000000000			
m_1	=	0.87867965644040	$\hat{m}_1$	=	1.17157287525380
m_2	=	0.29289321881340	$\hat{m}_2$	=	0.58578643762690
μ_1	=	0.79289321881340	$\hat{\mu}_1$	=	0.58578643762690
μ_2	=	0.50000000000000	$\hat{\mu}_2$	=	1.00000000000000

Table 3: Coefficients for the W-method.

system.

3.1 Validation of Proposed Algorithm

Validation is carried out using the double pendulum mechanism shown in Fig.1. Stiffness is induced by means of two rotational spring-damper-actuators (RSDA). The masses of the two pendulums are $\mathtt{m}_1 = 3$ and $\mathtt{m}_2 = 0.3$, the dimension of the pendulums are $\mathtt{L}_1 = 1$ and $\mathtt{L}_2 = 1.5$, the stiffness coefficients are $\mathtt{k}_1 = 400$ and $\mathtt{k}_2 = 3.E5$, and the damping coefficients are $\mathtt{c}_1 = 15$ and $\mathtt{c}_2 = 5.E4$. The zero-tension angles for the two RSDA elements are $\alpha_1^0 = 3\pi/2$ and $\alpha_2^0 = 0$. All units are SI.

In its initial configuration, the two degree of freedom dynamic system has a dominant eigenvalue with a small imaginary part and a real part of the order -10E5. Since the two pendulums are connected through two parallel revolute joints the problem is planar. In terms of initial conditions, the centers of mass (CM) of bodies 1 and 2 are located at $x_1^{CM} = 1$, $y_1^{CM} = 0$, and $x_2^{CM} = 3.4488887$, $y_2^{CM} = -0.388228$. In the initial configuration, the centroidal principal reference frame of body 1 is parallel with the global reference frame, while the centroidal principal reference frame of body 2 is rotated with $\theta_2 = 23\pi/12$ around an axis perpendicular on the plane of motion. For body 1, $\dot{x}_1^{CM} = \dot{y}_1^{CM} = \dot{\theta}_1^{CM} = 0$, while for body 2,

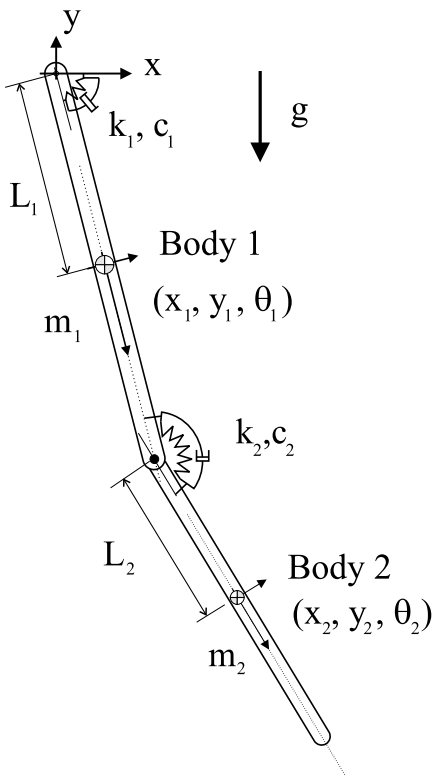

Figure 1: Double pendulum problem

$\dot{x}_2^{CM} = 3.8822857$, $\dot{y}_2^{CM} = 14.4888887$, and $\dot{\theta}_2^{CM} = 10$. All initial conditions are in SI units, and are consistent with the kinematic constraint equations at position and velocity levels (Eqs.(1a) and (1b)).

The first set of numerical experiments focuses on assessing the reliability of the step size control mechanism. The goal is to verify that user imposed levels of absolute and relative error are met by the simulation results. A *reference simulation* is first run using a very small constant integration step-size. Other simulations, run with different combinations of absolute and relative tolerances, are compared to the *reference simulation* to find the infinity norm of the error, the time at which this largest error occurred, and average error per time step. Suppose that n time steps are taken during the current simulation and that the variable whose accuracy is analyzed is denoted by e. The grid points of the current simulation are denoted by $t_{init} = t_1 < t_2 < \ldots < t_n = t_{end}$. If N is the number of time steps taken during the *reference simulation*; i.e., $T_{init} = T_1 < T_2 < \ldots < T_N = T_{end}$, assume that for the quantity of interest the computed reference values are E_j, for $1 \leq j \leq N$. For each $1 \leq i \leq n$, an integer $\mathbf{r}(i)$ is defined such that $T_{\mathbf{r}(i)} \leq t_i \leq T_{\mathbf{r}(i)+1}$. Using the reference values $E_{\mathbf{r}(i)-1}$, $E_{\mathbf{r}(i)}$, $E_{\mathbf{r}(i)+1}$, and $E_{\mathbf{r}(i)+2}$, cubic spline interpolation algorithm is used

to generate an interpolated value E_i^* at time t_i. If $\mathbf{r}(i) - 1 \leq 0$, the first four reference points are considered for interpolation, while if $\mathbf{r}(i) + 2 \geq N$, the last four reference points are used for interpolation. The error at time step i is defined as $\Delta_i = \mid E_i^* - e_i \mid$. For each tolerance set k, accuracy is measured by both the maximum $\Delta^{(k)}$ and the average $\overline{\Delta}^{(k)}$ trajectory errors, as well as by the percentage relative error

$$\Delta^{(k)} = \max_{1 \leq i \leq n} (\Delta_i) \ , \qquad \overline{\Delta}^{(k)} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \Delta_i^2} \ , \qquad \text{RelErr}[\%] = \frac{\Delta^{(k)}}{E^*} \times 100 \ ,$$

where $E^* = E_p$, with p defined such that $\Delta^{(k)} = \Delta_p$. Simulations are run for tolerances between 10^{-2} and 10^{-5}, a range that typically covers mechanical engineering accuracy requirements. The length of the simulation is 2 seconds. The time variation of the angle θ_1 is presented on the left of Fig.2. Notice that body 1 eventually stabilizes in the configuration $\theta_1 = 3\pi/2$, which is the zero-tension angle for the RSDA.

Table 4 contains error analysis information for angle θ_1. The first column contains the value of the tolerance with which the simulation is run. Relative and absolute tolerances ($Rtol_i$ and $Atol_i$ of Eqs.(4)) are set to 10^k, and they are applied for both position and velocity error control. The second column contains the time t^* at which the largest error $\Delta^{(k)}$ occurred. The third column contains the values of $\Delta^{(k)}$. Column four contains the relative error, and the last column shows the average trajectory error. Table 5 shows the number of integration steps selected by the numerical integrator for different values of the tolerance parameter k.

The most relevant information for step-size control validation is $\Delta^{(k)}$. If, for example, $k = -3$; i.e., accuracy of the order 10^{-3} is demanded, $\Delta^{(-3)}$ should have this order of magnitude. It can be seen from the results in Table 4 that this is the case for all tolerances. Note that these results are obtained with a non-zero relative tolerance. According to Eq.(4), depending on the magnitude of the variable being analyzed, the relative tolerance directly impacts the step-size control. Based on position results shown in Fig.2, the relative tolerance is multiplied by a value that for θ_1 oscillates between 4.0 and 6.0. Consequently, the actual upper bound of accuracy imposed on θ_1 fluctuates and reaches values up to $7 \cdot 10^{-k}$. Thus, the step-size controller is slightly conservative. For an explanation of this stiffness induced order reduction, the reader is referred to [2] or [6]. In the latter reference, the local truncation error $(\hat{y}_1 - y_1)$ in Eq.(4) is replaced by the scaled value $\delta = (I - h\gamma \partial f / \partial y)^{-1} (\hat{y}_1 - y_1)$. This step-size control strategy remains to be investigated.

Error analysis is also performed at the velocity level. The time variation of angular velocity $\dot{\theta}_1$ is shown in Fig.2. The angular velocity of body 1 fluctuates between -10 and 7 rad/s. As a result, values of $\Delta^{(k)}$ of

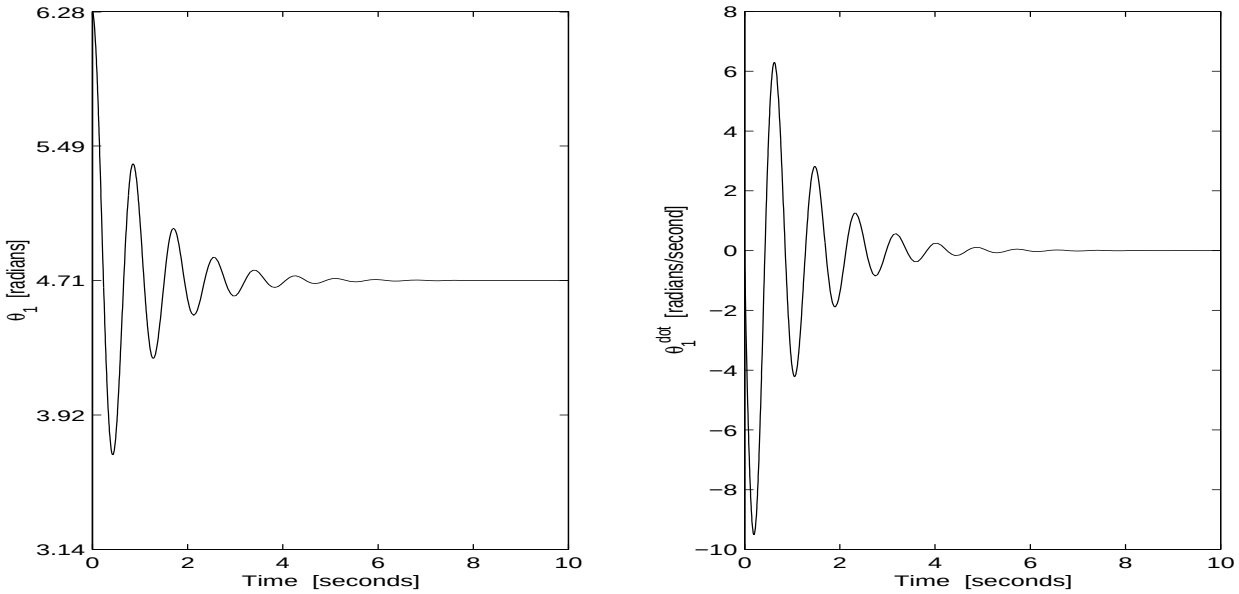

Figure 2: Time Variation of orientation θ_1 (Left) and of angular velocity $\dot{\theta_1}$ (Right) for Body 1.

Table 4: Position Error Analysis for the Double Pendulum Problem.

k	t^*	$\Delta^{(k)}$	RelErr[%]	$\overline{\Delta}^{(k)}$
-2	0.592127	5.223e-2	0.12126	3.234e-3
-3	0.599954	4.198e-3	0.00964	2.631e-4
-4	0.626135	4.916e-4	0.00108	2.946e-5
-5	1.065146	1.902e-5	0.00039	9.868e-6

up to the order 10^{k+1} are considered very good. Error analysis results for $\dot{\theta}_1$ are presented in Table 6. The step-size controller performs well, slightly on the conservative side.

The error analysis results presented suggest that the step-size controller employed is reliable, as the pre-imposed accuracy requirements are met or exceeded by the numerical results. In order to avoid unjustified CPU penalties, the algorithm may be improved for extremely stiff mechanical systems by adopting a modified step-size controller proposed in [6].

Table 5: Number of Integration Time Steps for the Double Pendulum Problem.

k	-2	-3	-4	-5	-6	-7
Steps	29	49	85	148	264	467

Table 6: Velocity Error Analysis for the Double Pendulum Problem.

k	t^*	$\Delta^{(k)}$	RelErr[%]	$\bar{\Delta^{(k)}}$
-2	0.795548	4.061e-2	1.84434	2.348e-2
-3	0.373114	3.792e-3	0.12340	2.181e-3
-4	0.217757	8.652e-4	0.00922	3.445e-4
-5	0.186183	2.343e-4	0.00246	9.357e-5

3.2 Performance Comparison with Explicit Integrator

In order to compare the performance of the proposed implicit algorithm with a SSODE algorithm based on a state of the art explicit integrator [6], a model of the US Army High Mobility Multipurpose Wheeled Vehicle (HMMWV) is considered for dynamic analysis. The HMMWV shown in Fig.3 is modeled using 14 bodies, as shown in Fig.3. In this figure, vertices represent bodies, while edges represent joints connecting the bodies of the system. Thus, vertex number 1 is the chassis, 2 and 5 are the right and left front upper control arms, 3 and 6 are the right and left front lower control arms, 9 and 12 are the right and left rear lower control arms, and 8 and 11 are the right and left rear upper control arms. Bodies 4, 7, 10, and 13 are the wheel spindles, and body 14 is the steering rack. Spherical joints are denoted by S, revolute joints by R, distance constraints by D, and translational joints by T. This set of joints imposes 79 constraint equations. One additional constraint equation is imposed on the steering system, such that the steering angle is zero; i.e., the vehicle drives straight. A total of 98 generalized coordinates are used to model the vehicle, which renders 18 degrees of freedom to the model.

Stiffness is induced in the model through means of four translational spring-damper actuators (TSDA). These TSDAs act between the front/rear and right/left upper control arms and the chassis. The stiffness

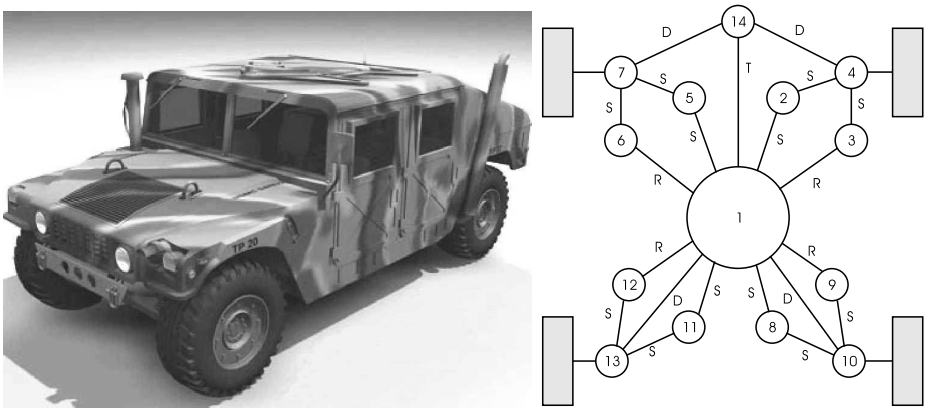

Figure 3: The HMMWV (left) and the associated model topology (right).

coefficient of each TSDA is 2E7 N/m, while the damping coefficient is 2E6 N·s/m. For the purpose of this numerical experiment, the tires of the vehicle are modeled as vertical TSDA elements with stiffness coefficient 296325 N/m and damping coefficient 3502 N·s/m. Finally, the dominant eigenvalue of the corresponding SSODE has a real component of approximately -2.6E5, and a small imaginary part.

Dynamic analysis of the model is carried out for the vehicle driving straight at 10mph over a bump. The shape of the bump is a half-cylinder of diameter 0.1m. Figure 4 shows the time variation of the vehicle chassis height. The front wheels hit the bump at time 0.5 seconds, and the rear wheels hit the bump at time 1.2 seconds. The length of the simulation in this plot is 5 seconds. Toward the end of the simulation (after 4 seconds), due to over-damping the, chassis height stabilizes at approximately $z_1 = 0.71$m.

The test problem is first run with an explicit integrator based on the code DEABM [6]. **Algorithm 2** below outlines the explicit integration approach used for SSODE integration of the equations of motion for the HMMWV model. The first 3 steps are identical to the ones in **Algorithm 1** in [4]. Step 4 computes the acceleration $\ddot{\mathbf{q}}$, by solving the linear system of Eq.(1d). A topology-based approach [5], that takes into account the sparsity of the coefficient matrix is used to solve for the generalized accelerations $\ddot{\mathbf{q}}$. The DDEABM integrator is then used to integrate for independent velocities $\dot{\mathbf{v}}_n$, and independent positions $\mathbf{v}_n$ [4]. The integrator is also used to integrate for the dependent coordinates $\mathbf{u}_n$, with the sole purpose of providing a good starting point during Step 6 that computes $\mathbf{u}_n$ by ensuring that the kinematic position constraint equations are satisfied; i.e., solving $\mathbf{\Phi}(\mathbf{v}_n, \mathbf{u}_n) = \mathbf{0}$. Likewise, dependent velocities $\dot{\mathbf{u}}_n$ are the solution of the linear system $\mathbf{\Phi}_\mathbf{u}(\mathbf{u}_n, \mathbf{v}_n)\dot{\mathbf{u}}_n = -\mathbf{\Phi}_\mathbf{v}(\mathbf{u}_n, \mathbf{v}_n)\dot{\mathbf{v}}_n$ [4], which thus guarantees that the generalized

velocities satisfy the kinematic velocity constraint equations. The dependent/independent partitioning of the generalized coordinates is checked during Step 7.

Algorithm 2

1. Initialize Simulation
2. Set Integration Tolerance
3. While (time < time-end) do
4. Get Acceleration
5. Apply Integration Step.
   Check Accuracy. Determine New Step-size
6. Recover Dependent Generalized Coordinates
7. Check Partition
8. End do

Timing results reported are obtained on an SGI Onyx computer with an R10000 processor. Computer times required by **Algorithm 2** are listed in Table 7. Results for the Rosenbrock Nystrom algorithm are presented in Table 8.

Table 7: Explicit Integrator Timing Results for the HMMWV Problem.

Tol	10^{-2}	10^{-3}	10^{-4}	10^{-5}
1 sec	3618	3641	3667	3663
2 sec	7276	7348	7287	7276
3 sec	10865	11122	10949	10965
4 sec	14480	14771	14630	14592

Results in Table 7 are typical for the situation when an explicit integrator is used for the numerical solution of a stiff IVP. For the stiff test problem considered, the performance limiting factor is stability of the explicit code. For a given simulation length, any tolerance in the range 1E-2 through 1E-5 results in

Table 8: Implicit Integrator Timing Results for the HMMWV Problem.

Tol	10^{-2}	10^{-3}	10^{-4}	10^{-5}
1 sec	5.6	13.2	40.7	172
2 sec	12.6	32.6	95	405
3 sec	13	36.3	105	422
4 sec	13.3	37	106	428

almost identical CPU times. The average explicit integration step-size turns out to be between 1E-5 and 1E-6, and it is not affected by accuracy requirements. The code is compelled to select very small step-sizes to assure stability of the integration process, and this is the criteria for step-size selection for a broad spectrum of tolerances. Only if extreme accuracy is imposed, does the step-size become limited based on accuracy considerations. In this context, note that the results in Table 8 indicate that stability is of no concern for the proposed algorithm, and solution accuracy solely determines the duration of the simulation. The integration step-size is automatically adjusted to keep integration error within the user prescribed limits. Figure 4 shows the time variation for the integration step-size when the absolute and relative errors at position and velocity levels are set to 10^{-3}. The y-axis for the step-size is provided at the right of Fig.4, on a logarithmic scale. In the lower half of the same figure, relative to the left y-axis is provided the time variation of the chassis height. Note that when the vehicle hits the bump; i.e., when in Fig.4 the z coordinate of the chassis increases suddenly, the step-size is simultaneously decreased to preserve accuracy of the numerical solution. On the other hand, for the region in which the road becomes flat; i.e., toward the end of the simulation, the integrator is capable of taking larger integration steps, thus decreasing simulation time.

4 Conclusions

A generalized coordinate partitioning based state-space implicit integration method is introduced for the dynamic analysis of multibody systems. In the companion paper [4] the derivation of a Rosenbrock-Nystrom family of methods for state space implicit integration was presented. Based on this, the paper defines a

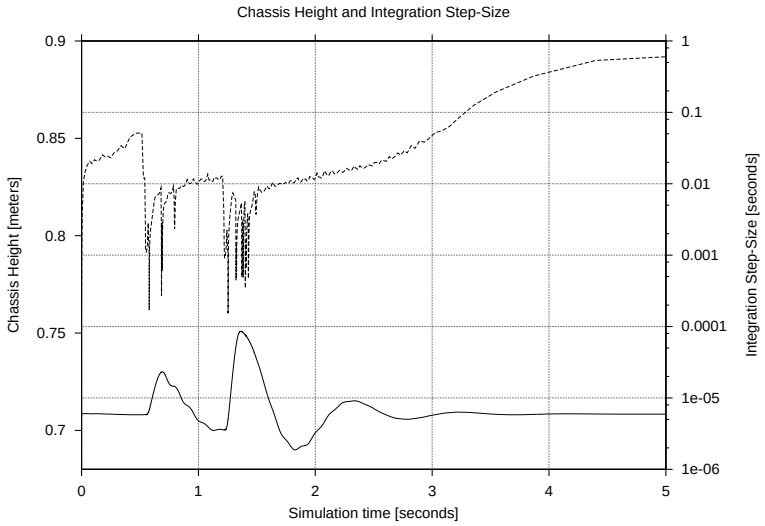

Figure 4: Chassis height and integration step-size

particular method based on a 4-stage order 4 L-stable Rosenbrock formula that has an order 3 embedded formula for error control. For a 14 body 18 degree of freedom vehicle, the proposed method is almost two orders of magnitude faster than an explicit integrator based method. The most restrictive condition associated with the use of the Rosenbrock formula employed is the requirement of an exact integration Jacobian. In this context, a formalism for systematically computing the state-space integration Jacobian is presented in [4]. When providing an exact Jacobian is not feasible, a lower order W-method is suggested as an alternative.

References

[1] E. Hairer, Norsett S. P., and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer-Verlag, Berlin Heidelberg New York, 1993.

[2] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin Heidelberg New York, 1996.

[3] E. J. Haug. *Computer-Aided Kinematics and Dynamics of Mechanical Systems.* Allyn and Bacon, Boston, London, Sydney, Toronto, 1989.

[4] A. Sandu, D. Negrut, E.J. Haug, F.A. Potra and C. Sandu. A Rosenbrock-Nystrom State Space Implicit Approach for the Dynamic Analysis of Mechanical Systems: I – Theoretical Formulation. *Submitted to Journal of Multibody Dynamics*, 2002.

[5] R. Serban, D. Negrut, E. J. Haug, and F. A. Potra. A Topology Based Approach for Exploiting Sparsity in Multibody Dynamics in Cartesian Formulation. *Mech. Struct.&Mach.*, 25(3):379–396, 1997.

[6] L. F. Shampine and H. A. Watts. The art of writing a Runge-Kutta code.II. *Appl. Math. Comput.*, 5:93–121, 1979.

List of Figure Captions

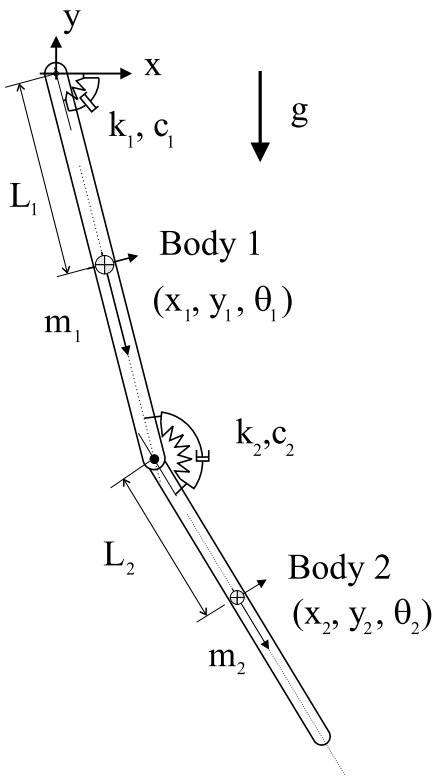

Figure 1. Double pendulum problem.

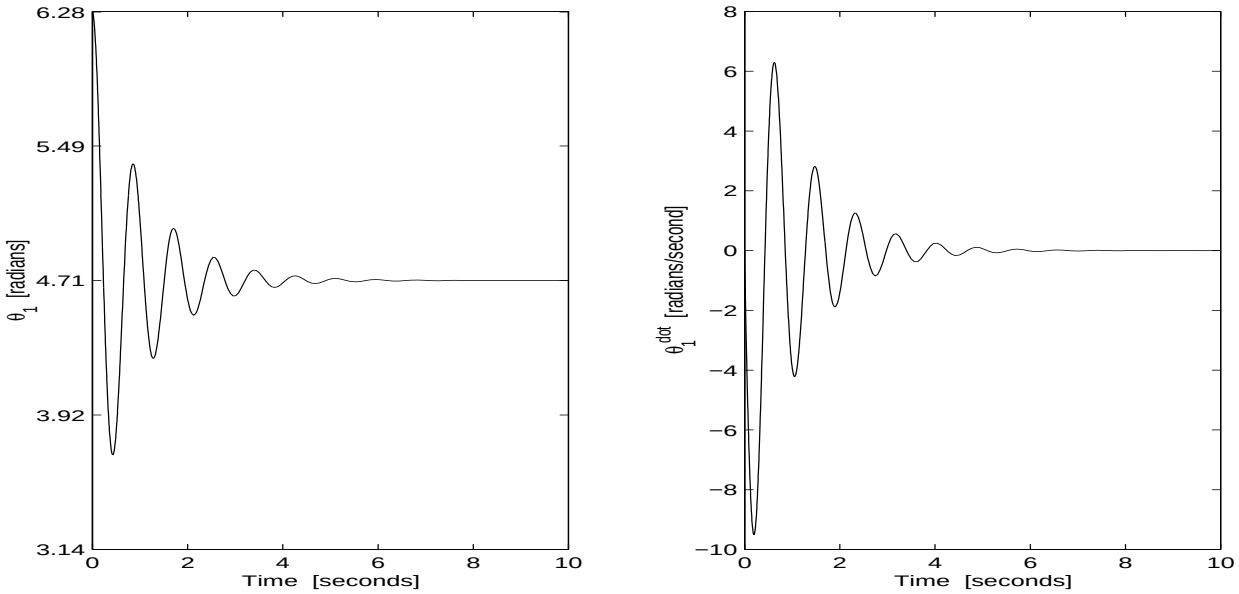

Figure 2. Time Variation of orientation θ_1 (Left) and of angular velocity $\dot{\theta}_1$ (Right) for Body 1.

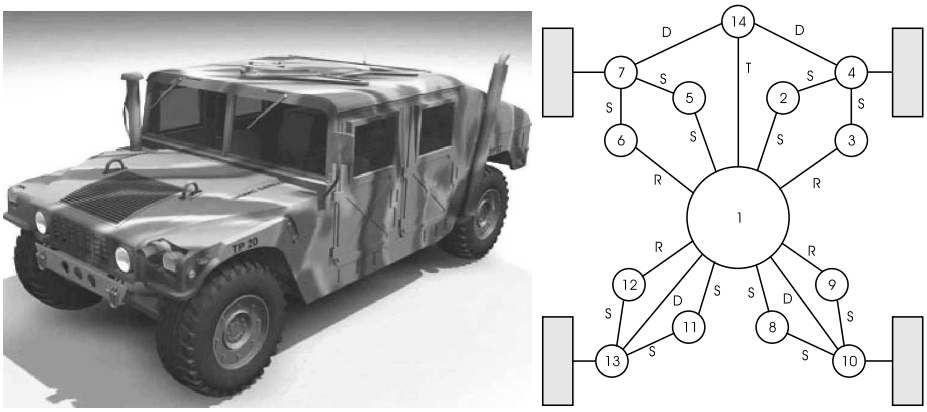

Figure 3. The HMMWV (left) and its topology representation (right).

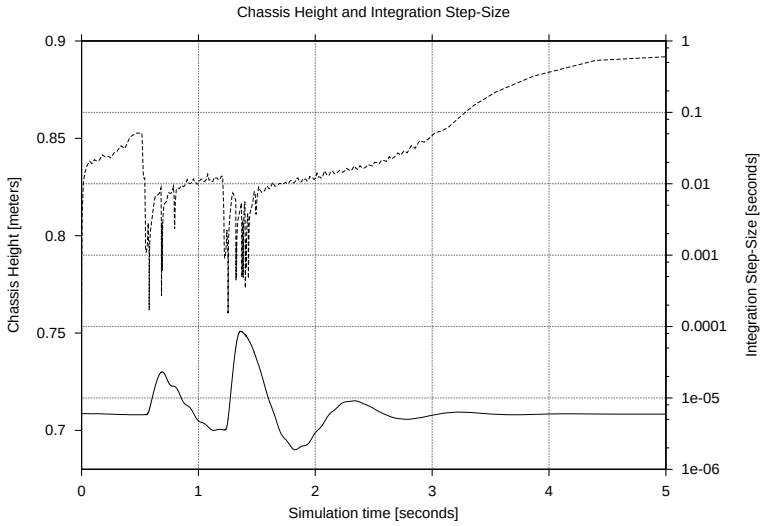

Figure 4. Chassis height and integration step-size