

IMECE2005-81770

## THE NEWMARK INTEGRATION METHOD FOR SIMULATION OF MULTIBODY SYSTEMS: ANALYTICAL CONSIDERATIONS

**B. Gavrea**

Department of Mathematics,  
University of Maryland, Baltimore County  
Baltimore, Maryland 21250  
Email: gavrea1@math.umbc.edu

**D. Negrut**

Mathematics and Computer Science Division  
Argonne National Laboratory  
Argonne, Illinois 60439  
Email: negrut@mcs.anl.gov

**F.A. Potra**

Department of Mathematics  
University of Maryland, Baltimore County  
Baltimore, Maryland 21250  
Email: potra@math.umbc.edu

### ABSTRACT

When simulating the behavior of a mechanical system, the time evolution of the generalized coordinates used to represent the configuration of the model is computed as the solution of a combined set of ordinary differential and algebraic equations (DAEs). There are several ways in which the numerical solution of the resulting index 3 DAE problem can be approached. The most well-known and time-honored algorithms are the direct discretization approach, and the state-space reduction approach, respectively. In the latter, the problem is reduced to a minimal set of potentially new generalized coordinates in which the problem assumes the form of a pure second order set of Ordinary Differential Equations (ODE). This approach is very accurate, but computationally intensive, especially when dealing with large mechanical systems that contain flexible parts, stiff components, and contact/impact. The direct discretization approach is less but nevertheless sufficiently accurate yet significantly faster, and it is the approach that is considered in this paper.

In the context of direct discretization methods, approaches based on the Backward Differentiation Formulas (BDF) have been the traditional choice for more than 20 years. This paper proposes a new approach in which BDF methods are replaced by the Newmark formulas. Local convergence analysis is carried out for the proposed method, and step-size control, error estimation, and nonlinear system solution related issues are discussed in detail. A series of two simple models are used to validate the method. The global convergence analysis and a computational-efficiency comparison with the most widely used numerical integrator available in the MSC.ADAMS commercial simulation

package are forthcoming. The new method has been implemented successfully for industrial strength Dynamic Analysis simulations in the 2005 version of the MSC.ADAMS software and used very effectively for the simulation of systems with more than 15,000 differential-algebraic equations.

### 1 GENERAL CONSIDERATIONS ABOUT THE NEWMARK METHOD

The Newmark method [1] is by far one of the most widely used integration method in the structural dynamics community for the numerical integration of a linear set of second Order Differential Equations (ODE). This problem is obtained at the end of a finite element discretization. Provided the finite element approach is linear, the equations of motion assume the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}(t) \quad (1)$$

The  $p \times p$  mass, damping, and stiffness matrices,  $\mathbf{M}$ ,  $\mathbf{C}$ , and  $\mathbf{K}$ , respectively, are constant, the force  $\mathbf{F} \in \mathbb{R}^p$  depends on time  $t$ , and  $\mathbf{q} \in \mathbb{R}^p$  is the set of generalized coordinates used to represent the configuration of the mechanical system. The attractive attributes associated with the Newmark method are: (a) the resulting second order ODEs that govern the time evolution of the system do not have to be reduced to first order, which leads to simpler implementation and a smaller dimension problem; (b) good stability properties and ability to adjust the amount of damping

introduced into the system; (c) the method has been tested and validated in a vast array of applications spanning many engineering fields. The Newmark family of integration formulas depends on two parameters  $\beta$  and  $\gamma$ :

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2} [(1 - 2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}] \quad (2a)$$

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h[(1 - \gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}] \quad (2b)$$

These formulas are used to discretize at time  $t_{n+1}$  the equations of motion (1)

$$\mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{C}\dot{\mathbf{q}}_{n+1} + \mathbf{K}\mathbf{q}_{n+1} = \mathbf{F}_{n+1} \quad (2c)$$

Note that based on Eqs. (2a) and (2b),  $\mathbf{q}_{n+1}$  and  $\dot{\mathbf{q}}_{n+1}$  are functions of the acceleration  $\ddot{\mathbf{q}}_{n+1}$ , which in Eq. (2c) remains the sole unknown quantity that is eventually computed as the solution of a linear system. This method is implicit and A-stable (stable in the whole left-hand plane) [2] provided [3]

$$\gamma \geq 1/2 \quad \beta \geq \frac{(\gamma + \frac{1}{2})^2}{4} \quad (3)$$

The only combination of  $\beta$  and  $\gamma$  that leads to second order accuracy is  $\gamma = \frac{1}{2}$ , and  $\beta = \frac{1}{4}$ . The method obtained is the trapezoidal method, which is therefore both A-stable and second order. The major drawback of the Newmark family is that it can not provide a formula that is A-stable, second order, and at the same time displays a desirable level of numerical damping. The compromise for the Newmark family is to keep the A-stability and have numerical damping at the expense of the integration order, which thus go down to one. A method that alleviates this limitation at the price of a slightly more convoluted formulation is the so called  $\alpha$ -method, also known as the Hilber-Hughes-Taylor (HHT) method [3]. A generalization of this method is provided in [4].

The remaining of this paper is organized as follows. Section 2 introduces the set of Differential Algebraic Equations (DAE) that govern the time evolution of mechanical systems (the constrained equations of motion), and discusses briefly approaches for their numerical solution. Section 3 presents the framework in which the Newmark method is applied for the solution of the index 3 DAE that arises in Multi-body Dynamics. In section 4, a set of three questions relevant to the robustness and efficiency of

the proposed algorithm are posed. The questions: (a) how to estimate the local integration error; (b) how to select the integration step-size; and (c) how to define a corrector stopping criteria are answered in a unitary framework. Section 5 presents the convergence proof for the proposed method, and discusses the order of the Newmark formulas when used in conjunction with the index 3 DAE that arises in Multi-body Dynamics. Section 6 presents two simple numerical experiments carried out with the proposed algorithm, aimed at validating the theoretical results derived in conjunction with the method. The paper concludes with a series of observations and directions of future work.

## 2 THE INDEX 3 DAE OF MULTI-BODY DYNAMICS

The state of a multi-body system at the position level is represented in this paper by an array  $\mathbf{q} = [q_1, \dots, q_n]^T$  of generalized coordinates. The velocity of the system is described by the array of generalized velocities  $\dot{\mathbf{q}} = [\dot{q}_1, \dots, \dot{q}_n]^T$ . There is a multitude of ways in which the set of generalized coordinates and velocities can be selected [5–8]. The generalized coordinates used here are Cartesian coordinates for position, and Euler angles for orientation of body centroidal reference frames. Thus, for each body  $i$  its position is described by the vector  $\mathbf{r}_i = [x_i, y_i, z_i]^T$ , while its orientation is given by the array of local 3-1-3 Euler angles [9],  $\mathbf{e}_i = [\psi_i, \theta_i, \phi_i]^T$ . Consequently, for a mechanical system containing  $n_b$  bodies,  $\mathbf{q} = [\mathbf{r}_1^T \mathbf{e}_1^T \dots \mathbf{r}_{n_b}^T \mathbf{e}_{n_b}^T]^T \in \mathbf{R}^p$ ,  $p = 6n_b$ . When compared with the alternative of using a set of relative generalized coordinates, the coordinates considered are convenient because of the rather complex formalism employed to obtain the Jacobian information required for implicit integration. Note that the set of position generalized coordinates  $\mathbf{q}$  is augmented with deformation modes when flexible bodies are present in the model. For notational simplicity the assumption is that there are no flexible bodies in the model, although as observed in practice this does not pose a problem with the proposed algorithm.

In any constrained mechanical system, joints connecting bodies restrict their relative motion and impose constraints on the generalized coordinates. Kinematic constraints are then formulated as algebraic expressions involving generalized coordinates,

$$\Phi(\mathbf{q}, t) = [\Phi_1(\mathbf{q}, t) \dots \Phi_m(\mathbf{q}, t)]^T = \mathbf{0} \quad (4a)$$

where  $m$  is the total number of constraint equations that must be satisfied by the generalized coordinates throughout the simulation. It is assumed here that the  $m$  constraint equations are independent. Although the implementation of the proposed method handles non-holonomic constraints, to keep the presentation simpler the case of holonomic constraints is assumed in what follows.

Differentiating Eq. (4a) with respect to time leads to the velocity

kinematic constraint equation

$$\Phi_{\mathbf{q}}(\mathbf{q}, t) \dot{\mathbf{q}} + \Phi_t(\mathbf{q}, t) = \mathbf{0} \quad (4b)$$

where the over-dot denotes differentiation with respect to time and the subscript denotes partial differentiation,  $\Phi_{\mathbf{q}} = \begin{bmatrix} \partial\Phi_i \\ \partial q_j \end{bmatrix}$ , for  $1 \leq i \leq m$ , and  $1 \leq j \leq n$ . Finally, the acceleration kinematic constraint equations are obtained by differentiating Eq. (4b) with respect to time,

$$\Phi_{q_i}(\mathbf{q}, t) \ddot{\mathbf{q}} + (\Phi_{\mathbf{q}}(\mathbf{q}, t) \dot{\mathbf{q}})_{\mathbf{q}} \dot{\mathbf{q}} + 2\Phi_{qt}(\mathbf{q}, t) \dot{\mathbf{q}} + \Phi_{tt}(\mathbf{q}, t) = \mathbf{0} \quad (4c)$$

Equations (4a)–(4c) characterize the admissible motion of the mechanical system.

The state of the mechanical system changes in time under the effect of applied forces. The time evolution of the system is governed by the Lagrange multiplier form of the constrained equations of motion [7],

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T(\mathbf{q}) \lambda = \mathbf{Q}(\dot{\mathbf{q}}, \mathbf{q}, t) \quad (4d)$$

where  $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{p \times p}$  is the generalized mass, and  $\mathbf{Q}(\dot{\mathbf{q}}, \mathbf{q}, t) \in \mathbb{R}^p$  is the action (as opposed to the reaction  $\Phi_{\mathbf{q}}^T(\mathbf{q}) \lambda$ ) force acting on the generalized coordinates  $\mathbf{q} \in \mathbb{R}^p$ . These equations are neither linear, nor ordinary differential as is the case in Eq. (1), first and foremost because the solution  $\mathbf{q}(t)$  must also satisfy the kinematic constraint equations in Eq. (4a). These constraint equations lead, in Eq. (4d), to the presence of the reaction force  $\Phi_{\mathbf{q}}^T(\mathbf{q}) \lambda$ , where  $\lambda \in \mathbb{R}^m$  is the Lagrange multiplier associated with the kinematic constraints.

In addition to the equations of motion and kinematic constraint equations, there are several classes of equations that need to be considered in a general purpose mechanical simulation package:

1. *User defined variables*, which can technically be regarded as aliases or definition equations. A set of  $n_v$  user defined variables  $\mathbf{V} \in \mathbb{R}^{n_v}$  is typically specified through an equation of the form

$$\mathbf{V} - \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{X}, \dot{\mathbf{X}}, \lambda, \mathbf{V}, \mathbf{F}, t) = \mathbf{0}_{n_v} \quad (5a)$$

and which during the solution sequence are solved (or rather evaluated) simultaneously with the equations of motion and the kinematic constraint equations. Here  $\mathbf{v} \in \mathbb{R}^{n_v}$  is a user

defined function that depends on other system states as indicated in Eq. (5a).

2. *External force definition*,  $\mathbf{F}$ , which allow the user to more conveniently define the set of  $n_f$  applied forces  $\mathbf{F} \in \mathbb{R}^{n_f}$  that act on the system. This is the mechanism through which a complex tire model can be hooked up with a vehicle model, or the avenue through which the user can define his/her own bushing elements, custom non-linear dampers, friction, etc.

$$\mathbf{F} - \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{X}, \dot{\mathbf{X}}, \lambda, \mathbf{V}, \mathbf{F}, t) = \mathbf{0}_{n_f} \quad (5b)$$

Equations (4a)–(4d) comprise a system of index 3 DAE [10]. It is known that differential-algebraic equations are not ordinary differential equations [11]. Analytical solutions of Eqs. (4a) and (4d) automatically satisfy Eqs. (4b) and (4c), but this is no longer true for numerical solutions. In general, the task of obtaining a numerical solution of the DAE of Eqs. (4a)–(4d) is substantially more difficult and prone to intense numerical computation than that of solving ordinary differential equations (ODE). For an account of relevant work in the area of numerical integration methods for the DAE of Multi-body Dynamics the reader is referred to [2, 10, 12–15] and references therein.

The theory and attractive features associated with the Newmark method have been derived in conjunction with a linear second order ODE. The only similarity between Eqs. (1) and (4d) is that they are both second order, and qualitatively obtained from Newton's second law. In [16] and more recently [17], for the purpose of stability and convergence analysis the constrained equations of motion are tackled in a stabilized index 2 DAE framework. The HHT and Newmark methods are also discussed in [8] and more recently in [18], where the proposed implementation is based on a technique that accounts for violations in the position and velocity constraints in a stabilization framework similar to the one proposed in [19].

There are also several Runge-Kutta based approaches for highly oscillatory mechanical system simulation that, like the Newmark family of methods, display the attractive attribute of selectively damping frequency at the high end of the spectrum. In [20], a Singly Diagonal Runge-Kutta (SDIRK) based method allows the user to choose, within certain bounds, the diagonal value in the formula, and thus control the amount of numerical damping associated with the algorithm. The role of the diagonal element in the formula becomes very similar to the role of the  $\alpha$  parameter in the  $\alpha$ -method [3]. An approach based on additive Runge-Kutta methods that has the potential to accurately handle highly oscillatory multi-body dynamics simulation was introduced in [21], and further discussed in [22]. These novel Runge-Kutta based algorithms are mathematically very sound, but more time is required for them to achieve, vis-a-vis industrial strength applications, the level of acceptance and trust currently associated with the well

established Newmark method.

### 3 THE PROPOSED ALGORITHM

The index 3 DAE problem of multi-body dynamics is neither linear, nor ordinary differential in nature and the Newmark method is thus applied for a different class of problems than what it was originally designed for. Rather than approaching the solution within an index 2 framework [16, 17] or using a stabilization approach [8, 18], the proposed algorithm uses the implicit Newmark formulas to discretize the equations of motion and requires that the position-level kinematic constraint equations be satisfied at the end of each time step. This is a direct index 3 approach and it requires at each integration time step the solution of a non-linear system of equations. The theoretical foundation of this method is provided by the convergence results presented in section 5.

By means of the Newmark formulas, the collection of differential and algebraic equations are discretized to obtain an algebraic non-linear system that is solved by means of a modified-Newton algorithm. In the most general case, the unknowns of interest are the generalized positions, velocities, and accelerations  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$ , respectively, the Lagrange multipliers  $\lambda$ , the applied force states  $\mathbf{F}$ , the user-defined variables (aliases)  $\mathbf{V}$ . Note that the generalized force  $\mathbf{Q}$  is obtained by projecting (via a linear transformation) the force states  $\mathbf{F}$  along the generalized coordinates  $\mathbf{q}$ ; *i.e.*,  $\mathbf{Q} = \Pi \mathbf{F}$ , where the projection operator  $\Pi = \Pi(\mathbf{q})$  depends on the choice of generalized coordinates.

For notational simplicity, when obvious, the dependency of some quantities on  $\mathbf{q}$  and/or  $\dot{\mathbf{q}}$  and/or time  $t$  will be omitted. Likewise, throughout this paper,  $\mathbf{I}_s$  denotes the identity matrix of dimension  $s$ .

$$(\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}})_{n+1} + (\Phi_{\mathbf{q}}^T \lambda)_{n+1} - (\mathbf{Q})_{n+1} = \mathbf{0} \quad (6a)$$

$$\Phi(\mathbf{q}_{n+1}, t_{n+1}) = \mathbf{0} \quad (6b)$$

$$\mathbf{V}_{n+1} - \mathbf{v}(\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \ddot{\mathbf{q}}_{n+1}, \mathbf{X}_{n+1}, \lambda_{n+1}, \mathbf{V}_{n+1}, \mathbf{F}_{n+1}, t_{n+1}) = \mathbf{0} \quad (6c)$$

$$\mathbf{F}_{n+1} - \mathbf{f}(\mathbf{q}_{n+1}, \dot{\mathbf{q}}_{n+1}, \ddot{\mathbf{q}}_{n+1}, \mathbf{X}_{n+1}, \dot{\mathbf{X}}_{n+1}, \lambda_{n+1}, \mathbf{V}_{n+1}, \mathbf{F}_{n+1}, t_{n+1}) = \mathbf{0} \quad (6d)$$

Everywhere in Eq. (6), the Newmark integration formulas of Eq. (2) are used to express  $\mathbf{q}$  and  $\dot{\mathbf{q}}$  as a function of  $\ddot{\mathbf{q}}$ . A Newton-like algorithm [23] is used to solve the resulting system of non-linear equations for the set of unknowns (in this order)  $\ddot{\mathbf{q}}$ ,  $\lambda$ ,  $\mathbf{V}$ , and  $\mathbf{F}$ . The iterative method requires at each iteration ( $k$ ) the solutions of the linear system

$$\begin{bmatrix} \Psi & \Phi_{\mathbf{q}}^T & 0 & -\Pi \\ \Phi_{\mathbf{q}} & 0 & 0 & 0 \\ -(\mathbf{v}_{\ddot{\mathbf{q}}} + \gamma h \mathbf{v}_{\dot{\mathbf{q}}} + \beta h^2 \mathbf{v}_{\mathbf{q}}) & -\mathbf{v}_{\lambda} \mathbf{I} - \mathbf{v}_{\mathbf{V}} & -\mathbf{v}_{\mathbf{F}} & \\ -(\mathbf{f}_{\ddot{\mathbf{q}}} + \gamma h \mathbf{f}_{\dot{\mathbf{q}}} + \beta h^2 \mathbf{f}_{\mathbf{q}}) & -\mathbf{f}_{\lambda} & -\mathbf{f}_{\mathbf{V}} & \mathbf{I} - \mathbf{f}_{\mathbf{F}} \end{bmatrix} \begin{bmatrix} \Delta \ddot{\mathbf{q}} \\ \Delta \lambda \\ \Delta \mathbf{V} \\ \Delta \mathbf{F} \end{bmatrix}^{(k)} = \begin{bmatrix} -\mathbf{e}_1 \\ -\mathbf{e}_2 \\ -\mathbf{e}_3 \\ -\mathbf{e}_4 \end{bmatrix}^{(k)} \quad (7)$$

where  $\mathbf{e}_i$  are the residuals in satisfying the set of discretized equations of motion, constraint equations, variable definition equations, and applied force definition equations, respectively.

$$\mathbf{e}_1 = (\mathbf{M}\ddot{\mathbf{q}})_{n+1} + (\Phi_{\mathbf{q}}^T \lambda - \mathbf{Q})_{n+1}$$

$$\mathbf{e}_2 = \frac{1}{\beta h^2} \Phi(\mathbf{q}, t)$$

$$\mathbf{e}_3 = \mathbf{V} - \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{X}, \lambda, \mathbf{V}, t)$$

$$\mathbf{e}_4 = \mathbf{F} - \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{V}, \mathbf{X}, \dot{\mathbf{X}}, t)$$

Likewise, the matrix  $\Psi$  in Eq. (7) is defined as

$$\Psi = \frac{\partial \mathbf{e}_1}{\partial \ddot{\mathbf{q}}} = \mathbf{M} + [(\mathbf{M}\ddot{\mathbf{q}})_{\mathbf{q}} + (\Phi_{\mathbf{q}}^T \lambda)_{\mathbf{q}} - \Pi_{\mathbf{q}} \mathbf{F}] \beta h^2 \quad (8)$$

All the quantities in  $\mathbf{e}_1$  through  $\mathbf{e}_4$  above are evaluated at time  $t_{n+1}$ . Finally, note that the non-linear equations associated with the position kinematic constraints are scaled by  $\frac{1}{\beta h^2}$  in order to improve the conditioning of the coefficient matrix in Eq. (7). This is a compromise reached after considering the following alternatives: (a) have the level-zero positions,  $\mathbf{q}$  be the unknowns (replacing  $\ddot{\mathbf{q}}$ ), but then some entries in the Jacobian matrix in Eq. (7) will have to be divided by  $\beta h^2$ ; (b) have  $\ddot{\mathbf{q}}$  be the unknowns, but then the second row in the Jacobian matrix comes multiplied by  $\beta h^2$ ; (c) do as in (b), except that the set of positions kinematic constraint equations are scaled by  $\frac{1}{\beta h^2}$ . Alternative (a) is what the default GSTIFF integrator currently uses in the MSC.ADAMS simulation package [24] (here entries get divided by a factor  $\beta_0 h$  rather than  $\beta h^2$ , as the second order equations of motion are reduced to an equivalent first order system of differential equations). On numerous occasions this has been observed to be the cause of numerical problems once the step-size becomes very small and consequently some entries in the Jacobian become extremely large. A bad Jacobian condition number ensues, and the quality of the Newton corrections becomes

poor. The alternative (b) was not embraced due to the fact that the problem at (a) plagues in a more subtle way this approach as well. If  $h$  becomes very small, the second row of the Jacobian matrix is scaled by  $\beta h^2$ , which practically makes all the entries in this row very small, from where the same large condition number situation ensues. Alternative (c) proved a good solution since typically the type of error that one sees in satisfying the position kinematic constraint equations is very small. It is never that these constraint equations are problematic in a simulation but rather some discontinuity in the model that causes the step-size  $h$  to assume small values. But if  $h$  is small, when advancing the simulation the position constraint violation stays very small, and the value of  $\mathbf{e}_2$  in Eq. (8) always remains very reasonable.

With the corrections computed as the solution of the linear system of Eq. (7), the numerical solution is improved at each iteration as  $\ddot{\mathbf{q}}^{(k+1)} = \ddot{\mathbf{q}}^{(k)} + \Delta\ddot{\mathbf{q}}^{(k)}$ ,  $\lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)}$ ,  $\mathbf{V}^{(k+1)} = \mathbf{V}^{(k)} + \Delta\mathbf{V}^{(k)}$ ,  $\mathbf{F}^{(k+1)} = \mathbf{F}^{(k)} + \Delta\mathbf{F}^{(k)}$ . The following sections present in detail the answer to three key questions; (a) when is the computed solution accurate enough, (b) how to select the integration step-size  $h$ , and (c) when to stop the Newton-like iterative process that computes at each integration step the unknowns  $\ddot{\mathbf{q}}$ ,  $\lambda$ ,  $\mathbf{F}$ , and  $\mathbf{V}$ . Recall that once  $\ddot{\mathbf{q}}$  is available, Eqs. (2a) and (2b) are used to evaluate  $\mathbf{q}$ , and  $\dot{\mathbf{q}}$ , respectively.

## 4 IMPLEMENTATION DETAILS

### 4.1 Estimating the local integration error

The strategy for evaluating the local integration error is based on a linearization of the equations of motion in Eq. (4d) along with an asymptotic expansion of the solution  $\mathbf{q}$ . This approach is similar to the one proposed in [25]. The discussion is going to focus on Eq. (1), since locally a linearization of Eq. (4d) leads to the previous form. Thus, Eq. (1) is rewritten as

$$\mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{C}\dot{\mathbf{q}}_{n+1} + \mathbf{K}\mathbf{q}_{n+1} = \mathbf{F}_{n+1} \quad (9)$$

For the purpose of computing the local integration error, the usual assumption is that the configuration at time  $t_n$ ; *i.e.*,  $(\mathbf{q}_n, \dot{\mathbf{q}}_n, \ddot{\mathbf{q}}_n)$ , is perfectly consistent. That is, it satisfies the equations of motion, along with the time derivatives of the equations of motion. The focus is exclusively on computing the error associated with advancing the simulation from  $t_n$  to  $t_{n+1}$  using the Newmark method. Since the configuration is considered to be consistent at time  $t_n$ , it will satisfy the equations of motion,

$$\mathbf{M}\ddot{\mathbf{q}}_n + \mathbf{C}\dot{\mathbf{q}}_n + \mathbf{K}\mathbf{q}_n = \mathbf{F}_n \quad (10)$$

as well the first time derivative of the equations of motion at time  $t_n$ , that is,

$$\mathbf{M}\ddot{\ddot{\mathbf{q}}}_n + \mathbf{C}\dot{\ddot{\mathbf{q}}}_n + \mathbf{K}\ddot{\mathbf{q}}_n = \dot{\mathbf{F}}_n \quad (11)$$

where  $\ddot{\ddot{\mathbf{q}}}_n$  formally represents the time derivative of the acceleration at time  $t_n$ . The fact that this quantity is typically not available is set aside for the time being and will be revisited later.

The Newmark integration formula of Eq. (2) is rewritten in the equivalent form

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\ddot{\mathbf{q}}_n + \beta h^2 \mathbf{x} \quad (12a)$$

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\ddot{\mathbf{q}}_n + h\gamma \mathbf{x} \quad (12b)$$

$$\ddot{\mathbf{q}}_{n+1} = \ddot{\mathbf{q}}_n + \mathbf{x} \quad (12c)$$

The quantity of interest is the unknown  $\mathbf{x}$ , which represents the change in the value of acceleration from time  $t_n$  to  $t_{n+1}$ . Once this quantity is available Eq. (12) is used to compute the configuration of the system at  $t_{n+1}$ . Note that  $\mathbf{q}_{n+1}$ ,  $\dot{\mathbf{q}}_{n+1}$  and  $\ddot{\mathbf{q}}_{n+1}$  are required to satisfy Eq. (9). The goal is then to compute an estimation of the error at the end of one integration step (the *local* integration error)

$$\delta_{n+1} = \mathbf{q}_{n+1} - \tilde{\mathbf{q}}_{n+1} \quad (13)$$

where  $\tilde{\mathbf{q}}_{n+1}$  is the *exact* solution of the initial value problem (IVP)

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F} \quad (14)$$

that starts in the configuration  $(\mathbf{q}_n, \dot{\mathbf{q}}_n, \ddot{\mathbf{q}}_n)$  at  $t = t_n$ .

Using Taylor's Theorem,  $\tilde{\mathbf{q}}_{n+1}$  is obtained as

$$\tilde{\mathbf{q}}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\ddot{\mathbf{q}}_n + \frac{h^3}{6}\ddot{\ddot{\mathbf{q}}}_n + O(h^4) \quad (15)$$

The local integration error  $\delta_{n+1}$  becomes available as soon the acceleration correction  $\mathbf{x}$  is available. In order to obtain an estimation for  $\mathbf{x}$ , from Eqs. (12) and (1),

$$\begin{aligned} \mathbf{M}(\ddot{\mathbf{q}}_n + \mathbf{x}) + \mathbf{C}(\dot{\mathbf{q}}_n + h\ddot{\mathbf{q}}_n + h\gamma\mathbf{x}) \\ + \mathbf{K}\left(\mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\ddot{\mathbf{q}}_n + \beta h^2\mathbf{x}\right) \\ = \mathbf{F}_n + h\dot{\mathbf{F}}_n + O(h^2) \end{aligned} \quad (16)$$

where Taylor's theorem was used to expand  $\mathbf{F}(t_n + h)$ . Using Eqs. (10) and (11), and ignoring the  $O(h^2)$  term  $\mathbf{K}\frac{h^2}{2}\ddot{\mathbf{q}}_n$  leads to

$$[\mathbf{M} + h\gamma\mathbf{C} + \beta h^2\mathbf{K}]\mathbf{x} = \mathbf{M}\ddot{\mathbf{q}}_n h + O(h^2) \quad (17)$$

Denoting  $\mathbf{D} = \mathbf{M} + \gamma h\mathbf{C} + \beta h^2\mathbf{K}$ , since  $\mathbf{D}^{-1} = \mathbf{M}^{-1} + O(h) \cdot \mathbf{I}_p$ , the equation

$$\mathbf{D}\mathbf{x} = \mathbf{M}\ddot{\mathbf{q}}_n h + O(h^2)$$

leads to

$$\mathbf{x} = \ddot{\mathbf{q}}_n h + O(h^2) \quad (18)$$

Using Eq. (12a),

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\ddot{\mathbf{q}}_n + \beta h^3\ddot{\mathbf{q}}_n + O(h^4) \quad (19)$$

Based on Eqs. (19) and (15)

$$\delta_{n+1} = \mathbf{q}_{n+1} - \tilde{\mathbf{q}}_{n+1} = h^3\left(\beta - \frac{1}{6}\right) \cdot \ddot{\mathbf{q}}_n + O(h^4) \quad (20)$$

Substituting for  $\ddot{\mathbf{q}}_n$  from Eq. (18) and dropping the higher order terms leads to

$$\delta_{n+1} = \left(\beta - \frac{1}{6}\right) h^2 \mathbf{x} \quad (21)$$

which provides an effective way of computing the local integration error since all the quantities that enter this equation are available at the end of the corrector stage.

## 4.2 The Accuracy Test

With  $\mathbf{q} \in \mathbb{R}^p$ , using for  $1 \leq i \leq p$  the notation  $c_i = \left(\beta - \frac{1}{6}\right) \ddot{\mathbf{q}}_{i,n}$ , and dropping the terms of order  $h^3$  or higher in Eq. (20) yields

$$\delta_{i,n+1} = c_i \cdot h^3 \quad (22)$$

Since  $c_i$  is not known, the integration error at the end of the one time step is actually approximated as

$$\delta_{i,n+1} = \left(\beta - \frac{1}{6}\right) h^2 \cdot \mathbf{x}_i \quad 1 \leq i \leq p \quad (23)$$

Based on this value of the local integration error, a *composite error* first needs to be defined. The proposed form is

$$e \equiv \sqrt{\frac{1}{p} \left[ \left( \frac{\delta_{1,n+1}}{Y_1^q} \right)^2 + \dots + \left( \frac{\delta_{p,n+1}}{Y_p^q} \right)^2 \right]} \quad (24)$$

where  $Y_i^q = \max(1, \max_{j=1, \dots, n+1} |q_{i,j}|)$ , and  $\delta_{i,n+1}$ ,  $1 \leq i \leq p$ , is the  $i^{th}$  component of the quantity computed in Eq. (23). The composite error is compared with the user prescribed error denoted here by  $\varepsilon$ . If  $e \leq \varepsilon$  the integration time step is accepted, otherwise it is rejected. Using the notation

$$\Psi \equiv \frac{p\varepsilon^2}{\left(\beta - \frac{1}{6}\right)^2}$$

error test is equivalently expressed as

$$\|\mathbf{x}\|_q^2 \leq \frac{\Psi}{h^4} \quad (25)$$

where  $\|\cdot\|_q$  represents a weighted norm [26] defined as  $\|\mathbf{x}\|_q \equiv \left[ \sum_{i=1}^p \left( \frac{x_i}{Y_i^q} \right)^2 \right]^{\frac{1}{2}}$ . If the condition of Eq. (25) is satisfied at the end of the corrector stage, the integration step is accepted, otherwise it is rejected. In either case, the next step-size is selected as indicated in subsection 4.3.

## 4.3 The Step-Size Selection

Step-size selection plays a central role in the numerical integration algorithm. If  $e_{new} \ll \varepsilon$ , effort is wasted in computing a

solution that unnecessarily exceeds the user demanded accuracy. In this case, a more aggressive step-size selection would result in larger values for  $h$  with potentially big CPU savings at the end of the simulation. At the other end of the spectrum, a step-size selection mechanism that is too aggressive leads to a large number of integration time steps at the end of which the user accuracy demands are not met. The effort to perform such an integration step is wasted, as the integration step is discarded for a new attempt with a more conservative step-size  $h$ . To strike the right note, every time the integration step-size is chosen the goal is for the error at the end of the integration step to be precisely equal to the one deemed acceptable by the user, and quantitatively defined by  $\varepsilon$ .

Key in the selection of a new step-size is Eq. (22), which indicates that the composite error is proportional to the cube of the step-size  $h$ . Ideally, the new step-size  $h_{new}$  is selected such that

$$\varepsilon = h_{new}^3 \left[ \frac{1}{p} \sum_{i=1}^p \left( \frac{c_i}{Y_i} \right)^2 \right]^{\frac{1}{2}}$$

Therefore,  $\frac{e}{\varepsilon} = \frac{h^3}{h_{new}^3}$ , from where

$$h_{new} = h \frac{\varepsilon^{\frac{1}{3}}}{\left\{ \left[ \frac{1}{p} \sum_{i=1}^p \left( \frac{\delta_{i,n+1}}{Y_i} \right)^2 \right]^{\frac{1}{2}} \right\}^{\frac{1}{3}}}$$

Based on Eq. (21),

$$h_{new} = h \frac{s \cdot \varepsilon^{\frac{1}{3}}}{\left[ \frac{1}{\sqrt{p}} \left( \beta - \frac{1}{\delta} \right) h^2 \cdot \|\mathbf{x}\|_q \right]^{\frac{1}{3}}}$$

where a safety factor  $s = 0.9$  was used to scale the value of the new step-size [12].

Defining  $\Theta = \frac{\|\mathbf{x}\|_q^2 \cdot h^4}{\psi}$ , the previous equation finally leads to

$$h_{new} = \frac{s h}{\Theta^{\frac{1}{6}}} \quad (26)$$

#### 4.4 The Correction Stage

The goal in this section is to determine a sensible stopping criteria for the corrector iterative process. The issue is how accurate should the quantity  $\mathbf{x}$  of Eqs. (12) be computed. This quantity is obtained as the solution of an iterative Newton-like algorithm that requires at least one evaluation of the residuals in Eq. (8), followed by a forward/backward substitution to retrieve the corrections in the unknowns. However, one corrector iteration might be as expensive as doing all of the above but preceded by a full blown evaluation and factorization of the coefficient matrix of the linear system of Eq. (7). These operations are expensive and should be kept to a minimum.

It is important to get an accurate solution of the non-linear discretization system because a sloppy solution would adversely impact the quality and stability of the numerical solution. In this context, suppose that  $\mathbf{x}$  is approximated by  $\mathbf{x}^{(k)}$ , *i.e.*, the value obtained after  $k$  corrector iterations. Equation (23) indicates that the integration error  $e$  is computed based on the value  $\mathbf{x}$ , and therefore  $\mathbf{x}^{(k)}$  will lead to a value of the error  $e^{(k)}$ . It is therefore important to have a good approximation  $\mathbf{x}^{(k)}$  for  $\mathbf{x}$ , if the algorithm is to produce a meaningful approximation  $e^{(k)}$  of the composite integration error  $e$ .

A second reason for having an accurate solution is that the stability and convergence results associated with a numerical integrator are derived under the assumption that the numerical solution is computed to the specifications of the integration formula; *i.e.*, there is no room left for errors in finding the numerical solution at the end of one integration step. Finding an approximate solution translates into solving a different initial value problem, which can be close or far from the original problem based on how accurate the non-linear system of Eq. (6) is solved, and the nature of the original initial value problem itself.

The corrector stopping criteria adopted here is that the relative difference between  $e$  and  $e^{(k)}$  will be smaller than a threshold value denoted by  $c$ . A typical value recommended in the literature is  $c = 0.001$  [12].

The local integration error at the end of one time step was shown to be  $e = \left[ \beta - \frac{1}{\delta} \right] \frac{h^2}{\sqrt{p}} \|\mathbf{x}\|_q$ . After iteration  $k$ , the approximation obtained is  $e^{(k)} = \left[ \beta - \frac{1}{\delta} \right] \frac{h^2}{\sqrt{p}} \|\mathbf{x}^{(k)}\|_q$ . The question is what should  $k$  be such that  $e^{(k)}$  is close to  $e$  within 0.1% ( $c = 0.001$ ); *i.e.*,

$$\left| \frac{e - e^{(k)}}{e} \right| \leq c \quad (27)$$

Since  $e$  is not available, the test above is replaced by

$$\left| \frac{e - e^{(k)}}{\varepsilon} \right| \leq c \quad (28)$$

where  $\varepsilon$  is the user prescribed error. Note that the goal of the step-size control is to keep  $e$  as close as possible to  $\varepsilon$ , so replacing Eq. (27) with Eq. (28) is acceptable. Therefore,

$$|e - e^{(k)}| \leq \left| \beta - \frac{1}{6} \right| \frac{h^2}{\sqrt{p}} \|\mathbf{x} - \mathbf{x}^{(k)}\|_q \quad (29)$$

and an approximation for  $\|\mathbf{x} - \mathbf{x}^{(k)}\|$  is needed. Since for the Newton-like method employed the convergence is linear, there is a constant  $\xi$  that for convergence must satisfy  $0 \leq \xi < 1$ , such that [23]

$$\|\Delta \mathbf{x}^{(k+1)}\|_q \leq \xi \cdot \|\Delta \mathbf{x}^{(k)}\|_q \quad (30)$$

where  $\Delta \mathbf{x}^{(k)}$  represents the correction at iteration  $k$ ,  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta \mathbf{x}^{(k)}$ . Then,

$$\begin{aligned} \mathbf{x}^{(k)} - \mathbf{x} &= [\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}] + [\mathbf{x}^{(k+1)} - \mathbf{x}^{(k+2)}] + \dots \\ \Rightarrow \|\mathbf{x}^{(k)} - \mathbf{x}\|_q &\leq \|\Delta \mathbf{x}^{(k)}\|_q + \|\Delta \mathbf{x}^{(k+1)}\|_q + \dots \\ &\leq \|\Delta \mathbf{x}^{(k)}\|_q (1 + \xi + \xi^2 + \dots) \end{aligned}$$

which leads to

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_q \leq \|\Delta \mathbf{x}^{(k)}\|_q \cdot \frac{1}{1 - \xi} \quad (31)$$

The value  $\xi$  is going to be approximated by

$$\xi \approx \xi_k = \frac{\|\Delta \mathbf{x}^{(k)}\|_q}{\|\Delta \mathbf{x}^{(k-1)}\|_q} \quad (32)$$

Based on Eq. (31)

$$\|\mathbf{x} - \mathbf{x}^{(k+1)}\|_q \leq \|\Delta \mathbf{x}^{(k)}\|_q \cdot \frac{\xi}{1 - \xi} \quad (33)$$

which indicates that the distance between the true solution  $\mathbf{x}$ , and the approximation of the solution obtained after applying  $k$  corrections; *i.e.*  $\mathbf{x}^{(k+1)}$ , is bounded from above by a quantity proportional to the norm of the most recent correction  $\|\Delta \mathbf{x}^{(k)}\|_q$ . Going

back to Eq. (29),

$$|e - e^{(k)}| \leq \left| \beta - \frac{1}{6} \right| \frac{h^2}{\sqrt{p}} \|\Delta \mathbf{x}^{(k)}\|_q \cdot \frac{\xi}{1 - \xi} \quad (34)$$

The condition of Eq. (28) is then satisfied as soon as

$$\frac{1}{\varepsilon} \left| \beta - \frac{1}{6} \right| \frac{h^2}{\sqrt{p}} \|\Delta \mathbf{x}^{(k)}\|_q \frac{\xi}{1 - \xi} \leq c$$

Equivalently,

$$\left( \frac{\xi}{1 - \xi} \right)^2 \|\Delta \mathbf{x}^{(k)}\|_q^2 \leq c^2 \cdot \frac{\Psi}{h^4} \quad (35)$$

Note that at the right of the inequality sign are quantities that remain constant during the corrector iterative process, while at the left are quantities that change at each iteration. Likewise, note that the stopping criteria of Eq. (35) can only be used at the end of the second iteration since it is only then that an approximation of the convergence rate  $\xi$  can be produced. In other words, the proposed approach will not be able to stop the iterative process after the first iteration. This is not a matter of great concern since models as simple as a one body pendulum are already non-linear.

#### 4.5 The Prediction Stage

When the Newton-like algorithm is used to find the solution of Eq. (8), a good starting point is essential both for convergence, and reducing the effort to find the approximation of the solution at time  $t_{n+1}$ . In [27], the generalized accelerations prediction is obtained by taking  $\ddot{\mathbf{q}}_{n+1} = \ddot{\mathbf{q}}_n$ , which is equivalent to setting  $\mathbf{x} = \mathbf{0}$  in Eq. (12). This strategy is replaced by one based on polynomial extrapolation, in which currently a polynomial of order up to three is used to produce an initial guess for all the unknowns. The approach used in [27] is thus obtained by setting the degree of the interpolation polynomial to zero. The polynomial extrapolation is based on Newton divided differences and it uses Horner's scheme for evaluation of the interpolant at time  $t_{n+1}$  [26]. The degree of the interpolant can be adjusted on the fly, and the mechanism to control this takes into account the smoothness of the solution.

#### 4.6 Summary of key formulas

The answers to the questions (a) what is the stopping criteria for the non-linear discretization algebraic system, (b) how is the integration error computed, and (c) how is the step-size controlled, is summarized below.

**Notation:**

$$\Psi \equiv \frac{p\varepsilon^2}{[\beta - \frac{1}{6}]^2} \quad \Theta = \frac{\|\mathbf{x}\|_q^2 \cdot h^4}{\Psi} \quad (36a)$$

**Prediction:** Performed based on divided differences (Newton interpolation and Horner's scheme for extrapolation at  $t_{n+1}$ ).

**Correction:** Linear convergence rate allows for computation of  $\xi$  (Eq. (32)). Stopping criteria:

$$\left(\frac{\xi}{1-\xi}\right)^2 \|\Delta \mathbf{x}^{(k)}\|_q^2 \leq c^2 \frac{\Psi}{h^4}, \quad (c = 0.001) \quad (36b)$$

**Accuracy Check:** Performed after corrector converged,

$$\Theta \leq 1 \quad (36c)$$

**Step-Size Selection:** With a safety factor  $s = 0.9$ ,

$$h_{new}^{(q)} = \frac{s h}{\Theta^{\frac{1}{6}}} \quad (36d)$$

**Dense output:** Based on third order Hermite interpolation [26], for  $\dot{\mathbf{q}}$ , and  $\mathbf{q}$

## 5 LOCAL CONVERGENCE ANALYSIS

In what follows we perform a local error analysis of the scheme when applied to the index 3 DAE (4a–4d). In the correction stage we perform one Newton iteration, with predictor  $\bar{\mathbf{a}}_{n+1} = \mathbf{a}_n$  (we will use  $\mathbf{a} := \ddot{\mathbf{q}}$  and  $\mathbf{v} := \dot{\mathbf{q}}$ ), so that the acceleration  $\mathbf{a}_{n+1}$  and the Lagrange multipliers  $\lambda_{n+1}$  are computed by solving the linear system:

$$\begin{bmatrix} \Psi & \Phi^T \\ \Phi & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_{n+1} - \mathbf{a}_n \\ \lambda_{n+1} - \lambda_n \end{bmatrix} = \begin{bmatrix} -\mathbf{e}_1 \\ -\mathbf{e}_2 \end{bmatrix}. \quad (37)$$

The linear system matrix as well as the right-hand side are computed using the predicted values  $\bar{\mathbf{a}}_{n+1} = \mathbf{a}_n$ ,  $\bar{\mathbf{v}}_{n+1} = \mathbf{v}_n + h\mathbf{a}_n$ ,  $\bar{\mathbf{q}}_{n+1} = \mathbf{q}_n + h\mathbf{v}_n + \frac{h^2}{2}\mathbf{a}_n$  and  $\bar{\lambda}_{n+1} = \lambda_n$ . Let  $(\tilde{\mathbf{q}}(t), \tilde{\lambda}(t))$  denote the true solution of (4a–4d). For the truncation error analysis we take  $\mathbf{q}_n = \tilde{\mathbf{q}}(t_n)$ ,  $\mathbf{v}_n = \dot{\tilde{\mathbf{q}}}(t_n)$ ,  $\mathbf{a}_n = \ddot{\tilde{\mathbf{q}}}(t_n)$  and  $\lambda_n = \tilde{\lambda}(t_n)$ .

We first show that locally, as  $h \rightarrow 0$ , the stopping criteria given by (36b) is satisfied after one iteration. For this to hold, it is sufficient to prove that the corrections  $(\Delta \mathbf{a}, \Delta \lambda)$  remain bounded after one Newton iteration when  $h \rightarrow 0$ . Let us denote by  $\mathbf{A} :=$

$\mathbf{A}(h)$  the matrix of (37). Using the smoothness of the problem data and Taylor's theorem we obtain:

$$\begin{aligned} \mathbf{A}(h) &= \begin{bmatrix} \Psi(\bar{\mathbf{q}}_{n+1}) & \Phi_q^T(t_{n+1}, \bar{\mathbf{q}}_{n+1}) \\ \Phi_q(t_{n+1}, \bar{\mathbf{q}}_{n+1}) & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{M}(\mathbf{q}_n) + O(h) & \Phi_q^T(t_n, \mathbf{q}_n) + O(h) \\ \Phi_q(t_n, \mathbf{q}_n) + O(h) & \mathbf{0} \end{bmatrix}. \end{aligned} \quad (38)$$

We also have that

$$\mathbf{e}_1 = \mathbf{M}(\mathbf{q}_n)\mathbf{a}_n + \Phi_q^T(\mathbf{q}_n)\lambda_n - \mathbf{Q}(t_n, \mathbf{v}_n, \mathbf{q}_n) + O(h),$$

and

$$\Phi(t_{n+1}, \mathbf{q}_n + h\mathbf{v}_n + \frac{h^2}{2}\mathbf{a}_n) = \Phi(t_{n+1}, \tilde{\mathbf{q}}(t_{n+1})) + O(h^3) = O(h^3).$$

Therefore the right-hand side of (37),  $\mathbf{e}(h) := -(\mathbf{e}_1^T \mathbf{e}_2^T)^T$ , satisfies the asymptotic expansion:  $\mathbf{e}(h) = \mathbf{e}_0 + O(h)$ , where  $\mathbf{e}_0$  is obtained from the above linearizations, and does not depend on the integration step  $h$ . For  $h$  sufficiently small we have

$$\mathbf{x}(h) = (\Delta \mathbf{a}^T \Delta \lambda^T)^T = \mathbf{A}(h)^{-1} \mathbf{e}(h),$$

and therefore,

$$\|\mathbf{x}(h)\| \leq \|\mathbf{A}(h)^{-1}\| \|\mathbf{e}(h)\|. \quad (39)$$

Taking  $h \rightarrow 0$  above gives that

$$\|\mathbf{x}(h)\| \leq C$$

for all  $h$  sufficiently small, where  $C$  is a constant independent of  $h$ . From the above estimate we conclude that the corrections  $\Delta \mathbf{a}$ ,  $\Delta \lambda$ , obtained after the first Newton iteration are bounded as  $h \rightarrow 0$ , which proves our claim.

In the rest of the section we analyze the local truncation error when one Newton iteration is performed. We first look at the level of accuracy that results when the method is applied to the true solution. Using Taylor's theorem, it is easy to see that

$$\begin{aligned} \tilde{\mathbf{q}}(t_{n+1}) &= \mathbf{q}_n + h\mathbf{v}_n + \frac{h^2}{2}((1-2\beta)\mathbf{a}_n + 2\beta\ddot{\tilde{\mathbf{q}}}(t_{n+1})) + O(h^3) \\ &= \mathbf{q}_n + h\mathbf{v}_n + \frac{h^2}{2}\mathbf{a}_n + h^2\beta(\ddot{\tilde{\mathbf{q}}}(t_{n+1}) - \mathbf{a}_n) + O(h^3) \\ &= \bar{\mathbf{q}}_{n+1} + h^2\beta(\ddot{\tilde{\mathbf{q}}}(t_{n+1}) - \mathbf{a}_n) + O(h^3). \end{aligned}$$

Using the fact that  $\Phi(\tilde{\mathbf{q}}(t_{n+1}), t_{n+1}) = 0$ , we obtain

$$\begin{aligned} 0 &= \Phi(\tilde{\mathbf{q}}(t_{n+1}), t_{n+1}) \\ &= \Phi(\bar{\mathbf{q}}_{n+1}, t_{n+1}) + h^2 \beta \Phi_q(\bar{\mathbf{q}}_{n+1}, t_{n+1})(\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_n) + O(h^3), \end{aligned}$$

and therefore

$$\Phi_q(\bar{\mathbf{q}}_{n+1}, t_{n+1})(\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_n) = -\frac{1}{\beta h^2} \Phi(\bar{\mathbf{q}}_{n+1}, t_{n+1}) + O(h) \quad (40)$$

From the first set of equations in (37) we have that the numerical solution  $(\mathbf{q}_{n+1}, \mathbf{v}_{n+1}, \mathbf{a}_{n+1}, \lambda_{n+1})$  satisfies

$$\begin{aligned} &\Psi(\mathbf{a}_{n+1} - \mathbf{a}_n) + \Phi_q^T(\bar{\mathbf{q}}_{n+1}, t_{n+1})(\lambda_{n+1} - \lambda_n) \\ &= -\mathbf{M}(\bar{\mathbf{q}}_{n+1})\mathbf{a}_n - \Phi_q^T(\bar{\mathbf{q}}_{n+1}, t_{n+1})\lambda_n + \mathbf{Q}(t_{n+1}, \bar{\mathbf{v}}_{n+1}, \bar{\mathbf{q}}_{n+1}). \end{aligned}$$

Simplifying the similar terms above leads to

$$\begin{aligned} &\mathbf{M}(\bar{\mathbf{q}}_{n+1})\mathbf{a}_{n+1} + \Phi_q^T(\bar{\mathbf{q}}_{n+1})\lambda_{n+1} \\ &- \mathbf{Q}(t_{n+1}, \bar{\mathbf{v}}_{n+1}, \bar{\mathbf{q}}_{n+1}) + h\mathbf{G}(\mathbf{a}_n - \mathbf{a}_{n+1}) = 0, \end{aligned} \quad (41)$$

where

$$\begin{aligned} &\mathbf{G} = \gamma \frac{\partial \mathbf{Q}}{\partial \mathbf{v}}(t, \mathbf{q}, \mathbf{v}) \\ &- \beta h \left[ (\mathbf{M}(\mathbf{q})\mathbf{a})_q(\mathbf{q}, \mathbf{a}) + (\Phi_q^T \lambda)_q(\mathbf{q}, t, \lambda) - \frac{\partial \mathbf{Q}}{\partial \mathbf{q}}(t, \mathbf{q}, \mathbf{v}) \right] \end{aligned} \quad (42)$$

with all the quantities above evaluated at  $t = t_{n+1}$ ,  $\mathbf{q} = \bar{\mathbf{q}}_{n+1}$ ,  $\mathbf{v} = \bar{\mathbf{v}}_{n+1}$ ,  $\mathbf{a} = \mathbf{a}_n$  and  $\lambda = \lambda_n$ .

For the true solution we have the following asymptotic estimates:

$$\begin{aligned} &\mathbf{M}(\bar{\mathbf{q}}_{n+1})\tilde{\mathbf{a}}(t_{n+1}) = \mathbf{M}(\tilde{\mathbf{q}}(t_{n+1}))\tilde{\mathbf{a}}(t_{n+1}) + O(h^3) \quad (43) \\ &\Phi_q^T(\bar{\mathbf{q}}_{n+1}, t_{n+1})\tilde{\lambda}(t_{n+1}) = \Phi_q^T(\tilde{\mathbf{q}}(t_{n+1}), t_{n+1})\tilde{\lambda}(t_{n+1}) + O(h^3) \quad (44) \\ &\mathbf{Q}(t_{n+1}, \bar{\mathbf{v}}_{n+1}, \bar{\mathbf{q}}_{n+1}) = \mathbf{Q}(t_{n+1}, \tilde{\mathbf{v}}(t_{n+1}), \tilde{\mathbf{q}}(t_{n+1})) + O(h^2) \quad (45) \end{aligned}$$

Using that

$$\begin{aligned} &\mathbf{M}(\tilde{\mathbf{q}}(t_{n+1}))\tilde{\mathbf{a}}(t_{n+1}) \\ &+ \Phi_q^T(\tilde{\mathbf{q}}(t_{n+1}), t_{n+1})\tilde{\lambda}(t_{n+1}) - \mathbf{Q}(t_{n+1}, \tilde{\mathbf{v}}(t_{n+1}), \tilde{\mathbf{q}}(t_{n+1})) = 0 \end{aligned}$$

together with (43–45) gives:

$$\begin{aligned} &\mathbf{M}(\bar{\mathbf{q}}_{n+1})\tilde{\mathbf{a}}(t_{n+1}) + \Phi_q^T(\bar{\mathbf{q}}_{n+1})\tilde{\lambda}(t_{n+1}) - \mathbf{Q}(t_{n+1}, \bar{\mathbf{v}}_{n+1}, \bar{\mathbf{q}}_{n+1}) \\ &+ h\mathbf{G}(\mathbf{a}_n - \tilde{\mathbf{a}}(t_{n+1})) = h\mathbf{G}(\mathbf{a}_n - \tilde{\mathbf{a}}(t_{n+1})) + O(h^2), \end{aligned}$$

which implies that equation (41) is satisfied up to  $O(h^2)$ , i.e.,

$$\begin{aligned} &\mathbf{M}(\bar{\mathbf{q}}_{n+1})\tilde{\mathbf{a}}(t_{n+1}) + \Phi_q^T(\bar{\mathbf{q}}_{n+1})\tilde{\lambda}(t_{n+1}) \\ &- \mathbf{Q}(t_{n+1}, \bar{\mathbf{v}}_{n+1}, \bar{\mathbf{q}}_{n+1}) + h\mathbf{G}(\mathbf{a}_n - \tilde{\mathbf{a}}(t_{n+1})) = O(h^2), \end{aligned} \quad (46)$$

The analysis above can be summarized by the following result

**Theorem 1.** *The true solution of the DAE (4a–4d) satisfies the following estimates:*

$$\tilde{\mathbf{q}}(t_{n+1}) = \bar{\mathbf{q}}(t_n) + h\tilde{\mathbf{v}}(t_n) + \frac{h^2}{2} ((1 - 2\beta)\tilde{\mathbf{a}}(t_n) + 2\beta\tilde{\mathbf{a}}(t_{n+1})) + O(h^3) \quad (47a)$$

$$\tilde{\mathbf{v}}(t_{n+1}) = \tilde{\mathbf{v}}(t_n) + h((1 - \gamma)\tilde{\mathbf{a}}(t_n) + \gamma\tilde{\mathbf{a}}(t_{n+1})) + O(h^2) \quad (47b)$$

$$\begin{aligned} &(\mathbf{M}(\bar{\mathbf{q}}_{n+1}) - h\mathbf{G})\tilde{\mathbf{a}}(t_{n+1}) + \Phi_q^T(\bar{\mathbf{q}}_{n+1})\tilde{\lambda}(t_{n+1}) \\ &- \mathbf{Q}(t_{n+1}, \bar{\mathbf{v}}_{n+1}, \bar{\mathbf{q}}_{n+1}) + h\mathbf{G}\mathbf{a}_n = O(h^2) \end{aligned} \quad (47c)$$

$$\Phi_q(\bar{\mathbf{q}}_{n+1}, t_{n+1})(\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_n) = -\frac{1}{\beta h^2} \Phi(\bar{\mathbf{q}}_{n+1}, t_{n+1}) + O(h) \quad (47d)$$

where  $\mathbf{G}$  is defined by (42).

In order to give the estimates for the local truncation error, we assume that the generalized mass matrix  $\mathbf{M}(\mathbf{q})$  is uniformly positive definite on the nullspace of the Jacobian of constraints (one may assume the uniform positive definiteness of  $\mathbf{M}$  on the whole space, but the condition above is obviously less restrictive). As mentioned before, we also assume that the Jacobian  $\Phi_q$  is full rank. To be more precise we assume that there is a matrix valued mapping  $\kappa: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n \times (n-m)}$  such that

$$\begin{aligned} &\Phi_q(\mathbf{q}, t)\kappa(\mathbf{q}, t) = 0, \quad \kappa(\mathbf{q}, t)^T \kappa(\mathbf{q}, t) = \mathbf{I}, \\ &\inf_{\mathbf{q}, t} \sigma(\Phi_q(\mathbf{q}, t)\Phi_q^T(\mathbf{q}, t)) > 0, \quad \inf_{\mathbf{q}, t} \sigma(\kappa(\mathbf{q}, t)^T \mathbf{M}(\mathbf{q})\kappa(\mathbf{q}, t)) > 0, \end{aligned} \quad (48)$$

where  $\sigma(\mathbf{A})$ , denotes the spectrum of the matrix  $\mathbf{A}$ . In other words the columns of the matrix  $\kappa(\mathbf{q}, t)$  form an orthonormal basis of  $\Phi_q(\mathbf{q}, t)^T$  and the eigenvalues of the symmetric positive

definite matrices  $\Phi_q(\mathbf{q}, t)\mathbf{M}(q)\Phi_q^T(\mathbf{q}, t)$  and  $\kappa^T(\mathbf{q}, t)\mathbf{M}(q)\kappa(\mathbf{q}, t)$  are bounded below by some positive constant. This implies that the operator norms of their inverses are bounded, i.e.,

$$\sup_{\mathbf{q}, t} \| (\Phi_q(\mathbf{q}, t)\Phi_q^T(\mathbf{q}, t))^{-1} \| < \infty, \quad \sup_{\mathbf{q}, t} \| (\kappa(\mathbf{q}, t)^T\mathbf{M}(q)\kappa(\mathbf{q}, t))^{-1} \| < \infty. \quad (49)$$

**Corollary 2.** Assume that conditions (48) are satisfied. Let  $\mathbf{q}_{n+1}$ ,  $\mathbf{v}_{n+1}$ ,  $\mathbf{a}_{n+1}$  and  $\lambda_{n+1}$  be the quantities obtained after one Newton iteration, with  $\mathbf{q}_n = \tilde{\mathbf{q}}(t_n)$ ,  $\mathbf{v}_n = \tilde{\mathbf{v}}(t_n)$ ,  $\mathbf{a}_n = \tilde{\mathbf{a}}(t_n)$  and  $\lambda_n = \tilde{\lambda}(t_n)$ . Then the following relations hold:

$$\begin{aligned} \tilde{\mathbf{q}}(t_{n+1}) &= \mathbf{q}_{n+1} + O(h^3), & \tilde{\mathbf{v}}(t_{n+1}) &= \mathbf{v}_{n+1} + O(h^2), \\ \tilde{\mathbf{a}}(t_{n+1}) &= \mathbf{a}_{n+1} + O(h), & \tilde{\lambda}(t_{n+1}) &= \lambda_{n+1} + O(h). \end{aligned} \quad (50)$$

*Proof.* Using Theorem 1 and the fact that the numerical solution  $(\mathbf{q}_{n+1}, \mathbf{v}_{n+1}, \mathbf{a}_{n+1}, \lambda_{n+1})$  satisfies exactly the estimates given in this theorem we obtain that

$$\tilde{\mathbf{q}}(t_{n+1}) - \mathbf{q}_{n+1} = \beta h^2 (\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_{n+1}) + O(h^3) \quad (51a)$$

$$\tilde{\mathbf{v}}(t_{n+1}) - \mathbf{v}_{n+1} = \gamma h (\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_{n+1}) + O(h^2) \quad (51b)$$

$$\begin{aligned} (\mathbf{M}(\tilde{\mathbf{q}}_{n+1}) - h\mathbf{G}) (\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_{n+1}) \\ + \Phi_q^T(\tilde{\mathbf{q}}_{n+1}) (\tilde{\lambda}(t_{n+1}) - \lambda_{n+1}) = O(h^2) \end{aligned} \quad (51c)$$

$$\Phi_q(\tilde{\mathbf{q}}_{n+1}, t_{n+1}) (\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_n) = O(h) \quad (51d)$$

Now let us denote  $\mathbf{A} = \Phi_q^T(\tilde{\mathbf{q}}_{n+1}, t_{n+1})$ ,  $\mathbf{B} = \kappa(\tilde{\mathbf{q}}_{n+1}, t_{n+1})$  and  $\hat{\mathbf{M}} = \mathbf{M}(\tilde{\mathbf{q}}_{n+1}) - h\mathbf{G}$ . Then there are unique vectors  $\mathbf{u}$  and  $\mathbf{w}$  such that  $\tilde{\mathbf{a}}(t_{n+1}) - \mathbf{a}_{n+1} = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{w}$ , and from (51a–51d), we obtain

$$\begin{aligned} \tilde{\mathbf{q}}(t_{n+1}) - \mathbf{q}_{n+1} &= \beta h^2 (\mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{w}) + O(h^3) \\ \tilde{\mathbf{v}}(t_{n+1}) - \mathbf{v}_{n+1} &= \gamma h (\mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{w}) + O(h^2) \\ \mathbf{A}^T \hat{\mathbf{M}} \mathbf{A} \mathbf{u} + \mathbf{A}^T \hat{\mathbf{M}} \mathbf{B} \mathbf{w} + \mathbf{A}^T \mathbf{A} (\tilde{\lambda}(t_{n+1}) - \lambda_{n+1}) &= O(h^2) \\ \mathbf{B}^T \hat{\mathbf{M}} \mathbf{A} \mathbf{u} + \mathbf{B}^T \hat{\mathbf{M}} \mathbf{B} \mathbf{w} &= O(h^2) \\ \mathbf{A}^T \mathbf{A} \mathbf{u} &= O(h) \end{aligned}$$

Now by using the above relations in reverse order, we get suc-

cessively

$$\begin{aligned} \mathbf{u} &= O(h), \quad \mathbf{w} = O(h), \quad \tilde{\lambda}(t_{n+1}) - \lambda_{n+1} = O(h), \\ \tilde{\mathbf{v}}(t_{n+1}) - \mathbf{v}_{n+1} &= O(h^2), \quad \tilde{\mathbf{q}}(t_{n+1}) - \mathbf{q}_{n+1} = O(h^3), \end{aligned}$$

which completes the proof of our corollary.  $\blacksquare$

## 6 NUMERICAL EXPERIMENTS

### 6.1 The simple pendulum

The first example consists of a simple planar pendulum. A point mass  $m$  ( $m = 1$ ) is attached to a massless rod of length  $L = 1$ ; all units are SI. The equations of motion are formulated in terms of the Cartesian coordinates  $(x, y)$  of  $m$ , the motion being constrained by  $\Phi(\mathbf{q}) := x^2 + y^2 - L^2 = 0$ . The initial position of the point mass is given by

$$\mathbf{q}_0 = (x_0, y_0) = \left( \sin\left(\frac{\pi}{3}\right), -\cos\left(\frac{\pi}{3}\right) \right),$$

and the initial velocity is taken to be zero. The system was simulated for an interval of  $T = 4$  (s). To obtain a reference solution  $(\mathbf{q}_{ref}, \mathbf{v}_{ref})$ , the equations of motion are reformulated as a system of ordinary differential equations which are numerically solved with the Matlab ODE solver `ode45`. To enforce a high accuracy for the reference solution the relative and absolute tolerances of the Matlab solver are set to `RelTol` =  $10^{-8}$  and `AbsTol` =  $10^{-16}$ , respectively.

For different values of the time-step  $h$ , we compute the error in position and velocity at the final time  $T = 4$ . To be more precise, in what follows we look at the errors  $\Delta_q^h = \|\mathbf{q}_h(T) - \mathbf{q}_{ref}(T)\|_2$  and  $\Delta_v^h = \|\mathbf{v}_h(T) - \mathbf{v}_{ref}(T)\|_2$ , where  $\mathbf{q}_h$  and  $\mathbf{v}_h$  are the numerical positions and velocities obtained with the Newmark scheme when applied to the corresponding DAE. The two tables below list the above errors as well as the rate of decrease in these errors as the time-step  $h$  is successively halved. More precisely this rates are defined by  $R_q^h = \frac{\Delta_q^h}{\Delta_q^{(h/2)}}$  and  $R_v^h = \frac{\Delta_v^h}{\Delta_v^{(h/2)}}$ . As it can be seen from Table 1, for  $\gamma = 1/2$  and  $\beta = 1/4$ , i.e., parameters that correspond to the trapezoidal method, the order of convergence both in positions and velocities is 2.

### 6.2 Double Pendulum

Validation of the convergence order is also carried out using the double pendulum mechanism shown in Fig.1. This is the same model that has been recently used in [28, 29]; a large

$h$	$\Delta_q^h$	$R_q^h$	$\Delta_v^h$	$R_v^h$
$2^{-4}$	$7.72\text{E}-2$	N/A	$7.46\text{E}-1$	N/A
$2^{-5}$	$1.49\text{E}-2$	$5.20\text{E}+0$	$2.94\text{E}-2$	$2.54\text{E}+1$
$2^{-6}$	$4.48\text{E}-3$	$3.31\text{E}+0$	$1.10\text{E}-2$	$2.68\text{E}+0$
$2^{-7}$	$1.13\text{E}-3$	$3.97\text{E}+0$	$3.42\text{E}-3$	$3.21\text{E}+0$
$2^{-8}$	$2.82\text{E}-4$	$4.00\text{E}+0$	$9.02\text{E}-4$	$3.79\text{E}+0$
$2^{-9}$	$7.05\text{E}-5$	$4.00\text{E}+0$	$2.29\text{E}-4$	$3.94\text{E}+0$
$2^{-10}$	$1.76\text{E}-5$	$4.00\text{E}+0$	$5.73\text{E}-5$	$3.99\text{E}+0$
$2^{-11}$	$4.41\text{E}-6$	$4.00\text{E}+0$	$1.44\text{E}-5$	$3.99\text{E}+0$

Table 1. Single Pendulum. Error results for  $\gamma = 1/2$ ,  $\beta = 1/4$ .

$h$	$\Delta_q^h$	$R_q^h$	$\Delta_v^h$	$R_v^h$
$2^{-4}$	$1.56\text{E}-1$	N/A	$1.13\text{E}+0$	N/A
$2^{-5}$	$6.21\text{E}-2$	$2.52\text{E}+0$	$7.38\text{E}-1$	$1.53\text{E}+0$
$2^{-6}$	$2.26\text{E}-2$	$2.75\text{E}+0$	$4.27\text{E}-1$	$1.73\text{E}+0$
$2^{-7}$	$8.19\text{E}-3$	$2.76\text{E}+0$	$2.31\text{E}-1$	$1.85\text{E}+0$
$2^{-8}$	$3.15\text{E}-3$	$2.60\text{E}+0$	$1.20\text{E}-1$	$1.92\text{E}+0$
$2^{-9}$	$1.31\text{E}-3$	$2.40\text{E}+0$	$6.12\text{E}-2$	$1.96\text{E}+0$
$2^{-10}$	$5.88\text{E}-4$	$2.23\text{E}+0$	$3.09\text{E}-2$	$1.98\text{E}+0$
$2^{-11}$	$2.77\text{E}-4$	$2.13\text{E}+0$	$1.55\text{E}-2$	$1.99\text{E}+0$

Table 2. Single Pendulum. Error results for  $\gamma = 3/4$ ,  $\beta = \frac{(\gamma+1/2)^2}{4}$ .

amount of stiffness is induced by means of two rotational spring-damper-actuators (RSDA). The masses of the two pendulums are  $m_1 = 3$  and  $m_2 = 0.3$ , the dimension of the pendulums are  $L_1 = 1$  and  $L_2 = 1.5$ , the stiffness coefficients are  $k_1 = 400$  and  $k_2 = 3.E5$ , and the damping coefficients are  $c_1 = 15$  and  $c_2 = 5.E4$ . The zero-tension angles for the two RSDA elements are  $\alpha_1^0 = 3\pi/2$  and  $\alpha_2^0 = 0$ . All units are SI.

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

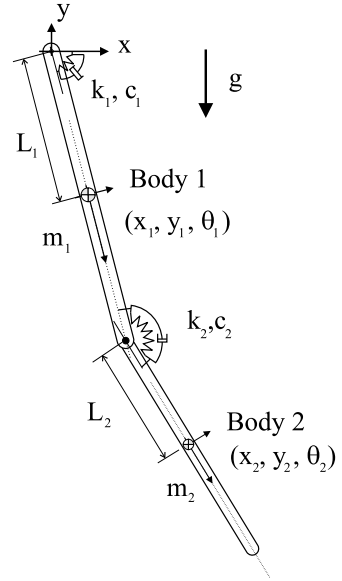


Figure 1. Double pendulum problem

around an axis perpendicular on the plane of motion. For body 1,  $\dot{x}_1^{CM} = \dot{y}_1^{CM} = \dot{\theta}_1^{CM} = 0$ , while for body 2,  $\dot{x}_2^{CM} = 3.8822857$ ,  $\dot{y}_2^{CM} = 14.4888887$ , and  $\dot{\theta}_2^{CM} = 10$ . All initial conditions are in SI units, and are consistent with the kinematic constraint equations at position and velocity levels (Eqs.(4a) and (4b)).

The differential equations that govern the time evolution of the six generalized coordinates used to model the double pendulum are

$$m_1 \ddot{x}_1 = -\lambda_1 - \lambda_3 \quad (52a)$$

$$m_1 \ddot{y}_1 = -\lambda_2 - \lambda_4 - m_1 g \quad (52b)$$

$$\frac{m_1 L_1^2}{3} \ddot{\theta}_1 = L_1 \sin(\theta_1) (\lambda_3 - \lambda_1) + L_1 \cos(\theta_1) (\lambda_2 - \lambda_4) + Q^{(1)} \quad (52c)$$

$$m_2 \ddot{x}_2 = \lambda_3 \quad (52d)$$

$$m_2 \ddot{y}_2 = \lambda_4 - m_2 g \quad (52e)$$

$$\frac{m_2 L_2^2}{3} \ddot{\theta}_2 = L_2 \sin(\theta_2) \lambda_3 - L_2 \cos(\theta_2) \lambda_4 + Q^{(2)} \quad (52f)$$

They are subjected to the algebraic constraints

$$\Phi(q) = \begin{bmatrix} x_1 - L_1 \cos(\theta_1) \\ y_1 - L_1 \sin(\theta_1) \\ x_1 + L_1 \cos(\theta_1) + L_2 \cos(\theta_2) - x_2 \\ y_1 + L_1 \sin(\theta_1) + L_2 \sin(\theta_2) - y_2 \end{bmatrix} = 0. \quad (52g)$$

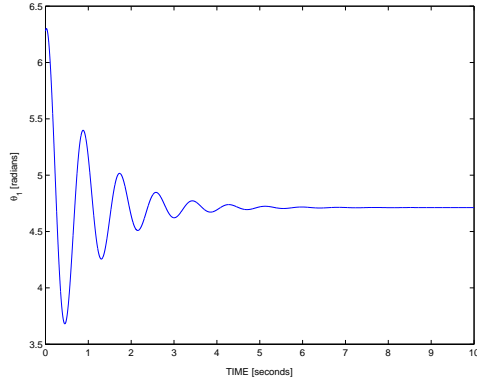


Figure 2. Time variation of orientation  $\theta_1$ , for  $\gamma = \frac{3}{4}$ ,  $\beta = \frac{(\gamma+1/2)^2}{4}$  and  $h = 0.001$

Here  $Q^{(1)} = \frac{3\pi}{2}k_1 - (k_1 + k_2)\theta_1 - (c_1 + c_2)\dot{\theta}_1 + k_2\theta_2 + c_2\dot{\theta}_2$ ,  $Q^{(2)} = k_2(\theta_1 - \theta_2) + c_2(\dot{\theta}_1 - \dot{\theta}_2)$ , and  $g = 9.81$  is the gravitational acceleration. To obtain a reference solution later used in calculating the error, the above DAE is reduced to a state space ODE in the two rotation angles  $\theta_1$  and  $\theta_2$ . The resulting system of second order ordinary differential equations assumes the form

$$4L_1^2\left(\frac{m_1}{3} + m_2\right)\ddot{\theta}_1 + 2m_2L_1L_2\cos(\theta_1 - \theta_2)\ddot{\theta}_2 = Q^{(1)}$$

$$-2m_2L_1L_2\sin(\theta_1 - \theta_2)\dot{\theta}_2^2 - (m_1 + 2m_2)gL_1\cos(\theta_1) \quad (53)$$

$$2m_2L_1L_2\cos(\theta_1 - \theta_2)\dot{\theta}_1 + \frac{4}{3}m_2L_2^2\ddot{\theta}_2 = Q^{(2)}$$

$$+ 2m_2L_1L_2\sin(\theta_1 - \theta_2)\dot{\theta}_1^2 - m_2gL_2\cos(\theta_2). \quad (54)$$

The reference solution ( $\mathbf{q}_{ref}, \mathbf{v}_{ref}$ ) is obtained by passing the above ODE to the Matlab solver `ode23s`. To enforce a high accuracy for the reference solution the relative and absolute tolerances of the Matlab solver are set to `RelTol` =  $10^{-8}$  and `AbsTol` =  $10^{-16}$ , respectively.

The time evolution of this model was simulated for  $T = 2$  (s), and an error analysis similar to the one done for the simple pendulum example is provided in Table 3.

The error analysis results clearly indicate that as predicted by the global convergence analysis [30], Newmark with a choice of parameters  $\gamma = 3/4$ ,  $\beta = \frac{(\gamma+1/2)^2}{4}$  is a first order method even when used for the integration of the index 3 DAEs associated with constrained mechanical systems. In [30], a set of real-life mechanical systems (engine models) are used to show improved efficiency for the Newmark method when compared to the BDF type integrators currently implemented in MSC.ADAMS [24]. Typically, it has been noticed that for engine models that lead to index 3 DAEs in excess of 16,000 equations, the proposed Newmark method reduced the simulation lengths by a factor from 2 to 4 times.

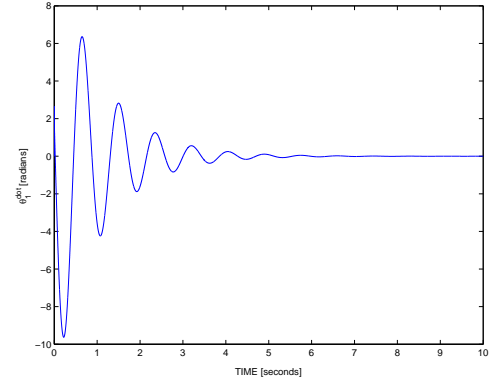


Figure 3. Time variation of angular velocity  $\dot{\theta}_1$ , for  $\gamma = \frac{3}{4}$ ,  $\beta = \frac{(\gamma+1/2)^2}{4}$  and  $h = 0.001$

$h$	$\Delta_q^h$	$R_q^h$	$\Delta_v^h$	$R_v^h$
$2^{-9}$	7.76E-2	N/A	1.59E-1	N/A
$2^{-10}$	1.26E-2	6.17E+0	9.39E-2	1.69E+0
$2^{-11}$	5.85E-3	2.15E+0	4.82E-2	1.95E+0
$2^{-12}$	2.82E-3	2.07E+0	2.44E-2	1.98E+0
$2^{-13}$	1.39E-3	2.04E+0	1.23E-2	1.99E+0
$2^{-14}$	6.85E-4	2.02E+0	6.16E-3	1.99E+0

Table 3. Double Pendulum. Error results for  $\gamma = 3/4$ ,  $\beta = \frac{(\gamma+1/2)^2}{4}$ .

## 7 CONCLUSIONS

The Newmark method used in structural dynamics was adapted in this paper for the numerical solution of index three differential algebraic equations of multi-body dynamics. Strategies for corrector stopping criteria, and error and step-size control were introduced, and a formal proof for local convergence of the method was presented in detail. A set of numerical experiments confirmed the expected convergence order. A forthcoming paper [30] will present an analysis of the global convergence behavior of the method, and compare it with a BDF type integrator. The proposed method has been implemented in the 2005 version of the commercial simulation package MSC.ADAMS and led to a significant reduction in the simulation time for large models containing flexible bodies and/or contact elements.

## 8 ACKNOWLEDGMENT

The work of the first and third author was supported in part by the National Science Foundation, Grant No. 0139701. The second author would like to thank Nicolae Orlandea whose passion and enthusiasm is a continuous source of motivation.

## REFERENCES

- [1] Newmark, N. M., 1959. "A method of computation for structural dynamics". *Journal of the Engineering Mechanics Division, ASCE*, pp. 67–94.
- [2] Ascher, U. M., and Petzold, L. R., 1998. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, PA.
- [3] Hilber, H. M., Hughes, T. J. R., and Taylor, R. L., 1977. "Improved numerical dissipation for time integration algorithms in structural dynamics". *Earthquake Eng. and Struct. Dynamics*, **5**, pp. 283–292.
- [4] Chung, J., and Hulbert, G. M., 1993. "A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- $\alpha$  method". *Transactions of ASME, Journal of Applied Mechanics*, **60**(2), pp. 371–375.
- [5] Featherstone, R., 1983. "The calculation of robot dynamics using articulated-body inertias". *International Journal of Robotics Research*(1), pp. 13–30.
- [6] de Jalon, J. G., Unda, J., and Avello, A. "Natural coordinates for the computer analysis of multibody systems". *Computer Methods in Applied Mechanics and Engineering*(56), pp. 309–327.
- [7] Haug, E. J., 1989. *Computer-Aided Kinematics and Dynamics of Mechanical Systems*. Prentice-Hall, Englewood Cliffs, New Jersey.
- [8] de Jalon, J. G., and Bayo, E., 1994. *Kinematic and Dynamic Simulation of Multi-body Systems. The real time challenge*. Springer-Verlag, Berlin.
- [9] Shabana, A. A., 1994. *Computational Dynamics*. John Wiley & Sons.
- [10] Brenan, K. E., Campbell, S. L., and Petzold, L. R., 1989. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland, New York.
- [11] Petzold, L. R., 1982. "Differential-algebraic equations are not ODE's". *SIAM J. Sci., Stat. Comput.*, **3**(3).
- [12] Hairer, E., and Wanner, G., 1991. *Solving Ordinary Differential Equations*, Vol. II of *Computational Mathematics*. Springer-Verlag.
- [13] Potra, F. A., 1993. "Implementation of linear multi-step methods for solving constrained equations of motion". *SIAM. Numer. Anal.*, **30**(3).
- [14] Lubich, C., Nowak, U., Pohle, U., and Engstler, C., 1995. "MEXX - numerical software for the integration of constrained mechanical multibody systems". *Mechanics of Structures and Machines*, **23**, pp. 473–495.
- [15] Eich-Sollner, E., and Fuhrer, C., 1998. *Numerical Methods in Multibody Dynamics*. Teubner-Verlag, Stuttgart.
- [16] Cardona, A., and Geradin, M., 1994. "Numerical integration of second order differential-algebraic systems in flexible mechanics dynamics". In *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, M. F. O. S. Pereira and J. A. C. Ambrosio, eds., Kluwer Academic Publishers.
- [17] Yen, J., Petzold, L., and Raha, S., 1996. A time integration algorithm for flexible mechanism dynamics: the DAE  $\alpha$ -method. Tech. Rep. TR96-024, Dept. of Comp. Sci., University of Minnesota.
- [18] Cuadrado, J., Dopico, D., Naya, M. N., and Gonzalez, M., 2004. "Penalty, semi-recursive and hybrid methods for MBS Real-Time dynamics in the context of structural integrators". *Multibody Systems Dynamics*(12), pp. 117–132.
- [19] Baumgarte, J., 1972. "Stabilization of constraints and integrals of motion in dynamical systems". *Computer Methods in Applied Mechanics and Engineering*, **1**, pp. 1–16.
- [20] Owren, B., and Simonsen, H., 1995. "Alternative integration methods for problems in structural dynamics". *Structural Dynamics. Comput. Meth. in Appl. Mech. and Eng.*, **122**, pp. 1–10.
- [21] Jay, L. O., 1998. "Structure preservation for constrained dynamics with super partitioned additive runge-kutta methods". *SIAM J. Sci. Comput.*, **20**, pp. 416–446.
- [22] Schaub, M., and Simeon, B., 2002. "Automatic h-scaling for the efficient time integration of stiff mechanical systems". *Multibody System Dynamics*, **8**, pp. 329–345.
- [23] Dennis, J., and Schnabel, R., 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ.
- [24] MSCsoftware, 2005. ADAMS User Manual.
- [25] Zienkiewicz, O., and Xie, Y., 1991. "A simple error estimator and adaptive time-stepping procedure for dynamic analysis". *Earthquake Engineering and Structural Dynamics*, **20**, pp. 871–887.
- [26] Atkinson, K. E., 1989. *An introduction to numerical analysis*, second ed. John Wiley & Sons Inc., New York.
- [27] Hughes, T. J. R., 1987. *Finite Element Method - Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey.
- [28] Negrut, D., Haug, E. J., and German, H. C., 2003. "An implicit Runge-Kutta method for integration of Differential-Algebraic Equations of Multibody Dynamics". *Multibody System Dynamics*, **9**(2), pp. 121–142.
- [29] Oghbaei, M., and Anderson, K., 2005. "A new implicit integration scheme for the simulation of stiff-complex mechanical systems-DETC2005-84471". In *Proceedings to the Fifth ASME International Conference on Multibody Systems, Nonlinear Dynamics and Control*, A. Shabana and K. Anderson, eds., ASME.
- [30] Negrut, D., Gavrea, B., and Potra, F., 2005. "Multibody dynamics simulation with Newmark integrators". In *Preparation*.