



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 158 (1998) 341–355

**Computer methods
in applied
mechanics and
engineering**

A time integration algorithm for flexible mechanism dynamics: The DAE α -method

Jeng Yen¹, Linda Petzold^{*,2}, Soumyendu Raha²

Department of Computer Science and Army High Performance Computing Research Center, University of Minnesota, Minneapolis, MN 55455, USA

Received 10 May 1996; revised 22 June 1997

Abstract

This paper introduces a new family of second-order methods for solving the index-2 differential-algebraic equations (DAEs) of motion for flexible mechanism dynamics. These methods, which extend the α -methods for ODEs of structural dynamics to DAEs, possess numerical dissipation that can be controlled by the user. Convergence and stability analysis is given and verifies that the DAE α -methods introduce no additional oscillations and preserve the stability of the underlying ODE system. Convergence of the Newton iteration, which can be a source of difficulty in solving nonlinear oscillatory systems with large stepsizes, is achieved via a coordinate-split modification to the Newton iteration. Numerical results illustrate the effectiveness of the new methods for simulation of flexible mechanisms. © 1998 Elsevier Science S.A.

1. Introduction

The numerical solution of flexible multibody systems is required for nonlinear dynamic analysis of articulated structures. These are structures in which kinematic connections permit large relative motion between components that undergo small elastic deformation. A source of difficulty in the solution of flexible multibody equations of motion is the coupling between the elastodynamic equations and the gross motion. Simulation of flexible multibody systems has been an active research topic for the last two decades. Many of the developments in flexible multibody systems have been implemented in the multibody dynamic analysis codes [19,21]. For such systems, an important feature of the solution is the nonlinear oscillations induced by the elastodynamics equations. Moreover, since the governing equations of flexible multibody systems are often modeled using algebraic constraints, the numerical solution of DAEs is required [1,29]. This paper is concerned with the numerical solution of the DAEs of constrained flexible multibody motion by applying the family of α -methods for linear structural dynamics problems.

Recent developments in the simulation of multibody systems have created a variety of methodologies for efficient automation of the process of constructing the highly nonlinear equations of motion of constrained multibody mechanical systems [10]. The governing equations of motion are the well-known Euler–Lagrange equations,

$$M(q)\ddot{q} - f(\dot{q}, q, t) + G^T(q, t)\lambda = 0 \quad (1.1a)$$

$$g(q, t) = 0 \quad (1.1b)$$

* Corresponding author.

¹ Dept. of System Engineering, United Defense, LP, 4800 East River Road, Minneapolis, MN 55421, USA.

² Dept. of Mechanical and Environmental Engineering, Engr. 2 Bldg., Room 2355, University of California, Santa Barbara, Santa Barbara, CA 93106-5070, USA.

where q are the generalized coordinates, M is the mass-inertia matrix, λ are the Lagrange multipliers that couple the kinematic constraints g with the Newton–Euler equations of motion, and $G = \partial g / \partial q$ is the constraint Jacobian. For more details on the construction of the Euler–Lagrange equations of constrained multibody systems, we refer to [10,14,20,21]. While many contributions to the literature have focused on the efficient generation of (1.1), in recent years there has been progress in the numerical solution of differential-algebraic equations [1,9,8]. Two basically different approaches have been proposed for integration of (1.1). The first involves direct application of *stiffly stable* numerical integration, such as backward differentiation formulas (BDF), to various forms of DAEs derived from (1.1). The second approach is based on the solution of the *state-space* form of (1.1). Either solution should satisfy the constraint equations (1.1b), which requires special care. Solution techniques for multibody systems have been proposed in [5,6,15,18,23,27].

Time integration algorithms for solving structural dynamics problems have been developed since the late 1950s [17]. General requirements and the foundations of these methods have been well-documented [11,12,24,3]. Although their main application area is to linear structural dynamics, these methods can be directly applied to the initial value problems of *nonlinear* second-order ordinary differential equations (ODEs).

$$\ddot{q} = f(\dot{q}, q, t). \quad (1.2)$$

Accuracy and stability analysis hold for the numerical methods, provided the discretized nonlinear equations have been solved accurately, i.e. within a small enough tolerance. For example, the HHT- α method [11] for (1.2) is given by

$$a_{n+1} = (1 + \alpha)f_{n+1} - \alpha f_n \quad (1.3a)$$

$$q_{n+1} = q_n + h v_n + h^2 \left(\left(\frac{1}{2} - \beta \right) a_n + \beta a_{n+1} \right) \quad (1.3b)$$

$$v_{n+1} = v_n + h((1 - \gamma)a_n + \gamma a_{n+1}) \quad (1.3c)$$

where the subscript denotes the corresponding function value at the time, e.g. $q_n = q(t_n)$, $h = t_{n+1} - t_n$ is the time step, $\alpha \in [-\frac{1}{3}, 0]$, $\beta = (1 - \alpha)^2 / 4$, and $\gamma = \frac{1}{2} - \alpha$. It is well-known that the HHT- α family is second-order accurate and A-stable. Numerical dissipation is maximum for $\alpha = -0.3$, and zero for $\alpha = 0$. Controllable numerical dissipation and unconditional stability are needed to deal with the high-frequency modes which often result from standard finite element spatial discretization. For nonlinear oscillations, these properties are also required in the solution of flexible multibody systems. Rather than using ad hoc mode-selection processes, this approach is more desirable because the elimination of higher frequencies is controlled by selection of the parameters.

This paper is concerned with the numerical solution of constrained flexible multibody motion by the family of α -methods. Attempts have been made to apply the HHT- α -method (1.3) directly to (1.1) and its index-2 form [2]. However, these methods are plagued by oscillations which are largely unphysical. Here, we extend the α -method to DAEs in a way which does not introduce additional oscillations. The basic idea of the proposed method is to use the *underlying* ODEs of DAEs and the projection onto the constraint manifold. The main effort is in developing numerical schemes which exploit the structure of multibody dynamic equations, to maintain the stability and reduce the computational cost as compared to a straightforward application. To treat the internal oscillations in the constrained systems, we use a modified Newton-type iteration based on the *Coordinate-Split* formulation of multibody systems [27].

In Section 2, we show the direct application of the α -method to nonlinear multibody DAEs. A projection step to enforce the position and velocity constraints is used for maintaining the stability and accuracy of the numerical solution. A convergence analysis of this approach is given. In Section 3, we describe an effective form of the α -method for constrained flexible multibody equations. In addition, we describe an efficient Newton-type iteration for dealing with the high-frequency oscillations that arise from small elastic deformation. Numerical experiments illustrating the effectiveness of the DAE- α method are given in Section 4.

2. Numerical integration of flexible multibody systems

The basic form of the constrained multibody equations of motion is given by (1.1), which is a DAE of index-3 (see [1] for the definition of index). Due to the problems of numerical instability in solving index-3

DAEs, most of the solution techniques for (1.1) have been developed using differentiation of the constraints (1.1b). Assuming that the kinematic constraint equations are twice differentiable, the analytic solution of (1.1) must satisfy

$$M(q)\ddot{q} + G^T(q, t)\lambda - f(\dot{q}, p, t) = 0 \quad (2.1a)$$

$$G(q, t)\ddot{q} + \frac{\partial G\dot{q}}{\partial q} \dot{q} + 2 \frac{\partial^2 g}{\partial q \partial t} \dot{q} + \frac{\partial^2 g}{\partial t^2} = 0 \quad (2.1b)$$

$$G(q, t)\dot{q} + \frac{\partial g}{\partial t} = 0 \quad (2.1c)$$

$$g(q, t) = 0 \quad (2.1d)$$

where (2.1c) and (2.1b) are the velocity and acceleration constraints, respectively. They are obtained by once and twice differentiating the position constraint (2.1d).

Solving (2.1a) and (2.1b) for $\dot{v} = \ddot{q}$, the underlying ODE of (2.1) is obtained

$$\ddot{q} = \phi(\dot{q}, q, t) = M^{-1}(f - G^T\lambda) \quad (2.2)$$

where $\lambda = (GM^{-1}G^T)^{-1}(GM^{-1}f - \eta)$ and $\eta = -((\partial G\dot{q}/\partial q)\dot{q} + 2(\partial^2 g/\partial q \partial t)\dot{q} + \partial^2 g/\partial t^2)$. For simplicity of notation, we assume that M is invertible. Any conventional numerical integration method can be applied to (2.2), however the numerical solution will not preserve the constraints (2.1c) and (2.1d). To enforce the constraints, the numerical solution can be projected onto the position constraint manifold, which is defined as $\mathcal{M}_q = \{q \mid g(q, t) = 0, \forall t\}$. For an approximation \tilde{q} of $q \in \mathcal{M}_q$, \tilde{q} is projected onto the manifold \mathcal{M}_q via

$$M(q)(q - \tilde{q}) + G^T(q, t)\tilde{v} = 0 \quad (2.3a)$$

$$g(q, t) = 0. \quad (2.3b)$$

This nonlinear system can be solved using Newton-type methods, since its Jacobian is nonsingular near the constraint manifold [10,15]. The approximation \tilde{v} is projected onto the velocity constraint manifold, $\mathcal{M}_v = \{v \mid G(q, t)v = 0, \forall q \in \mathcal{M}_q, \forall t\}$, by solving the linear system

$$M(q)(v - \tilde{v}) + G^T(q, t)\tilde{\mu} = 0 \quad (2.4a)$$

$$G(q, t)v = 0. \quad (2.4b)$$

In (2.3) and (2.4), \tilde{v} and $\tilde{\mu}$ are the Lagrange multipliers associated with the projections. Carrying out the simultaneous solution of (2.2), (2.3) and (2.4), e.g. (q, \tilde{v}) and $(v, \tilde{\mu})$, yields the results (q, v) that are consistent with the analytical solution of (2.1).

REMARK 2.1. The projection of \tilde{q} onto the position constraint manifold has been proposed by many authors [18,23]. A variant of (2.3) was used in the half-explicit extrapolation method [15]

$$M(\tilde{q})(q - \tilde{q}) + G^T(\tilde{q}, t)\tilde{v} = 0,$$

where \tilde{q} is obtained by extrapolation.

Applying the α -method (1.3) to (2.2), and projecting the numerical solution to (2.3) and (2.4) yields

$$a_{n+1} - (1 + \alpha)\phi_{n+1} + \alpha\phi_n = 0 \quad (2.5a)$$

$$M(\tilde{q}_{n+1})(q_{n+1} - \tilde{q}_{n+1}) + G^T(\tilde{q}_{n+1}, t_{n+1})\tilde{v}_{n+1} = 0 \quad (2.5b)$$

$$M(\tilde{q}_{n+1})(v_{n+1} - \tilde{v}_{n+1}) + G^T(\tilde{q}_{n+1}, t_{n+1})\tilde{\mu}_{n+1} = 0 \quad (2.5c)$$

$$G(q_{n+1}, t_{n+1})v_{n+1} = 0 \quad (2.5d)$$

$$g(q_{n+1}, t_{n+1}) = 0 \quad (2.5e)$$

where

$$\tilde{q}_{n+1} = q_n + hv_n + h^2 \left(\left(\frac{1}{2} - \beta \right) a_n + \beta a_{n+1} \right) \quad (2.6a)$$

$$\tilde{v}_{n+1} = v_n + h((1 - \gamma)a_n + \gamma a_{n+1}). \quad (2.6b)$$

The solution of (2.5) consists of the acceleration, velocity and position, e.g. $(a_{n+1}, v_{n+1}, q_{n+1})$, and the multipliers (ν_{n+1}, μ_{n+1}) of the projection. For simplicity, we assume that the kinematic constraints do not depend explicitly on time, e.g. scleronomic constraints. The accuracy and stability of (2.5) can be shown by the standard analysis of (1.3) and Gauss–Newton iteration of the constraints.

THEOREM 2.1. Suppose g is smooth, e.g. $g \in C^2$, $G = \partial g / \partial q$ has full row rank on the position constraint manifold, and GMG^T is invertible. Suppose ϕ in (2.2) is Lipschitz continuous in q and v . There exists h_0 such that for $h \leq h_0$ the solutions q_{n+1} and v_{n+1} of (2.5) satisfy

$$\|q_{n+1} - q(t_{n+1})\| = O(h^2), \quad \|v_{n+1} - v(t_{n+1})\| = O(h) \quad (2.7)$$

given consistent starting values $(q_0, v_0, a_0, \tilde{\nu}_0, \tilde{\mu}_0)$.

PROOF. Applying the numerical integration of (1.3) to the underlying ODE of (2.1), we denote $\tilde{X}_{n+1} = [\tilde{q}_{n+1}, h\tilde{v}_{n+1}, h^2\tilde{a}_{n+1}]$, where

$$\tilde{a}_{n+1} = (1 + \alpha)\phi(\tilde{v}_{n+1}, \tilde{q}_{n+1}, t_{n+1}) - \alpha\phi(\tilde{v}_n, \tilde{q}_n, t_n),$$

and \tilde{q}_{n+1} and \tilde{v}_{n+1} are obtained from (2.6) using \tilde{a}_{n+1} . Linearization of the HHT α -method yields

$$\tilde{X}_{n+1} = A_n \tilde{X}_n, \quad n \in \{0, 1, 2, \dots, N\}, \quad (2.8)$$

where A_i for $i \in \{0, 1, 2, \dots, N\}$ is the local amplification matrix. For a linear function $\phi = M^{-1}Kq$, standard analysis [11] shows that the local truncation error of \tilde{X}_{n+1} is $O(h^3)$, with α, β, γ given in (1.3). Since ϕ is Lipschitz, eigenvalue analysis for the linear model function can be applied to the nonlinear case via the linearization of (2.8), see [2] for details.

Since GMG^T is nonsingular and G has full row rank, locally there exists $P(q)$ such that $P(q)G^T(q) = 0$. Premultiplying (2.5b), (2.5c) by $\tilde{P}_{n+1} = P(\tilde{q}_{n+1})$ yields

$$\begin{pmatrix} \tilde{P}_{n+1} M(\tilde{q}_{n+1}) (q_{n+1} - \tilde{q}_{n+1}) \\ \tilde{P}_{n+1} M(\tilde{q}_{n+1}) (v_{n+1} - \tilde{v}_{n+1}) \end{pmatrix} = 0,$$

which is a state-space ODE of (1.1). The state-space solution together with the constraints (2.5d), (2.5e) is well-defined. In addition, the local error of (q_{n+1}, hv_{n+1}) is the same order as that of $(\tilde{q}_{n+1}, h\tilde{v}_{n+1})$, provided that the nonlinear system is solved with an error tolerance which is less than the truncation error. For more details on the proof, we refer to [25] (Theorem 3, pp. 559–561). Since \tilde{X}_{n+1} is obtained from the second-order α -method, this implies that the local error of $[q_{n+1}, hv_{n+1}]$ is $O(h^2)$, provided that the nonlinear equations are solved with an error tolerance $O(h^3)$. Denoting $X_{n+1} = [q_{n+1}, hv_{n+1}, h^2 a_{n+1}]$, we conclude that the local truncation error of X_{n+1} is $O(h^3)$. It follows that (2.7) holds for $h \leq h_0$, where h_0 depends on the Lipschitz constant of ϕ and on the Jacobian and Hessian of g . \square

REMARK 2.2. The above results hold for many of the numerical integration methods of structural dynamics, including the Newmark β -method [17], and a variety of the α -methods [12,24,3].

Note that the projection \tilde{P}_{n+1} can be replaced by $P_{n+1} = P(q_{n+1})$ in the above analysis [18]. Therefore, we may use $M(q_{n+1})$ and $G^T(q_{n+1})$ in (2.5b), (2.5c) instead of $M(\tilde{q}_{n+1})$ and $G^T(\tilde{q}_{n+1})$.

REMARK 2.3. For the one-step α -methods, the error of the multipliers $(\tilde{\nu}_{n+1}, \tilde{\mu}_{n+1})$ does not depend on accumulated errors of the constraints from the past state variables. Consider the expansion of the constraints $g(\tilde{q}_{n+1})$ about q_n

$$g(\tilde{q}_{n+1}) = g(q_{n+1}) + G(q_{n+1})(q_{n+1} - \tilde{q}_{n+1} + \xi_{n+1})$$

for some $\|G_{n+1}\xi_{n+1}\| = O(\|q_{n+1} - \tilde{q}_{n+1}\|^2)$. Using the projection (2.3), the associated multiplier $\tilde{\nu}_{n+1}$ is given by

$$\tilde{\nu}_{n+1} = (G_{n+1}M_{n+1}^{-1}G_{n+1}^T)^{-1}(g(\tilde{q}_{n+1}) - G_{n+1}\xi_{n+1}). \quad (2.9)$$

Similarly, the projection (2.4) yields the multiplier

$$\tilde{\mu}_{n+1} = (G_{n+1}M_{n+1}^{-1}G_{n+1}^T)^{-1}(G_{n+1}\tilde{\nu}_{n+1}). \quad (2.10)$$

Eq. (2.5) illustrate a straightforward application of the α -method to the constrained multibody systems. However, the computational cost of this approach is relatively large for moderate- to large-scale multibody systems. Each iteration of (2.5) requires the inversion of the mass-inertia matrix M and the formation of f and decomposition of $GM^{-1}G^T$ in the computation of ϕ_{n+1} , e.g. the underlying ODE (2.2). To remedy the high cost of (2.5), we retain the *Lagrange multiplier* λ in (1.1) to avoid the inversion of large matrices. This is achieved because the linear algebra involved in the computation of ϕ_{n+1} is the same as that of the projection steps (2.3) and (2.4). It will be shown in the next section how this approach leads to the DAE α -method which applies the discretization schemes of the α -methods for linear structural dynamics to the index-2 DAEs of the constrained multibody equations of motion.

3. The DAE α -method

One can reduce the computational cost of (2.5) by eliminating the explicit solution for ϕ_{n+1} . Substituting the right-hand side of (2.2) into (2.5a) yields

$$a_{n+1} = (1 + \alpha)M_{n+1}^{-1}(f_{n+1} - G_{n+1}^T\lambda_{n+1}) - \alpha\phi_n \quad (3.1)$$

where $\phi_n = M_n^{-1}(f_n - G_n^T\lambda_n)$. Substituting (3.1) into (2.6) yields

$$\tilde{q}_{n+1} = q_n + hv_n + h^2\left[\left(\frac{1}{2} - \beta\right)a_n - \beta\alpha\phi_n\right] + h^2\hat{\beta}M_{n+1}^{-1}(f_{n+1} - G_{n+1}^T\lambda_{n+1}) \quad (3.2a)$$

$$\hat{v}_{n+1} = v_n + h[(1 - \gamma)a_n - \gamma\alpha\phi_n] + h\hat{\gamma}M_{n+1}^{-1}(f_{n+1} - G_{n+1}^T\lambda_{n+1}) \quad (3.2b)$$

where $\hat{\beta} = \beta(1 + \alpha)$ and $\hat{\gamma} = \gamma(1 + \alpha)$. Substitute (3.2) into (2.5b, 2.5c) to obtain the DAE α -method

$$M_{n+1}(q_{n+1} - \hat{q}_n) - \hat{\beta}h^2f_{n+1} + G_{n+1}^T\nu_{n+1} = 0 \quad (3.3a)$$

$$M_{n+1}(v_{n+1} - \hat{v}_n) - \hat{\gamma}hf_{n+1} + G_{n+1}^T\mu_{n+1} = 0 \quad (3.3b)$$

$$G_{n+1}v_{n+1} = 0 \quad (3.3c)$$

$$g(q_{n+1}) = 0 \quad (3.3d)$$

where

$$\hat{q}_n = q_n + hv_n + h^2\left[\left(\frac{1}{2} - \beta\right)a_n - \beta\alpha\phi_n\right] \quad (3.4a)$$

$$\hat{v}_n = v_n + h[(1 - \gamma)a_n - \gamma\alpha\phi_n], \quad (3.4b)$$

and $a_0 = \phi_0$ and $a_n = (1 + \alpha)\phi_n + \alpha\phi_{n-1}$ for $n \geq 1$. Note that ν_{n+1} and μ_{n+1} in (3.3) comprise $h^2\hat{\beta}\lambda_{n+1}$ and the corresponding *correction* terms by the projections.

Comparing (3.3) to (2.5), the solutions of the state variables q_{n+1} and v_{n+1} differ by the use of two slightly different projections. The solution of (3.3) is obtained from the projection along the direction of $M_{n+1}^{-1}G_{n+1}^T$, while the solution of (2.5) is obtained from the projection along the direction of $\tilde{M}_{n+1}^{-1}\tilde{G}_{n+1}^T$. On a smooth manifold, e.g. $g \in C^2$, the distance between the solutions can be bounded by $O(h)$ for $h \leq h_0$ [18,25]. However, the *algebraic* variables ν_{n+1} and μ_{n+1} contain also the scaled Lagrange multiplier λ of (1.1) which was not accounted for by $\tilde{\nu}_{n+1}$ and $\tilde{\mu}_{n+1}$ in (2.5b) and (2.5c), respectively. Thus, convergence analysis of the *algebraic* variables (ν_{n+1}, μ_{n+1}) remains for the accuracy and stability of the DAE α -method (3.3). This can be established via the results for the application of *stiffly stable* numerical integration methods to index-2 DAEs [9].

THEOREM 3.1. Let the assumptions of Theorem 2.1 hold, and the initial value (q_0, v_0) at t_0 satisfy

$$\|q_0 - q(t_0)\| = O(h^3), \quad \|v_0 - v(t_0)\| = O(h^2)$$

and $g(q_0) = 0$, $G(q_0)v_0 = 0$. The corresponding initial values of the multipliers are

$$\nu_0 = \hat{\beta}h^2\lambda_0, \quad \mu_0 = \hat{\gamma}h\lambda_0,$$

where $\lambda_0 = \lambda(t_0)$ of (1.1). There exists h_0 such that for $h \leq h_0$ the solution of (3.3) satisfies

$$\|q_n - q(t_n)\| = O(h^2), \quad \|v_n - v(t_n)\| = O(h)$$

for all $n \in \{0, 1, 2, \dots, N\}$.

PROOF. The algebraic variables of (3.3) can be written as

$$\nu_{n+1} = \tilde{\nu}_{n+1} + \hat{\beta}h^2\lambda_{n+1},$$

$$\mu_{n+1} = \tilde{\mu}_{n+1} + \hat{\gamma}h\lambda_{n+1}$$

where $\tilde{\nu}_{n+1}$ and $\tilde{\mu}_{n+1}$ are due to the projections of \tilde{q}_{n+1} and \tilde{v}_{n+1} onto the corresponding constraint manifolds, i.e. (2.3) and (2.4), respectively. Using the compact form of (2.8), e.g. $X_{n+1} = A_n X_n$, the discretization of (3.3) can be expressed by

$$q_{n+1} = q_n + hv_n + \frac{h^2}{2}a_n + \delta_{n+1} + G_{n+1}^T \tilde{\nu}_{n+1} \quad (3.5a)$$

$$v_{n+1} = v_n + ha_n + \frac{\gamma}{\beta h} \delta_{n+1} + G_{n+1}^T \tilde{\mu}_{n+1} \quad (3.5b)$$

$$a_{n+1} = a_n + \frac{1}{\beta h^2} \delta_{n+1} \quad (3.5c)$$

where

$$\delta_{n+1} = \beta h^2[(1 + \alpha)M_{n+1}^{-1}(f_{n+1} - G_{n+1}^T \lambda_{n+1}) - \alpha \phi_n - a_n].$$

Applying the test equation $\ddot{q} = -\omega^2 q$ to (3.5), three eigenvalues of the amplification matrix A_n as $h\omega^2 \rightarrow \infty$ are of the magnitudes [2]

$$\left\{ \frac{\alpha}{1 + \alpha}, \frac{1 - \alpha}{1 + \alpha}, \frac{1 - \alpha}{1 + \alpha} \right\}$$

for $\beta = (1 - \alpha)^2/4$ and $\gamma = \frac{1}{2} - \alpha$. Thus, the α -method is stiffly stable for all $\alpha \in [-\frac{1}{3}, 0)$. Convergence of stiffly stable integration methods for index-2 DAEs is well-known [9]. These results yield the convergence of the algebraic variables in (3.5). In addition, for the special case $\alpha = 0$, i.e. the *implicit trapezoidal rule*, the eigenvalues corresponding to $(\tilde{q}_{n+1}, h\tilde{v}_{n+1})$ are 1 and 0. Convergence analysis has also been carried out for this case (for example see [13]).

Using these results and Theorem 2.1, the local errors of ν_{n+1} and μ_{n+1} can be obtained from (2.9) and (2.10), respectively. Consequently, (3.1) implies that

$$\|\lambda_{n+1} - \lambda(t_{n+1})\| = O(h)$$

where $\lambda(t_{n+1})$ is defined by (2.2). For $\|X_n - X(t_n)\| = O(h^3)$, we conclude that

$$\|\tilde{q}_{n+1} - q(t_{n+1})\| = O(h^3), \quad \|\tilde{v}_{n+1} - v(t_{n+1})\| = O(h^2)$$

where \tilde{q}_{n+1} and \tilde{v}_{n+1} are the solutions of (3.2). Thus, the solution of (3.3) satisfies $\|q_{n+1} - q(t_{n+1})\| = O(h^2)$ and $\|v_{n+1} - v(t_{n+1})\| = O(h)$, $\forall n \in 0, 1, \dots, N-1$. \square

3.1. Coordinate-split formulation

Alternatively, we can directly apply the α -method to an index-1 form of (1.1) using nonlinear projection operators, e.g. an annihilation matrix $P(q)$ such that $P(q)G^T(q) = 0$. Premultiplying (3.3a) and (3.3b) by such a matrix yields

$$P_{n+1}(M_{n+1}(q_{n+1} - \hat{q}_n) - \hat{\beta}h^2f_{n+1}) = 0 \quad (3.6a)$$

$$P_{n+1}(M_{n+1}(v_{n+1} - \hat{v}_n) - \hat{\gamma}hf_{n+1}) = 0 \quad (3.6b)$$

$$G_{n+1}v_{n+1} = 0 \quad (3.6c)$$

$$g(q_{n+1}) = 0. \quad (3.6d)$$

Accuracy and stability of the α -methods for ODE are preserved, since the algebraic variables have been annihilated.

There is a potential gain in efficiency for this formulation due to the size-reduction of the nonlinear system, compared to (3.3). *An important practical consequence of (3.6) is that (v, μ) have been eliminated from the DAE.* Thus, the Newton iteration convergence test in a numerical implementation of (3.6) no longer needs to include those multipliers, which can cause problems in the direct numerical solution of (3.3). One could in principle also consider removing (v, μ) from the Newton convergence test in the solution of (3.3), however it is not usually possible to justify this action. Elimination of these variables from the Newton convergence test in the solution of (3.3) can lead to a code which sometimes produces incorrect solutions. It is the fact that multiplying by the *nonlinear* $P(q)$ eliminates (v, μ) from the nonlinear system, which allows these variables to be excluded from the tests in the solution of (3.6). However, the computation of the matrix $P(q)$ and its derivative can be costly if not impossible. We have dealt with this problem by an efficient Newton-type iteration of (3.6) using a family of the CS-matrix $P(q)$ [27].

To obtain a cheap representation of $P(q)$ and its derivative, the CS-operator is computed by the solution of a class of pseudo-inverses of the constraint Jacobian $G(q)$.

DEFINITION 3.1 (Coordinate-Splitting Matrix). Let X and Y be the matrices whose columns constitute the standard basis such that $\|(G(q)Y)^{-1}\|$ is bounded in a neighborhood U_0 of q_0 . The *coordinate-splitting matrix* is defined by

$$P(q) = X^T - Q(q)^T Y^T = X^T(I - G(q)^T(G(q)Y)^{-1}Y^T) \quad (3.7)$$

where $Q(q) = (G(q)Y)^{-1}G(q)X$.

REMARK 3.1. From the construction of the CS matrix $P(q)$, one can easily see that $P(q)G^T(q) = 0$ for all q , i.e. $P(q)$ is orthogonal to range (G^T) . Furthermore, the row vectors of $P(q)$ are orthonormal, i.e. $P(q)^T P(q)$ is the identity.

The computation of $P(q)$ can be carried out using the LU -factorization of the constraint Jacobian matrix. Applying Gaussian elimination with row-pivoting to G^T yields

$$E_m \cdots E_1 G^T = L_m \cdots L_1 U \quad (3.8)$$

where E_i is the elementary permutation and L_i is a Gauss transformation, $i \in \{1, 2, \dots, m\}$. From the factorization, we have

$$[Y, X] = E = E_m \cdots E_1. \quad (3.9)$$

Using the standard solution technique by LU -decomposition, the projected vector $P(q)r$ can be computed in a straightforward and relatively cheap way. In addition, the derivative $(dP(q)r)/dq$ can be computed by the same factorization of G^T and the intermediate result $s = -(GY)^{-1}Y^T r$ from the computation of $P(q)r$.

REMARK 3.2. Alternatively, one can apply QR -factorization to G^T for the computation of $P(q)r$. Using QR -factorization with partial column pivoting [7], we obtain

$$G^T \tilde{E}_1 \cdots \tilde{E}_m = \tilde{L}_1 \cdots \tilde{L}_m \tilde{U} \quad (3.10)$$

where \tilde{E}_i is the elementary permutation, \tilde{L}_i is the Householder matrix, $i \in \{1, 2, \dots, m\}$, and \tilde{U} is upper triangular. The last $(n - m)$ columns of the orthogonal matrix $\tilde{L} = \Pi_{i=1}^m \tilde{L}_i$ constitute a basis for the null-space of G . Thus, we can write

$$[\tilde{Y}, \tilde{X}] = \tilde{L} = \tilde{L}_1 \cdots \tilde{L}_m. \quad (3.11)$$

Note that \tilde{X} and \tilde{Y} are usually subsets of the standard basis in \mathbb{R}^n .

An efficient implementation of Newton-type iteration for (3.6) has been presented in [27], where the Jacobian of the nonlinear system was derived. More importantly, a modification of the Jacobian was obtained to enhance the convergence of Newton-type iterations for high-frequency oscillations. This is achieved by eliminating the term

$$\frac{dP(q)r}{dq} = P(q) \frac{dG^T(q)s}{dq} \equiv 0,$$

where $s = -(GY)^{-1}Y^T r$, from the Jacobian. This term represents the sensitivity of the *potential force* due to the constraints. When the solution is not close to an *equilibrium* on the constraint manifold, this term can be dominant in the Jacobian of the nonlinear system. Consequently, the iterative solution is forced to resolve the high-frequency oscillations if the constraint violation is large compared to the amplitude of the high modes. In highly oscillatory flexible multibody systems, this problem can occur at larger time integration stepsizes. Using this modification of the CS iteration, abbreviated *CM*, the DAE α -method permits a fast convergence of the solution. In addition, the computational cost of the CS-matrix is insignificant compared to one iteration of (3.3) for DAEs with a small number of constraints, such as large-scale flexible multibody systems. We will present several numerical examples involving high-frequency vibrations to illustrate the advantages of the *CM* iteration in the next section.

4. Numerical examples

The DAE HHT α -method in (3.3) can be used for any numerical integration scheme for linear structural dynamics. For the numerical solution of the test problems, we will use the DAE formulas which are extensions of the *generalized α -method* [3]. The generalized α -method for (1.2) is given by

$$(1 - \alpha_m)a_{n+1} + \alpha_m a_n = (1 - \alpha_f)f_{n+1} + \alpha_f f_n, \quad (4.1)$$

where the position and velocity states are defined by (1.3b) and (1.3c), respectively. The parameters of the method are obtained by a user-specified value of high-frequency dissipation $\rho \in [0, 1]$ (i.e. $\rho = 0$ maximum dissipation, $\rho = 1$ none), such that

$$\alpha_m = \frac{2\rho - 1}{\rho + 1}, \quad \alpha_f = \frac{\rho}{\rho + 1},$$

$$\gamma = \frac{1}{2} - \alpha_m + \alpha_f, \quad \beta = \frac{(1 - \alpha_m + \alpha_f)^2}{4}.$$

This yields a second-order, A-stable family of numerical integration methods. Using (4.1) in (3.3) or (3.6), we need only replace (3.4) by

$$\hat{q}_n = q_n + hv_n + h^2 \left[\left(\frac{1}{2} - \frac{\beta}{1 - \alpha_m} \right) a_n - \frac{\beta \alpha_f}{1 - \alpha_m} \phi_n \right] \quad (4.2a)$$

$$\hat{v}_n = v_n + h \left[\left(1 - \frac{\gamma}{1 - \alpha_m} \right) a_n - \frac{\gamma \alpha_f}{1 - \alpha_m} \phi_n \right], \quad (4.2b)$$

and $\hat{\beta} = \beta(1 - \alpha_f)/(1 - \alpha_m)$, $\hat{\gamma} = \gamma(1 - \alpha_f)/(1 - \alpha_m)$. The generalized α -method is parametrized by the size of

the dominant eigenvalue(s) ρ of its amplification matrix $A(-(h\omega)^2)$ as $h\omega \rightarrow 0$. It gives the most comprehensive range of the user-specified value of the spectral radius in the high-frequency limit, while low-frequency dissipation is minimized [3].

4.1. Stiff spring pendulum

To demonstrate the user-specified high-frequency dissipation in the generalized α -method, we use the *stiff pendulum* example, as shown schematically in Fig. 1. The pendulum is attached to a fixed point with a massless spring. The gravitation force is acting on the negative y-axis. In Cartesian coordinates, the equations of motion of this model, with unit mass and gravity, can be written

$$0 = \dot{x} - u \quad (4.3a)$$

$$0 = \dot{y} - v \quad (4.3b)$$

$$0 = \dot{u} + x\lambda \quad (4.3c)$$

$$0 = \dot{v} + y\lambda - 1.0 \quad (4.3d)$$

$$\epsilon^2 \lambda = \frac{\sqrt{x^2 + y^2} - 1.0}{\sqrt{x^2 + y^2}} \quad (4.3e)$$

where the stiff spring of natural length 1.0 and stiffness $1/\epsilon^2$ is attached to the center of mass of the pendulum. Denote by h the stepsize, δ the maximum deflection of the spring, and the potential energy of the system

$$V = \frac{1}{2\epsilon^2} (\sqrt{x^2 + y^2} - 1)^2.$$

Then, the frequency ω of the system is proportional to $\|dV\|$ such that

$$\frac{1}{\epsilon} \sqrt{\frac{\delta}{1+\delta}} \leq \omega \leq \frac{1}{\epsilon} \sqrt{\frac{\delta}{1-\delta}}.$$

As $\epsilon \rightarrow 0$, the dominant pair of eigenvalues approaches $\pm\infty$ along the imaginary axis. This problem is discussed in more detail in [26].

The numerical solution of the index-1 DAE (4.3) has been carried out applying the generalized α -method to (3.3). Using the initial values $(x_0, y_0, u_0, v_0) = (0.9, 0, 0, 0)$, $\epsilon^2 = 10^{-3}$, and Newton iteration with tolerance 10^{-10} , the results of several combinations of the high frequency dissipation ρ and steplength h for a 0 to 5 s simulation are given. The total energy is plotted in Fig. 2, with the high-frequency dissipation parameter ρ varying from 1 to 0. The solution trajectory on the coordinate plane is shown in Fig. 3 with the stepsize varying from 10^{-3} to 10^{-1} . The results illustrate the effective damping of the DAE- α method. As $\rho \rightarrow 0$ or the stepsize h is increased, the numerical solution approaches the equilibrium of the spring while maintaining the slow swing, e.g. see Fig. 3. The controllable damping of the DAE α -method is demonstrated by the stiff spring

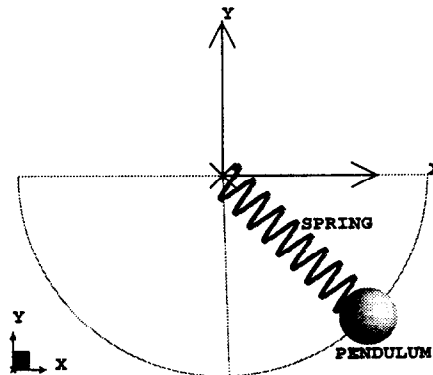
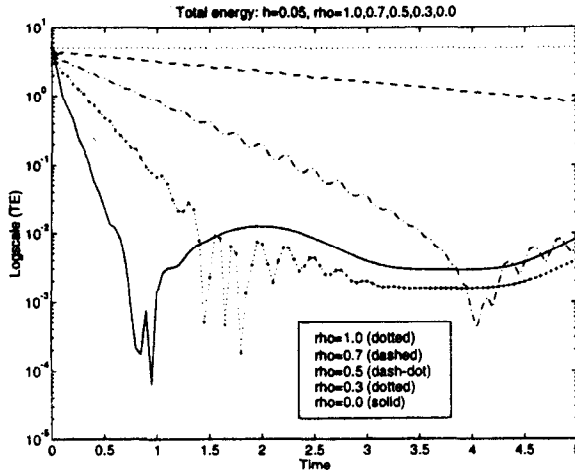
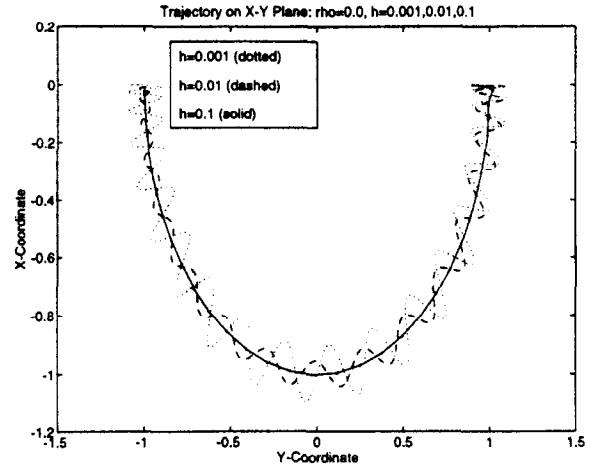


Fig. 1. Stiff spring pendulum.

Fig. 2. Stiff spring pendulum example; $\epsilon^2 = 10^{-3}$, $TOL = 10^{-10}$.Fig. 3. Stiff spring pendulum example; $\epsilon^2 = 10^{-3}$, $TOL = 10^{-10}$.

pendulum example. However, the stepsize may need to be restricted to obtain convergence of the Newton iteration for solving the nonlinear equations at each time step. In the next subsection, we present a nonlinear bushing example, which illustrates that the *CM* iteration applied to (3.6) achieves better convergence than solution of (3.3) via Newton iteration, thus allowing effective damping of high-frequency oscillations by taking larger time steps.

4.2. Nonlinear bushing force

The next example is a nonlinear three-variable bushing example. Using 2-D Cartesian coordinates, the problem may be represented as

$$0 = \ddot{x} - \frac{1}{\epsilon^2} f_x \quad (4.4a)$$

$$0 = \ddot{y} - \frac{1}{\epsilon^2} f_y - 1 \quad (4.4b)$$

$$0 = \ddot{\theta} + \frac{1}{10\epsilon^2} \theta - \frac{1}{2} \left(\cos \theta \left(\frac{1}{\epsilon^2} f_y + 1 \right) - \sin \theta \frac{1}{\epsilon^2} f_x \right) \quad (4.4c)$$

$$0 = x^2 + y^2 - 1 \quad (4.4d)$$

where $f_x = \frac{1}{2} - x + \frac{1}{2} \cos \theta$, $f_y = -y + \frac{1}{2} \sin \theta$ and $\epsilon = 10^{-5}$. This problem was solved using both the stabilized index-2 DAE (3.3) and the *CM* iteration applied to (3.6). Both the formulations work well for the problem. The *CM* iteration permits a larger stepsize and therefore a much quicker dissipation. Some results are compared in Figs. 4 and 5. With the initial values $x = 0.8$, $y = 0.6$ and $\theta = 0.0$ and the tolerance for the Newton convergence 10^{-7} , the figures show the response of the θ variable for different parametric values. The simulation was run for $T = 40\pi \times 10^{-5}$ s, i.e. roughly 6 cycles of θ and around 20 cycles for x and y .

The *CS* iteration is implemented using

$$P = \begin{pmatrix} 1 & -y/x & 0 \\ 0 & 0 & 1 \end{pmatrix}^T,$$

since $x \approx 1$ during the simulation. In Table 1, we list the damping measurement of the solution of the *CM* iteration and that of (3.3). It is measured by the ratio of the initial energy and the energy retained at the end of the simulation. Because a straightforward measurement of damping from the displacement response is difficult, the energy norm has been conveniently used. The last row of Table 1 shows the maximum stepsize that can be

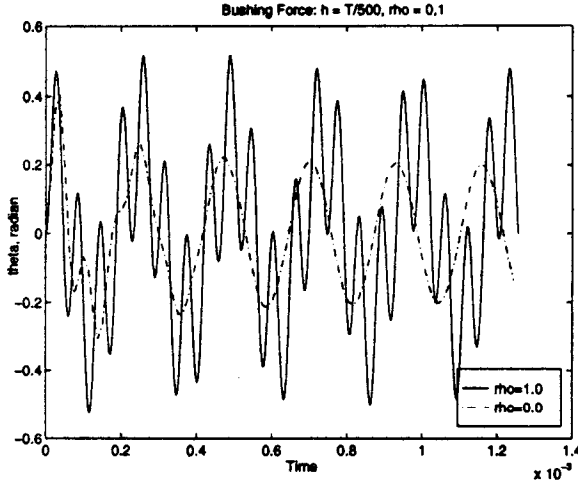
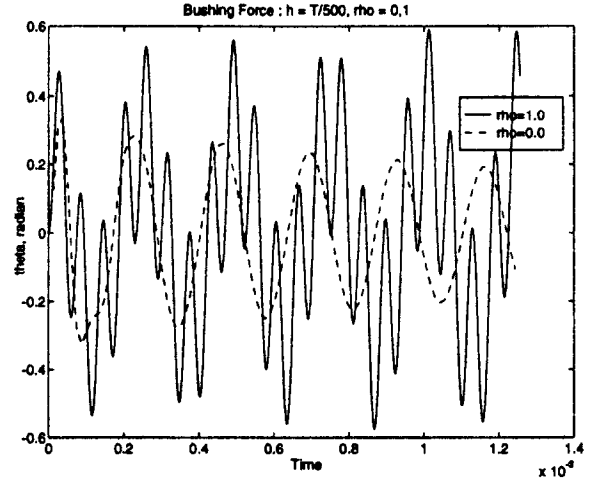
Fig. 4. Bushing example, stabilized index-2 formulation; $\epsilon = 10^{-5}$, $\text{TOL} = 10^{-7}$.Fig. 5. Bushing example, *CM* iteration for 500 timesteps; $\epsilon = 10^{-5}$, $\text{TOL} = 10^{-7}$.

Table 1

Final energy vs. initial energy for the bushing problem; $T = 40\pi \times 10^{-5}$

| No. steps | $\rho = 0$ (<i>CM</i>) | $\rho = 0$ | $\rho = \frac{1}{2}$ (<i>CM</i>) | $\rho = \frac{1}{2}$ | $\rho = \frac{3}{4}$ (<i>CM</i>) | $\rho = \frac{3}{4}$ |
|------------|--------------------------|-----------------|------------------------------------|----------------------|------------------------------------|----------------------|
| 200 | 0.10 | 0.15 | 0.35 | 0.56 | 0.76 | 0.90 |
| 500 | 0.24 | 0.30 | 0.40 | 0.65 | 0.81 | 0.93 |
| 1200 | 0.30 | 0.41 | 0.51 | 0.76 | 0.89 | 0.96 |
| max. h_0 | $\frac{T}{40}$ | $\frac{T}{100}$ | $\frac{T}{70}$ | $\frac{T}{150}$ | $\frac{T}{80}$ | $\frac{T}{175}$ |

taken without causing a failure in Newton convergence. For this example, the *CM* iteration allows stepsizes twice as large as those of the index-2 formulation. In addition, the *CM* solution exhibits more rapid convergence towards the *slow solution*, e.g. approaching a smaller energy level than that of (3.3). This is illustrated in Table 1, where the same h and ρ were used for both *CM* and (3.3) solutions.

4.3. Flexible slider-crank

A flexible slider-crank mechanism is presented schematically in Fig. 6. The mechanism consists of a rigid crank connected with a flexible rod driving a slider on a straight line. Three coordinates (ϕ_1, ϕ_2, x_3) are used for the rigid body equations of motion, and the flexible rod is modeled by a finite element approximation of a linearized Bernoulli–Euler beam. For the details of this model, we refer to [22].

Using a simple beam obtained from the linearization with respect to the vertical displacement by setting the longitudinal displacement to zero, the two constraint equations are

$$g(q) = \begin{pmatrix} l_1 \sin \phi_1 + l_2 \sin \phi_2 + q_{2k+1} \cos \phi_2 \\ x_3 - l_1 \cos \phi_1 - l_2 \cos \phi_2 + q_{2k+1} \sin \phi_2 \end{pmatrix}$$

where l_1 is the length of the crank, l_2 is the length of the rod, q_j are the nodal coordinates induced by the k elements and q_{2k+1} is the nodal coordinate at the *boundary*. Since the constraint involves only the boundary nodes, the computation of $P(q)$ for the *CS* formulation can be reduced to

$$\bar{P}(q) = \bar{X}^T - \begin{pmatrix} 0 & \cos \phi_2 \\ 1 & \sin \phi_2 \end{pmatrix} \begin{pmatrix} l_1 \cos \phi_1 & l_2 \cos \phi_2 - q_{2k+1} \sin \phi_2 \\ l_1 \sin \phi_1 & l_2 \sin \phi_2 + q_{2k+1} \cos \phi_2 \end{pmatrix}^{-1} \bar{Y}^T \quad (4.5)$$

where \bar{X}^T consists of the column vectors e_3 and e_{2k+1} and \bar{Y}^T consists of e_1 and e_2 such that e_j is the j th

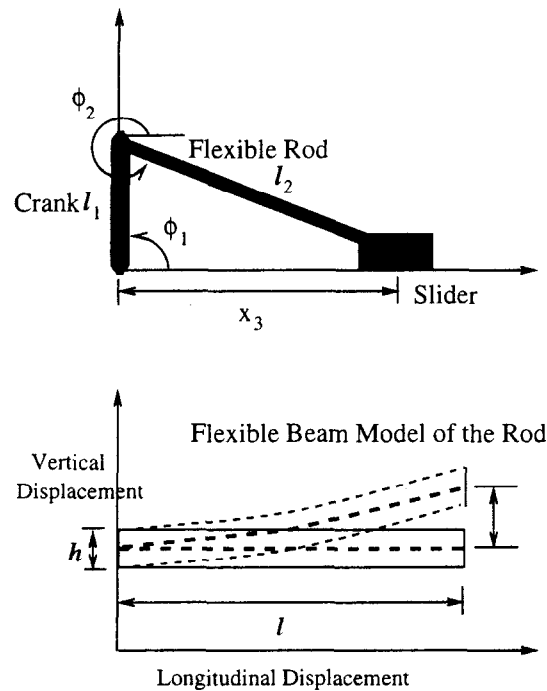


Fig. 6. Slider-crank mechanism with a flexible rod.

standard base of \mathbb{R}^{2k+5} . The matrix $\bar{P}(q)$ is applied directly to the 4×1 vectors of $X_s^T(M(q) - f)$ with $X_s^T = (e_1, e_2, e_3, e_{2k+1})$. The rest of the $2k+1$ equations remain unchanged. The computational cost of the CS-matrix \bar{P} is equivalent to that of \bar{P} . Increasing the number of elements, the computation of \bar{P} remains unchanged.

A one-second simulation was run with a driving torque of 0.2 N-m from 0–0.3 s. The initial velocity and all position values are zero except $x_3 = 0.45$. The Newton iteration tolerance was set to 10^{-9} . The spatial discretization of the flexible rod was carried out using $k = 4$, where the magnitude of the maximum eigenvalue is about 1.3×10^4 . Using (4.5) in the CM iteration, Newton convergence can be attained with larger stepsizes compared to the index-2 DAE form, see Table 2. The high-frequency oscillation of the nodal coordinates is shown in Fig. 7, where the amplitude of the vibrations is small compared to the rigid motion. The numerical solution corresponding to the slow motion, i.e. (ϕ_1, ϕ_2, x_3) , is preserved by the iterations of CM and stabilized index-2 DAE, as shown by the ϕ_2 plots in Fig. 8. The stepsize $h = \frac{1}{2000}$ is about one period of the highest frequency. Applying stronger damping to the high frequency modes, the solution at the midpoint of the flexible rod is presented in Fig. 9 with $\rho = 0.5, 0.0$ and $h = \frac{1}{500}$. The high modes are eliminated in the numerical solution, comparing to Fig. 7. For various combinations of ρ and h , the total energy plots are presented in Figs. 10 and 11. At various levels of damping corresponding to the different values of ρ and h , only the higher modes

Table 2

Maximum stepsize: CM vs. index-2 DAE, $\rho = 0.75$

| Newton solution tolerance | h_0 (CM) | h_0 (index-2 DAE) |
|---------------------------|-----------------|---------------------|
| 10^{-5} | $\frac{1}{100}$ | $\frac{1}{175}$ |
| 10^{-7} | $\frac{1}{250}$ | $\frac{1}{500}$ |
| 10^{-9} | $\frac{1}{400}$ | $\frac{1}{800}$ |

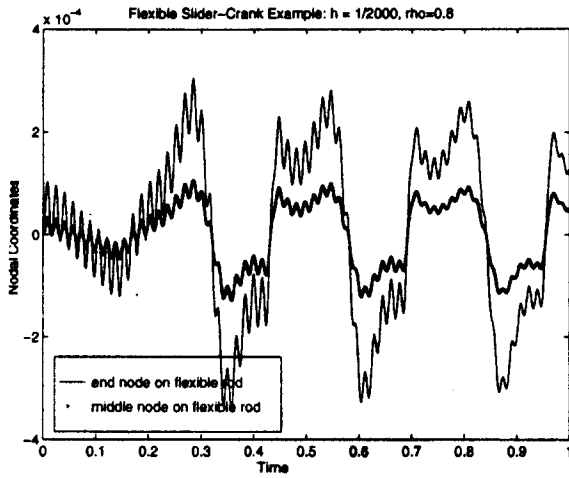


Fig. 7. Nonlinear oscillations in nodal coordinates.

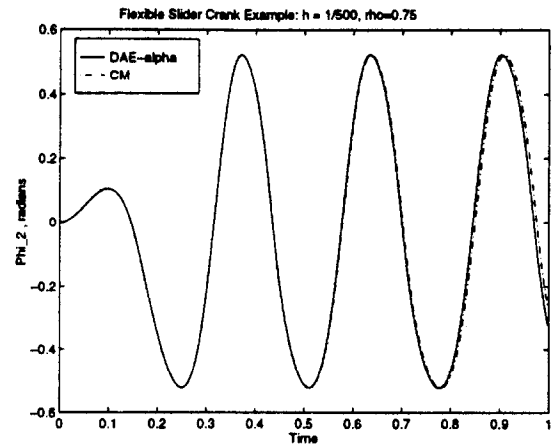


Fig. 8. Rigid motion of coordinate ϕ_2 .

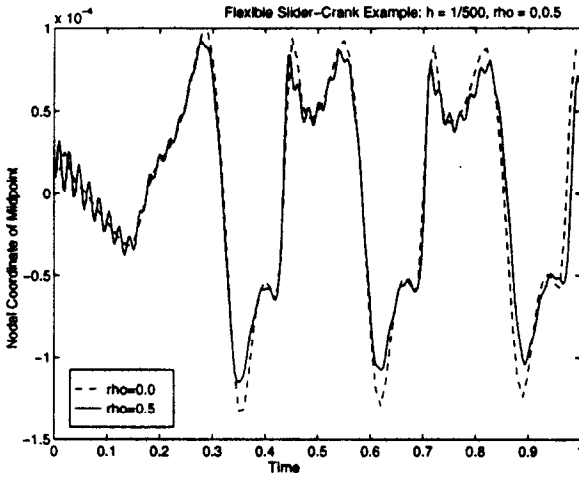


Fig. 9. Damped vibration on midpoint of flexible rod.

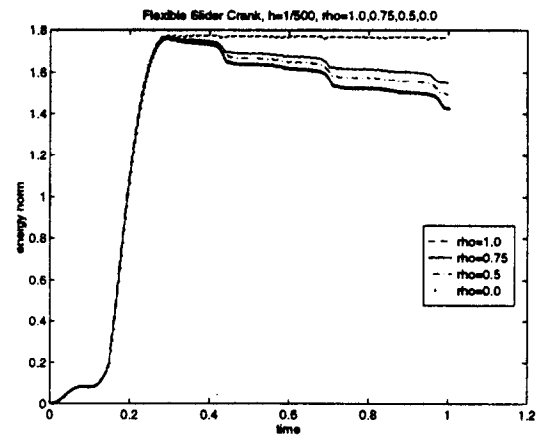


Fig. 10. Total energy comparison, $\rho = 0, 0.5, 0.75$.

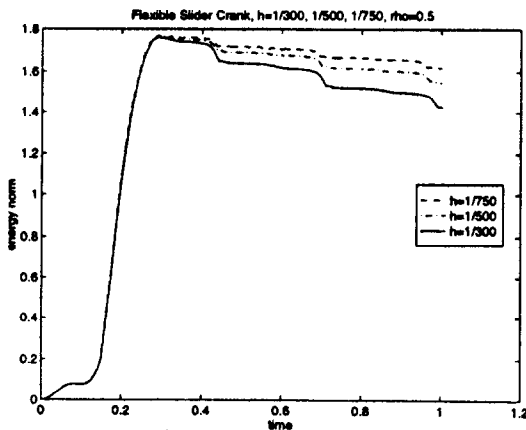


Fig. 11. Total energy comparison, $h = \frac{1}{300}, \frac{1}{500}, \frac{1}{750}$.

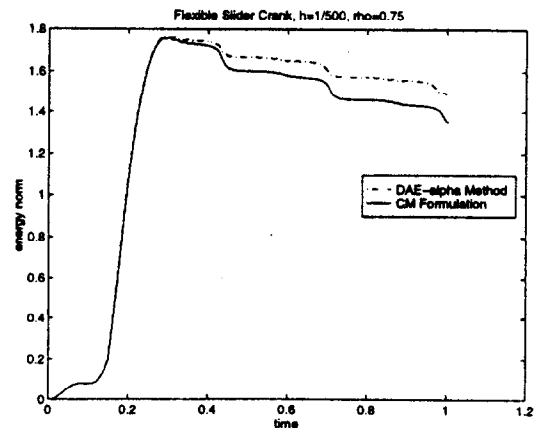


Fig. 12. Total energy comparison, *CM* vs. stabilized index-2 DAE.

are removed in the numerical solution. It is shown in Fig. 12 that faster numerical dissipation of the high modes is achieved via the *CM* iteration, where $h = \frac{1}{500}$ and $\rho = 0.75$.

5. Conclusions

A new family of second-order methods has been introduced for solving the equations of motion for flexible mechanism dynamics. These DAE α -methods possess numerical dissipation that can be controlled by the user. It was shown that the DAE α -methods introduce no additional oscillations and are stable and convergent.

Convergence of the Newton iteration can be a serious problem in solving nonlinear oscillatory mechanical systems with dissipative implicit numerical methods. The source of the difficulties is explained in more detail in [28]. In [27] a modified coordinate-split (*CM*) iteration was introduced which overcomes this problem. Numerical results demonstrate that the DAE α -methods, implemented via the *CM* iteration, offer an effective new tool for the simulation of flexible mechanisms.

We caution that any method which possesses numerical dissipation should be used with care on oscillatory systems. Damping should be attempted only for those oscillations which are not of physical interest or importance.

Acknowledgment

The work was partially sponsored by the Army High Performance Computing Research Center at the University of Minnesota, under the auspices of the Department of Army, Army Research Laboratory cooperative agreement numbers DAAH04-95-2-0003/contract number DDAH04-95-C-0008, and by ARO contract number DAAH04-94-6-0208 and DAAH04-94-6-0213, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

References

- [1] K.E. Brenan, S.L. Campbell and L.R. Petzold, Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, 2nd edition (SIAM, 1995).
- [2] A. Cardona and M. Géradin, Time integration of the equations of motion in mechanical analysis, *Comput. Struct.* 33 (1989) 801–820.
- [3] J. Chung and G.M. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- α method, *ASME J. Appl. Mech.*, No. 93-APM-20 (1993).
- [4] R.R. Craig, Jr., *Structural Dynamics—An Introduction to Computer Methods* (Wiley, 1981).
- [5] C. Führer and B.J. Leimkuhler, Numerical solution of differential-algebraic equations for constrained mechanical motion, *Numer. Math.* 59 (1991) 55–69.
- [6] C.W. Gear, G.K. Gupta and B.J. Leimkuhler, Automatic integration of the Euler–Lagrange equations with constraints, *J. Comput. Appl. Math.* 12 (1985) 77–90.
- [7] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd edition (Johns Hopkins University Press, 1989).
- [8] E. Griepentrog and R. März, *Differential-Algebraic Equations and their Numerical Treatment* (Teubner, Leipzig, 1986).
- [9] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems* (Springer-Verlag, Berlin, 1991).
- [10] E.J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems, Vol. I: Basic Methods* (Allyn-Bacon, 1989).
- [11] H.H. Hilber, T.J.R. Hughes and R.L. Taylor, Improved numerical dissipation for time integration algorithms in structural dynamics, *Earthquake Engrg. Struct. Dyn.* 5 (1977) 283–292.
- [12] C. Hoff and P.J. Pahl, Development of an implicit method with numerical dissipation from a generalized single-step algorithm for structural dynamics, *Comput. Methods Appl. Mech. Engrg.* 67 (1988) 367–385.
- [13] L. Jay, Convergence of a class of Runge–Kutta methods for differential-algebraic systems of index 2, *BIT* 33 (1993) 137–150.
- [14] T.R. Kane and D.A. Levinson, *Dynamics: Theory and Applications* (McGraw-Hill, New York, 1985).
- [15] Ch. Lubich, Extrapolation integrators for constrained multibody systems, *IMPACT of Comput. Sci. Engrg.* 3 (1991) 213–234.
- [16] Ch. Lubich, Integration of stiff mechanical systems by Runge–Kutta methods, *ZAMP* 44 (1993) 1022–1053.
- [17] N.M. Newmark, A method of computation for structural dynamics, *ASCE J. Engrg. Mech. Div.* 85 (1959) 67–94.
- [18] F. Potra and W.C. Rheinboldt, On the numerical solution of the Euler–Lagrange equations, *J. Mech. Struct. Mach.* 19 (1991) 1–18.
- [19] M.S. Pereira and J.A.C. Ambrósio, eds., *Computer Aided Analysis of Rigid and Flexible Mechanical Systems*, NATO—Advanced Study Institute, Tróia, Portugal, 27 June–9 July, 1993.

- [20] R.E. Roberson and R. Schwertassek, *Dynamics of Multibody Systems* (Springer, Berlin, 1988).
- [21] W. Schiehlen, ed., *Advanced Multibody System Dynamics* (Kluwer Academic Publishers, Stuttgart, 1993).
- [22] B. Simeon, Modelling of a flexible slider crank mechanism by a mixed system of DAEs and PDEs, *Math. Model. Syst.* 2 (1996) 1–18.
- [23] R.A. Wehage and E.J. Haug, Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, *ASME J. Mech. Des.* 134 (1982) 247–255.
- [24] W.L. Wood, M. Bossak and O.C. Zienkiewicz, An alpha modification of Newmark's method, *Int. J. Numer. Methods Engrg.* 15 (1980) 1562–1566.
- [25] J. Yen, Constrained equations of motion in multibody dynamics as ODEs on manifolds, *SINUM* 30 (1993) 553–568.
- [26] J. Yen and L.R. Petzold, Numerical solution of nonlinear oscillatory multibody dynamic systems, in: D.F. Griffiths and G.A. Watson, eds., *Proc. 1995 Biennial Conference on Numerical Analysis*, Pitman Research Notes in Mathematics, Vol. 344 (Addison Wesley Longman Ltd., 1996) 209–225.
- [27] J. Yen and L.R. Petzold, An efficient Newton-type iteration for the numerical solution of highly oscillatory constrained multibody dynamic systems, *SIAM J. Scient. Comput.*, to appear.
- [28] L.R. Petzold, L.O. Jay and J. Yen, Numerical solution of highly-oscillatory ordinary differential equations, *Acta Numerica* (1997) 437–483.
- [29] W.S. Yoo and E.J. Haug, Dynamics of articulated structures, Part I theory, *J. Struct. Mech.* 14(1) (1986) 105–126.