PROOF COPY [CND-07-1179] 007902CND

Scaling of Constraints and Augmented Lagrangian Formulations in Multibody Dynamics Simulations

Olivier A. Bauchau[1]
e-mail: olivier.bauchau@ae.gatech.edu

Alexander Epple
e-mail: alexander.epple@gatech.edu

Daniel Guggenheim School of Aerospace Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332

Carlo L. Bottasso
Dipartimento di Ingegneria Aerospaziale,
Politecnico di Milano,
Milano, Italy
e-mail: carlo.bottasso@polimi.it

This paper addresses practical issues associated with the numerical enforcement of constraints in flexible multibody systems, which are characterized by index-3 differential algebraic equations (DAEs). The need to scale the equations of motion is emphasized; in the proposed approach, they are scaled based on simple physical arguments, and an augmented Lagrangian term is added to the formulation. Time discretization followed by a linearization of the resulting equations leads to a Jacobian matrix that is independent of the time step size, h; hence, the condition number of the Jacobian and error propagation are both $\mathcal{O}(h^0)$: the numerical solution of index-3 DAEs behaves as in the case of regular ODEs. Since the scaling factor depends on the physical properties of the system, the proposed scaling decreases the dependency of this Jacobian on physical properties, further improving the numerical conditioning of the resulting linearized equations. Because the scaling of the equations is performed before the time and space discretizations, its benefits are reaped for all time integration schemes. The augmented Lagrangian term is shown to be indispensable if the solution of the linearized system of equations is to be performed without pivoting, a requirement for the efficient solution of the sparse system of linear equations. Finally, a number of numerical examples demonstrate the efficiency of the proposed approach to scaling. [DOI: 10.1115/1.3079826]

1 Introduction

Multibody systems are characterized by two distinguishing features: system components undergo finite relative rotations and these components are connected by mechanical joints that impose restrictions on their relative motion. Finite rotations introduce geometric nonlinearities, hence, multibody systems are inherently nonlinear. Mechanical joints result in algebraic constraints leading to a set of governing differential algebraic equations (DAEs). Bauchau and Laulusa [1,2] presented a comprehensive review of the many formulations and numerical techniques that have been used to enforce constraints in multibody systems. Orlandea et al. [3,4] developed an approach to the analysis of multibody systems based on the direct solution of the governing index-3 DAEs. To overcome the numerical problems associated with the solution of DAEs, numerically dissipative time integrators were used that are specifically designed for stiff problems. It is interesting to note that this early approach proposes a purely numerical solution to the challenges posed by Lagrange's equations of the first kind: stiff integrators are used to deal with DAEs.

Gear and co-worker [5,6] studied DAEs extensively and concluded in 1984 that "If the index does not exceed 1, automatic codes [...] can solve the problem with no trouble." Furthermore, "If [...] the index is greater than one, the user should be encouraged to reduce it." These recommendations stem from the well-known fact that the amplification of small errors and perturbations in the solution of DAEs causes severe numerical difficulties. For example, Petzold and Lötstedt [7] showed that the index-3 DAEs characteristic of constrained multibody systems are severely ill conditioned for small time step sizes when discretized using backward differentiation formulas. Their analysis indicates that, unless corrective actions are taken, the condition number of the iteration matrix is $\mathcal{O}(h^{-3})$, where h is the time step size. Furthermore, errors propagate in the displacement, velocity, and multiplier fields at rates of $\mathcal{O}(h^{-1})$, $\mathcal{O}(h^{-2})$, and $\mathcal{O}(h^{-3})$, respectively.

These observations prompted the multibody community to engage along two distinct avenues of research. First, coordinate reduction techniques that eliminate Lagrange's multipliers all together, reducing the DAEs to ODEs. Second, index reduction techniques that reduce the governing equations of motion to index-1 equations. For instance, Borri et al. [8] developed a general index reduction procedure that splits the solution of systems represented by index-3 DAEs into separate ODE and algebraic problems. Clearly, such procedures are only attractive when leading to computationally efficient algorithms. In recent years, however, the direct solution of index-3 DAEs has regained popularity, especially when finite element formulations are used to model flexible multibody systems. Because of the large number of degrees of freedom (dofs) involved in these formulations and the likely presence of high frequencies associated with the spatial discretization process, time integration relies almost exclusively on implicit schemes such as the HHT integrator [9], or more recently, the generalized-α scheme [10].

In view of the difficulties associated with the solution of index-3 DAEs, considerable effort was devoted to the development of time integration techniques suitable for large flexible multibody systems. Cardona and co-worker [11,12] showed that the classical Newmark [13] trapezoidal rule is unconditionally *unstable* for linear systems in the presence of constraints. However, the use of dissipative algorithms such as HHT [9] resulted in stable behavior, even for nonlinear systems. Further work by Farhat et al. shows that both HHT and generalized-α [10] methods achieve stability for a class of constrained hybrid formulations. In these approaches, stabilization of the integration process is inherently associated with the dissipative nature of the algorithms.

[1]Corresponding author.

Contributed by the Design Engineering Division of ASME for publication in the JOURNAL OF COMPUTATIONAL AND NONLINEAR DYNAMICS. Manuscript received November 30, 2007; final manuscript received August 4, 2008; published online xxxxx-xxxxx-xxxxx. Review conducted by Kurt S. Anderson.

Copyright © 2009 by ASME
PROOF COPY [CND-07-1179] 007902CND

While stability is mathematically proved for linear systems, there is no guarantee when it comes to nonlinear systems [14].

While dissipative time integration schemes seem to be indispensable to the successful integration of constrained dynamical systems modeled with index-3 DAEs, scaling of the governing equations and constraints seems to be an equally important technique, which is, in fact, hardly new. In the framework of engineering optimization, scaling of constraint equations is a well-known practice that is recommended in numerous textbooks, such as Refs. [15,16]. In his 1984 textbook, Vanderplaats [17] specifically mentioned: "Often, numerical difficulties are encountered because one constraint function is of different magnitude or changes more rapidly than the others and therefore dominates the optimization process. [...,] we have normalized the constraints so they become of order of unity. This improves the conditioning of the optimization problem considerably, and *should always be done when formulating the problem.*" Although engineering optimization and multibody dynamics simulation are numerically similar problems that must both deal with constraints, it is disturbing to note that scaling of the constraint equations is rarely mentioned in multibody dynamics papers or textbooks.

Within the framework of multibody dynamics, Petzold and Lötstedt [7] discussed a simple scaling transformation of the index-3 governing equations, which yielded a condition number of the iteration matrix of $\mathcal{O}(h^{-2})$ and an improvement of one order in the errors for all solution fields. Although the sensitivity to perturbations is reduced with respect to the unscaled problem, difficulties can still be expected in practice. Cardona and Géradin [18] showed that the condition number of the iteration matrix obtained from the HHT integrator is of $\mathcal{O}(h^{-4})$ and stated that "If we try to solve this problem without scaling, the Newton algorithm will not converge since round-off errors would become of the same order as the Newton correction itself." To remedy this problem, they proposed a symmetric scaling of the equations of motion that render the condition number of the system matrix independent of the time step size and of the mean value of the mass matrix. A more systematic analysis of the scaling procedure was discussed by Bottasso et al. [19] who proposed a simple scaling transformation for the index-3 DAEs describing constrained multibody dynamical systems. The approach amounts to a left and right preconditioning of the iteration matrix, in an effort to decrease solution sensitivity to perturbation propagation. A remarkable result was obtained: both error propagation and iteration matrix conditioning are $\mathcal{O}(h^0)$, and hence, the behavior of the numerical solution of index-3 DAEs is identical to that of regular ODEs. Bottasso et al. [20] later extended the same ideas to the Newmark family of integration schemes and provided a better theoretical foundation to explain how perturbations affect the solution process.

In this paper, physical arguments are used to derive a simple scaling procedure that is directly applied to the governing equations of motion, before the time discretization is performed, and an augmented Lagrangian term is added to the formulation. Application of any time discretization scheme followed by a linearization of the resulting nonlinear algebraic equations then lead to a Jacobian matrix that is independent of the time step size, h; hence, the condition number of the Jacobian and error propagations are both $\mathcal{O}(h^0)$: the numerical solution of index-3 DAEs behaves as in the case of regular ODEs. Since the scaling factor depends on the physical properties of the system, the proposed scaling decreases the dependency of this Jacobian on physical properties, further improving the numerical conditioning of the resulting linearized equations. Finally, the additional benefits stemming from the augmented Lagrangian term are discussed. Specifically, this term enables the use of sparse solvers that do not rely on pivoting for the stable and accurate solution of the linearized equations of motion. Finally, a number of numerical examples demonstrate the efficiency of the proposed approach to scaling.

2 Scaling of the Equations of Motion

In this section, very simple, physical arguments are used to scale the index-3 DAEs characteristic of multibody systems, which are written here in the following form:

$$M_{(n\times n)}\frac{d^2\underline{q}_{(n)}}{dt^2}+B^T_{(n\times m)}\underline{\lambda}_{(m)}=\underline{F}_{(n)} \tag{1a}$$

$$\underline{\mathcal{C}}_{(m)}=0 \tag{1b}$$

where $M=M(\underline{q},t)$ is the symmetric positive definite mass matrix, and $\underline{F}=\underline{F}(\underline{q},\dot{\underline{q}},t)$ is the array of dynamic and externally applied forces. The system features n generalized coordinates stored in array $\underline{q}$, t denotes time, and the subscripts indicate the sizes of the corresponding arrays. It is assumed that the system is subjected to m holonomic constraints, $\underline{\mathcal{C}}=\underline{\mathcal{C}}(\underline{q},t)$; for simplicity of the exposition, the constraints are all assumed to be holonomic, but the derivation presented here equally applies to nonholonomic constraints or a mixture thereof. The array of Lagrange's multipliers used to enforce these constraints is denoted $\underline{\lambda}$. As expected, due to the presence of Lagrange's multipliers, these equations form a set of index-3 DAEs, as defined by Petzold and co-workers [6,21,7]. To ease the following discussion, the damping and stiffness matrices will be explicitly shown in the equations of motion, and Eqs. (1*a*) and (1*b*) are restated as

$$M\frac{d^2q}{dt^2}+D_{(n\times n)}\frac{dq}{dt}+K_{(n\times n)}\underline{q}+B^T\underline{\lambda}=\underline{G}_{(n)} \tag{2a}$$

$$\underline{\mathcal{C}}_{(m)}=0 \tag{2b}$$

where $D=D(\underline{q})$ is the damping matrix, $K=K(\underline{q})$ is the stiffness matrix, and $\underline{G}=\underline{G}(\underline{q},\dot{\underline{q}},t)$ is the array of remaining dynamic and externally applied forces.

At first, following the advice of Vanderplaats [17] for optimization problems, the constraints are normalized so as to become of order of unity. This can be readily achieved by introducing normalized generalized coordinates, $\hat{q}$, such that $\underline{q}=\ell_r\underline{\hat{q}}$, where ℓ_r is a reference or characteristic length of the system. For dynamical systems, it is also important to introduce a normalized time variable, τ, such that $t=h\tau$, where h is the time step size. Note that the equations of motion, Eqs. (2*a*) and (2*b*), have not yet been discretized in time, but the time step size is anticipated to become an important characteristic time of the problem from a numerical standpoint. The equations of motion now become

$$M\ddot{\underline{\hat{q}}}+hD\dot{\underline{\hat{q}}}+h^2K\underline{\hat{q}}+B^Th^2\underline{\lambda}=h^2\underline{G} \tag{3a}$$

$$\underline{\mathcal{C}}=0 \tag{3b}$$

It is clear that matrices M, D, K, and B, as well as arrays $\underline{G}$ and $\underline{\mathcal{C}}$, are now expressed in terms of the normalized generalized coordinates. Matrices M, D, and K have been multiplied by ℓ_r; for simplicity, the same notation is used from here on. The notation $(\dot{\cdot})$ is used to denote a derivative with respect to the nondimensional time τ. The equations of motion, Eq. (3*a*), were multiplied by h^2 to avoid division by a potentially small number, h^2.

A cursory examination of the normalized equations of motion, Eqs. (3*a*) and (3*b*), reveals two obvious numerical problems. First, if the mass and/or damping and/or stiffness of the system become large, one or more of the first three terms of the equations of motion will become large, whereas the constraint equations remain unchanged. In other words, for systems with large mass, damping, or stiffness, the constraint equations become "invisible" to the numerical process. Second, the unknowns of the problem are of different orders of magnitude: displacements are typically very small quantities, whereas Lagrange multipliers are force quantities, and hence, typically much larger.

PROOF COPY [CND-07-1179] 007902CND

The first problem is easily solved by multiplying the constraint equations, Eq. (3*b*), by a scalar factor, called the scaling factor, s, so that the constraint equations and the equations of motion, Eq. (3*a*), become of comparable magnitudes. Clearly, selecting $s=m_r+d_r h+k_r h^2$ accomplishes this goal. In this expression, m_r, d_r, and k_r represent characteristic mass, damping, and stiffness coefficients of the system, which can be selected as $m_r=\|M\|_\infty$, $d_r=\|D\|_\infty$, and $k_r=\|K\|_\infty$; another convenient choice is to select m_r, d_r, and k_r as the average of the diagonal terms of the mass, damping, and stiffness matrices, respectively. The second problem can be solved by scaling Lagrange's multipliers by writing $h^2\underline{\lambda}=s\underline{\hat{\lambda}}$. Clearly, in view of Newton's law, selecting $s=m_r+d_r h+k_r h^2$, makes $\underline{\hat{\lambda}}$ a quantity of magnitude comparable to that of displacement quantities. The equations of motion of the problem, Eqs. (3*a*) and (3*b*), now become

$$M\underline{\ddot{q}}+hD\underline{\dot{q}}+h^2K\underline{q}+B^Ts\underline{\hat{\lambda}}=h^2\underline{G} \tag{4a}$$

$$s\underline{\mathcal{C}}=0 \tag{4b}$$

It is important to understand that the techniques used here are well-known scaling techniques for systems of equations, as discussed in textbooks on matrix computations. For instance, Golub and Van Loan [22] state: "The basic recommendation is that the scaling of the equations and unknowns must proceed on a problem-by-problem basis. General scaling strategies are unreliable. It is best to scale (if at all) on the basis of what the source problem proclaims about the significance of each a_{ij} [i.e., each matrix entry]." In the proposed scaling strategy, the scaling factor was selected on the basis of physical arguments about the nature and order of magnitude of each term appearing in the equations of motion.

At this point, it is convenient to simplify the notation and write the scaled governing equations of index-3 multibody systems as

$$M\underline{\ddot{q}}+B^Ts\underline{\hat{\lambda}}=h^2\underline{F} \tag{5a}$$

$$s\underline{\mathcal{C}}=0 \tag{5b}$$

where the scaling factor is defined as

$$s=m_r+d_r h+k_r h^2 \tag{6}$$

It is important to remember that the notation $(\dot{\cdot})$ indicates a derivative with respect to the nondimensional time $\tau=t/h$, and that all generalized coordinates have been normalized by the reference length ℓ_r.

3 The Augmented Lagrangian Term

An augmented Lagrangian term is now added to the scaled formulation of the equations of motion, as proposed by Bayo et al. [23,24]

$$M\underline{\ddot{q}}+B^Ts\underline{\hat{\lambda}}+B^T\rho s\underline{\mathcal{C}}=h^2\underline{F} \tag{7a}$$

$$s\underline{\mathcal{C}}=0 \tag{7b}$$

The penalty factor, ρs, was defined as the product of the scaling factor, see Eq. (6), by ρ; for $\rho=1$, the penalty factor is equal to the scaling factor. A modified Lagrange multiplier $\underline{\hat{\mu}}=\underline{\hat{\lambda}}+\rho\underline{\mathcal{C}}$ is introduced to simplify the above equations, which become

$$M\underline{\ddot{q}}+B^Ts\underline{\hat{\mu}}=h^2\underline{F} \tag{8a}$$

$$s\underline{\mathcal{C}}=0 \tag{8b}$$

Note that the equations were scaled first, then the augmented Lagrangian term was added. Had this latter term be added from the onset, the penalty factor would become h^2p, i.e., the penalty factor would vanish for small time step sizes, negating any advantage this term could have. It is possible to include the augmented Lagrangian term from the onset of the formulation by using a penalty factor written as $\bar{\rho}s=\rho s/h^2$, which yields results identical to those presented here.

4 Time Discretization of the Equations

To understand the implications of the scaling factor and augmented Lagrangian term presented above, the equations of motion will now be discretized in the time domain. A simple midpoint scheme is used for this task

$$M(\underline{\hat{v}}_f-\underline{\hat{v}}_i)+B_m^Ts\underline{\hat{\mu}}_m=h^2\underline{F}_m \tag{9a}$$

$$\underline{\hat{q}}_f-\underline{\hat{q}}_i=(\underline{\hat{v}}_i+\underline{\hat{v}}_f)/2 \tag{9b}$$

$$s\underline{\mathcal{C}}_m=0 \tag{9c}$$

Subscripts $(\cdot)_i$ and $(\cdot)_f$ indicate quantities at the beginning and end times of the time step, denoted by t_i and t_f, respectively, $B_m=(B_i+B_f)/2$, $\underline{\mathcal{C}}_m=(\underline{\mathcal{C}}_i+\underline{\mathcal{C}}_f)/2$, $\underline{F}_m=(\underline{F}_i+\underline{F}_f)/2$, and $\underline{\hat{\mu}}_m$ are the midpoint modified Lagrange multipliers. Equation (9*b*) is the discretized velocity-displacement relationship obtained from the midpoint rule; with the present notation, $\underline{\hat{v}}=\underline{\hat{\dot{q}}}=d\underline{\hat{q}}/d\tau=hd\underline{\hat{q}}/dt$. In view of the scaling of the time dimension performed in Sec. 3, the formulas associated with time discretization are independent of the time step size, which is, in fact, taken to be unity; see Eq. (9*b*), for example. This means that the time step size dependency of the various terms of the equations of motion indicated in Eqs. (8*a*) and (8*b*) will not be affected by the time discretization, no matter what time integration scheme is used. AQ: #4

The unknown velocity, $\underline{\hat{v}}_f$, is easily eliminated from the discretized equations, leading to

$$2M(\underline{\hat{q}}_f-\underline{\hat{q}}_i-\underline{\hat{v}}_i)+B_m^Ts\underline{\hat{\mu}}=h^2\underline{F}_m \tag{10a}$$

$$s\underline{\mathcal{C}}_m=0 \tag{10b}$$

Next, these nonlinear algebraic equations will be solved using a Newton–Raphson iterative process based on the following set of linear algebraic equations:

$$\hat{J}\Delta\underline{\hat{x}}=-\underline{\hat{b}} \tag{11}$$

The Jacobian of the system, $\hat{J}$, is

$$\hat{J}=\begin{bmatrix}2M+s(B^T\underline{\hat{\mu}})_{,\underline{\hat{q}}}-h^2\underline{F}_{,\underline{\hat{q}}} & sB^T\\ s\underline{\mathcal{C}}_{,\underline{\hat{q}}} & 0\end{bmatrix}_m \tag{12}$$

$$=\begin{bmatrix}\hat{J}_{11} & \hat{J}_{12}\\ \hat{J}_{21} & 0\end{bmatrix} \tag{13}$$

where the notation $(\,)_{,\underline{\hat{q}}}$ was used to indicate a derivative with respect to the generalized coordinates, and the subscript $[\cdot]_m$ indicates that the Jacobian matrix is evaluated at the midpoint. The corrections to the unknowns of the problem are $\Delta\underline{\hat{x}}^T=\lfloor\Delta\underline{\hat{q}}_f^T,\Delta\underline{\hat{\lambda}}_m^T\rfloor$, and the residual array is

$$\underline{\hat{b}}=\begin{bmatrix}2M(\underline{\hat{q}}_f-\underline{\hat{q}}_i-\underline{\hat{v}}_i)+B^Ts\underline{\hat{\mu}}-h^2\underline{F}\\ s\underline{\mathcal{C}}\end{bmatrix}_m \tag{14}$$

It is important to realize that the asymptotic behavior of the Newton corrections $\Delta\underline{\hat{x}}$ as the time step size tends to zero depends on the asymptotic behavior of both the Jacobian, $\hat{J}$, and the right hand side, $\underline{\hat{b}}$. In fact,

PROOF COPY [CND-07-1179] 007902CND

$$\lim_{h\to 0}(\hat{J}\Delta\hat{\underline{x}}) = \lim_{h\to 0}(\hat{J})\lim_{h\to 0}(\Delta\hat{\underline{x}}) = -\lim_{h\to 0}\hat{\underline{b}} \tag{15}$$

and therefore, if $\lim_{h\to 0}(\hat{J})=\mathcal{O}(h^0)$ and $\lim_{h\to 0}(\hat{\underline{b}})=\mathcal{O}(h^0)$, then $\lim_{h\to 0}(\hat{\underline{x}})=\mathcal{O}(h^0)$.

The following results are easily obtained from an examination of Eqs. (12) and (14):

$$\hat{J} = \begin{bmatrix} \mathcal{O}(h^0) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & 0 \end{bmatrix} \quad \text{and} \quad \hat{\underline{b}} = \begin{bmatrix} \mathcal{O}(h^0) \\ \mathcal{O}(h^0) \end{bmatrix} \tag{16}$$

Furthermore, it is readily verified that the inverse Jacobian matrix is

$$\hat{J}^{-1} = \begin{bmatrix} \mathcal{O}(h^0) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^0) \end{bmatrix} \tag{17}$$

It then follows that the condition number of the Jacobian matrix, $\kappa(\hat{J})=\|\hat{J}\|_\infty\|\hat{J}^{-1}\|_\infty$, is clearly independent of the time step size, $\kappa(\hat{J})=\mathcal{O}(h^0)$. And in view of Eqs. (11) and (15), it follows that

$$\Delta\hat{\underline{q}}_f = \mathcal{O}(h^0), \quad \Delta\hat{\underline{\lambda}}_m = \mathcal{O}(h^0) \tag{18}$$

Of course, scaling of the variables has to be considered when the criterion for convergence of Newton iterations is evaluated.

This behavior is markedly different from what happens when scaling of the equations is not performed. Indeed, applying the midpoint time discretization to the unscaled augmented equations of motion, Eqs. (1*a*) and (1*b*), leads to

$$\frac{2M}{h^2}\left(\underline{q}_f - \underline{q}_i - h\frac{\mathrm{d}\underline{q}_i}{\mathrm{d}t}\right) + B_m^T\underline{\mu}_m = \underline{F}_m \tag{19a}$$

$$\underline{\mathcal{C}}_f = 0 \tag{19b}$$

where the unscaled modified Lagrange multiplier is defined as $\underline{\mu} = \underline{\lambda} + \rho\underline{\mathcal{C}}$. A Newton–Raphson approach is taken again to solve this set of nonlinear algebraic equations; linearization leads to $J\Delta\underline{x} = -\underline{b}$, where the Jacobian of the system, J, is

$$J = \begin{bmatrix} 2M/h^2 + (B^T\underline{\mu})_{,\underline{q}} - \underline{F}_{,\underline{q}} & B^T \\ \underline{\mathcal{C}}_{,\underline{q}} & 0 \end{bmatrix}_m \tag{20}$$

and the residual array is

$$\underline{b} = \begin{bmatrix} \frac{2M}{h^2}\left(\underline{q}_f - \underline{q}_i - h\frac{d\underline{q}_i}{dt}\right) + B^T\underline{\mu} - \underline{F} \\ \underline{\mathcal{C}} \end{bmatrix}_m \tag{21}$$

The following results are easily obtained from examination of Eqs. (20) and (21):

$$J = \begin{bmatrix} \mathcal{O}(h^{-2}) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & 0 \end{bmatrix} \quad \text{and} \quad \underline{b} = \begin{bmatrix} \mathcal{O}(h^{-2}) \\ \mathcal{O}(h^0) \end{bmatrix} \tag{22}$$

In the Appendix, it is shown that the inverse Jacobian matrix is

$$J^{-1} = \begin{bmatrix} \mathcal{O}(h^2) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^{-2}) \end{bmatrix} \tag{23}$$

It then follows that the condition number of the Jacobian matrix, $\kappa(J)$, exhibits a strong dependency on the time step size, $\kappa(J)=\mathcal{O}(h^{-4})$, and

$$\Delta\underline{q}_f = \mathcal{O}(h^0), \quad \Delta\underline{\lambda}_m = \mathcal{O}(h^{-2}) \tag{24}$$

5 Two Simple Examples

Two very simple examples are described in this section, to illustrate applications of the proposed procedure. Consider a simple pendulum of length ℓ and bob of mass m, as depicted in Fig. 1. This problem will be treated with two generalized coordinates: the bob's horizontal and vertical Cartesian coordinates, denoted by q_1 and q_2, respectively. Since the system features a single degree of freedom, a single constraint must be enforced: the pendulum arm must remain of constant length, ℓ. The governing equations of problem I are

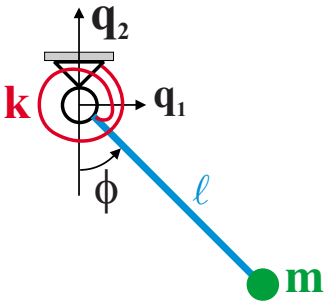

Fig. 1 Simple pendulum

$$M\frac{d^2q}{dt^2} + B^T\underline{\lambda} = 0 \tag{25a}$$

$$\underline{\mathcal{C}} = 0 \tag{25b}$$

where $M=\mathrm{diag}(m,m)$, $B=2\underline{q}^T$, $\underline{\mathcal{C}}=\underline{q}^T\underline{q}-\ell^2$, and $\underline{\lambda}=\lambda_1$. The Jacobian of the unscaled system is readily obtained from Eqs. (25*a*) and (25*b*) as

$$J = \begin{bmatrix} 2M/h^2 + (B^T\underline{\lambda})_{,\underline{q}} & B^T \\ \underline{\mathcal{C}}_{,\underline{q}} & 0 \end{bmatrix}_m \tag{26}$$

These equations of motion can be scaled then augmented using the proposed approach, and with the help of the midpoint time discretization method, the Jacobian of the linearized system then becomes

$$\hat{J} = \begin{bmatrix} 2M + s(B^T\hat{\underline{\mu}})_{,\hat{\underline{q}}} & sB^T \\ s\underline{\mathcal{C}}_{\hat{\underline{q}}} & 0 \end{bmatrix}_m \tag{27}$$

It is readily verified that all blocks of this Jacobian and of the corresponding right hand side are $\mathcal{O}(h^0)$. For this simple problem, this is true even without the augmented Lagrangian term, i.e., for $\rho=0$.

Next, consider the same problem with an additional root torsional spring of stiffness constant k, as depicted in Fig. 1. This problem will be treated with three generalized coordinates: the bob's horizontal and vertical Cartesian coordinates and the root rotation angle, ϕ. Since the system features a single degree of freedom, two constraints must be enforced, the pendulum arm must remain of constant length, ℓ, and angle ϕ can be obtained from elementary trigonometric considerations. The governing equations of problem II are

$$M\frac{d^2q}{dt^2} + B^T\underline{\lambda} = 0 \tag{28a}$$

$$k\phi + \mathcal{C}_{2,\phi}\lambda_2 = 0 \tag{28b}$$

$$\underline{\mathcal{C}} = 0 \tag{28c}$$

where $C_\phi=\cos\phi$, $S_\phi=\sin\phi$, $\underline{\lambda}^T=\lfloor\lambda_1,\lambda_2\rfloor$, $\underline{\mathcal{C}}^T=\lfloor\mathcal{C}_1,\mathcal{C}_2\rfloor=\lfloor\underline{q}^T\underline{q}-\ell^2, q_1C_\phi+q_2S_\phi\rfloor$, and

$$B = \begin{bmatrix} 2q_1 & C_\phi \\ 2q_2 & S_\phi \end{bmatrix} \tag{29}$$

Note that the relative root rotation angle, ϕ, is an *algebraic variable*, which, in contrast with the Lagrange multipliers $\underline{\lambda}$, explicitly appears in the constraint equations, Eq. (28*b*). This equation simply represents the static equilibrium of the spring and

PROOF COPY [CND-07-1179] 007902CND

hence, involves no time derivative of this angle. The explicit definition of the relative displacements and rotations at joints, as additional algebraic variables, represents an important detail for the practical implementation of multibody dynamics formulations [25]. It allows for the introduction of springs and/or dampers in the joints, as was done in this model problem, and furthermore, the time history of joint relative motions can be driven according to suitably specified time functions. The Jacobian of the unscaled system is readily obtained from Eqs. (28*a*)–(28*c*) as

$$J = \begin{bmatrix} 2M/h^2 + (B^T\underline{\lambda})_{,\underline{q}} & (B^T\underline{\lambda})_{,\phi} & B^T \\ (\mathcal{C}_{2,\phi}\lambda_2)_{,\underline{q}} & k + (\mathcal{C}_{2,\phi}\lambda_2)_{,\phi} & \underline{\mathcal{C}}^T_{,\phi} \\ \underline{\mathcal{C}}_{,\underline{q}} & \underline{\mathcal{C}}_{,\phi} & 0 \end{bmatrix}_m \tag{30}$$

These equations of motion can be scaled and augmented using the proposed approach, and with the help of the midpoint time discretization method, the Jacobian of the linearized system then becomes

$$\hat{J} = \begin{bmatrix} 2M + s(B^T\hat{\underline{\mu}})_{,\hat{\underline{q}}} & s(B^T\hat{\underline{\mu}})_{,\phi} & sB^T \\ s(\mathcal{C}_{2,\phi}\hat{\underline{\mu}}_2)_{,\hat{\underline{q}}} & h^2k + s(\mathcal{C}_{2,\phi}\hat{\underline{\mu}}_2)_{,\phi} & s\underline{\mathcal{C}}^T_{,\phi} \\ s\underline{\mathcal{C}}_{\hat{\underline{q}}} & s\underline{\mathcal{C}}_{\phi} & 0 \end{bmatrix}_m \tag{31}$$

Here again it is readily verified that all blocks of this Jacobian and of the corresponding right hand side are $\mathcal{O}(h^0)$. The key to this proof is in the fact that

$$s\hat{\underline{\mu}} = s\hat{\underline{\lambda}} + s\rho\underline{\mathcal{C}} = h^2\underline{\lambda} + s\rho\underline{\mathcal{C}} = \mathcal{O}(h^0) \tag{32}$$

In contrast with the previous example, the augmented Lagrangian term is indispensable to achieving this result; indeed, if $\rho=0$

$$s\hat{\underline{\mu}} = s\hat{\underline{\lambda}} = h^2\underline{\lambda} = \mathcal{O}(h^2) \tag{33}$$

Clearly, the proposed scaling of the unknowns and equations is sufficient to achieve time step size independent Jacobians when the problem only features Lagrange multipliers among its algebraic variables. However, when the problem also involves additional algebraic variables, such as the relative rotation of the second example, the scaling of the unknowns and of the equations must be used in conjunction with the augmented Lagrangian formulation to achieve time step size independent solutions.

6 Relationship to the Preconditioning Approach of Bottasso et al.

A preconditioning approach for index-3 DAEs was proposed by Bottasso et al. [19,20]. The starting point of their development is the Jacobian matrix resulting from the linearization of the governing Eqs. (1*a*) and (1*b*). The Jacobian is multiplied by left and right preconditioning matrices, denoted as L and R, respectively, such that $\bar{J}=LJR$, where $L=\mathrm{diag}(h^{\alpha_i})$ and $R=\mathrm{diag}(h^{\beta_i})$. The powers of the time step size, i.e., the coefficients α_i and β_i, are selected to render the preconditioned Jacobian, $\bar{J}$, independent of h. To prevent confusion, it must be noted the scaling factor defined in the present work, s, and that defined by Bottasso et al., s' (but noted s in Refs. [19,20]), are different: $s'=s/h^2$.

For problem I, the preconditioning and scaling approaches yield identical Jacobians if the preconditioning matrices are selected as $L=\mathrm{diag}(h^2,s)$ and $R=\mathrm{diag}(1,s/h^2)$. For problem II, identical Jacobians are obtained by selecting $L=\mathrm{diag}(h^2,h^2,s)$ and $R=\mathrm{diag}(1,1,s/h^2)$. Clearly, left and right preconditioning matrices can be found that will yield identical Jacobians for the two approaches.

For problem II, a time step size independent Jacobian is only obtained with the addition of an augmented Lagrangian term; indeed, without these terms, the Jacobian becomes

$$\bar{J} = \begin{bmatrix} 2M + (B^Th^2\underline{\lambda})_{,\underline{q}} & (B^Th^2\underline{\lambda})_{,\phi} & B^T \\ (\mathcal{C}_{2,\phi}h^2\lambda_2)_{,\underline{q}} & h^2k + (\mathcal{C}_{2,\phi}h^2\lambda_2)_{,\phi} & \underline{\mathcal{C}}^T_{,\phi} \\ \underline{\mathcal{C}}_{,\underline{q}} & \underline{\mathcal{C}}_{,\phi} & 0 \end{bmatrix}_m \tag{34}$$

Clearly, not all blocks of this Jacobian are $\mathcal{O}(h^0)$. The reasons why this feature is desirable is discussed in Sec. 7. While the use of the augmented Lagrangian term was not addressed in Refs. [19,20], it is clear that if such term is added to the equations of motion from the onset of the formulation, the two methods become entirely equivalent.

7 Benefits of the Augmented Lagrangian Formulation

In practical implementations of the finite element method, the linearized set of governing equations is solved in two steps [26,22]: first, the system Jacobian is factorized as $J=LDL^T$, where L is a lower triangular matrix and D a diagonal matrix, and second, the solution is found by back substitution. The advantage of this approach is that it preserves the banded structure of the Jacobian, if its factorization is performed *without pivoting*. In general, the factorization of the Jacobian without pivoting is numerically unstable, unless the Jacobian is symmetric and positive definite [22]. This is always the case for the stiffness and mass matrices of structures because they can be derived from the minimization of quadratic energy functionals; hence, factorizations without pivoting, also called "skyline solvers," are used systematically in finite element codes.

However, the Jacobian matrices of constrained multibody systems are not identical to the mass and stiffness matrices of structures. Consider the Jacobian obtained without the augmented Lagrangian term given by Eq. (34), and note the presence of the factor h^2 along some columns of the matrix. Consider next the very simple linear system, $J\underline{x}=\underline{b}$, where

$$J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & h^2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \underline{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{35}$$

which shares the characteristics of Eq. (34); although symmetric, the Jacobian is not positive definite. It is easy to show that the condition number of this Jacobian is unity, and for $h=0.001$, the exact solution is $x_1=x_2=1$, and $x_3=0.999999$. Using finite precision arithmetic with five significant digits, solution of the system with *full pivoting* yields $x_1=x_2=1$, and $x_3=0.99999$, whereas solution *without pivoting* leads to an incorrect answer, $x_1=1$, $x_2=10$, and $x_3=0.99999$. Clearly, when using a skyline solver, i.e., when factorization of the Jacobian is performed without pivoting, the condition number of the system matrix is not a good indicator of the accuracy of the solution.

While a low condition number is a necessary condition for obtaining accurate solutions of linear problems, it is not a sufficient condition when skyline solvers are used. Consider the problem II Jacobian matrices defined in Eqs. (31) and (34), obtained with and without the augmented Lagrangian term, respectively. Because of the presence of the multiplicative factor, h^2, across entire columns of the Jacobian in Eq. (34), pivoting will be required to ensure accurate solutions. On the other hand, all the submatrices of the Jacobian obtained from the present scaling approach, see Eq. (31), are independent of the time step size, enabling the safe use of skyline solvers.

The augmented Lagrangian term of the proposed formulation was shown above to be key to achieving time step size independent Jacobians, see Eq. (12). The Hessian of the system, see Eq. (13), can be expressed as $\hat{J}_{11}=2M+s(B^T\hat{\underline{\lambda}})_{,\hat{\underline{q}}}-h^2\underline{F}_{,\hat{\underline{q}}}+s\rho B^TB$, where the last term represents the contribution of the penalty term, which provides two further benefits.

First, consider problem II described earlier and assume the system to be at rest at $t=0$. Since the first Lagrange multiplier represents the tension in the rod and the second the moment in the

PROOF COPY [CND-07-1179] 007902CND

spring, it is clear that $\underline{\lambda}=0$ at $t=0$. Hence, in the absence of a penalty term, i.e., for $\rho=0$, the Jacobian of the linearized system at that instant becomes

$$\hat{J}=\begin{bmatrix} 2M & 0 & sB^T \\ 0 & 0 & s\underline{\mathcal{C}}_{,\phi}^T \\ s\underline{\mathcal{C}}_{\hat{q}} & s\underline{\mathcal{C}}_{\phi} & 0 \end{bmatrix}_m \tag{36}$$

Although this Jacobian is not singular, a skyline solver will obviously fail if pivoting is not used. Clearly, if a skyline solver is used, the augmented Lagrangian term is indispensable to the success of the simulation's first time step.

Second, Gill et al. [27] showed that there always exists a ρ^* such that the Hessian of the augmented Lagrangian, $\hat{J}_{11}$, is positive definite for all $\rho>\rho^*$. As mentioned earlier, positive definiteness is key to the reliable use of skyline solvers: this implies that the subsystem $\hat{J}_{11}\Delta\underline{\hat{x}}^*=-\underline{\hat{b}}^*$, where $\underline{\hat{x}}^*$ and $\underline{\hat{b}}^*$ are vectors of appropriate dimensions, can be solved without pivoting. Experience shows that $\rho=1$ is a good choice; this implies that the penalty factor is taken as equal to the scaling factor.

Finally, now that it has been proven that the Hessian of the augmented Lagrangian, $\hat{J}_{11}$, can be factorized without pivoting, it must also be proved that the complete solution can be obtained without pivoting. At first, consider a system with a single constraint: $\hat{J}_{12}$ and $\hat{J}_{21}$ are then column and row vectors, respectively. Since the constraint matrix is assumed to be of full rank, its single column, $\hat{J}_{12}$, must contain a least one nonzero element, and hence, factorization without pivoting can safely proceed. Mathematical induction then implies that factorization without pivoting can proceed for systems with an arbitrary number of constraints, for as long as columns and rows of $\hat{J}_{12}$ and $\hat{J}_{21}$, respectively, are linearly independent, a property that is guaranteed by the fact that the constraint matrix is of full rank.

As a last note of interest, the proof presented above assumes that the degrees of freedom of the system are segregated: first, all the generalized coordinates of the system, then, Lagrange's multipliers. In practice, this ordering is not desirable because it does not minimize the bandwidth of the system of equations. It can be easily shown that generalized coordinates and Lagrange's multipliers can be interspersed, as desired for minimization of the bandwidth, while still using a skyline solver. The only requirement is that Lagrange's multipliers must be placed after the generalized coordinates that participate in the corresponding constraint equation, as was already observed by Cardona [12].

8 Using Other Time Integration Schemes

While the proposed scaling method has been presented so far within the framework of the midpoint time integration scheme, it is easily extended to the more advanced integration methods, which are used for the analysis of realistic mechanical systems. Consider, for example, the generalized-α method [10] applied to the scaled general equations of motion of a multibody system, see Eqs. (8*a*) and (8*b*). The resulting discretization is

$$M\underline{\hat{A}}+B^Ts(\underline{\hat{\Lambda}}+\rho\underline{\mathcal{C}})=h^2\underline{F} \tag{37a}$$

$$s\underline{\mathcal{C}}=0 \tag{37b}$$

Here, the mass matrix, constraints, constraint Jacobian, and forces are evaluated using the following variables:

$$\underline{\hat{Q}}=(1-\alpha_f)\underline{\hat{q}}_{n+1}+\alpha_f\underline{\hat{q}}_n \tag{38a}$$

$$\underline{\hat{V}}=(1-\alpha_f)\underline{\hat{v}}_{n+1}+\alpha_f\underline{\hat{v}}_n \tag{38b}$$

$$\underline{\hat{A}}=(1-\alpha_m)\underline{\hat{a}}_{n+1}+\alpha_m\underline{\hat{a}}_n \tag{38c}$$

Table 1 Condition numbers of the iteration matrix, $\kappa(J)$, at convergence of the last time step for various time steps sizes. Scaling 1 is for $s=1$ and Scaling 2 is for s, as in Eq. (6). AQ: #11

h	No scaling	Scaling 1	Scaling 2
1×10^{-1}	4×10^{4}	10	12
5×10^{-2}	6×10^{5}	8.9	13
1×10^{-2}	3×10^{8}	9.2	14
5×10^{-3}	5×10^{9}	9.2	14
1×10^{-3}	3×10^{12}	9.2	14
5×10^{-4}	5×10^{13}	9.2	14
1×10^{-4}	3×10^{16}	9.2	14
5×10^{-5}	5×10^{17}	9.2	14
1×10^{-5}	3×10^{20}	9.2	14

$$\underline{\hat{\Lambda}}=(1-\alpha_f)\underline{\hat{\lambda}}_{n+1}+\alpha_f\underline{\hat{\lambda}}_n \tag{38d}$$

$$T=(1-\alpha_f)\tau_{n+1}+\alpha_f\tau_n \tag{38e}$$

which are the algorithmic displacements, velocities, accelerations, Lagrange's multipliers, and time, respectively. The corresponding variables at the end of the time step are related to their values at the beginning of the time step through the following expressions:

$$\underline{\hat{q}}_{n+1}=\underline{\hat{q}}_n+\underline{\hat{v}}_n+\underline{\hat{a}}_n/2+\underline{k}_{\hat{q}} \tag{39a}$$

$$\underline{\hat{v}}_{n+1}=\underline{\hat{v}}_n+\underline{\hat{a}}_n+\gamma\underline{k}_{\hat{q}}/\beta \tag{39b}$$

$$\underline{\hat{a}}_{n+1}=\underline{\hat{a}}_n+\underline{k}_{\hat{q}}/\beta \tag{39c}$$

$$\underline{\hat{\lambda}}_{n+1}=\underline{\hat{\lambda}}_n+\underline{k}_{\hat{\lambda}} \tag{39d}$$

$$\tau_{n+1}=\tau_n+1 \tag{39e}$$

where $\underline{k}_{\hat{q}}$ and $\underline{k}_{\hat{\lambda}}$ are the increments in displacements and Lagrange multipliers. Note that the time step size does not appear in these expressions because the nondimensional time variable has been selected in such a manner that $\Delta\tau=\Delta t/h=1$. Linearization of Eqs. (37*a*) and (37*b*) with respect to these increments yields a system of algebraic equations identical to Eq. (11) with a Jacobian matrix presenting the same structure as in Eq. (13), where the submatrices are $\hat{J}_{11}=(1-\alpha_m)/\beta M+h^2(1-\alpha_f)\gamma/\beta\underline{F}_{,\underline{\hat{v}}}+h^2(1-\alpha_f)\underline{F}_{,\underline{\hat{q}}}+s(B^T\underline{\hat{\mu}})_{,\underline{\hat{q}}}$, $\hat{J}_{12}=s(1-\alpha_f)B^T$, and $\hat{J}_{21}=s(1-\alpha_f)\underline{\mathcal{C}}_{,\underline{\hat{q}}}$, respectively, and their asymptotic behavior is independent of the time step size, as was observed for the simple midpoint scheme.

The developments presented above can be repeated for other integration schemes, such as the well-known HHT scheme [9], implicit Runge–Kutta methods including the class of Radau schemes [28], or backward difference formulas (BDFs) [29]. In all cases, the application of the time integration scheme to the proposed scaled equations, see Eqs. (8*a*) and (8*b*), leads to a Jacobian matrix that is independent of the time step size.

9 Numerical Examples

The performance of the proposed scaling method will be illustrated by means of simple examples first. Consider the simple pendulum problem described in Sec. 5, with $m=1$ kg, $k=10$ N m/rad, and $\ell=1$ m, simulated within the time range $t\in[0,1]$ s. Table 1 lists the condition numbers of iteration matrix, $\kappa(J)$, at convergence of the last time step, for time step size $h\in[10^{-1},10^{-5}]$ s. These results clearly demonstrate the need for scaling: the condition number rapidly degrades with decreasing time step sizes in the absence of scaling. AQ: #6

Next, the same problem is solved with a fixed time step size, $h=0.01$ s, and fixed spring stiffness constant $k=10$ N m/rad, but for a range of mass values, $m\in[10^{-2},10^{4}]$ kg. Table 2 lists the

PROOF COPY [CND-07-1179] 007902CND

Table 2 Condition numbers of the iteration matrix at convergence of the last time step. Scaling 1 is for $s=1$ and Scaling 2 is for s, as in Eq. (6).

Mass	No scaling	Scaling 1	Scaling 2
10^{-2}	3×10^{6}	2×10^{1}	13
10^{-1}	3×10^{8}	9×10^{0}	14
10^{0}	3×10^{10}	4×10^{2}	14
10^{1}	3×10^{12}	3×10^{4}	14
10^{2}	3×10^{14}	3×10^{6}	14
10^{3}	3×10^{16}	3×10^{8}	14
10^{4}	3×10^{18}	3×10^{10}	14

condition numbers of iteration matrix at convergence of the last time step. Here again, the need for scaling is obvious: as the mass of the system increases, the condition number of the Jacobian matrix increases. This example highlights the importance of scaling the problem with respect to its dependency on physical properties. Note the rapid rise of the condition number for the case $s=1$, whereas the use of s, as in Eq. (6), makes the condition number of the Jacobian independent of the value of the mass. Of course, varying the spring stiffness constant would yield similar results.

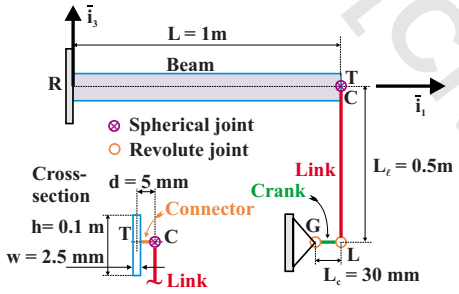

Fig. 2 Beam actuated by a tip crank

The last example deals with a realistic flexible multibody system consisting of a cantilevered beam actuated by a crank mechanism, as depicted in Fig. 2. The beam of length $L=1$ m has a rectangular cross section of depth $h=0.1$ m and width $w=2.5$ mm; it is made of aluminum, Young's modulus $E=73$ GPa, and Poisson's ratio $\nu=0.3$. This beam is modeled by eight cubic beam elements. The tip of the beam is connected to a spherical joint at point **C** by means of a short connector modeled by two cubic elements and featuring physical properties identical to those of the beam. In turn, the spherical joint is connected to a flexible steel link of length $L_\ell=0.5$ m with a hollow circular cross section of outer radius $R_o=15$ mm and thickness $t=8$ mm. Next, the link connects to a crank of length $L_c=30$ mm through a revolute joint at point **L**; the cross section of the crank is identical to that of the link. Finally, the crank is attached to the ground by means of a revolute joint at point **G**. Points **G**, **L**, and **C** define the plane of the crank-link mechanism, which is offset by a distance $d=5$ mm from the plane $(\bar{i}_1,\bar{i}_3)$ of the cantilevered beam. The relative rotation of the revolute joint at point **G** is prescribed as $\phi=1.6(1-\cos 2\pi t/T)$ rad, where $T=1.6$ s.

As the crank rotates upward, the vertical transverse shear force in the beam increases, and the beam suddenly buckles laterally. Figure 3 shows the three displacement components at the beam's midpoint: at about 0.05 s in the simulation, the lateral displacement component, u_2, suddenly increases. Lateral buckling is accompanied by a rotation of the beam's midsection. The following observations will be made concerning this simulation. First, in the absence of augmented Lagrangian terms, the simulation failed at the first iteration of the first time step. Indeed, as shown earlier, the skyline solver used in the solution process is unable to deal with the structure of the system Jacobian. Next, augmented Lagrangian terms were included in the simulation, but no scaling was used, i.e., $s=1$ was selected. In this case, the skyline solver was able to factorize the Jacobian at the first time step, however, iterations failed to converge because of the poor conditioning of the system. Finally, when using the proposed scaling, the simulation ran smoothly, as shown in Fig. 3.

The same example will also be used to demonstrate the applicability of the proposed scaling to various time integration

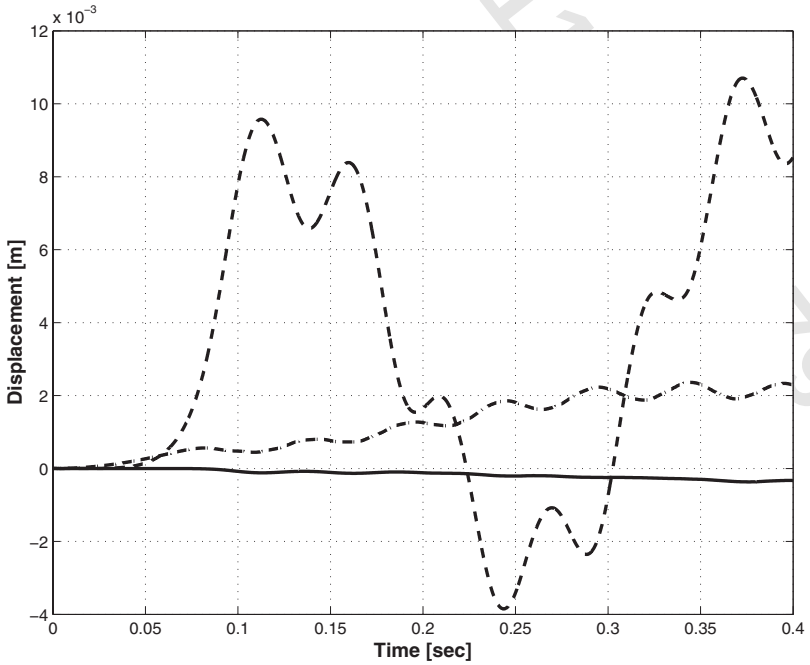

Fig. 3 Displacement components at the beam's midspan: u_1: solid line, u_2: dashed line, and u_3: dashed-dotted line

PROOF COPY [CND-07-1179] 007902CND

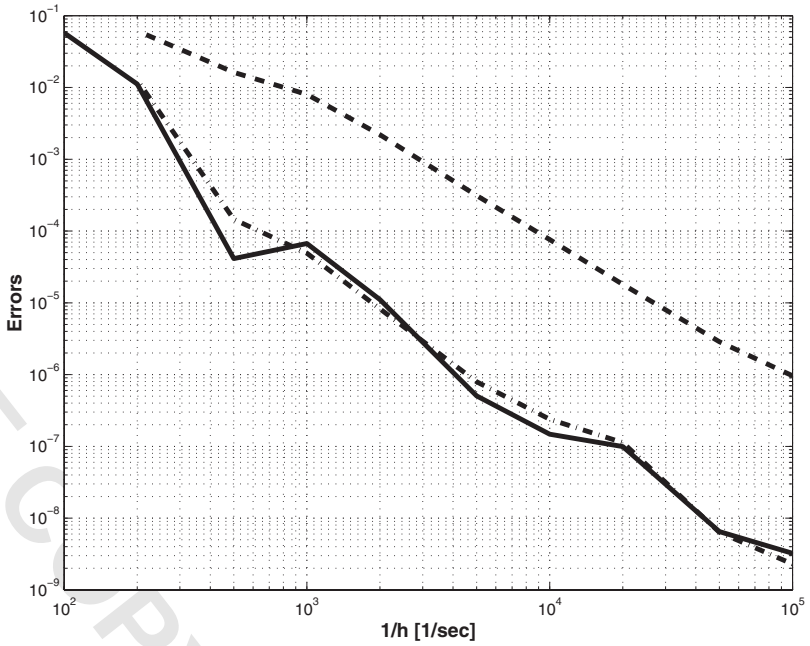

Fig. 4 Convergence characteristics of three integration schemes: Radau IIA: solid line, energy decaying scheme: dashed-dotted line, and HHT: dashed line

schemes. Simulations were run with three integration schemes: the Radau IIA scheme [28], the energy decaying scheme [30], and the HHT scheme [9]. Figure 4 demonstrates the convergence characteristics of the three schemes by plotting the solution error as a function of the inverse of the time step size. Errors were computed with respect to a reference solution obtained by using the Radau IIA scheme with a time step size $h=5$ μs. Note the good convergence of all three schemes, even for very small time step sizes.

10 Conclusion

For the past several decades, the numerical solution of DAEs has been known to be fraught with difficulties, mainly due to their undesirable behavior for vanishingly small time step sizes. Previous papers have demonstrated that scaling of both equations of motion and solution fields can cure this problem. The present paper sheds additional light on this important matter, and has established the following facts. (1) Scaling can be performed at the level of the equations of motion, prior to time discretization. By curing problems a priori, benefits are reaped for all time integration algorithms. (2) The proposed scaling factor depends both on time step size and system physical properties, further improving the numerical conditioning of the problem. (3) In many multibody formulations, algebraic variables stem not only from the presence of Lagrange multipliers, but also from the definition of additional algebraic variables such as relative motions. In such cases, scaling in conjunction with an augmented Lagrangian term was shown to yield time step size independent Jacobians. (4) The combined use of scaling with an augmented Lagrangian term also enables the safe use of sparse linear equation solvers that do not rely on pivoting to ensure stable accurate solutions. While finite element codes routinely rely on such skyline solvers, their safe use for DAEs has been justified in this paper and has considerably improved the efficiency of the solution process; this point is seldom addressed in literature.

Although further theoretical work is needed before more general conclusions can be drawn, specific facts emerge from the work presented in this paper and in Refs. [18–20]. (1) High index DAEs, once properly scaled, are not more difficult to integrate than ODEs. Unless leading to computational savings, there is no reason to avoid Lagrange multipliers, the main source of algebraic variables. (2) While numerous researchers have advocated the use of specific time integration schemes to overcome the ill conditioning of the linearized index-3 equations, the present work shows that these problems can be resolved a priori for all stable integration schemes. Furthermore, scaling does not alter the basic properties of time integration schemes. If an integration scheme is energy preserving, its application to scaled equations of motion will still preserve energy; if a scheme is robust enough to deal singular configurations, it will remain so when applied to the scaled governing equations. (3) Promoting index reduction techniques to avoid the perceived numerical problems associated with DAEs might be ill advised: the present results indicate that these techniques are not required. Furthermore, they might create difficulties that were not present in the original formulation based on DAEs; for instance, index reduction techniques often enforce constraints through their higher order derivatives, leading to the drift phenomenon, which does not affect the direct solution of high index DAEs. While the drift problem may be alleviated or completely eliminated by the use of projections onto the constraint manifold, the present index-3 approach is conceptually simpler and possibly more efficient since it does not incur in the extra costs of computing and applying projection operators.

Appendix

The inverse of the Jacobian matrix defined by Eq. (20) can be written as

$$J^{-1}=\begin{bmatrix} h^2X_{11} & X_{12} \\ X_{21} & h^{-2}X_{22} \end{bmatrix} \tag{A1}$$

where matrices $X_{11}=J_{11}^{-1}(I-J_{12}AJ_{21}J_{11}^{-1})$, $X_{12}=AJ_{21}J_{11}^{-1}$, $X_{21}=J_{11}^{-1}J_{12}A$, $X_{22}=-A$, and $A=(J_{21}J_{11}^{-1}J_{12})^{-1}$ are independent of the time step size. In these expressions, the following notation was used for the partitions of the Jacobian matrix: $J_{11}=2M+h^2(B_m^T\underline{\mu}_m)_{,\underline{q}}-h^2\underline{F}_{m,\underline{q}}$, $J_{12}=B_m^T$, and $J_{21}=\underline{C}_{f,\underline{q}}$. The above result can be easily verified by matrix multiplication. It then follows that

PROOF COPY [CND-07-1179] 007902CND

$$J^{-1} = \begin{bmatrix} \mathcal{O}(h^2) & \mathcal{O}(h^0) \\ \mathcal{O}(h^0) & \mathcal{O}(h^{-2}) \end{bmatrix} \tag{A2}$$

In view of Eq. (22), it is clear that $\|J\|_\infty = \mathcal{O}(h^{-2})$, whereas Eq. (A2) implies $\|J^{-1}\|_\infty = \mathcal{O}(h^{-2})$; it then follows that $\kappa(J) = \|J\|_\infty \|J^{-1}\|_\infty = \mathcal{O}(h^{-4})$.

References

[1] Laulusa, A. and Bauchau, O., 2008, "Review of Classical Approaches for Constraint Enforcement in Multibody Systems," ASME J. Comput. Nonlinear Dyn., **3**(1), p. 011004.

[2] Bauchau, O. and Laulusa, A., 2008, "Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems," ASME J. Comput. Nonlinear Dyn., **3**(1), pp. 011005.

[3] Orlandea, N., Chace, M., and Calahan, D., 1977, "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems. Part I," ASME J. Eng. Ind., **99**(3), pp. 773–779.

[4] Orlandea, N., Calahan, D., and Chace, M., 1977, "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems. Part II," ASME J. Eng. Ind., **99**(3), pp. 780–784.

[5] Gear, C., 1984, "Differential-Algebraic Equations," *Computer Aided Analysis and Optimization of Mechanical Systems Dynamics*, E. Haug, ed., Springer-Verlag, Berlin, pp. 323–334.

[6] Gear, C., and Petzold, L., 1984, "ODE Methods for the Solution of Differential/Algebraic Systems," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **21**(4), pp. 716–728.

[7] Petzold, L., and Lötstedt, P., 1986, "Numerical Solution of Nonlinear Differential Equations With Algebraic Constraints. II: Practical Implications," SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput., **7**(3), pp. 720–733.

[8] Borri, M., Trainelli, L., and Croce, A., 2006, "The Embedded Projection Method: A General Index Reduction Procedure for Constrained System Dynamics," Comput. Methods Appl. Mech. Eng., **195**(50–51), pp. 6974–6992.

[9] Hilber, H., Hughes, T., and Taylor, R., 1977, "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," Earthquake Eng. Struct. Dyn., **5**, pp. 283–292.

[10] Chung, J., and Hulbert, G., 1993, "A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The generalized-α Method," ASME J. Appl. Mech., **60**, pp. 371–375.

[11] Cardona, A., and Géradin, M., 1989, "Time Integration of the Equations of Motion in Mechanism Analysis," Comput. Struct., **33**(3), pp. 801–820.

[12] Cardona, A., 1989, "An Integrated Approach to Mechanism Analysis," Ph.D. thesis, Université de Liège, Belgium.

[13] Newmark, N., 1959, "A Method of Computation for Structural Dynamics," J. Engrg. Mech. Div., **85**, pp. 67–94.

[14] Hughes, T., 1983, "Analysis of Transient Algorithms With Particular Reference to Stability Behavior," *Computational Methods for Transient Analysis*, T. Belytschko and T. Hughes, eds., North-Holland, Amsterdam, pp. 67–155.

[15] Fox, R., 1971, *Optimization Methods for Engineering Design*, Addison-Wesley, Reading, MA.

[16] Reklaitis, G., Ravindran, A., and Ragsdell, K., 1983, *Engineering Optimization. Methods and Applications*, Wiley, New York.

[17] Vanderplaats, G., 1984, *Numerical Optimization Techniques for Engineering: With Applications*, McGraw-Hill, New York.

[18] Cardona, A., and Géradin, M., 1994, "Numerical Integration of Second Order Differential-Algebraic Systems in Flexible Mechanism Dynamics," *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems* (NATO ASI Series), J. Ambrosio and M. S. Pereira, eds., Kluwer, Dordrecht, The Netherlands, pp. 501–529.

[19] Bottasso, C., Bauchau, O., and Cardona, A., 2007, "Time-Step-Size-Independent Conditioning and Sensitivity to Perturbations in the Numerical Solution of Index Three Differential Algebraic Equations," SIAM J. Sci. Comput. (USA), **29**(1), pp. 397–414.

[20] Bottasso, C., Dopico, D., and Trainelli, L., "On the Optimal Scaling of Index-Three DAEs in Multibody Dynamics," Multibody Syst. Dyn., to be published.

[21] Lötstedt, P., and Petzold, L., 1986, "Numerical Solution of Nonlinear Differential Equations With Algebraic Constraints I: Convergence Results for Backward Differentiation Formulas," Math. Comput., **46**(174), pp. 491–516.

[22] Golub, G., and Van Loan, C., 1989, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD.

[23] Bayo, E., García de Jalón, J., and Serna, M., 1988, "A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems," Comput. Methods Appl. Mech. Eng., **71**, pp. 183–195.

[24] Bayo, E., García de Jalón, J., Avello, A., and Cuadrado, J., 1991, "An Efficient Computational Method for Real Time Multibody Dynamic Simulation in Fully Cartesian Coordinates," Comput. Methods Appl. Mech. Eng., **92**, pp. 377–395.

[25] Bauchau, O., Bottasso, C., and Nikishkov, Y., 2001, "Modeling Rotorcraft Dynamics With Finite Element Multibody Procedures," Math. Comput. Modell., **33**(10–11), pp. 1113–1137.

[26] Bathe, K., 1996, *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, NJ.

[27] Gill, P., Murray, W., Saunders, M., and Wright, M., 1984, "Sequential Quadratic Programming Methods For Nonlinear Programming," *Computer-Aided Analysis and Optimization of Mechanical System Dynamics*, E. Haug, ed., Springer-Verlag, Berlin, pp. 679–697.

[28] Hairer, E., and Wanner, G., 1996, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer, Berlin.

[29] Gear, C., 1971, "Simultaneous Numerical Solution of Differential-Algebraic Equations," IEEE Trans. Circuit Theory, **18**(1), pp. 89–95.

[30] Bauchau, O., 1998, "Computational Schemes for Flexible, Nonlinear Multi-Body Systems," Multibody Syst. Dyn., **2**(2), pp. 169–225.

PROOF COPY [CND-07-1179] 007902CND

PROOF COPY [CND-07-1179] 007902CND

NOT FOR PRINT! FOR REVIEW BY AUTHOR NOT FOR PRINT!

AUTHOR QUERIES — 007902CND

#1 Au: Please supply zip code for Milano, Italy.
#2 Au: Please define ODE if possible.
#3 Au: Please define HHT if possible.
#4 Au: Please check our change from “the previous section” to “Sec. 3” to ensure that this is what you are referring to.
#5 Au: please use correct symbol for all single primes throughout.
#6 Au: Kindly verify our change from “in section” to “in Sec. 5”.
#7 Au: Please check our change from “up” to “upward” to ensure that your meaning is preserved.
#8 Au: Kindly verify page range in Refs. 1 and 2.
#9 Author: Please check-page number should be 283 instead of 282 in Ref. 9.
#10 Au: Please update Ref. 20.
#11 Au: Please verify our changes to Tables 1 and 2.

PROOF COPY [CND-07-1179] 007902CND