SIAM J. SCI. COMPUT.
Vol. 19, No. 5, pp. 1513–1534, September 1998

007

AN EFFICIENT NEWTON-TYPE ITERATION FOR THE NUMERICAL SOLUTION OF HIGHLY OSCILLATORY CONSTRAINED MULTIBODY DYNAMIC SYSTEMS*

JENG YEN† AND LINDA R. PETZOLD‡

Abstract. In this paper we present a coordinate-split (CS) technique for the numerical solution of the equations of motion of constrained multibody dynamic systems. We show how the CS technique can be implemented within the context of commonly used solution methods, for increased efficiency and reliability.

A particularly challenging problem for multibody dynamics is the numerical solution of highly oscillatory nonlinear mechanical systems. Highly stable implicit integration methods with large stepsizes can be used to damp the oscillation, if it is of small amplitude. However, the standard Newton iteration is known to experience severe convergence difficulties which force a restriction of the stepsize. We introduce a modified coordinate-split (CM) iteration which overcomes these problems. Convergence analysis explains the improved convergence for nonlinear oscillatory systems, and numerical experiments illustrate the effectiveness of the new method.

Key words. initial value problems, differential-algebraic equations, multibody systems, highly oscillatory systems, error bound

AMS subject classifications. 65L05, 65L70

PII. S1064827596297227

1. Introduction.

The equations of motion of a constrained multibody system can be written as [10]

$$\dot{q} - v = 0, \tag{1.1a}$$

$$M(q)\dot{v} + G^T(q)\lambda - f(v, q, t) = 0, \tag{1.1b}$$

$$g(q) = 0, \tag{1.1c}$$

where $q = [q_1, q_2, ..., q_n]$ are the *generalized coordinates*, $\lambda = [\lambda_1, \lambda_2, ..., \lambda_m]$ are the *Lagrange multipliers*, $M(q) \in \mathbb{R}^{n\times n}$ is the mass-inertia matrix, $f \in \mathbb{R}^n$ is the force applied to the system, $v = \frac{dq}{dt}$ is the velocity, and $\dot{v} = \frac{d^2q}{dt^2}$ is the acceleration vector. The constraints $g = [g_1, g_2, ..., g_m]$ are m twice-differentiable functions of q, whose Jacobian,

$$G(q) = \left[\frac{\partial g_i}{\partial q_j}\right] \in \mathbb{R}^{m\times n}\ ,\ m \le n, \tag{1.2}$$

* Received by the editors September 15, 1996; accepted for publication (in revised form) March 26, 1997; published electronically April 22, 1998. This research was partially sponsored by the Army High Performance Computing Research Center under the auspices of the Department of Army, Army Research Laboratory cooperative agreement DAAH04-95-2-0003/contract DDAH04-95-C-0008, and by ARO contract DAAH04-94-6-0208.

http://www.siam.org/journals/sisc/19-5/29722.html

† Department of Computer Science and Army High Performance Computing Research Center, University of Minnesota, Minneapolis, MN 55415. Current address: Department of System Engineering, United Defense, LP, 4800 East River Road, Minneapolis, MN 55421 (yen@cs.umn.edu).

‡ Department of Computer Science and Army High Performance Computing Research Center, University of Minnesota, Minneapolis, MN 55455. Current address: Department of Mechanical and Environmental Engineering, Engr. 2 Building, Room 2355, University of California, Santa Barbara, Santa Barbara, CA 93106-5070 (petzold@engineering.ucsb.edu). The work of this author was partially sponsored by ARO contract DAAL03-92-G-0247, DOE contract DE-FG02-92ER25130, and the Minnesota Supercomputer Institute.

is assumed to be of full row rank. We assume that $G(q)M(q)G^T(q)$ is symmetric and positive definite for every $q \in \mathbb{R}^n$ to obtain a consistent physics represented by (1.1). The system (1.1) has $n-m$ degrees of freedom. Equation (1.1) is a well-known index-3 DAE [3, 13].

Many methods have been proposed for modeling multibody systems. Direct numerical integration of the index-3 DAE (1.1) suffers from the well-known difficulties inherent in the solution of high-index DAEs [13]. One way to lower the index involves introducing derivatives of the constraint $g(q)$, along with additional Lagrange multipliers μ. This yields the *stabilized index-2* or *Gear–Gupta–Leimkuhler (GGL)* formulation of the constrained equations of motion [8]

$$\dot{q} - v + G^T(q)\mu = 0, \tag{1.3a}$$

$$M(q)\dot{v} + G^T(q)\lambda - f(v, q, t) = 0, \tag{1.3b}$$

$$G(q)v = 0, \tag{1.3c}$$

$$g(q) = 0, \tag{1.3d}$$

which has been widely used in simulation. The Lagrange multiplier variables λ and μ fulfill the role of projecting the solution onto the *position* (1.3d) and the *velocity* (1.3c) constraints, respectively. Equations (1.3) and related systems have been solved by a variety of methods. Here we will consider solution by implicit numerical methods such as backward differentiation formulas (BDF) or implicit Runge–Kutta methods (in particular, RADAU). A closely related approach is based on explicitly projecting the numerical solution onto the constraints [19, 22, 25, 26] and involves many of the same issues for the implementation that are considered here.

Many of the numerical methods for multibody systems solve the system (1.3) directly. It is also possible to eliminate the Lagrange multipliers and reduce the size of the system to the number of degrees of freedom. One way to accomplish this begins with the stabilized index-2 system (1.3). Suppose that $G(p)$ is full-rank on the manifold $\mathcal{M} = \{q \in \mathbb{R}^n \mid g(q) = 0\}$. Then one can find an annihilation matrix $P(q) \in \mathbb{R}^{(n-m)\times n}$ such that $P(q)G^T(q) = 0 \ \forall q \in \mathcal{M}$. Premultiplying (1.3a) and (1.3b) by $P(q)$ yields an index-1 DAE

$$P(q)(\dot{q} - v) = 0, \tag{1.4a}$$

$$P(q)(M(q)\dot{v} - f(v, q, t)) = 0, \tag{1.4b}$$

$$G(q)v = 0, \tag{1.4c}$$

$$g(q) = 0. \tag{1.4d}$$

There is a potential gain in efficiency for this formulation due to the size-reduction of the nonlinear system, compared to (1.3). *An important practical consequence of* (1.4) *is that* (μ, λ) *have been eliminated from the DAE, via multiplication of* (1.3a), (1.3b) *by the nonlinear* $P(q)$. Thus, the error test and Newton iteration convergence

test in a numerical implementation of (1.4) no longer need to include (μ, λ). These higher-index variables can cause problems in the direct numerical solution of (1.3). One could in principle also consider removing (μ, λ) from the test in the solution of (1.3), however, it is not usually possible to justify this action, particularly in the case of the Newton convergence test. Elimination of these variables from the Newton convergence test in the solution of (1.3) can lead to a code which sometimes produces incorrect solutions. It is the fact that multiplying by the *nonlinear* $P(q)$ eliminates (μ, λ) from the nonlinear system which allows these variables to be excluded from the tests in the solution of (1.4).

Direct numerical solution of (1.4) presents some challenges. First we must have a means of generating $P(q)$ which is reliable and cheap. Further, we note that the Jacobian matrix for the Newton iteration involves terms which arise from the derivatives of $P(q)$. We need a means of generating the Jacobian matrix. Finally, practical issues such as the error test and Newton convergence test must be considered. Under these considerations, our approach to obtaining a representation of $P(q)$ is based on a *coordinate-splitting* of the variables. This family of projections can be computed by applying a common matrix factorization, such as LU-decomposition, to the constraint Jacobian. Without an additional matrix decomposition, the same factorization is reused to generate the system Jacobian of (1.4). The coordinate-splitting also permits a natural modification to the regular error test in the DAE solver, which leads to an efficient solution of (1.3).

A widely used method which is related in the sense of also making use of a splitting of the coordinates is the generalized coordinate partitioning method [25]. In order to obtain the independent generalized coordinates, the Jacobian $G(q)$ is calculated and an LU-decomposition of G is carried out to identify a nonsingular $m \times m$ submatrix GY, where the generalized coordinate q is partitioned into $q = Xx + Yy$ such that $X \in \mathbb{R}^{n\times p}$ and $Y \in \mathbb{R}^{n\times m}$. The columns of X and Y constitute the standard basis for $\mathbb{R}^n$. The implicit function theorem assures that the dependent variables y can be uniquely determined by the constraint $g(q)$, at least locally, as a function of the remaining generalized coordinates x, i.e., $y = h(x)$. Variables in x thus are declared to be independent coordinates. The twice-differentiable function $h : \mathbb{R}^{(n-m)} \to \mathbb{R}^m$ satisfies $g(x, h(x)) = 0$ for arbitrary x in some neighborhood of q. Differentiating twice the constraints (1.1c) with respect to time, then multiplying (1.1b) by GM^{-1} and substituting for $G\dot{v}$ from the twice-differentiated constraints to solve for λ, we obtain

$$\lambda(q, v) = (GM^{-1}G^T)^{-1}\left(GM^{-1}f + \frac{dGv}{dq}v\right).$$

Substituting the solution $\lambda(q, v)$ into (1.1), and eliminating the dependent variables y, $\dot{y}$, and $\ddot{y}$, yields n-m differential equations

$$\hat{M}(x, h(x))\ddot{x} = \hat{f}\left(\dot{x}, \frac{dh}{dx}\dot{x}, x, h(x), t\right), \tag{1.5}$$

which is a *state-space form* representation of (1.1) [7, 14]. However, this differs substantially from the approach we outline here because $P(q)$ *associated with generalized coordinate partitioning is not orthogonal to* $G^T(q)$. *Hence the index-reduction by differentiating the constraints and projecting to the invariant space must be carried out explicitly.* In particular, this requires forming the derivative of the velocity constraints (i.e., the acceleration constraints) explicitly. A method that is closely related to the

generalized coordinate partitioning method has been proposed in [9, 19, 20, 21, 22], where the projector is chosen to be an orthonormal basis of the local tangent space of the constraint manifold.

Direct numerical solution of (1.4) via our coordinate-split (CS) approach yields an efficient and reliable method for solving equations of motion for many multibody mechanical systems. This approach is advantageous in part because it allows the Lagrange multiplier variables to be safely removed from the Newton convergence test. This is particularly important in the solution of highly oscillatory mechanical systems, where the Lagrange multiplier variables are very badly behaved; they act like the second derivatives of the highly oscillatory position variables.

Multibody systems with high-frequency nonlinear oscillations are an important and challenging computational problem. Highly oscillatory components are often used to model devices with strong potential energy. Typical examples of such problems arise from modeling flexible multibody mechanical, and molecular dynamic systems. For many problems, oscillations of a sufficiently small amplitude are not important for the model, but they severely restrict the stepsize for numerical methods. For these types of problems, stiffly stable implicit numerical integration methods can be used to damp out the oscillation [17, 24]. However, the stepsize may still be severely restricted due to difficulties in converging the Newton iteration for larger stepsizes. We have studied this class of oscillating problems in [27, 18]. The solutions are composed of a *low-amplitude* high-frequency oscillation around the *smooth solution* [17, 23]. Along the smooth solution, the eigenstructure of the local Jacobian matrix varies smoothly. However, along the solutions which are nearby to the smooth solution, the local eigenstructure oscillates with the high frequency and is very badly behaved. The standard Newton iteration inside a damping numerical method starts from a *predictor* which is on a nearby solution, and attempts to find the smooth solution. It evaluates its Jacobian matrix on the nearby solution, which determines the direction it takes toward the smooth solution. Unfortunately, these Jacobian matrices do not yield good directions for nonlinear oscillating problems as described above, unless the predictor is already extremely close to the smooth solution. Thus, the standard Newton method must be coupled with a severe reduction in the time step to achieve an adequate predictor. We emphasize that a damping numerical method in this situation *finds the smooth solution*. If a detailed approximation to the oscillation is desired, then another type of method, with small stepsizes to track the oscillation, should be used.

In section 2, we outline the CS iteration carried out by an efficient method for $P(q)$ and its derivative. For the purpose of finding the smooth solution, we introduce a modification to the Newton iteration, i.e., the modified coordinate-split (CM) iteration. This iteration is easy to implement, effective for nonoscillatory problems, and particularly effective for nonlinear highly oscillatory problems. The basic idea of the CM iteration is that there are terms in the Jacobian which involve derivatives of the projection onto the constraint manifold. These terms are large and complicated to compute, but small on the *slow manifold* [23]. The CM iteration sets these terms to zero, yielding a reliable direction towards the smooth solution for the Newton-type iteration. We prove the convergence of the CM iteration and give an error estimate for the numerical solution. A damping numerical method, together with the CM modification to the Newton iteration which greatly improves the rate of convergence, and the CS technique which allows the Lagrange multiplier variables to be safely removed from the Newton convergence test, offer an efficient technique for computing the smooth solution of highly oscillatory multibody DAE systems. In section 3 we

describe in more detail the structure of nonlinear oscillatory mechanical systems and derive estimates for the rates of convergence of the CS and CM iterations applied to these oscillatory systems. The difference in convergence rate explains why the CM iteration is highly effective for oscillatory systems, and shows that its rate of convergence for nonoscillatory systems is similar to that of the CS iteration. In section 4, numerical experiments are given which demonstrate the effectiveness of these methods, particularly for oscillatory nonlinear mechanical systems.

2. The CS technique. In this section we present the CS technique, which defines $P(q)$ in (1.4a) and (1.4b) via coordinate-splitting, and computes this matrix cheaply. Although at first glance it would appear that implementation of this method would be difficult due to complications in computing the derivatives of $P(q)$, we show that the special form of the pseudoinverse can be used to give a much simpler derivation of the Jacobian.

The construction of the annihilation matrix $P(q)$ involves the computation of a class of pseudoinverses of the constraint Jacobian $G(q)$. Given a smooth vector-valued function $r(q)$, an effective way to obtain the projected vector $P(q)r$ is to use a splitting of the original coordinates.

DEFINITION 2.1 (CS matrix). *Let X and Y be the matrices whose columns constitute the standard Cartesian basis of $\mathbb{R}^{n\times n}$ such that $\|(G(q)Y)^{-1}\|$ is bounded in a neighborhood U_0 of some q_0, which is on the manifold $\mathcal{M} = \{q \in \mathbb{R}^n \mid g(q) = 0\}$. The $p \times n$* CS matrix *for* (1.1) *is defined by*

$$P(q) = X^T - Q(q)^T Y^T = X^T(I - G(q)^T(G(q)Y)^{-T}Y^T), \tag{2.1}$$

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

Remark 2.1. Note that X and Y are piecewise constant with respect to $q \in \mathcal{M}$. From the construction of the CS matrix $P(q)$, one can easily see that $P(q)G^T(q) = 0$ for all $q \in \mathbb{R}^n$; i.e., $P(q)$ is orthogonal to range(G^T). Furthermore, the row vectors of $P(q)$ are orthonormal; i.e., $P(q)^T P(q) = I_p$ where I_p is the identity matrix in $\mathbb{R}^p$.

The computation of $P(q)$ can be carried out using the LU-factorization or QR-factorization of the constraint Jacobian matrix [12]. Then the projected vector $P(q)r$ can be computed in a straightforward way. In addition, directly applying the formulas given in [11] (Theorem 4.3, pp. 420), we obtain

$$\left(\frac{dP(q)}{dq}\right) r = P(q)\left(\frac{dG^T(q)}{dq}\right) s, \text{ with } s = -(GY)^{-T}Y^T r, \tag{2.2}$$

where $P(q)$ is defined by (2.1) and $r(q) \in \mathbb{R}^{n\times 1}$.

Using (2.1) and (2.2), the CS iteration for solving the nonlinear system at each time step can be carried out. Applying, for example, a kth-order BDF formula to (1.4) yields the nonlinear system

$$P(q_n)h(\rho_h q_n - v_n) = 0, \tag{2.3a}$$

$$P(q_n)h(M(q_n)\rho_h v_n - f(v_n, q_n, t_n)) = 0, \tag{2.3b}$$

$$G(q_n)v_n = 0, \tag{2.3c}$$

$$g(q_n) = 0, \tag{2.3d}$$

where ρ_h is the discretization operator, $\rho_h q_n = \frac{1}{h}\sum_{i=0}^{k} \alpha_i q_{n-i}$, with α_i the coefficients of the BDF method and h the stepsize of the time discretization. Given an initial prediction $(q_n^{(0)}, v_n^{(0)})$, applying Newton-type methods to (2.3) requires the solution of a linear system

$$J(q_n, v_n)(\Delta q_n, \Delta v_n) = -r(q_n, v_n) \tag{2.4}$$

such that Δq_n and Δv_n are the increments of q_n and v_n,

$$J(q_n, v_n) = \begin{bmatrix} P(q_n)[(\frac{\partial G^T(q_n)}{\partial q_n})s_1 + h\frac{\partial \rho_h q_n}{\partial q_n}] & -hP(q_n) \\ P(q_n)[(\frac{\partial G^T(q_n)}{\partial q_n})s_2 + \frac{\partial r_2(q_n,v_n)}{\partial q_n}] & P(q_n)\frac{\partial r_2(q_n,v_n)}{\partial v_n} \\ \frac{\partial (G(q_n)v_n)}{\partial q_n} & G(q_n) \\ G(q_n) & 0 \end{bmatrix}, \tag{2.5}$$

and

$$r(q_n, v_n) = [P_n r_1, P_n r_2, G_n v_n, g_n]$$

where $s_1 = -(GY)^{-T}Y^T r_1$, $s_2 = -(GY)^{-T}Y^T r_2$, $r_1 = h(\rho_h q_n - v_n)$, and $r_2 = h(M(q_n)\rho_h v_n - f(v_n, q_n, t_n))$. In our implementation, the Jacobian J may be fixed for several steps. The strategies of DASSL are used to decide when a new evaluation of J is needed.

To analyze the solution of (2.4), we rewrite the first two equations of (2.4); i.e., corresponding to the derivatives of (2.3a) and (2.3b),

$$\begin{bmatrix} P(q_n) & 0 \\ 0 & P(q_n) \end{bmatrix} \left(J_h(q_n, v_n) \begin{bmatrix} \Delta q_n \\ \Delta v_n \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \right) = 0, \tag{2.6}$$

where the $2n \times 2n$ matrix J_h is

$$J_h(q_n, v_n) = \begin{bmatrix} h\frac{d\rho_h q_n}{dq_n} + (\frac{dG_n^T}{dq_n})s_1 & -hI \\ h(\frac{dM(q_n)}{dq_n})\rho_h v_n - h\frac{\partial f_n}{\partial q_n} + (\frac{dG_n^T}{dq_n})s_2 & hM\frac{d\rho_h v_n}{dv_n} - h\frac{\partial f_n}{\partial v_n} \end{bmatrix}. \tag{2.7}$$

Since for small enough h, J_h is invertible under the assumption of $M(q_n)$ nonsingular, the solution of (2.4) is well defined. To simplify the notation, (2.4) is rewritten as[1]

$$\begin{bmatrix} \mathcal{P}_n J_h \\ \mathcal{G}_n \end{bmatrix} \begin{bmatrix} \Delta q_n \\ \Delta v_n \end{bmatrix} = -r_n, \tag{2.8}$$

[1] To contrast the CS formulation to the generalized coordinate partitioning technique [25], we note that generalized coordinate partitioning applies the discretization formula directly to (1.5). This requires explicitly forming and satisfying the acceleration constraints, and solving for the Lagrange multipliers. The CS iteration (2.8) does not directly use the acceleration equations or solve for the intermediate solutions of λ. This is advantageous because it eliminates the work of forming the acceleration equations and because in highly oscillatory mechanical systems, λ can be very badly behaved and is a source of numerical difficulties.

where

$$\mathcal{P}_n = \begin{bmatrix} P_n & 0 \\ 0 & P_n \end{bmatrix},$$

$$\mathcal{G}_n = \begin{bmatrix} \frac{\partial (G_n v_n)}{\partial q} & G_n \\ G_n & 0 \end{bmatrix},$$

$$r_n = \begin{bmatrix} P_n r_1 \\ P_n r_2 \\ G_n v_n \\ g_n \end{bmatrix}.$$

We remark that the CS iteration leads to a natural and reliable error estimator for the numerical integration method. In particular, the local error estimator can be based on the independent coordinates and velocities only, since the increments of the dependent variables $\Delta y_n \equiv Y \Delta q_n$ and $\Delta z_n \equiv Y \Delta v_n$ are bounded by $\Delta x_n \equiv X \Delta q_n$ and $\Delta w_n \equiv X \Delta v_n$, e.g.,

$$\|\Delta y_n\| \leq \|(G_n Y)^{-1}\|(\|G_n X\|\|x_n\| + \|g_n\|),$$

$$\|\Delta z_n\| \leq \|(G_n Y)^{-1}\|(\|G_n X\|\|w_n\| + \|\eta_n\|),$$

where g_n is the residual of (2.3d) and $\eta_n = \frac{d(G v_n)}{dq} \Delta q_n + G_n v_n$. For small enough $\|g_n\|$ and $\|\eta_n\|$, the convergence of (2.4) implies that $(G_n Y)$ is well conditioned. Thus, the modified error estimate is a reliable indicator of the local errors in q_n and v_n [26].

The CS iteration is an efficient formulation of the Newton iteration for multibody systems, so standard convergence theorems apply. We will now state conditions to ensure the convergence of the CS iteration in the context of numerical integration methods for (1.4). For simplicity we consider, instead of the GGL formulation of the constrained equations of motion (1.3), the structurally similar system

$$\dot{q} - f(q,t) + G^T(q)\lambda = 0, \tag{2.9a}$$

$$g(q) = 0, \tag{2.9b}$$

since the convergence of numerical methods applied to (2.9) can be trivially extended to (1.3). Applying stiffly stable numerical methods, the convergence result is well known; see [13, pp. 494–498]. Convergence of discretization methods for the index-1 system,

$$P(q)(\dot{q} - f(q,t)) = 0, \tag{2.10a}$$

$$g(q) = 0, \tag{2.10b}$$

obtained by applying the CS matrix $P(q)$ to (2.9a), is also well developed. By the construction of $P(q)$, it is easy to see that the solution of the CS iteration is equivalent to that of the local *state-space ODE* of the independent coordinate x [7].

Applying a linear discretization operator ρ_h with stepsize h to (2.10) and scaling (2.10a) by h yields a nonlinear system

$$\phi(q) = \begin{bmatrix} P(q)r_h(q) \\ g(q) \end{bmatrix}, \tag{2.11}$$

where the *residual* function is

$$r_h(q,t) = h(\rho_h(q) - f(q,t)). \tag{2.12}$$

The convergence of the CS iteration can be carried out on a smooth constraint manifold $\mathcal{M}$. Assume that for any $q_0 \in \mathcal{M}$, there exist h_0 and $X \in \mathbb{R}^{p\times n}$ and $Y \in \mathbb{R}^{m\times n}$ such that

$$\begin{aligned} A = \begin{bmatrix} P(q)J_h \\ G(q) \end{bmatrix} \text{ is invertible, } \ \|A^{-1}\| \le C_0, \\ \|(G(q)J_h^{-1}Y)^{-1}\| \le C_1 \ \ \forall h \le h_0, \end{aligned} \tag{2.13}$$

where $J_h = \frac{\partial r_h(q)}{\partial q} + (\frac{dG^T}{dq})s$, $s = -(GY)^{-T}Y^T r$, for some $C_0 > 0$, and $C_1 > 0$. Similarly, we assume that

$$\|G(q_1)^T - G(q_2)^T\| \le C_2\|q_1 - q_2\| \tag{2.14}$$

for some C_2, where q_1 and q_2 are in a neighborhood $U(q_0)$ of q_0.

Convergence of the CS iteration can be assured for a sufficiently accurate initial guess. Under the assumptions (2.13) and (2.14), a sufficient condition for convergence requires that the numerical integration satisfy

$$\left\| \left(\frac{dr_h}{dq} + \left(\frac{dG^T}{dq} \right) s \right)^{-1} \right\| = \|J_h^{-1}\| \le C_3 \tag{2.15}$$

for some $C_3 > 0$ in a neighborhood of the true solution q^* [6].

For highly oscillatory dynamic systems, the stepsize of a numerical method may be restricted by the error test to follow the oscillation. Applying a stiffly stable numerical method to (2.10), such as BDF of order ≤ 2, one may be able to damp the oscillation and take a larger stepsize to follow the trajectory of the equilibrium, i.e., $f - G^T s = 0$. However, convergence of the Newton iteration requires a further restriction on the stepsize. Depending on how close the predictor is to the equilibrium of highly oscillatory components, the Newton direction imposed by the Jacobian can excite the high-frequency oscillations. When applying the Newton method directly to the discretization of the Lagrangian form (2.9), an even more severe problem in Newton convergence is observed, and is illustrated by the numerical experiments in section 4. The limitation on the stepsize due to the Newton convergence failures for highly oscillatory nonlinear multibody systems can be overcome via a modification to the CS iteration which we call the CM iteration.

The CM iteration. In large-scale multibody mechanical systems, most of the unwanted oscillations are due to the *noise* of high-frequency forces, where the amplitude is well below the solution tolerance. However, small perturbations in the position can cause drastic changes in the Newton direction. This results in difficulties for convergence of Newton-type methods. To remedy this problem in the CS iteration, we

reduce the noise from the oscillations by setting $(\frac{dP(q)}{dq})r = 0$ in the Newton iteration matrix, since it is the main source contributing to the rapidly changing Newton direction. This term is large away from the smooth solution but negligible on the smooth solution. The CM iteration approximates the Newton direction of (2.10) via an oblique projection to the unconstrained ODE

$$P(q_0)(\dot{q} - f(q,t)) = 0 \tag{2.16}$$

for a q_0 close to the solution q, e.g., $G(q_0)G^T(q)$ invertible. When applying a stiffly stable numerical integrator to highly oscillatory problems, this modification, for q_0 close enough to the smooth solution, overcomes the difficulties in the CS iteration.

Applying a stiffly stable discretization method to (2.16) coupled with constraint (2.10b) leads to the nonlinear system

$$\phi_0(q) = \begin{bmatrix} P(q_0)r_h(q) \\ g(q) \end{bmatrix}. \tag{2.17}$$

A convergence result for the modified CS iteration, denoted by CM, is given next.

We first give an upper bound on the difference between the derivative of the projected vector $P(q)r(q)$ and the projected derivative $P(q)(\frac{dr(q)}{dq})$.

LEMMA 2.1. *Suppose conditions* (2.13) *and* (2.14) *hold. Then for* $q \in D(q_0, \varrho_0) \subseteq U(q_0)$*, where* $D(q_0, \varrho_0)$ *is the disc in* $\mathbb{R}^n$ *with center* q_0 *and radius* ϱ_0*,*

$$\left\| \frac{d}{dq}[P(q)r(q)] - P(q)\frac{dr(q)}{dq} \right\| \leq \varrho_0 C_1 C_2 \|Y^T r(q)\|. \tag{2.18}$$

Proof. From (2.2),

$$\left\| \frac{d}{dq}[P(q)r(q)] - P(q)\left(\frac{dr(q)}{dq}\right) \right\| = \left\| P(q)\left(\frac{dG(p)^T}{dq}\right)s \right\|.$$

Since the row vectors of $P(q)$ are p orthonormal vectors in $\mathbb{R}^n$, applying the Cauchy inequality gives

$$\left\| P(q)\left(\frac{dG(p)^T}{dq}\right)s \right\| \leq \left\| \left(\frac{dG(q)^T}{dq}\right)s \right\| \leq \varrho_0 C_2 \|(GY)^{-T}Y^T r(q)\|$$

for all $q \in D(q_0, \varrho_0) \subseteq U(q_0)$. Condition (2.13), with $h \to 0$, implies the result in (2.18). ∎

Convergence of the CM iteration and an estimate of the distance between the solutions of (2.11) and (2.17) are given in the following.

THEOREM 2.1. *Suppose conditions* (2.13)–(2.15) *hold for some* h_0*, such that* $\{q_j\} \to q^*$ *is generated by the CS iteration. Choosing* $\bar{q}_0 = q_0$*, the sequence* $\{\bar{q}_k\}$ *generated by the CM iteration*

$$\bar{q}_{k+1} = \bar{q}_k - \bar{J}(\bar{q}_0)^{-1}\phi_0(\bar{q}_k), \tag{2.19}$$

where $\bar{J} = \frac{d\phi_0}{dq}$*, converges to* $\bar{q}^*$*. Furthermore, the distance between* $\bar{q}^*$ *and* q^* *is bounded above by*

$$\|\bar{q}^* - q^*\| \leq C(\|r_h(q^*)\|\|q^* - q_0\| + \|q^* - q_0\|^2 + \|\bar{q}^* - q_0\|^2) \tag{2.20}$$

for some moderate constant C.

Proof. Convergence of $\{\bar{q}_k\}$ follows immediately by noting that the CM iteration is an inexact Newton method [5] with the property that the terms which the CM modification deletes from the Jacobian, hence the perturbation to the Newton iteration, can be made as small as we want by taking the initial guess accurate enough.

Next, we show the estimate (2.20). Since the CS and CM iterations converge to q^* and $\bar{q}^*$, respectively, we have

$$0 = \phi(q^*) = \phi_0(\bar{q}^*).$$

Subtracting $\left[\begin{smallmatrix} P(q_0)r_h(q^*) \\ 0 \end{smallmatrix}\right]$ from both sides of the above equation we obtain

$$\begin{bmatrix} (P(q^*) - P(q_0))r_h(q^*) \\ g(q^*) \end{bmatrix} = \begin{bmatrix} P(q_0)(r_h(\bar{q}^*) - r_h(q^*)) \\ g(\bar{q}^*) \end{bmatrix}.$$

Expanding around q_0 yields

$$\begin{aligned} P(q_0)\bar{J}_h(q_0)(\bar{q}^* - q^*) = -\frac{1}{2}P(q_0)\left(\frac{d\bar{J}_h\bar{\zeta}}{dq}\bar{\zeta} - \frac{d\bar{J}_h\zeta}{dq}\zeta\right) + \left(\frac{dP}{dq}\right)\zeta r_h(q^*) \\ + \frac{1}{2}\left(\frac{d^2P}{dq^2}\right)\zeta^2 r_h(q^*), \end{aligned}$$

where $\bar{\zeta} = \bar{q}^* - q_0$, $\zeta = q^* - q_0$, and $\bar{J}_h = \frac{d}{dq}r_h(q)$. In matrix form, the above equations become

$$\bar{J}(q_0)(\bar{q}^* - q^*) = \begin{bmatrix} \frac{dP}{dq}r_h\zeta + O(\bar{\zeta}^2 + \zeta^2) \\ O(\bar{\zeta}^2 + \zeta^2) \end{bmatrix}.$$

By the assumption that $\bar{J}$ is invertible, we can write

$$\|\bar{q}^* - q^*\| \le C(\|r_h(q^*)\|\|q_0 - q^*\| + \|q^* - q_0\|^2 + \|\bar{q}^* - q_0\|^2). \qquad \square$$

Note that since $\frac{dP}{dq} \equiv 0$ and $\frac{dG}{dq} \equiv 0$ when the constraints $g(q)$ are linear, $\{q_k\}$ of the CS iteration and $\{\bar{q}_k\}$ of the CM iteration are the same in that case. In general, the rate of convergence of the CM iteration is *superlinear*, using the Dennis–Moré characterization theorem [6].

3. Rate of convergence for highly oscillatory multibody systems. High-frequency oscillatory forces often appear in the modeling of vehicle suspension systems, modal analysis in structural dynamics, or modeling of oscillations in computer-aided engineering, etc. For simplicity, we consider the constrained dynamic system of (1.1) with a dominant oscillatory force

$$M(q)\dot{v} + G^T\lambda + \frac{1}{\epsilon}\eta(q) - f(v, q, t) = 0, \tag{3.1a}$$

$$g(q) = 0, \tag{3.1b}$$

where $\frac{1}{\epsilon}$ may be, for example, the coefficients of stiff springs; i.e., $0 < \epsilon \ll 1$. In practice, $\eta(q)$ is usually oblique towards $\mathrm{Ker}P(q)$; i.e., the oscillatory force(s) acts on both the independent and the dependent coordinates. For the purpose of obtaining a

smooth solution with large stepsizes, we will show that the CM iteration can be very effective for many classes of nonlinear oscillatory forces.

Deformation forces are the most common potential forces that can produce small-amplitude high-frequency oscillations. They are usually linear with respect to the local coordinates [4, 28]. For these reasons, we consider as a model the class of oscillatory forces of the form

$$\eta(q,t) = B(t)(q - b_0(t)), \tag{3.2}$$

where the components of B and b_0 are slowly varying. We note that in the more complicated situation where B is nonlinear with respect to q, there may be restrictions on the stepsize in addition to those which will be derived in the following.

Using a linear oscillatory force, the Lagrange equations of motion of the multibody system can be written as

$$M(q)\dot{v} + \frac{1}{\epsilon}B(q - b_0) + G^T\lambda - f(v,q,t) = 0 \tag{3.3}$$

where $\frac{1}{\epsilon} \gg \|\frac{\partial f}{\partial(q,v)}\|$. From assumption (2.14) on the constraint manifold, we can also assume that

$$\frac{1}{\epsilon} \gg \max_{\|u_1\|,\|u_2\|=1} \left\| \frac{dG(q)u_1}{dq} u_2 \right\| \tag{3.4}$$

for all q.

In the context of the CS iteration, the problem of convergence of the Newton iteration can be explained by analyzing the *reduced potential* function. The reduced potential of (3.1) is

$$V^r(q) = g(q)^T(GY)^{-T}Y^T r \tag{3.5}$$

where $r = f - M\ddot{q} - \frac{1}{\epsilon}B(q - b_0)$. The reduced potential force generated by (3.5) is

$$\nabla V^r(q) \equiv \frac{dV^r}{dq} = G^T(GY)^{-T}Y^T r. \tag{3.6}$$

At each iteration, the reduced potential force acts along the normal direction of the constraint manifold. The gradient of the correction term yields

$$\nabla^2 V^r(q) = (I - G^T(GY)^{-T}Y^T)\left(\frac{dG^T(q)}{dq}\right)s \tag{3.7}$$

where $s = (GY)^{-T}Y^T r$. Applying Y^T to (3.7) gives

$$Y^T\nabla^2 V^r(q) = Y^T(I - G^T(GY)^{-T}Y^T)\left(\frac{dG^T(q)}{dq}\right)s = 0,$$

and applying X^T to (3.7) yields

$$X^T\nabla^2 V^r(q) = P(q)\left(\frac{dG^T(q)}{dq}\right)s.$$

When high-frequency oscillations appear in the system, e.g., $\epsilon \to 0$, the reduced potential force also becomes oscillatory if $Y^T r$ is nonzero. This is the general case

when the solution is not at an equilibrium position. Nevertheless, convergence of the CS iteration can be achieved by using a small enough stepsize [17].

For small stepsizes, a convergence result for the CM iteration can be obtained provided the assumptions of Theorem 2.1 are valid. In many applications, following the oscillations is *not* of interest. Instead, one wants to use a large time step to damp out the oscillations of small amplitude but high frequency. For this reason, we now consider only the multistep numerical integration methods that are *strictly stable* at infinity and A-stable, such as the lower order (i.e., ≤ 2) BDF methods [13]. The convergence of L-stable implicit Runge–Kutta methods to the smooth solution of the highly oscillatory ODE of multibody mechanical systems can be found in [17]. Here we focus on the convergence of the CM iteration for constrained multibody systems with oscillatory forces when applying the above-mentioned linear multistep methods.

Numerical solutions on the slow manifold can be evaluated using the equilibrium of (3.1), i.e., the *slow* solution [2, 15] satisfies

$$\eta(q) - \epsilon(f(v,q) - \nabla V^r(q) - M(q)\dot{v}) = 0,$$

and the smooth solution is its asymptotic expansion to some order of ϵ around the manifold $\{q \mid \eta(q) = 0\}$. In the linear form, the smooth solution of (3.1) is not far from $B(q - b_0) = 0$ since $\frac{1}{\epsilon} \gg \|\frac{\partial f}{\partial(q,v)}\|$. For the strongly damped numerical solution q_n, $B(q_n - b_0) \to O(\epsilon)$ as $t_n \to \infty$. During the iterative solution onto the slow manifold, the constraints may not be satisfied, which causes a large reaction force in the form of (3.6). This may cause oscillations in the CS iteration, while the CM iteration annihilates these nonlinear oscillations generated by the reduced potential. This yields a superior performance of the CM iteration as compared to the CS iteration for computing the smooth solution of (3.1). The result is explained in the following.

THEOREM 3.1. *Consider the numerical solution of* (3.1)*, with $\eta(q)$ linear, and h the stepsize of the stiffly stable multistep integration method. Assume the starting values (q_0, v_0) lie within a δ_0-neighborhood of the smooth solution. If both the CS and CM iterations converge to q^* and $\bar{q}^*$, respectively, then the rate of convergence of the CS iteration $\sigma^{(CS)}$ compared to that of the CM iteration $\sigma^{(CM)}$ is given by*

$$\sigma^{(CS)} = \frac{h}{\epsilon} O(\delta_k) + O(\delta_k) + O(h),$$

where $\delta_k = \|[\begin{smallmatrix} q_k \\ v_k \end{smallmatrix}] - [\begin{smallmatrix} q^* \\ v^* \end{smallmatrix}]\|$, and

$$\sigma^{(CM)} = O(\bar{\delta}_k) + O(h),$$

where $\bar{\delta}_k = \|[\begin{smallmatrix} \bar{q}_k \\ \bar{v}_k \end{smallmatrix}] - [\begin{smallmatrix} \bar{q}^* \\ \bar{v}^* \end{smallmatrix}]\|$. When the constraints are linear, the two iterations yield the same result.

Proof. Recall that the CS iteration is a Newton iteration for the solution of

$$\phi(q,v) = [P(q)r_1, P(q)r_2, G(q)v, g(q)]^T = 0,$$

where $r_1 = h(\rho_h q - v)$ and $r_2 = hM(q)\rho_h v + \frac{h}{\epsilon}\eta(q) - hf$. Using the notation defined in (2.8), and comparing the Jacobian matrix for the CS iteration at iteration k with the Jacobian at the solution of the CS iteration, we obtain

$$\begin{aligned} J - J^* &= \begin{bmatrix} \mathcal{P}(q_k)J_h(q_k,v_k) \\ \mathcal{G}(q_k,v_k) \end{bmatrix} - \begin{bmatrix} \mathcal{P}(q^*)J_h(q^*,v^*) \\ \mathcal{G}(q^*,v^*) \end{bmatrix} \\ &= \begin{bmatrix} (\mathcal{P}(q_k) - \mathcal{P}(q^*))J_h(q^*,v^*) + \mathcal{P}(q_k)(J_h(q_k,v_k) - J_h(q^*,v^*)) \\ \mathcal{G}(q_k,v_k) - \mathcal{G}(q^*,v^*) \end{bmatrix}. \end{aligned} \tag{3.8}$$

For equation (3.1) solved by a multistep method with leading coefficient β using the CS iteration,

$$J_h = \begin{bmatrix} \beta I + K_1(q_k, v_k) & -hI \\ h\beta \frac{dM(q_k)}{dq}\rho_h v_k + \frac{h}{\epsilon}B(t) + K_2(q_k, v_k) - h\frac{\partial f}{\partial q}(v_k, q_k) & \beta M(q_k) - h\frac{\partial f}{\partial v}(v_k, q_k) \end{bmatrix}, \tag{3.9}$$

where

$$\begin{bmatrix} K_1(q_k, v_k) \\ K_2(q_k, v_k) \end{bmatrix} = \begin{bmatrix} (\frac{dG^T(q_k)}{dq})s_1(q_k, v_k) \\ (\frac{dG^T(q_k)}{dq})s_2(q_k, v_k) \end{bmatrix},$$

and $s_1 = -(GY)^{-T}Y^T r_1$, $s_2 = -(GY)^{-T}Y^T r_2$. Thus, from (3.8), the local rate of convergence of the CS iteration is given, via the contraction mapping theorem [6], by

$$\sigma^{CS} = \|I - J^{-1}J^*\| = \|J^{-1}(J - J^*)\| = \frac{h}{\epsilon}O(\delta_k) + O(\delta_k) + O(h).$$

Now, recall that the CM iteration is a Newton iteration for the solution of

$$\phi_0(q, v) = [P(q_0)r_1, P(q_0)r_2, G(q)v, g(q)]^T = 0.$$

Comparing the Jacobian matrix for the CM iteration at iteration k with the Jacobian at the solution of the CM iteration, we obtain

$$\begin{aligned} \bar{J} - \bar{J}^* &= \begin{bmatrix} \mathcal{P}(q_0)\bar{J}_h(\bar{q}_k, \bar{v}_k) \\ \mathcal{G}(\bar{q}_k, \bar{v}_k) \end{bmatrix} - \begin{bmatrix} \mathcal{P}(q_0)\bar{J}_h(\bar{q}^*, \bar{v}^*) \\ \mathcal{G}(\bar{q}^*, \bar{v}^*) \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{P}(q_0)(\bar{J}_h(\bar{q}_k, \bar{v}_k) - \bar{J}_h(\bar{q}^*, \bar{v}^*)) \\ \mathcal{G}(\bar{q}_k, \bar{v}_k) - \mathcal{G}(\bar{q}^*, \bar{v}^*) \end{bmatrix}, \end{aligned} \tag{3.10}$$

where for problem (3.1),

$$\bar{J}_h = \begin{bmatrix} \beta I & -hI \\ h\beta \frac{dM(\bar{q}_k)}{dq}\rho_h \bar{v}_k + \frac{h}{\epsilon}B(t) - h\frac{\partial f}{\partial q}(\bar{v}_k, \bar{q}_k) & \beta M(\bar{q}_k) - h\frac{\partial f}{\partial v}(\bar{v}_k, \bar{q}_k) \end{bmatrix}. \tag{3.11}$$

Noting that in a $\bar{\delta}_k$-neighborhood of the smooth solution,

$$\bar{J}_h(\bar{q}^*, \bar{v}^*) = \begin{bmatrix} \beta I & -hI \\ h\beta \frac{dM(\bar{q}^*)}{dq}\rho_h \bar{v}^* + \frac{h}{\epsilon}B(t) - h\frac{\partial f}{\partial q}(\bar{v}^*, \bar{q}^*) & \beta M(\bar{q}^*) - h\frac{\partial f}{\partial v}(\bar{v}^*, \bar{q}^*) \end{bmatrix},$$

equation (3.10) yields

$$\bar{J} - \bar{J}^* = \begin{bmatrix} \mathcal{P}(q_0)(\bar{J}_h(\bar{q}_k, \bar{v}_k) - \bar{J}_h(\bar{q}^*, \bar{v}^*)) \\ \mathcal{G}(\bar{q}_k, \bar{v}_k) - \mathcal{G}(\bar{q}^*, \bar{v}^*) \end{bmatrix} = O(\bar{\delta}_k) + O(h).$$

Thus,

$$\sigma^{CM} = O(\bar{\delta}_k) + O(h).$$

By a similar argument, this result holds for all linear numerical integration methods. □

4. Numerical experiments.

4.1. Point-mass with oscillatory force. The first example is a simple constrained multibody system under the influence of a highly oscillatory force. Consider a unit point-mass constrained to the two-dimensional unit circle, using $q = [x, y]^T$, the velocity $v = \frac{dq}{dt} = [w, z]^T$, and the constraint equation $g(q) = \frac{1}{2}(x^2 + y^2 - 1)$. The equations of motion are

$$\begin{aligned} \dot{w} + x\lambda - f^x &= 0, \\ \dot{z} + y\lambda - f^y &= 0, \end{aligned}$$

where $f = [f^x, f^y]^T$ is the applied force. Differentiating the constraint $g(q) = \frac{1}{2}(x^2 + y^2 - 1)$ twice with respect to time, an explicit form of the multiplier λ is obtained by

$$\lambda = \frac{1}{x^2 + y^2}(xf^x + yf^y + w^2 + z^2).$$

For a highly oscillatory force $f(q)$, one can see that $\lambda(t)$ is oscillating with the frequency of $f(q)$, and with amplitude proportional to the magnitude of $\|f\|$.

The numerical experiments are carried out using BDF of order ≤ 2 in DASSL [3], where the local error estimation has been modified. For the stabilized index-2 DAE (1.3) denoted by GGL, the local error is estimated using only the position, i.e., q. Thus, when oscillations in the position variables are of amplitude smaller than the error tolerance, DASSL will attempt to increase the stepsize. Moreover, we have also included some experiments where the Newton convergence test of GGL has been modified to exclude the multipliers. The corresponding numerical solution is denoted by GGL*. For the CS and CM iterations, the local error is estimated using the independent variable $X^T q$, as recommended in [27]. The CM iteration updates the matrix $P(\hat{q})$ when a new Jacobian is required.

Linear oscillation. Let a unit gravitational force act along the negative y-direction, and apply a linear oscillatory force

$$f = \begin{bmatrix} \frac{1}{\epsilon}x \\ \frac{1}{\epsilon^2}(y+1) - 1.0 \end{bmatrix}.$$

There is a stable *equilibrium* at $q = [0, -1]$ and $v = [0, 0]$. The natural frequency of the system is $\omega = \frac{1}{\epsilon}$, and no dissipative force is present.

The numerical solution has been carried out with a moderate solution tolerance $ATOL = RTOL = TOL = 10^{-3}$. For a 0- to 0.25-second simulation, the results of several combinations of the stiffness coefficient ϵ are presented in Table 4.1, where the initial values are $q = [0.0, -1.0]$ and $v = [1.0, 0]$. The CM iterations show better efficiency than those of CS, GGL, and GGL* in all cases, i.e., comparing the numbers of function and Jacobian evaluations in Table 4.1. In the table, $etfs$ and $ctfs$ denote the number of failures of the error test and Newton convergence test, respectively, in DASSL. Comparing the results of GGL* with those of GGL, we observe an improved Newton convergence. As $\epsilon \to 0$, i.e., for higher frequencies of the oscillation, the CM iteration becomes even more efficient. In Figure 4.1, we plot the total energy of each numerical solution. The CM iteration achieves the strongest damping because DASSL is able to increase the stepsize faster with the CM iteration.

Linear spring force. In the next test, we replace the linear oscillatory force in the previous constrained system by a spring force

$$\frac{1}{\epsilon^2}\frac{l - l_0}{l}\begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix},$$

TABLE 4.1
Results of the constrained point-mass with a linear oscillatory force.

Method	TOL	ϵ	*No. steps*	*No. fevals*	*No. jevals*	*No. etfs*	*No. ctfs*
GGL	10^{-3}	10^{-4}	446	1713	263	47	0
GGL^*	10^{-3}	10^{-4}	478	1304	127	45	0
CS	10^{-3}	10^{-4}	381	1036	106	46	0
CM	10^{-3}	10^{-4}	373	963	96	38	0
GGL	10^{-3}	10^{-5}	431	1726	261	34	2
GGL^*	10^{-3}	10^{-5}	500	1324	139	52	2
CS	10^{-3}	10^{-5}	447	1281	116	56	2
CM	10^{-3}	10^{-5}	456	1180	115	58	1
GGL	10^{-3}	10^{-6}	427	1693	275	41	8
GGL^*	10^{-3}	10^{-6}	478	1247	123	44	4
CS	10^{-3}	10^{-6}	414	1139	119	38	5
CM	10^{-3}	10^{-6}	325	844	81	34	1

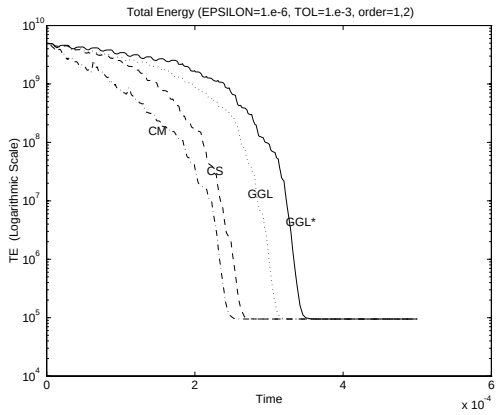

FIG. 4.1. *Total energy comparison of linear oscillatory force example*; $\epsilon = 10^{-6}$ *and* $TOL = 10^{-3}$.

where $l = \sqrt{(x-x_0)^2 + (y-y_0)^2}$ for (x_0, y_0), the attachment point of the spring, l_0 the natural length, and $\frac{1}{\epsilon^2}$ is the stiffness coefficient, as shown schematically in Figure 4.2. For unit mass and unit gravitational force, we set the spring attached to $(x_0, y_0) = (0, -0.5)$ and the natural length $l_0 = 0.4$, such that the equilibrium is at $(0, -1, 0, 0)$.

Using the initial conditions $[0.04471, -0.999, 0, 0]$, the results of the 0- to 0.05-second simulation by the GGL, GGL*, CS, and CM iterations are shown in Table 4.2. Because DASSL is able to increase the stepsize, and hence damp the solution faster with the CM iteration, the CM method is quite effective in these tests. In Figure 4.3, we plot the total energy of each solution. The numerical solutions of x, w, and λ are presented in Figure 4.4.

4.2. Two-body pendulum with bushing. The second example is a two-body pendulum in two-dimensional Cartesian coordinates. Six generalized coordinates, $q = [x_1, y_1, \theta_1, x_2, y_2, \theta_2]^T$, locate the centers of mass and the orientation of the bodies. The first body is grounded, and the second body is constrained such that the distance between a point A of the first body and another point B of the second body is fixed, and its orientation is held constant. This leads to five constraint equations:

TABLE 4.2
Results of the constrained point-mass with an oscillatory linear spring force.

Method	TOL	ϵ	$No.\ steps$	$No.\ fevals$	$No.\ jevals$	$No.\ etfs$	$No.\ ctfs$
GGL	10^{-3}	10^{-4}	109	383	56	13	0
GGL^*	10^{-3}	10^{-4}	109	337	45	13	0
CS	10^{-3}	10^{-4}	150	474	58	20	0
CM	10^{-3}	10^{-4}	72	208	31	7	0
GGL	10^{-3}	10^{-5}	105	416	68	11	0
GGL^*	10^{-3}	10^{-5}	105	387	47	12	0
CS	10^{-3}	10^{-5}	135	455	56	18	0
CM	10^{-3}	10^{-5}	66	205	32	4	0
GGL	10^{-4}	10^{-4}	822	2534	337	78	0
GGL^*	10^{-4}	10^{-4}	818	2572	242	82	0
CS	10^{-4}	10^{-4}	943	2727	217	66	0
CM	10^{-4}	10^{-4}	193	577	56	14	0

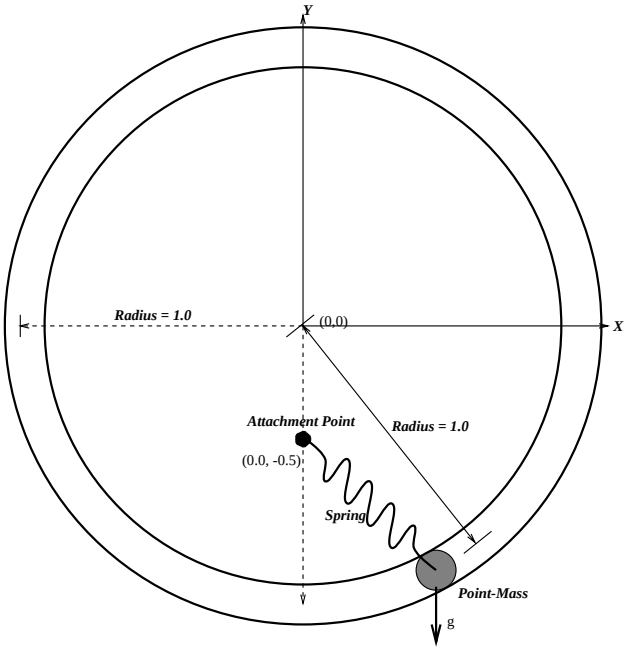

FIG. 4.2. *Constrained point-mass with a linear spring.*

$$g_1 = x_1, \tag{4.1a}$$

$$g_2 = y_1, \tag{4.1b}$$

$$g_3 = \theta_1, \tag{4.1c}$$

$$g_4 = d^{AB^T} d^{AB} - l^{AB}, \tag{4.1d}$$

$$g_5 = \theta_2 - \theta, \tag{4.1e}$$

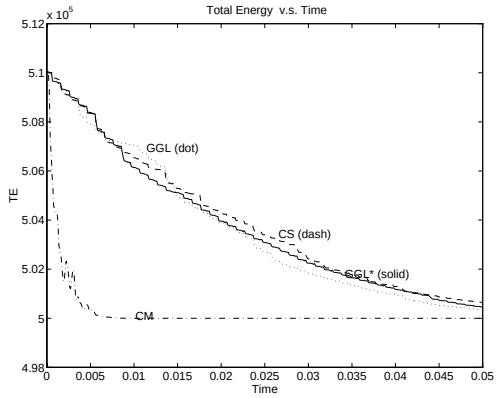

FIG. 4.3. *Total energy comparison of oscillatory spring example*; $\epsilon = 10^{-4}$ *and* $TOL = 10^{-4}$.

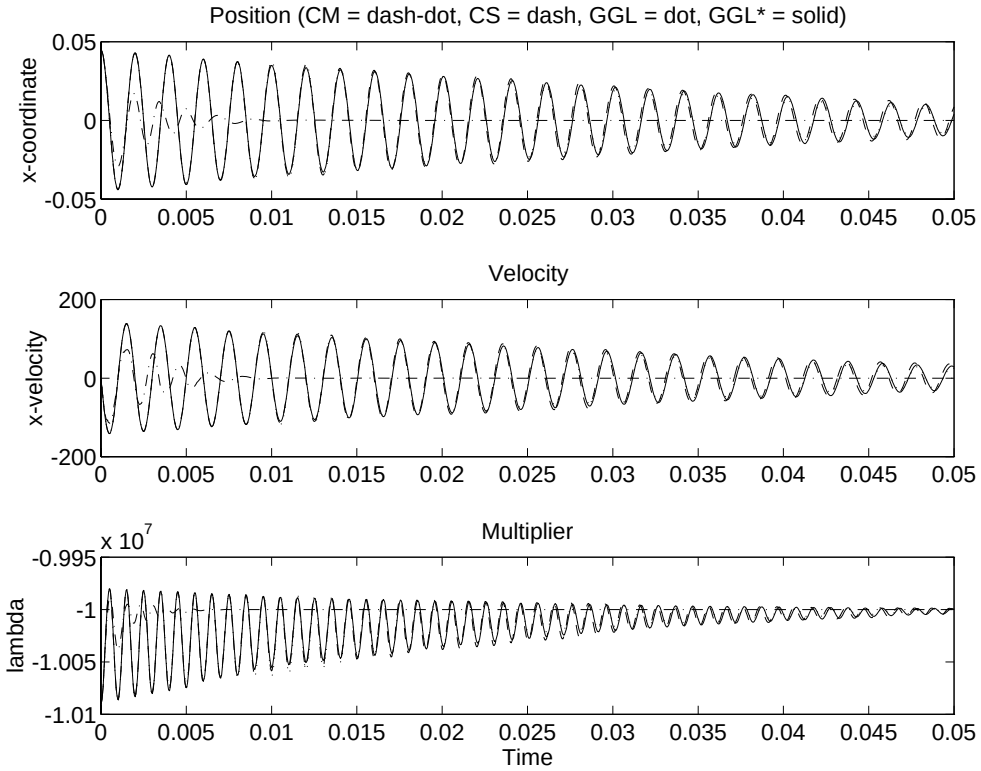

FIG. 4.4. *Results of oscillatory spring example*; $\epsilon = 10^{-4}$ *and* $TOL = 10^{-4}$.

where l^{AB} and θ are constant, and

$$d^{AB} = \begin{bmatrix} x_1 + a_1 \cos\theta_1 - a_2 \sin\theta_1 - (x_2 + b_1 \cos\theta_2 - b_2 \sin\theta_2) \\ y_1 + a_1 \sin\theta_1 + a_2 \cos\theta_1 - (y_2 + b_1 \sin\theta_2 + b_2 \cos\theta_2) \end{bmatrix}$$

such that $A = [a_1, a_2]$ and $B = [b_1, b_2]$ in the local reference coordinate systems of body 1 and body 2, respectively. In this example, we use $A = [0, 0]$, $B = [0, 0]$, $l^{AB} = 1$, and $\theta = 0$ for the constraint equations (4.1).

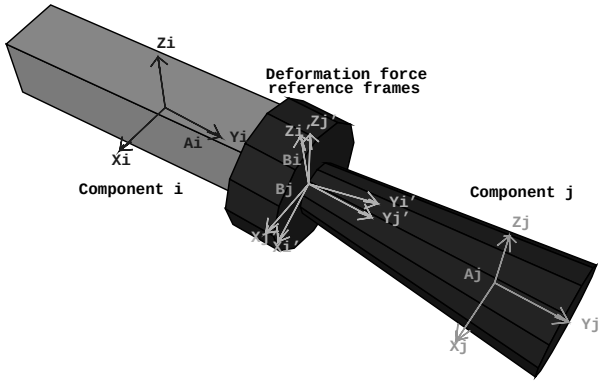

FIG. 4.5. *Deformation force of a flexible body.*

We apply a nonlinear oscillatory force formulated using *nonlinear beam* theory [4]. This type of force arises commonly in flexible multibody dynamics [28]. As described in (3.2), the deformation force between the ith and jth components is a function of the relative displacement of the reference frames X_i'-Y_i'-Z_i' and X_j'-Y_j'-Z_j', as shown schematically in Figure 4.5. Typically, the relative displacement is measured by

$$d_{ij} = r_j + A_j s_j' - r_i - A_i s_i', \tag{4.2}$$

where s_i' and s_j' are constant vectors with respect to the origins of the force reference frames in their respective *local* coordinate systems, i.e., X_i-Y_i-Z_i and X_j-Y_j-Z_j, where r_i, r_j are the corresponding origins in a *global* coordinate system and A_i and A_j are the transformation matrices from the *global* to the *local* coordinate system [10]. The relative angles, $\Theta_{ij} = [\psi_{ij}, \theta_{ij}, \phi_{ij}]^T$, are calculated as

$$A_{ij} = (A_i B_i)^T A_j B_j,$$

$$\psi_{ij} = -A_{ij}(2,3),$$

$$\theta_{ij} = A_{ij}(1,3),$$

$$\phi_{ij} = \arctan\left(\frac{A_{ij}(2,1)}{A_{ij}(2,2)}\right),$$

where $A_{ij}(k,l)$ is the component of the kth row and lth column of A_{ij}. The matrix A_{ij} is the relative orientation matrix of two force reference frames; i.e., B_i and B_j

TABLE 4.3
Results of two-body pendulum with a bushing force, $\epsilon = 10^{-3}$, $c^x = c^y = c^\theta = 10$.

Method	TOL	*No. steps*	*No. fevals*	*No. jevals*	*No. etfs*	*No. ctfs*
GGL	10^{-3}	59	141	46	0	12
*GGL**	10^{-3}	62	154	48	0	13
CS	10^{-3}	62	156	48	0	13
CM	10^{-3}	62	156	48	0	13
GGL	10^{-4}	61	136	27	1	5
*GGL**	10^{-4}	77	193	65	1	16
CS	10^{-4}	77	193	65	1	16
CM	10^{-4}	77	193	65	1	16
GGL	10^{-5}	108	259	77	1	21
*GGL**	10^{-5}	87	215	54	0	13
CS	10^{-5}	87	215	54	0	13
CM	10^{-5}	87	215	54	0	13

TABLE 4.4
Results of bushing problem, $\epsilon = 10^{-6}$, $c^x = c^y = c^\theta = 10$.

Method	TOL	*No. steps*	*No. fevals*	*No. jevals*	*No. etfs*	*No. ctfs*
GGL	10^{-4}	5267	10536	7899	0	2633
*GGL**	10^{-4}	2251	4900	3360	1	1120
CS	10^{-4}	2252	4901	3361	1	1120
CM	10^{-4}	20	40	7	0	0

are constant. The relative velocity is the time derivative of the relative displacement $\dot{d}_{ij} = \frac{d}{dt} d_{ij}$, and the relative angular velocity is $\omega_{ij} = \omega_j - \omega_i$, where ω_i and ω_j are the angular velocities of bodies i and j, respectively.

Using the above-defined notation, the force acting between the ith and jth components due to the deformation can be written as

$$f_{ij} = A_i B_i (K^f (A_i B_i)^T d_{ij} + C^f (A_i B_i)^T \dot{d}_{ij}),$$

where K^f is a 3×3 structural stiffness matrix and C^f is the 3×3 damping coefficient matrix. Similarly, the torque acting between the components is

$$\tau_{ij} = A_i B_i (K^\tau \Theta_{ij} + C^\tau (A_i B_i)^T \omega_{ij}),$$

where K^τ and C^τ are analogous to K^f and C^f. Note that the force and torque in this form are linear functions of the relative displacement (d_{ij}, Θ_{ij}) and the relative velocity $(\dot{d}_{ij}, \Omega_{ij})$.

Here, the two-dimensional *bushing* force has stiffness matrix

$$K^f = \frac{1}{\epsilon} \begin{bmatrix} k^x & 0 & 0 \\ 0 & k^y & 0 \\ 0 & 0 & k^\theta \end{bmatrix},$$

where k^x, k^y, and k^θ are $O(1)$, and damping matrix

$$C^f = \begin{bmatrix} c^x & 0 & 0 \\ 0 & c^y & 0 \\ 0 & 0 & c^\theta \end{bmatrix},$$

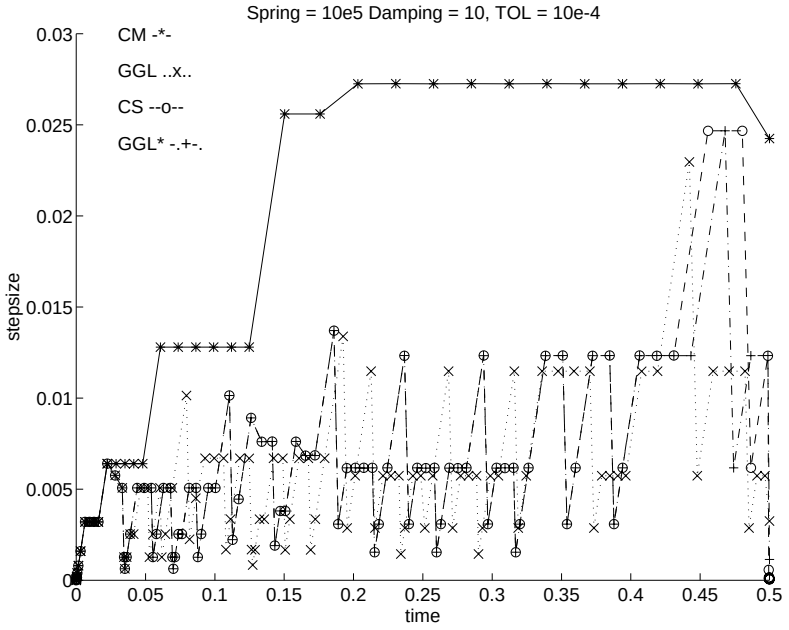

FIG. 4.6. *Time steps used in solving the bushing problem,* $\epsilon = 10^{-5}$, $c^x = c^y = c^\theta = 10$.

where c^x, c^y, and c^θ are $O(1)$. For this two-dimensional bushing example,

$$B(\theta) = \frac{1}{\epsilon}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & k^\theta \end{bmatrix} + |k^x - k^y| \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta & 0 \\ \cos\theta\sin\theta & \sin^2\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The attachment points of the force device are $s_1' = [0.5, 0]$ and $s_2' = [-0.5, 0]$ in the body-fixed reference frames of bodies 1 and 2, respectively. The bushing force introduces oscillatory applied forces, causing small oscillations of the numerical solution and yielding highly oscillatory multipliers in the index-2 DAE (1.3). The multipliers associated with the highly oscillatory components exhibit high-frequency oscillations with large amplitude. The standard convergence test of the Newton iteration depends heavily on these multipliers. Therefore, we modified the convergence test in DASSL to exclude the test for the multipliers. In addition, the multipliers are computed in the GGL* by applying the pseudoinverse $(GY)^{-T}Y^T$ to $r_1 = v^{(0)} - \rho_h q^{(0)}$ and to $r_2 = f(q^{(0)}, v^{(0)}, t) - M\rho_h v^{(0)}$, where $(q^{(0)}, v^{(0)})$ is the predictor in DASSL, and ρ_h is the discretization operator of BDF. The local error is estimated by the predictor-corrector difference of $(X^T q, X^T v)$ for CS, CM, and GGL*, and of (q, v) for GGL.

Using the initial values $q = [0, 0, 0, 9.9989\text{e}-1, -1.4852\text{e}-2, 0]$ and $v = [0, 0, 0, -6.75\text{e}{-5}, -4.5444\text{e}{-3}, 0]$, numerical results with $ATOL = RTOL = TOL$ are shown in Table 4.3, where $\epsilon = 10^{-3}$, $k^x = k^y = k^\theta = 1$, and $c^x = c^y = c^\theta = 10$. For this moderate stiffness $\epsilon = 10^{-3}$, all the methods perform well. Note that the constraint violation with these initial values is $O(10^{-3})$. This implies that the difference of the constraint reaction force at the initial value q_0 and that of a q^* on the constraint manifold is $\delta \approx \|q_0 - q^*\| = O(10^{-2})$, e.g., $\delta = O(\sqrt{10^{-3}})$. According to Theorem 3.1, the rate of convergence of the CS iteration is proportional to $\frac{\delta}{\epsilon}O(h)$. Increasing numbers of convergence test failures in DASSL are expected as $\epsilon \to 0$. In this example, fre-

quent convergence test failures occured when $\epsilon \leq 10^{-5}$ and $TOL \leq 10^{-3}$. We observe the same difficulties in the Newton convergence of the GGL and GGL* iterations. On the other hand, the CM iteration with its better Newton convergence, as explained by Theorem 3.1, is able to take much larger time steps, and the nonlinear oscillation is damped effectively. In Table 4.4, the results of $\epsilon = 10^{-6}$, $k^x = k^y = k^\theta = 1$, and $c^x = c^y = c^\theta = 10$ are shown. In Figure 4.6, we plot the stepsize taken by DASSL for GGL, GGL*, CS, and CM using the stiffness coefficient $\epsilon = 10^{-5}$.

REFERENCES

[1] U. Ascher and L. R. Petzold, *Projected implicit Runge–Kutta methods for differential-algebraic equations*, SIAM J. Numer. Anal., 28 (1991), pp. 1097–1120.

[2] N. N. Bogoliubov and Y. A. Mitropolski, *Asymptotic Methods in the Theory of Nonlinear Oscillations*, Hindustan Publishing, Delhi, India, 1961.

[3] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, 2nd ed., SIAM, Philadelphia, 1995.

[4] R. R. Craig, *Structural Dynamics*, John Wiley, New York, 1981.

[5] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[6] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[7] E. Eich, C. Führer, and J. Yen, *On the error control for multistep methods applied to ODEs with invariants and DAEs in multibody dynamics*, J. Mech. Struct. Mach., 23 (1995), pp. 17–30.

[8] C. W. Gear, G. K. Gupta, and B. J. Leimkuhler, *Automatic integration of the Euler-Lagrange equations with constraints*, J. Comp. Appl. Math., 12 (1985), pp. 77–90.

[9] E. Griepentrog, *Index Reduction Methods for Differential-Algebraic Equations*, Seminarberichte Nr. 92-1, Fachbereich Mathematik, Humboldt-Universität zu Berlin, 1992.

[10] H. Goldstein, *Classical Mechanics*, 2nd ed., Addison-Wesley, Reading, MA, 1980.

[11] G. H. Golub and V. Pereyra, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, 1989.

[13] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations* II: *Stiff and Differential-Algebraic Problems*, Springer-Verlag, Berlin, 1991.

[14] E. J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems Volume* I: *Basic Methods*, Allyn-Bacon, Boston, 1989.

[15] J. Kevorkian and J. D. Cole, *Perturbation Methods in Applied Mathematics*, Springer-Verlag, New York, 1981.

[16] C. Lanczos, *The Variational Principles of Mechanics*, University of Toronto Press, ON, Canada, 1949.

[17] Ch. Lubich, *Integration of stiff mechanical systems by Runge-Kutta methods*, Z. Angew. Math. Phys., 44 (1991), pp. 1022–1053.

[18] L. R. Petzold, L. O. Jay, and J. Yen, *Numerical solution of highly oscillatory ordinary differential equations*, in Acta Numerica, A. Iserles, ed., Cambridge University Press, Cambridge, UK, 1997, pp. 437–484.

[19] F. Potra and W. C. Rheinboldt, *On the numerical solution of the Euler-Lagrange equations*, J. Mech. Struct. Mach., 19 (1991), pp. 1–18.

[20] F. Potra and J. Yen, *Implicit numerical integration for Euler-Lagrange equations via tangent space parametrization*, J. Mech. Struct. Mach., 19 (1991), pp. 76–98.

[21] P. J. Rabier and W. C. Rheinboldt, *On the Numerical Solution of the Euler-Lagrange Equations*, Technical Report ICMA-93-117, Univ. of Pittsburgh, PA, 1993.

[22] S. Reich, *On a geometric interpretation of differential-algebraic equations*, Circ. Syst. Signal Proc., 9 (1990), pp. 367–382.

[23] S. Reich, *A Free Energy Approach to the Torsion Dynamics of Macromolecules*, Preprints SC 95-17, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1995.

[24] S. Schneider, *Convergence results for multistep Runge-Kutta on stiff mechanical systems*, Numer. Math., 69 (1995), pp. 495–508.

[25] R. A. Wehage and E. J. Haug, *Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems*, ASME J. Mech. Design, 134 (1982), pp. 247–255.
[26] J. Yen, *Constrained equations of motion in multibody dynamics as ODEs on manifolds*, SIAM J. Numer. Anal., 30 (1993), pp. 553–568.
[27] J. Yen and L. R. Petzold, *Computational challenges in the solution of nonlinear oscillatory multibody dynamic systems*, in Proceedings of the 1995 Biennial Conference on Numerical Analysis, D. F. Griffiths and G. A. Watson, eds., Pitman Research Notes in Mathematics 344, Addison Wesley Longman Ltd., 1996, pp. 209–224.
[28] W. S. Yoo and E. J. Haug, *Dynamics of articulated structures, Part* I *Theory*, J. Mech. Struct. Mach., 14 (1986), pp. 105–126.