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

Dan Negrut†
MSCsoftware Inc.,
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.

November 26, 2002

---

* This research was supported in part by the US Army Tank-Automotive Research, Development, and Engineering Center (DoD contract number DAAE07-94-R004), a multiuniversity Center led by the University of Michigan.

† Corresponding Author.
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. In the companion paper [4] we developed a state-space based method for the numerical solution of the resulting DAE. The numerical method uses a linearly-implicit time stepping formula of Rosenbrock type, which is suitable for medium accuracy integration of stiff systems. In this paper we discuss choices of method coefficients and present 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 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

In this paper we present a method for solving the dynamics of a multibody system. The generalized coordinates $\mathbf{q}$ used are Cartesian coordinates for position and Euler parameters for orientation of body centroidal reference frames. Kinematic constraints are formulated as algebraic expressions involving generalized coordinates,

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

where $m$ is the total number of independent constraint equations that must be satisfied by the generalized coordinates throughout the simulation. The number of degrees of freedom $ndof$ is thus the difference between the number of generalized coordinates $n$ and the number of constraints $ndof = n - m$. Differentiating Eq.(1a) with respect to time leads to the velocity kinematic constraint equation

$$\mathbf{\Phi}_q(\mathbf{q}) \dot{\mathbf{q}} = 0 ,$$  \hspace{1cm} (1b)

where the over dot denotes differentiation with respect to time and the subscript denotes partial differentiation.

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

\[
\begin{bmatrix}
M(q) & \Phi(q)(q)^T \\
\Phi(q) & 0
\end{bmatrix}
\begin{bmatrix}
\ddot{q} \\
\lambda
\end{bmatrix}
= \begin{bmatrix}
Q^A(q, \dot{q}, t) \\
\tau(q, \dot{q})
\end{bmatrix}.
\]  

(1c)

Equations (1a)–(1c) comprise a system of differential-algebraic equations (DAE). In the companion paper [4] we developed an integration algorithm 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.

In this paper we present practical Rosenbrock-Nystrom algorithms based on a fourth order L-stable Rosenbrock method and on a second order method that can accomodate inexact Jacobians. Numerical experiments for two problems indicate that the approach is valid and efficient for medium accuracy integration of stiff state space equations.

2 The Proposed Algorithm

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

\[
y_{n+1} = y_n + \sum_{i=1}^{s} b_i k_i ,
\]

(2a)

\[
k_i = h f \left( t_n + \alpha_i h, y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + \gamma_i h^2 \frac{\partial f}{\partial y} (t_n, y_n) + hJ \sum_{j=1}^{i} \gamma_{ij} k_j ,
\]

(2b)

where the coefficients \( \alpha, \gamma \) 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 \( \hat{b_i} \).

\[
\hat{y}_{n+1} = y_n + \sum_{i=1}^{s} \hat{b}_i k_i
\]

(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 right below a user prescribed absolute and/or
relative tolerance. If $\mathbf{y} \in \mathbb{R}^s$, at time step $n+1$ the error in component $i$ is kept smaller than a composite error tolerance $sc_i$

$$|y_{n+1}^i - \tilde{y}_{n+1}^i| < sc_i, \quad sc_i = Atol_i + \max(|y_n^i|, |y_{n+1}^i|) \cdot Rtol_i$$

(4)

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

$$err = \left( \frac{1}{s} \sum_{i=1}^{s} \frac{(y_{n+1}^i - \tilde{y}_{n+1}^i)^2}{sc_i^2} \right)^{1/2}$$

(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 $err \approx Ch^{\hat{p}+1}$, where $C$ is a constant depending on the choice of formulas and $q = \min(p, \hat{p})$. Optimally, $err = 1$ and therefore $1 \approx Ch^{\hat{p}+1}$. The optimal step-size is computed then as

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

(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 $facmax$,

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

(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. In terms of function evaluations, this makes the proposed Rosenbrock-Nystrom method competitive with the trapezoidal method, whenever the latter requires 3 or more iterations for convergence. Moreover, the trapezoidal method is of order 2 and only weakly A-stable.

With $\beta_i^j = \sum_{j=1}^{i-1} \beta_{ij}$, the defining coefficients $\alpha_{ij}$, $\gamma_{ij}$, and $b_i$ of an order 4 Rosenbrock method of Eqs.(2a-
2b) are subject to the following order conditions [2]:

\[ b_1 + b_2 + b_3 + b_4 = 1 \]  \hspace{1cm} (8a)

\[ b_2 \beta_2^2 + b_3 \beta_3^2 + b_4 \beta_4^2 = 1/2 - \gamma \]  \hspace{1cm} (8b)

\[ b_2 \alpha_2^2 + b_3 \alpha_3^2 + b_4 \alpha_4^2 = 1/3 \]  \hspace{1cm} (8c)

\[ b_3 \beta_3 \beta_2^2 + b_4 (\beta_{42} \beta_2^2 + \beta_{43} \beta_3^2) = 1/6 - \gamma + \gamma^2 \]  \hspace{1cm} (8d)

\[ b_2 \alpha_2^3 + b_3 \alpha_3^3 + b_4 \alpha_4^3 = 1/4 \]  \hspace{1cm} (8e)

\[ b_3 \alpha_3 \alpha_2 \beta_2^2 + b_4 \alpha_4 (\alpha_{42} \beta_2^2 + \alpha_{43} \beta_3^2) = 1/8 - \gamma/3 \]  \hspace{1cm} (8f)

\[ b_3 \beta_3 \alpha_2 \alpha_2^2 + b_4 (\beta_{42} \alpha_2^2 + \beta_{43} \alpha_3^2) = 1/12 - \gamma/3 \]  \hspace{1cm} (8g)

\[ b_4 \beta_{43} \beta_2 \beta_2^2 = 1/24 - \gamma/2 + 1.5\gamma^2 - \gamma^3 \]  \hspace{1cm} (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 \( \hat{y}_i = y_0 + \sum_{i=1}^{s} \hat{b}_i k_i \). The order conditions for the order 3 algorithm are as indicated as Eqs. (8a–8d), and they lead to the system

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
0 & \beta_2^2 & \beta_3^2 & \beta_4^2 \\
0 & \alpha_2^2 & \alpha_3^2 & \alpha_4^2 \\
0 & 0 & \beta_{32} \beta_2^2 & \beta_{42} \beta_2^2 + \beta_{43} \beta_3^2
\end{bmatrix}
\begin{bmatrix}
\hat{b}_1 \\
\hat{b}_2 \\
\hat{b}_3 \\
\hat{b}_4
\end{bmatrix}
= \begin{bmatrix}
1 \\
1/2 - \gamma \\
1/3 \\
1/6 - \gamma + \gamma^2
\end{bmatrix}
\]  \hspace{1cm} (9)

If the coefficient matrix in Eq. (9) is non-singular, uniqueness of the solution of this linear system implies \( b_1 = \hat{b}_1 \). 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 the condition

\[ \beta_{32} \beta_2^2 (\beta_4^2 \alpha_4^2 - \beta_4^2 \alpha_2^2) = (\beta_2^2 \alpha_3^2 - \beta_2^2 \alpha_2^2) (\beta_{42} \beta_2^2 + \beta_{43} \beta_3^2) \]  \hspace{1cm} (10)

The number of coefficients that must be determined is 17; the diagonal coefficient \( \gamma \), six coefficients \( \gamma_{ij} \), six coefficients \( \alpha_{ij} \), 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

\[ \alpha_{41} = \alpha_{31} , \hspace{0.5cm} \alpha_{42} = \alpha_{32} , \hspace{0.5cm} \alpha_{43} = 0 , \]  \hspace{1cm} (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

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

(12)

A second order 5 condition is satisfied by imposing the condition

$$b_4\beta_3\alpha_3^2(\alpha_3 - \alpha_2) = 1/20 - \gamma/4 - \alpha_2 (1/12 - \gamma/3)$$

(13)

Next, two conditions are chosen as

$$b_3 = 0, \quad \alpha_2 = 2\gamma,$$

(14)

to make the task of finding the defining coefficients $\alpha_{ij}$, $\gamma_{ij}$, and $b_i$ more tractable. Finally, the last condition regards the choice of the diagonal element $\gamma$. 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, accurate up to 25 digits, is provided below, along with the coefficients $\hat{b}_i$ of the order 3 embedded formula.

<table>
<thead>
<tr>
<th>$\gamma = 0.57281606$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{21} = 1.14563212$</td>
</tr>
<tr>
<td>$\alpha_{31} = 0.520920789130629029328516$</td>
</tr>
<tr>
<td>$\alpha_{32} = 0.134294186842304800149232$</td>
</tr>
<tr>
<td>$\alpha_{41} = 0.520920789130629029328516$</td>
</tr>
<tr>
<td>$\alpha_{42} = 0.134294186842304800149232$</td>
</tr>
<tr>
<td>$\alpha_{43} = 0.0$</td>
</tr>
<tr>
<td>$b_1 = 0.324534707891734513474196$</td>
</tr>
<tr>
<td>$b_2 = 0.049086544787523308684633$</td>
</tr>
<tr>
<td>$b_3 = 0.0$</td>
</tr>
<tr>
<td>$b_4 = 0.626378747320742177841171$</td>
</tr>
</tbody>
</table>
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 below.

| $\theta_{21}$ = | 1.14563212 | $\alpha_{21}$ = | 0.200000000000000000000000000000 |
| $\theta_{31}$ = | 0.789509168215638629626980 | $\alpha_{31}$ = | 1.867921494982371323474476 |
| $\theta_{22}$ = | 0.13429418642504800149232 | $\alpha_{32}$ = | 0.23444556851723885002322 |
| $\theta_{41}$ = | 0.789509168215638629626980 | $\alpha_{41}$ = | 1.867921494982371323474476 |
| $\theta_{42}$ = | 0.13429418642504800149232 | $\alpha_{42}$ = | 0.23444556851723885002322 |
| $\theta_{43}$ = | 0.0 | $\alpha_{43}$ = | 0.0 |
| $c_{21}$ = | -7.137649943349979830369260 | $\delta_{21}$ = | -1.19236100112011394170520 |
| $c_{31}$ = | 2.58092266650965771488050 | $\delta_{31}$ = | 1.47050254409780714633870 |
| $c_{32}$ = | 0.651629887302032023387417 | $\delta_{32}$ = | 0.348105837679204490016704 |
| $c_{41}$ = | -2.13715266636619116806370 | $\delta_{41}$ = | 0.0037650943555561579874 |
| $c_{42}$ = | -0.321469531339951037796241 | $\delta_{42}$ = | -0.1097624867581032556753098 |
| $c_{43}$ = | -0.694966049282445225157329 | $\delta_{43}$ = | -0.228031035973133829477744 |
| $m_{1}$ = | 2.255566228604565243728840 | $\tilde{m}_{1}$ = | 2.06839160527583734258670 |
| $m_{2}$ = | 0.28705563194157607662630 | $\tilde{m}_{2}$ = | 0.23868132067532797936493 |
| $m_{3}$ = | 0.435311963379983213402707 | $\tilde{m}_{3}$ = | 0.36337334535391708261747 |
| $m_{4}$ = | 1.093507656403247803214820 | $\tilde{m}_{4}$ = | 0.36657127936155144309163 |
| $\mu_{1}$ = | 1.59275081940958342074900 | $\hat{\mu}_{1}$ = | 1.434903971848209472627100 |
| $\mu_{2}$ = | 0.195938266310250609693329 | $\hat{\mu}_{2}$ = | 0.222978672588698369045153 |
| $\mu_{3}$ = | 0.0 | $\hat{\mu}_{3}$ = | 0.12459686414702949774897 |
| $\mu_{4}$ = | 0.626378747320742177841171 | $\hat{\mu}_{4}$ = | 0.20996809789304321311906 |
| $\gamma_{2}$ = | -1.769177067112013949170520 | $a_{2}$ = | 1.145632120 |
| $\gamma_{3}$ = | 0.759293964293209853670967 | $a_{3}$ = | 0.655214975973133829477748 |
| $\gamma_{4}$ = | -0.104904624009558032606743 | $a_{4}$ = | 0.655214975973133829477748 |

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 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 the table below.

<table>
<thead>
<tr>
<th>$\gamma = 1.70710678118650$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1 = 0.00000000000000$</td>
</tr>
<tr>
<td>$\alpha_2 = 1.00000000000000$</td>
</tr>
<tr>
<td>$a_{21} = 0.58578643762690$</td>
</tr>
<tr>
<td>$\delta_{11} = 1.70710678118650$</td>
</tr>
<tr>
<td>$\delta_{21} = -2.41421356237310$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\theta_{21} = 1.00000000000000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1 = 0.87867965644040$</td>
</tr>
<tr>
<td>$m_2 = 0.29289321881340$</td>
</tr>
<tr>
<td>$\mu_1 = 0.79289321881340$</td>
</tr>
<tr>
<td>$\mu_2 = 0.50000000000000$</td>
</tr>
</tbody>
</table>

This paper is concerned with the implementation of a Rosenbrock-Nystrom based method. Algorithm 1 based on the 4 stage, order 4 L-stable Rosenbrock formula introduced is presented below. The W-method can be similarly implemented by replacing the corresponding coefficients of the Rosenbrock-Nystrom formula with the coefficients provided in the previous table. Details of this implementation and the performance of such a method are not presented here.

### 3 Numerical Experiments

A set of numerical experiments is first carried out to validate the proposed algorithm. 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 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 \( m_1 = 3 \) and \( m_2 = 0.3 \), the dimension of the pendulums are \( L_1 = 1 \) and \( L_2 = 1.5 \), the stiffness coefficients are \( k_1 = 400 \) and \( k_2 = 3 \times 10^5 \), and the damping coefficients are \( C_1 = 15 \) and \( C_2 = 5 \times 10^4 \). 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 \(-10^5\). 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, \( \dot{x}_2^CM = 3.8822857, \dot{y}_2^CM = 14.4888887, \) 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 \( r(i) \) is defined such that \( T_{r(i)} \leq t_i \leq T_{r(i)+1} \).

Using the reference values \( E_{r(i)-1}, E_{r(i)}, E_{r(i)+1}, \) and \( E_{r(i)+2} \), cubic spline interpolation algorithm is used to generate an interpolated value \( E_i^* \) at time \( t_i \). If \( r(i) - 1 \leq 0 \), the first four reference points are considered.
for interpolation, while if \( r(i) + 2 \geq N \), the last four reference points are used for interpolation. The error at time step \( i \) is defined as \( \Delta_i = \| E^*_i - e_i \| \). 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

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

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 \( \theta_1 \) 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 1 contains error analysis information for angle \( \theta_1 \). 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 \( 10^4 \), 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
Figure 2: Time Variation of orientation $\theta_1$ (Left) and of angular velocity $\dot{\theta}_1$ (Right) for Body 1.

contains the relative error, and the last column shows the average trajectory error. Table 2 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 1 that this is the case for all tolerances. Notice 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 loosens or tightens the step-size control. Based on results shown on the left of Fig.2, the relative tolerance is multiplied by a value that oscillates between 4.0 and 6.0. Consequently, the actual upper bound of accuracy imposed on $\theta_1$ fluctuates and reaches values up to $7 \cdot 10^{-3}$. Thus, the step-size controller is slightly conservative. For an explanation of this stiffness induced order reduction, the reader is referred to [2]. To remedy this, in [6], a scaling of the truncation error that enters in Eq.(4) the computation of the new step-size is recommended. In this context, the quantity $(\hat{y}_1 - y_1)$ 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 on the right of Fig.2. The angular velocity of body 1 fluctuates between -10 and 7 rad/s. As a result, values of $\Delta^{(k)}$ of up to the order $10^{k+1}$ are considered very good. Error analysis results for $\dot{\theta}_1$ are presented in
Table 1: Position Error Analysis for the Double Pendulum Problem.

<table>
<thead>
<tr>
<th>k</th>
<th>$t^*$</th>
<th>$\Delta^{(k)}$</th>
<th>RelErr [%]</th>
<th>$\Delta^{(k)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>0.592127</td>
<td>5.223e-2</td>
<td>0.12126</td>
<td>3.234e-3</td>
</tr>
<tr>
<td>-3</td>
<td>0.599954</td>
<td>4.198e-3</td>
<td>0.00964</td>
<td>2.631e-4</td>
</tr>
<tr>
<td>-4</td>
<td>0.626135</td>
<td>4.916e-4</td>
<td>0.00108</td>
<td>2.946e-5</td>
</tr>
<tr>
<td>-5</td>
<td>1.065146</td>
<td>1.902e-5</td>
<td>0.00039</td>
<td>9.868e-6</td>
</tr>
</tbody>
</table>

Table 2: Number of Integration Time Steps for the Double Pendulum Problem.

<table>
<thead>
<tr>
<th>k</th>
<th>-2</th>
<th>-3</th>
<th>-4</th>
<th>-5</th>
<th>-6</th>
<th>-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steps</td>
<td>29</td>
<td>49</td>
<td>85</td>
<td>148</td>
<td>264</td>
<td>467</td>
</tr>
</tbody>
</table>

Table 3: 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. The step-size control mechanism used indicates that using an embedded formula for local error estimation is suitable, since for the test problem considered 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 the step-size controller proposed in [6].

Table 3: Velocity Error Analysis for the Double Pendulum Problem.

<table>
<thead>
<tr>
<th>k</th>
<th>$t^*$</th>
<th>$\Delta^{(k)}$</th>
<th>RelErr [%]</th>
<th>$\Delta^{(k)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>0.795548</td>
<td>4.061e-2</td>
<td>1.84434</td>
<td>2.348e-2</td>
</tr>
<tr>
<td>-3</td>
<td>0.373114</td>
<td>3.792e-3</td>
<td>0.12340</td>
<td>2.181e-3</td>
</tr>
<tr>
<td>-4</td>
<td>0.217757</td>
<td>8.652e-4</td>
<td>0.00922</td>
<td>3.445e-4</td>
</tr>
<tr>
<td>-5</td>
<td>0.186183</td>
<td>2.343e-4</td>
<td>0.00246</td>
<td>9.357e-5</td>
</tr>
</tbody>
</table>
3.2 Performance Comparison with Explicit Integrator

In order to compare the performance of the proposed implicit algorithm with a state of the art SSODE explicit integrator, 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 coefficient of each TSDA is $2E07$ N/m, while the damping coefficient is $2E06N \cdot s/m$. 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 $3502N \cdot s/m$. 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 $z_1 = 0.71m$.

The test problem is first run with an explicit integrator based on the code DEABM [6]. **Algorithm 2** outlines the explicit integration approach used for SSODE integration of the equations of motion for the HMMWV model.
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

The first 3 steps are identical to the ones in Algorithm 1. Step 4 computes the acceleration \( \ddot{\mathbf{q}} \), by solving the linear system of Eq.(1c). 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 \( \mathbf{v}_n \). 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 \( \Phi(\mathbf{v}_n, \mathbf{u}_n) = \mathbf{0} \). Likewise, dependent velocities \( \dot{\mathbf{u}}_n \) are the solution of the linear system \( \Phi_u(\mathbf{u}_n, \mathbf{v}_n)\dot{\mathbf{u}}_n = -\Phi_v(\mathbf{u}_n, \mathbf{v}_n)\dot{\mathbf{v}}_n \), 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.

Timing results reported are obtained on an SGI Onyx computer with an R10000 processor. Computer times required by Algorithm 2 are listed in Table 4. Results for the Rosenbrock Nystrom algorithm are presented in Table 5.

Table 4: Explicit Integrator Timing Results for the HMMWV Problem.

<table>
<thead>
<tr>
<th>Tol</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 sec</td>
<td>3618</td>
<td>3641</td>
<td>3667</td>
<td>3663</td>
</tr>
<tr>
<td>2 sec</td>
<td>7276</td>
<td>7348</td>
<td>7287</td>
<td>7276</td>
</tr>
<tr>
<td>3 sec</td>
<td>10865</td>
<td>11122</td>
<td>10949</td>
<td>10965</td>
</tr>
<tr>
<td>4 sec</td>
<td>14480</td>
<td>14771</td>
<td>14630</td>
<td>14592</td>
</tr>
</tbody>
</table>

Table 5: Implicit Integrator Timing Results for the HMMWV Problem.

<table>
<thead>
<tr>
<th>Tol</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
<th>$10^{-5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 sec</td>
<td>5.6</td>
<td>13.2</td>
<td>40.7</td>
<td>172</td>
</tr>
<tr>
<td>2 sec</td>
<td>12.6</td>
<td>32.6</td>
<td>95</td>
<td>405</td>
</tr>
<tr>
<td>3 sec</td>
<td>13</td>
<td>36.3</td>
<td>105</td>
<td>422</td>
</tr>
<tr>
<td>4 sec</td>
<td>13.3</td>
<td>37</td>
<td>106</td>
<td>428</td>
</tr>
</tbody>
</table>

Results in Table 4 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 any tolerance in the range 1E-2 through 1E-5 and any given simulation length, CPU times are almost identical. 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 when extremely severe accuracy constraints are imposed on integration, does the step-size
become limited by accuracy considerations. In this context, note that the results in Table 5 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.

![Chassis Height and Integration Step-Size](image)

Figure 4: Chassis height and integration step-size

4 Conclusions

A generalized coordinate partitioning based state-space implicit integration method is presented for dynamic analysis of multibody systems. In the companion paper [4] we presented a derivation of Rosenbrock-Nystrom methods for state space integration of multibody systems. In this paper we present a particular method
based on a 4-stage Rosenbrock type formula which is L-stable and has an embedded order 3 formula for error control. For a 14 body 18 degree of freedom vehicle, the proposed algorithm is almost two orders of magnitude faster than an explicit integrator based method. There is room for improvement as far as the step-size controller is concerned, as it is seen to be conservative when very stiff models are run under stringent accuracy requirements. The most restrictive condition imposed on the user by the Rosenbrock formula employed is the requirement of an exact integration Jacobian. A formalism for computing the analytical integration Jacobian required is presented in [4]. When providing an exact Jacobian is not feasible, a lower order W-method is suggested as an alternative.

References


