Applied Numerical Mathematics 10 (1992) 397–413
North-Holland

APNUM 349

ODAE methods for the numerical solution of Euler–Lagrange equations

Linda R. Petzold

Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, USA

Florian A. Potra

Department of Mathematics, University of Iowa, Iowa City, IA 52242, USA

Abstract

Petzold, L.R. and F.A. Potra, ODAE methods for the numerical solution of Euler–Lagrange equations, Applied Numerical Mathematics 10 (1992) 397–413.

In a series of recent papers [5–7] it is shown that many of the numerical methods for solving Euler–Lagrange equations can be viewed as generalized solutions of an overdetermined differential-algebraic equation (ODAE). For a model with linear constraints, it is shown that if the ODAE is solved by a certain iteration technique in conjunction with BDF discretization, then the corresponding numerical solution coincides with the numerical solution obtained by applying the same BDF scheme to a state-space form of the original Euler–Lagrange equation. In addition, it is shown that using this iteration to solve the ODAE is equivalent to numerically solving a DAE which arises from extending the mechanical system by adding derivatives of the constraints and additional Lagrange multipliers to ensure that those constraints are satisfied. In this paper we examine these equivalences to determine to what extent they continue to hold for problems which are more general than the linear model.

Keywords. Euler–Lagrange equations; differential-algebraic equations; overdetermined differential-algebraic equations; multibody systems.

1. Introduction

In the last years we have witnessed a renewed interest in the field of kinematics and dynamics of multibody systems. Several methods for the automatic generation of the equations of motion of very complex systems have been developed. Some of them are already integrated in general-purpose computer codes (see e.g. [11]). Also real-time simulators for mechanical systems have recently been built [4]. These developments have increased the need for fast and reliable numerical methods for solving the equations of motion of constrained mechanical systems. These equations, often called the Euler–Lagrange equations, are a classical example

Correspondence to: L.R. Petzold, Department of Computer Science, University of Minnesota, 200 Union Street SE, Rm. 4-192, Minneapolis, MN 55455, USA. E-mail: petzold@cs.umn.edu.

of mixed differential-algebraic equations (DAEs). A systematic study of numerical algorithms for solving DAEs has begun only in the 1980s (see [3]).

In a series of recent papers [5–7] it is shown that many of the numerical methods for solving Euler–Lagrange equations can be viewed as *generalized solutions* of an overdetermined differential-algebraic equation (ODAE). For a special linear model, it is shown that if the ODAE is solved by a special oblique projection technique in conjunction with Newton's method and BDF discretization, then the corresponding numerical solution coincides with the numerical solution obtained by applying the same BDF scheme to a certain *state-space form* of the original Euler–Lagrange equation. This motivates the name of *ssf-iteration* given to the numerical method. In addition, for the linear model it is shown that using the ssf-iteration to solve the ODAE is equivalent to numerically solving a DAE which arises from extending the mechanical system by adding derivatives of the constraints and additional Lagrange multipliers to ensure that those constraints are satisfied [8,9].

The objective of this paper is to examine the equivalences shown in [5–7] to determine to what extent they continue to hold for problems which are more general than the linear model. In Section 2 we introduce the equations of motion of constrained multibody systems. In Section 3 we review some terminology on the solution of ODAE systems, and explore extended DAE systems which have been proposed in [5–7], and a new formulation which has also been considered, in a more general framework, in [1]. In Section 4 we develop some basic results for state-space forms. Finally, in Section 5 we examine the relationship between numerical methods based on solving the state-space form, the ODAE, the extended DAEs, and methods based on local parameterization, and determine the extent to which they are equivalent. In particular, we show that the tangent plane parameterization methods introduced in [16,17] are equivalent to the method of projected invariants for a certain projection, which was suggested recently in [1] to overcome numerical stability problems associated with the extended DAEs in [5–7]. These two methods are in turn equivalent to a certain solution of the ODAE.

Before proceeding further, it is useful to make some notational conventions. We denote by $\mathbb{R}^n$ the n-dimensional Euclidian space. An element $x \in \mathbb{R}^n$ is a column vector of the form $x = (x_1, x_2, \ldots, x_n)^T$. Let $y = (y_1, y_2, \ldots, y_m)^T$ be an element of $\mathbb{R}^m$. Then we define the augmented vector

$$(x, y) = (x^T, y^T)^T = (x_1, \ldots, x_n, y_1, \ldots, y_m)^T \in \mathbb{R}^{n+m}.$$

If $\sigma : \mathbb{R}^{n+m} \to \mathbb{R}^k$ is a differentiable mapping with components $\sigma_j : \mathbb{R}^{n+m} \to \mathbb{R}$, $j = 1, \ldots, k$, i.e.

$$\sigma(x, y) = (\sigma_1(x, y), \ldots, \sigma_k(x, y))^T,$$

then the Jacobian matrix of σ with respect to x is defined as

$$D_x\sigma(x, y) = \begin{pmatrix} \dfrac{\partial}{\partial x_1}\sigma_1(x, y) & \cdots & \dfrac{\partial}{\partial x_n}\sigma_1(x, y) \\ \vdots & & \vdots \\ \dfrac{\partial}{\partial x_1}\sigma_k(x, y) & \cdots & \dfrac{\partial}{\partial x_n}\sigma_k(x, y) \end{pmatrix}.$$

The Jacobian matrix $D_y\sigma(x, y)$ is defined in a similar way. For simplicity, we write $D\sigma(x, y)$ to denote $D(x, y)\sigma(x, y)$, the Jacobian matrix with respect to all variables.

2. Equations of motion of constrained multibody systems

The equations of motion of constrained multibody systems can be written in the following general form

$$M(t, p)\ddot{p} + G(t, p)^{\mathrm{T}}\lambda = f(t, p, \dot{p}), \tag{2.1a}$$

$$g(t, p) = 0. \tag{2.1b}$$

Here the *position vector* $p \in \mathbb{R}^{n_p}$ contains the *generalized coordinates* chosen to describe the multibody system. The number n_p of these coordinates as well as their physical interpretation depends on the particular formulation. Correspondingly, $M(t, p)$ is a symmetric $n_p \times n_p$ matrix called the *generalized mass matrix*, $\lambda \in \mathbb{R}^{n_\lambda}$ is the vector of *Lagrange multipliers* or *constraint forces* coupled to the system, $f(t, p, \dot{p})$ is the vector of *generalized external forces*, $g : \mathbb{R} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_\lambda}$ is the mapping defining the kinematic (position-level) constraints, and $G(t, p) = D_p g(t, p)$ is the Jacobian matrix of $g(t, p)$ with respect to p.

For a given constrained mechanical system there are many equations of the form (2.1) that accurately describe the motion of the system. The dimension, physical interpretation, and the structure of the quantities entering (2.1) may vary significantly from one formulation to another. For example with some formulations, like the *Cartesian formulation* (see [10]), the mass matrix $M(t, p)$ will be rather sparse, which is not the case with most other formulations.

By differentiating twice the kinematic constraints (2.1b) we obtain

$$G(t, p)\dot{p} + D_t g(t, p) = 0, \tag{2.2}$$

$$G(t, p)\ddot{p} + z(t, p, \dot{p}) = 0, \tag{2.3}$$

where

$$z(t, p, \dot{p}) = \big(D_p G(t, p)\dot{p}\big)\dot{p} + 2\big(D_t G(t, p)\big)\dot{p} + D_{tt} g(t, p).$$

It is known that, if

$$\text{Rank } G(t, p) = n_\lambda, \tag{2.4a}$$

$$a^{\mathrm{T}} M(t, p) a > 0, \quad \forall a \in \text{Ker } G(t, p), \tag{2.4b}$$

then the Euler–Lagrange equation (2.1) has a unique maximally extended solution for any consistent initial values $p(0)$, $\dot{p}(0)$, $\lambda(0)$ (see [10]). Clearly any such solution will also satisfy the constraints (2.2) and (2.3). The assumption (2.4) implies that the matrix

$$\begin{pmatrix} M(t, p) & G(t, p)^{\mathrm{T}} \\ G(t, p) & 0 \end{pmatrix} \tag{2.5}$$

is nonsingular. Therefore the linear system in $\ddot{p}$ and λ formed by (2.1a) and (2.3) has a unique solution

$$\ddot{p} = \eta(t, p, \ddot{p}), \tag{2.6}$$

$$\lambda = \zeta(t, p, \dot{p}). \tag{2.7}$$

For any p_0, $\dot{p}_0 \in \mathbb{R}^{n_p}$, the ordinary differential equation (ODE) (2.6) has a unique solution satisfying $p(0) = p_0$ and $\dot{p}(0) = \dot{p}_0$. If, in addition, the initial values p_0 and $\dot{p}_0$ are such that the constraints (2.1b) and (2.2) are satisfied for $t = 0$, $p = p_0$, and $\dot{p} = \dot{p}_0$, then this unique solution, together with λ given by (2.7) form a solution of the Euler–Lagrange equations (2.1). Therefore, theoretically, the problem of solving the Euler–Lagrange equations (2.1) is reduced to the problem of solving the ODE (2.6) with consistent initial values. However it has been observed that the numerical solution obtained by applying standard ODE solvers to (2.6) no longer satisfies the kinematic constraints (2.1b). This *drift off the constrained manifold* is not admissible in most practical applications. The challenge is therefore to construct reliable methods for obtaining an approximate solution of the Euler–Lagrange equations which satisfies the kinematic constraints within any given tolerance. Some such methods will be discussed in the next section.

3. DAEs and ODAEs

The Euler–Lagrange equations discussed above are second-order equations. It is convenient to rewrite them as first-order by introducing the variable $v = \dot{p}$. Equations (2.1), (2.2), and (2.3) become

$$\dot{p} - v = 0, \tag{3.1a}$$

$$M(t, p)\dot{v} + G(t, p)^{\mathrm{T}}\lambda - f(t, p, v) = 0, \tag{3.1b}$$

$$G(t, p)\dot{v} + z(t, p, v) = 0, \tag{3.1c}$$

$$G(t, p)v + D_t g(t, p) = 0, \tag{3.1d}$$

$$g(t, p) = 0. \tag{3.1e}$$

Let us consider the vector $x = (p, v, \lambda)$ of dimension $n_x = 2n_p + n_\lambda$ and let us denote by $F(t, x, \dot{x})$ the vector defined by the left-hand sides of (3.1a), (3.1b), and (3.1e). Then the Euler–Lagrange equations can be written as

$$F(t, x, \dot{x}) = 0. \tag{3.2}$$

Any equation of the form (3.2) with F a given mapping of $\mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ into $\mathbb{R}^{n_x}$ is called a mixed differential-algebraic equation (DAE). On the other hand the system (3.1) can be written in the form

$$\Psi(t, x, \dot{x}) = 0, \tag{3.3}$$

where Ψ is a mapping of $\mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ into $\mathbb{R}^{\gamma}$ with $\gamma = n_x + 2n_\lambda$. Any equation of the form (3.3) with $\Psi : \mathbb{R}^{2n_x+1} \to \mathbb{R}^{\gamma}$, $\gamma > n_x$, is called an overdetermined differential-algebraic equation (ODAE). The latter terminology was introduced in [5–7].

In our case (3.3) is obtained from (3.2) by differentiating twice equation (3.1e). In general, we say that the ODAE (3.3) is *generated* by the DAE (3.2) if all equations of (3.2) are contained in (3.3) and the other equations of (3.3) are obtained by differentiating some of the equations of (3.2) with respect to t.

From this definition it follows that, if (3.3) is generated by (3.2), then $x = x(t)$ is a solution of (3.2) if and only if it is a solution of (3.3).

The index of a general DAE of the form (3.2) is defined as *the minimum number of times that all, or part of* (3.2) *must be differentiated with respect to t in order to determine* $\dot{x}$ *as a continuous function of* x *and* t (cf. [3, p. 17]). The index of a general ODAE of the form (3.3) can be defined in a similar way. In our case, i.e. where (3.2) represents the Euler–Lagrange equations (3.1a), (3.1b), and (3.1e), and condition (2.4) is satisfied, the index is 3. Indeed, in the previous section we have obtained (2.6) and (2.7) by differentiating (2.1b) twice and rearranging. One more differentiation of (2.7) with respect to t, combined with (2.6), will explicitly express $\dot{x}$ as a function of x and t. Using the same argument we deduce that the ODAE (3.3) (hence (3.1)) has index 1.

The differentiation of constraints is a standard method for reducing the index of a DAE. However in order to keep the system *square*, i.e. to stay within the DAE framework, some of the original constraints have to be discarded. Therefore the numerical solution of the lower-index DAE may severely violate those constraints. Moreover the lower-index DAE, while containing all solutions of the original DAE, may have other solutions as well (cf. [3]). As we have seen above this is not the case if we work with lower-index ODAEs generated by the original DAE.

Of course there are many ways in which a lower-index ODAE can be generated by a DAE. For example in [6] one considers the following ODAE generated by the Euler–Lagrange equations (3.2) (hence (3.1a), (3.1b), and (3.1e)), under the supplementary assumption that the matrix $M(t, p)$ is symmetric positive-definite (SPD):

$$\dot{p} - v = 0, \tag{3.4a}$$

$$M(t, p)\dot{v} + G(t, p)^{\mathrm{T}}\lambda - f(t, p, \dot{p}) = 0, \tag{3.4b}$$

$$G(t, p)M(t, p)^{-1}f(t, p, v) - H(t, p)\lambda + z(t, p, v) = 0, \tag{3.4c}$$

$$G(t, p)v + D_t g(t, p) = 0, \tag{3.4d}$$

$$g(t, p) = 0, \tag{3.4e}$$

where

$$z(t, p, v) = \big(D_p G(t, p)v\big)v + 2\big(D_t G(t, p)\big)v + D_{tt} g(t, p), \tag{3.5a}$$

$$H(t, p) = G(t, p)M(t, p)^{-1}G(t, p)^{\mathrm{T}}. \tag{3.5b}$$

The equations of (3.1) and (3.4) coincide with the exception of the middle one: (3.1c) is a differential equation while (3.4c) is an algebraic equation. Equation (3.4c) is obtained by solving (3.1b) for $\dot{v}$ and then substituting into (3.1c). We note that if $M(t, p)$ is symmetric positive-definite, then so is $H(t, p)$. It is convenient to write (3.4) in the form

$$\hat{\Psi}(t, x, \dot{x}) = 0, \tag{3.6}$$

where the mapping $\hat{\Psi}: \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{\gamma}$, $\gamma = n_x + 2n_\gamma$, is defined by the left-hand side of (3.4).

The equations (3.3) and (3.6) are overdetermined but consistent because they are satisfied by any solution of (2.1). This is no longer the case for their discretized versions. For example, by applying a BDF scheme to (3.3) we replace the differentiation operator $\dot{x}(t_n) = D_t x(t_n)$ by the corresponding difference operator $\dot{x}_n = \rho x_n$ defined by

$$\dot{x}_n := \rho x_n := \frac{\alpha_0}{h} x_n + \bar{\rho} x_n := \frac{\alpha_0}{h} x_n + \frac{1}{h} \sum_{i=1}^{k} \alpha_i x_{n-i}, \tag{3.7}$$

where the α_i are the BDF coefficients and h is the time step used in the integration (i.e. $t_n = t_{n-1} + h$). Assuming that x_{n-i}, $i = 1, \ldots, k$, are known, the current value x_n of the numerical solution of (3.3) is obtained as the "solution" of the nonlinear system

$$\psi(x_n) := \Psi(t_n, x_n, \rho x_n) = 0. \tag{3.8}$$

The same scheme applied to (3.6) defines x_n as the "solution" of

$$\hat{\psi}(x_n) := \hat{\Psi}(t_n, x_n, \rho x_n) = 0. \tag{3.9}$$

Both (3.8) and (3.9) are overdetermined nonlinear algebraic systems and they will in general be inconsistent so that they have no (classical) solution. Formally Newton's method applied to (3.8) consists of "solving" the linear system in Δx

$$D\psi\left(x_n^{(m)}\right) \Delta x + \psi\left(x_n^{(m)}\right) = 0, \tag{3.10}$$

and setting

$$x_n^{(m+1)} = x_n^{(m)} + \Delta x. \tag{3.11}$$

If we apply Newton's method to (3.9), then the linear system (3.10) is replaced by

$$D\hat{\psi}\left(x_n^{(m)}\right) \Delta x + \hat{\psi}\left(x_n^{(m)}\right) = 0. \tag{3.12}$$

Both (3.10) and (3.12) are overdetermined inconsistent linear systems. In [5] the notion of the ssf-solution of an overdetermined linear system is introduced, by requiring that the residual of the system lies in certain lower-dimensional subspaces. More precisely we say that s^* is the ssf-solution of the overdetermined linear system

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} s = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \tag{3.13}$$

where A_1 is an $N \times N$ matrix, and A_2 is an $M \times N$ matrix with $M < N$, if the following conditions are satisfied

$$A_1 s^* - b_1 \in \text{Range } A_2^{\mathrm{T}}. \tag{3.14a}$$

$$A_2 s^* = b_2. \tag{3.14b}$$

It is proved that the above conditions determine s^* uniquely, provided that the matrix $V_2^{\mathrm{T}} A_1 V_2$ is nonsingular, where V_2 is a matrix whose columns form a basis for the null space of A_2. The systems (3.10) and (3.11) are of the form (3.13) with A_1 being the sub-Jacobian corresponding to the first three equations in (3.1) and (3.4) respectively.

Suppose that Δx is chosen to be the ssf-solution of the linear system (3.10) and that the sequence $\{x_n^{(m)}\}_{m=0}^{\infty}$ defined by (3.10)–(3.11) converges to x_n^*. Then x_n^* is called the *ssf-solution of the nonlinear system* (3.8). We also say that x_n^* is the *BDF ssf-solution of the ODAE* (3.1). The ssf-solution of the nonlinear system (3.9) or, equivalently, the BDF ssf-solution of the ODAE (3.4) is defined in a similar way.

Another way of "squaring" the ODAEs (3.1) and (3.4) is by introducing additional multipliers and transforming them into DAEs of increased dimension. Thus a new multiplier μ is associated with (3.1e) and (3.4e) and another new multiplier τ is associated with (3.1d) and (3.4d). The introduction of μ and τ is equivalent to a projection of the differential equations, as described in [5–7]. We note that the Jacobian with respect to p of the latter equation may be written as

$$L(t, p, v) := D_p(G(t, p)v) + D_t G(t, p). \tag{3.15}$$

Corresponding to (3.1) we have the DAE:

$$\dot{p} - v + G(t, p)^{\mathrm{T}}\mu + L(t, p, v)^{\mathrm{T}}\tau = 0, \tag{3.16a}$$

$$M(t, p)\dot{v} + G(t, p)^{\mathrm{T}}\lambda + G(t, p)^{\mathrm{T}}\tau - f(t, p, v) = 0, \tag{3.16b}$$

$$G(t, p)\dot{v} + z(t, p, v) = 0, \tag{3.16c}$$

$$G(t, p)v + D_t g(t, p) = 0, \tag{3.16d}$$

$$g(t, p) = 0. \tag{3.16e}$$

Similarly we associate with the ODAE (3.4) the following DAE:

$$\dot{p} - v + G(t, p)^{\mathrm{T}}\mu + L(t, p, v)^{\mathrm{T}}\tau = 0, \tag{3.17a}$$

$$M(t, p)\dot{v} + G(t, p)^{\mathrm{T}}\lambda + G(t, p)^{\mathrm{T}}\tau - f(t, p, v) = 0, \tag{3.17b}$$

$$G(t, p)M(t, p)^{-1}f(t, p, v) - H(t, p)\lambda + z(t, p, v) = 0, \tag{3.17c}$$

$$G(t, p)v + D_t g(t, p) = 0, \tag{3.17d}$$

$$g(t, p) = 0. \tag{3.17e}$$

In both (3.16) and (3.17), $z(t, p, v)$ is defined as in (3.5a). For holonomic constraints (3.16) was considered in [5], while (3.17) was considered in [6,7].

Clearly any solution of (3.1) and (3.4) gives a solution of (3.16) and (3.17) with $\mu = \tau = 0$. In what follows we show that any solution $(p, v, \lambda, \mu, \tau)$ of (3.17) must have $\mu = \tau = 0$, while (3.16) may have solutions with μ and/or τ different from zero. This contradicts the claim made in the proof of [5, Theorem 3.1]. However, we note that this version of the equations does not appear in the later paper [7].

Lemma 3.1. *The DAE system* (3.16) *may have solutions with* μ *and/or* τ *different from zero.*

Proof. By differentiating (3.16d) we obtain

$$G(t, p)\dot{v} + D_t G(t, p)(v + \dot{p}) + D_p(G(t, p)v)\dot{p} + D_{tt} g(t, p) = 0,$$

and by using (3.5a) and (3.15) we get

$$G(t, p)\dot{v} + L(t, p, v)(\dot{p} - v) + z(t, p, v) = 0. \tag{3.18}$$

Differentiation of (3.16e) gives

$$G(t, p)\dot{p} + D_t g(t, p) = 0. \tag{3.19}$$

By subtracting (3.16d) from (3.19) we have

$$G(t, p)(\dot{p} - v) = 0, \tag{3.20}$$

so that premultiplication of (3.16a) by $G(t, p)$ yields the equation

$$G(t, p)G(t, p)^{\mathrm{T}}\mu + G(t, p)L(t, p, v)^{\mathrm{T}}\tau = 0. \tag{3.21}$$

Also by comparing (3.16c) and (3.18) we get

$$L(t, p, v)(\dot{p} - v) = 0, \tag{3.22}$$

and by premultiplying (3.16a) by $L(t, p, v)$ we obtain

$$L(t, p, v)G(t, p)^{\mathrm{T}}\mu + L(t, p, v)L(t, p, v)^{\mathrm{T}}\tau = 0. \tag{3.23}$$

From (3.21) and (3.23) we deduce that, if the matrix

$$\begin{pmatrix} G(t, p) \\ L(t, p, v) \end{pmatrix}\begin{pmatrix} G(t, p) \\ L(t, p, v) \end{pmatrix}^{\mathrm{T}} \tag{3.24}$$

is nonsingular, then any solution $(p, v, \lambda, \mu, \tau)$ of (3.16) has $\mu = \tau = 0$, and therefore $x = (p, v, \lambda)$ is a solution of (3.1). In this case the index of (3.16) is 2. In case the matrix (3.24) is singular, then for any solution μ and τ of the system (3.21) and (3.23) we have

$$G(t, p)^{\mathrm{T}}\mu = L(t, p, v)^{\mathrm{T}}\tau = 0. \tag{3.25}$$

It follows that $x = (p, v, \lambda + \tau)$ is a solution of (3.1). □

Lemma 3.2. *Any solution of* (3.17) *must have* $\mu = \tau = 0$ (*for M symmetric positive-definite*).

Proof. For $M(t, p)$ invertible, (3.17c) is equivalent to saying that

$$\lambda = \zeta(t, p, v), \tag{3.26}$$

where ζ is the function defined in (2.7). By premultiplying (3.17b) with $G(t, p)M(t, p)^{-1}$ and using (3.17c) we obtain

$$G(t, p)\dot{v} + H(t, p)\tau + z(t, p, v) = 0. \tag{3.27}$$

If we differentiate (3.17d), (3.17e), and (3.26) we obtain (3.18), (3.19), and

$$\dot{\lambda} = D_t\zeta(t, p, v) + D_p\zeta(t, p, v)\dot{p} + D_v\zeta(t, p, v)\dot{v}. \tag{3.28}$$

As before, it follows that μ and τ satisfy equation (3.21). Also by subtracting (3.18) from (3.27) we obtain

$$L(t, p, v)(\dot{p} - v) - H(t, p)\tau = 0, \tag{3.29}$$

and by substituting $\dot{p} - v$ from (3.17a) we get

$$L(t, p, v)G(t, p)^{\mathrm{T}}\mu + \big(H(t, p) + L(t, p, v)L(t, p, v)^{\mathrm{T}}\big)\tau = 0. \tag{3.30}$$

The matrix of the linear system (3.21) and (3.30) is nonsingular whenever $G(t, p)$ has full rank. This can be seen by noticing that $H(t, p)$ is symmetric positive-definite in this case, and by using a Schur complement of the nonsingular block $G(t, p)G(t, p)^{\mathrm{T}}$ in the matrix of this system. According to (2.4a) it follows that $\dot{p} = v$ and $\dot{v} = \eta(t, p, v)$ where η is the function from (2.6). Substituting in (3.28) we obtain

$$\dot{\lambda} = \psi(t, p, v). \tag{3.31}$$

Thus, any solution $(p, v, \lambda, \mu, \tau)$ of the DAE (3.17) has $\mu = \tau = 0$ and $x = (p, v, \lambda)$ is a solution of (3.4). □

There are a number of practical problems associated with the numerical solution of (3.17). In the construction of (3.17) we have assumed that the matrix $M(t, p)$ is invertible. With some formulations of the Euler–Lagrange equations the matrix $M(t, p)$ satisfies (2.4b) but may be singular [14]. Moreover, even for nonsingular $M(t, p)$, (3.17c) is expensive and/or inconvenient to evaluate because it requires a decomposition of M on every function call. (Although symbolic techniques are available for solving linear systems involving the matrix M in $O(n)$ operations [20], their implementation requires considerable expertise, and the resulting function calls are still relatively expensive.)

Another problem for the numerical solution of (3.17) has to do with stability. In [1], it is shown that for linear DAEs with the structure of (3.17), the stepsize of numerical methods such as BDF and many implicit Runge–Kutta may need to be restricted to maintain stability. This can happen if for example $M(t, p)$ has both large and small eigenvalues (a corresponding physical situation is a heterogeneous multibody system, i.e. a system which includes bodies of very different masses [21]). The stability restriction arises because the direct discretization of the DAE by these methods corresponds to a discretization of an *essential underlying ODE* (*EUODE*) [1] [1], where one of the terms has been discretized *explicitly*. If this term is large (usually, this causes the EUODE to be stiff), the stepsize must be restricted in order to maintain numerical stability. In [1] it is shown that under minimal assumptions there is no problem with stability for numerical methods applied to semi-explicit Hessenberg [3] systems, [2] provided that the matrix which is used for the projection (in (3.17) this is the matrix which multiplies the Lagrange multipliers μ and τ) is the transpose of the constraint matrix. Unfortunately, this is not the case for (3.17), because of the term in (3.17b) which multiplies τ. Indeed, examples are given in [1] of systems of the form (3.17) where the stepsize of BDF must be restricted to maintain stability.

An additional practical difficulty for (3.17) is the evaluation of the term $L^{\mathrm{T}}\tau$ in (3.17a). The evaluation of L requires the analytical derivatives of G. These are not generally available, and

[1] The essential underlying ODE is a set of ODEs of minimal dimension which describes the motion of the system. In the terminology of the next section, it is a state-space form induced by the matrix A, where A is a basis for the null space of G.

[2] To bring the system (3.17) to Hessenberg form, solve (3.17c) for λ and insert into (3.17b), then multiply (3.17b) by M^{-1}. The constraints are (3.17c) and (3.17d).

their generation requires substantial software and additional expense. The following formulation works without the L-term. However for some problems with oscillations, inclusion of the L-term appears to be desirable [2].

To overcome these difficulties, the DAE can be formulated as follows:

$$\dot{p} - v + G(t, p)^{\mathrm{T}}\mu = 0, \tag{3.32a}$$

$$M(t, p)\dot{v} + G(t, p)^{\mathrm{T}}\lambda + M(t, p)G(t, p)^{\mathrm{T}}\tau - f(t, p, v) = 0, \tag{3.32b}$$

$$G(t, p)\dot{v} + G(t, p)G(t, p)^{\mathrm{T}}\tau + z(t, p, v) = 0, \tag{3.32c}$$

$$G(t, p)v + D_t g(t, p) = 0, \tag{3.32d}$$

$$g(t, p) = 0. \tag{3.32e}$$

This method has been also suggested recently in [1], where it is a special case of the method called *projected invariants*. A straightforward extension of the stability analysis in [1] shows that the stepsize of BDF applied to this formulation does not need to be restricted to maintain numerical stability, even if the constraints become nearly rank-deficient, provided that G satisfies certain minimal assumptions (i.e. if $G^{\mathrm{T}} = U'\Sigma V^{\mathrm{T}}$ is the singular value decomposition of G^{T}, then U' must be of moderate size). In Section 5 we will see that BDF applied to the projected invariants formulation is equivalent to the tangent plane parameterization method proposed in [16,17], provided that the parameterization is updated continuously, and that these methods are in turn equivalent to a certain solution of the ODAE. Finally, we see below that the system (3.32) is well-defined.

Lemma 3.3. *Any solution of* (3.32) *must have* $\mu = \tau = 0$ (*for M invertible*).

Proof. By differentiating (3.32d) and (3.32e) we obtain again (3.18) and (3.19). It follows that (3.20) holds in our case too. Then by premultiplying (3.32a) by $G(t, p)$ we obtain that

$$G(t, p)G(t, p)^{\mathrm{T}}\mu = 0. \tag{3.33}$$

Hence, according to (2.4a) it follows that $\mu = 0$. Now, (3.32a) implies that $\dot{p} = v$ and from (3.18) it follows that

$$G(t, p)\dot{v} + z(t, p, v) = 0. \tag{3.34}$$

By substituting in (3.32c) we deduce the same as before, that the second additional multiplier τ must also be equal to zero. □

In the next section we discuss the notion of a local state-space form for general nonlinear DAEs. Using the framework developed in [16,17] we construct local state-space forms for nonlinear Euler–Lagrange equations that satisfy (2.4). The numerical solution obtained by applying a BDF scheme to the local state-space form may be interpreted as a "generalized solution" of (3.8) and (3.9).

4. State-space forms

The concept of solvability of a general DAE of the form (3.2) is defined in [3]. That definition has in some sense a local character, but it can be easily "globalized" as follows:

Definition 4.1. Let $\mathcal{T}$ be an open interval of $\mathbb{R}$, Ω a connected open subset of $\mathbb{R}^{2n_x+1}$, and F a smooth mapping from Ω to $\mathbb{R}^{n_x}$. Then (3.2) is solvable on $\mathcal{T}$ in Ω if there is a family of solutions $\xi(t, c)$, $c \in \mathcal{C}$, where $\mathcal{C}$ is an r-dimensional manifold such that:

(1) $\xi(t, c)$ is defined on all of $\mathcal{T}$ for each $c \in \mathcal{C}$,
(2) $(t, \xi(t, c), D_t\xi(t, c)) \in \Omega$ for $(t, c) \in \mathcal{T} \times \mathcal{C}$;
(3) if $x(t)$ is any other solution of (3.2) with $(t, x(t), \dot{x}(t)) \in \Omega$, then $x(t) = \xi(t, c)$ for some $c \in \mathcal{C}$;
(4) the graph of ξ as a function of (t, c) is an $(r+1)$-dimensional manifold.

In what follows we will always assume that $0 \in \mathcal{T}$. If the manifold $\mathcal{C}$ has an atlas composed of a single chart, i.e. if there is an open set $\tilde{\Omega} \subset \mathbb{R}^r$ and a diffeomorphism $\theta : \tilde{\Omega} \to \mathcal{C}$, then the above definition reduces to the definition given in [3, p. 16]. If (3.2) is an ODE, then clearly $r = n_x$ and $\mathcal{C} = \mathbb{R}^{n_x}$. In general $r < n_x$ and $\mathcal{C}$ can be viewed as a representation of the manifold of *consistent initial values*; r represents the number of "differential variables", and $n_x - r$ the number of "algebraic variables". For example, if (3.2) (equivalently (3.1a), (3.1b), and (3.1e)) are the Euler–Lagrange equations, then the manifold $\mathcal{C}$ may be defined as

$$\begin{aligned}\mathcal{C} = \{c = (p_0, v_0, \lambda_0) \in \mathbb{R}^{n_x} :\ & g(0, p_0) = 0, \\ & G(0, p_0)v_0 + D_t g(0, p_0) = 0, \\ & \lambda_0 = \zeta(0, p_0, v_0)\},\end{aligned} \tag{4.1}$$

where ζ is the function of (2.7). In this case $r = 2(n_p - n_\lambda)$ and the manifold $\mathcal{M}$ considered in (4) of the above definition reduces to

$$\begin{aligned}\mathcal{M} = \{(t, p, v, \lambda) :\ & g(t, p) = 0,\ G(t, p)v + D_t g(t, p) = 0, \\ & \lambda = \xi(t, p, v)\}.\end{aligned} \tag{4.2}$$

We note that the manifold $\mathcal{C}$ defined by (4.1) is the *initial data manifold* considered in [16]. Following [13] we define the local state-space form of a DAE as follows:

Definition 4.2. We say that (3.2) has a *local state-space form* at a point $(t_c, x_c) \in \mathcal{M}$ if there is an open neighborhood $\mathcal{T}_c \subset \mathcal{T}$ of t_c, an open neighborhood $\mathcal{U} \subset \mathbb{R}^r$ of 0, an open neighborhood $\mathcal{V} \subset \mathbb{R}^{n_x}$ of x_c, a diffeomorphism

$$\hat{\Phi} : \mathcal{T}_c \times \mathcal{U} \to \mathcal{W} := (\mathcal{T}_c \times \mathcal{V}) \cap \mathcal{M}$$

of the form

$$\hat{\Phi} = (t, \phi(t, u)),$$

and a Lipschitz continuous mapping

$$\omega : \mathcal{T}_c \times \mathcal{U} \to \mathbb{R}^r$$

such that:

(1) If $(t_0, x_0) \in \mathscr{W}$ and if $u = u(t)$ is the solution of the initial value problem (IVP)

$$\dot{u} = \omega(t, u), \qquad u(t_0) = u_0, \tag{4.3}$$

$$(t_0, u_0) = \hat{\Phi}^{-1}(t_0, x_0),$$

then $x = x(t) := \phi(t, u(t))$ is a solution of (3.2) satisfying $x(t_0) = x_0$.

(2) If $x = x(t)$ is any smooth solution of (3.2), then for any $t_0 \in \mathscr{T}_c$ there is a $u_0 \in \mathscr{U}$ such that $x(t) = \phi(t, u(t))$ with $u = u(t)$ the solution of the IVP (4.3). The pair (Φ, ω) is called a *local* state-space form of (3.2) at (t_c, x_c). If $\mathscr{T}_c = \mathscr{T}$ and $\mathscr{W} = \mathscr{M}$, then we say that the state-space form (ϕ, ω) is *global*.

In what follows we show that the class of local parameterizations considered in [16,17] provide local state-space forms at any $(t_c, x_c) \in \mathscr{M}$. Although the construction is rather general, for the purpose of this paper we restrict ourselves to the case where (3.2) are the Euler–Lagrange equations (3.1a), (3.1b) and (3.1e). Together with the manifold $\mathscr{M}$ given by (4.2) we also consider the $(n_p - n_\lambda)$-dimensional manifold $\mathscr{N}$ defined by the original kinematic constraints (2.1b):

$$\mathscr{N} = \{(t, p) \in \mathscr{T} \times \mathbb{R}^{n_p} : g(t, p) = 0\}. \tag{4.4}$$

The following result is proved in [17].

Proposition 4.3. *Suppose that all the mappings M, f, and g in (2.1) are of class C^ρ, $\rho \geq 2$ (i.e. they are ρ-times continuously differentiable), and assume that (2.4) is satisfied. Let $(t_c, p_c) \in \mathscr{N}$ and let A be an $n_p \times n_\lambda$ matrix such that the augmented square matrix*

$$\begin{pmatrix} G(t_c, p_c) \\ A^{\mathrm{T}} \end{pmatrix} \tag{4.5}$$

is nonsingular.

Then there is an open neighborhood $\mathscr{U}_1$ of the origin of $\mathbb{R}^{n_p - n_\lambda}$, an open neighborhood $\mathscr{V}_1 \subset \mathbb{R}^{n_p}$ of p_c, and an open neighborhood $\mathscr{T}_c \subset \mathscr{T}$ of t_c such that for any $q \in \mathscr{U}_1$ and $t \in \mathscr{T}_c$ the nonlinear system

$$g(t, p) = 0, \tag{4.6a}$$

$$A^{\mathrm{T}}(p - p_c) = q \tag{4.6b}$$

has a unique solution

$$p = \phi_1(t, q) \in \mathscr{V}_1, \tag{4.7}$$

where $(t, p) \in \mathscr{N}$. Moreover the mapping

$$\Phi_1 : \mathscr{T}_c \times \mathscr{U}_1 \to (\mathscr{T}_c \times \mathscr{V}_1) \cap \mathscr{N},$$

$$\Phi_1(t, q) = (t, \phi_1(t, q)) \tag{4.8}$$

is a $C^{\rho-1}$-diffeomorphism.

Proof. Without loss of generality we may assume that the neighborhoods $\mathcal{T}_c$, $\mathcal{U}_1$, and $\mathcal{V}_1$ from Proposition 4.3 are such that the matrix

$$\begin{pmatrix} G(t, p) \\ A^T \end{pmatrix} \tag{4.9}$$

is nonsingular for all $t \in \mathcal{T}_c$ and $p \in \mathcal{V}_1$.

Let (t_c, p_c) be as in Proposition 4.3, and let v_c and λ_c be such that $(t_c, p_c, v_c, \lambda_c) \in \mathcal{M}$. Then for any $t \in \mathcal{T}_c$, $q \in \mathcal{U}_1$, and $s \in \mathbb{R}^{n_p - n_\lambda}$ the linear system

$$\begin{pmatrix} G(t, \phi_1(t, q)) \\ A^T \end{pmatrix} v = \begin{pmatrix} -D_t g(t, \phi_1(t, q)) \\ s \end{pmatrix} \tag{4.10}$$

has a unique solution

$$v = \phi_2(t, q, s). \tag{4.11}$$

Let ζ be the function from (2.7) and denote

$$\phi_3(t, q, s) = \zeta(t, \phi_1(t, \phi_1(t, q), \phi_2(t, q, s)), \tag{4.12a}$$

$$u = (q, s), \qquad \mathcal{U} = \mathcal{U}_1 \times \mathbb{R}^{n_p - n_\lambda}, \qquad \mathcal{V} = \mathcal{V}_1 \times \mathbb{R}^{n_p + n_\lambda}, \tag{4.12b}$$

$$\phi(t, u) = (\phi_1(t, q), \phi_2(t, q, s), \phi_3(t, q, s)), \tag{4.12c}$$

$$\Phi(t, u) = (t, \phi(t, u)). \tag{4.12d}$$

Then the mapping $\Phi : \mathcal{T}_c \times \mathcal{U} \to (\mathcal{T}_c \times \mathcal{V}) \cap \mathcal{M}$ is a C^{p-2}-diffeomorphism. Clearly

$$(t, u) = \Phi^{-1}(t, p, v, \lambda), \quad u = (q, s) \tag{4.13}$$

if and only if

$$q = A^T(p - p_c), \qquad s = A^T v. \tag{4.14}$$

Now let $x = x(t)$ be a solution of (3.2) and let $t_0 \in \mathcal{T}_c$. Clearly $x(t) \in \mathcal{M}$ for $t \in \mathcal{T}_c$. Taking eventually a smaller $\mathcal{T}_c$ we may assume that $x(t) \in \mathcal{M} \cap (\mathcal{T}_c \times \mathcal{V})$ for $t \in \mathcal{T}_c$. Let us define the functions

$$q(t) = A^T(p(t) - p_c), \tag{4.15a}$$

$$s(t) = A^T v(t), \tag{4.15b}$$

$$u(t) = (q(t), s(t)). \tag{4.15c}$$

This is equivalent to

$$x(t) = \phi(t, u(t)), \quad t \in \mathcal{T}_c. \tag{4.16}$$

It is easily seen that the function $u = u(t)$ defined by (4.15) is the solution of an initial value problem of the form (4.3), where

$$\omega(t, u) = \left(s, A^T \eta(t, \phi_1(t, q), \phi_2(t, q, s))\right)^T \tag{4.17}$$

with η the mapping defined in (2.6) and $x_0 = x(t_0)$. Conversely, let $(t_0, x_0) \in \mathcal{V}$ and let $u = u(t)$ be the solution of the initial value problem (4.3). Then the function $x = x(t)$ defined by

(4.16) is the solution of (3.2) which satisfies $x(t_0) = x_0$. Thus the pair (ϕ, ω) constructed above is a local state-space form of (3.2) in the sense of Definition 4.2. □

This state-space form is called the *standard first-order local state-space form induced by the matrix A*. (Henceforth we will call this the state-space form induced by A.) This name is motivated by the fact that condition (4.5) uses only the first derivative of the kinematic constraints. In [18], state-space forms have been considered which use the second derivatives of the kinematic constraints as well. However, we will not consider these here because it is unclear whether they are advantageous.

5. Numerical integration

In the remainder of this paper (3.2) will always denote the Euler–Lagrange equations (3.1a), (3.1b) and (3.1e). Let $x = x(t)$ be a solution of this equation and let $x_c = x(t_c)$ be a "current point" on this solution. Let (Φ, ω) be the local state-space form of (3.2) at (t_c, x_c) considered in the previous section, and let $u = u(t)$ be the solution of the initial value problem (4.3) with $(t_0, x_0) = (t_c, x_c)$. Then (4.15) is satisfied. Let h be a positive real number and N a positive integer such that

$$t_j = t_c + jh \in \mathcal{T}_c, \quad j = 0, 1, \ldots, N.$$

Assume that for some n, $1 \leqslant n \leqslant N$, some approximations

$$u_j = (q_j, s_j) \approx u(t_j), \quad j = 0, 1, \ldots, n-1,$$

have already been computed. The BDF scheme (3.7) applied to (4.3) defines the new approximation $u_n = (q_n, s_n)$ as the solution of the nonlinear system

$$\rho q_n = s_n, \tag{5.1a}$$

$$\rho s_n = A^{\mathrm{T}} \eta(t_n, \phi_1(t_n, q_n), \phi_2(t_n, q_n, s_n)). \tag{5.1b}$$

We note that (5.1) defines the solution by BDF of the state-space form induced by A. If

$$x_j = (p_j, v_j, \lambda_j) = \phi(t_j, u_j), \tag{5.2}$$

then

$$q_j = A^{\mathrm{T}}(p_j - p_c), \qquad s_j = A^{\mathrm{T}} v_j. \tag{5.3}$$

Substituting in (5.1), noting that the BDF coefficients satisfy $\alpha_0 + \alpha_1 + \cdots + \alpha_k = 0$, and using (4.6)–(4.12) we deduce that, *for A constant*, p_n and v_n are uniquely determined by the following nonlinear system:

$$A^{\mathrm{T}}(\rho p_n - v_n) = 0,$$

$$A^{\mathrm{T}}(\rho v_n - \eta(t_n, p_n, v_n)) = 0,$$

$$G(t_n, p_n) v_n + D_t g(t_n, p_n) = 0,$$

$$g(t_n, p_n) = 0.$$

The quantity $w_n = \eta(t_n, p_n, v_n)$ is obtained as the solution of a linear system (see (2.6) and (2.7)). It follows that $(p_n, v_n, \lambda_n, w_n)$ is the solution of the augmented nonlinear system

$$A^{\mathrm{T}}(\rho p_n - v_n) = 0, \tag{5.4a}$$

$$A^{\mathrm{T}}(\rho v_n - w_n) = 0, \tag{5.4b}$$

$$M(t_n, p_n)w_n + G(t_n, p_n)^{\mathrm{T}}\lambda_n - f(t_n, p_n, v_n) = 0, \tag{5.4c}$$

$$G(t_n, p_n)w_n + z(t_n, p_n, v_n) = 0, \tag{5.4d}$$

$$G(t_n, p_n)v_n + D_t g(t_n, p_n) = 0, \tag{5.4e}$$

$$g(t_n, p_n) = 0. \tag{5.4f}$$

Equations (5.4) define the solution by BDF of a class of local parameterizations discussed in [16,17]. For A given by a set of unit vectors chosen such that

$$\begin{pmatrix} G \\ A^{\mathrm{T}} \end{pmatrix}$$

is nonsingular, this is equivalent to solving the state-space form by the method of generalized coordinate partitioning [22]. For A which is a basis for the null space of $G(t_n, p_n)$, this is the tangent plane parameterization method introduced in [16,17].

Now, (5.4) is a system of $3n_p + n_\lambda$ equations in $3n_p + n_\lambda$ unknowns which, under our assumptions, has a unique solution satisfying $(t_n, p_n, v_n, \lambda_n) \in \mathscr{W}$. This solution may be viewed as a *generalized* solution of the overdetermined system (3.8). To see this, let us denote $x_n = (p_n, v_n, \lambda_n)$ and consider the residual

$$(r_1, r_2, r_3, r_v, r_p) = \psi(x_n), \tag{5.5}$$

where r_1, r_2, r_3, r_v, and r_p correspond to (3.1a), (3.1b), (3.1c), (3.1d), and (3.1e), respectively. We have clearly

$$r_1 \in \operatorname{Ker} A^{\mathrm{T}}, \tag{5.6a}$$

$$\begin{pmatrix} r_2 \\ r_3 \end{pmatrix} \in \begin{pmatrix} M(t_n, p_n) \\ G(t_n, p_n) \end{pmatrix} \operatorname{Ker} A^{\mathrm{T}}, \tag{5.6b}$$

$$(r_v, r_p) = 0. \tag{5.6c}$$

It is easily proved that under our assumptions there is a unique x_n satisfying (5.5)–(5.6) and $(t_n, x_n) \in \mathscr{W}$. Thus the solution of (5.4) may be considered a "generalized solution" of (3.8). It may also be viewed as a "generalized" solution of the overdetermined system (3.9). Indeed by solving (5.4c) for w_n and substituting in (5.4d) we obtain

$$G(t_n, p_n)M(t_n, p_n)^{-1} f(t_n, p_n, v_n) - H(t_n, p_n)\lambda_n + z(t_n, p_n, v_n) = 0.$$

If follows that the residual

$$(\hat{r}_1, \hat{r}_2, \hat{r}_\lambda, \hat{r}_v, \hat{r}_p) = \hat{\psi}(x_n) \tag{5.7}$$

satisfies

$$\hat{r}_1 \in \text{Ker } A^T, \tag{5.8a}$$

$$\hat{r}_2 \in M(t_n, p_n) \text{ Ker } A^T, \tag{5.8b}$$

$$\left(\hat{r}_\lambda, \hat{r}_v, \hat{r}_p\right) = 0. \tag{5.8c}$$

Again it can be shown that x_n is uniquely defined by (5.7)–(5.8) and $(t_n, x_n) \in \mathscr{W}$. Any x_n satisfying (5.7)–(5.8) is called an A-ssf solution of the ODAE (3.9). Thus we see that the local parameterization methods are equivalent to the A-ssf solution of the ODAE (3.9).

Let us choose the matrix A such that

$$\text{Range } A = \text{Ker } G(t_c, p_c). \tag{5.9}$$

Using (5.6) it follows that $x_n = (p_n, v_n, \lambda_n)$ is a solution of (5.4) if and only if there are uniquely determined multipliers $\mu_n, \tau_n \in \mathbb{R}^{n_\lambda}$ such that:

$$\rho p_n - v_n + G(t_c, p_c)^T \mu_n = 0, \tag{5.10a}$$

$$M(t_n, p_n)\rho v_n + G(t_n, p_n)^T \lambda_n + M(t_n, p_n)G(t_c, p_c)^T \tau_n - f(t_n, p_n, v_n) = 0, \tag{5.10b}$$

$$G(t_n, p_n)\rho v_n + G(t_n, p_n)G(t_c, p_c)^T \tau_n + z(t_n, p_n, v_n) = 0, \tag{5.10c}$$

$$G(t_n, p_n)v_n + D_t g(t_n, p_n) = 0, \tag{5.10d}$$

$$g(t_n, p_n) = 0. \tag{5.10e}$$

For $t_c = t_n$, (5.10) defines the solution by BDF of the method of projected invariants, for a certain projection.

6. Conclusions

To summarize, we note that for the model problem in [5–7], all of the methods we have discussed are equivalent because it is implicitly assumed there that the constraint matrix G is constant. However, in the general case where $G = G(t, p)$, there are important differences in some of the formulations which can dramatically effect the performance of numerical methods. Certain methods are equivalent, and seem to offer the best possibilities for the foundation of robust solution methodology.

For A constant, as in the method of generalized coordinate partitioning [22], the following methods are equivalent: solution of the local state-space form induced by A, local parameterization based on A, and solving the A-ssf of the ODAE. A robust implementation of these methods depends on algorithms for detecting the need for updating the coordinate basis A.

For A which is not constant but which forms a basis for the null space of $G(t, p)$, and for BDF or other similar numerical methods, the tangent plane parameterization (5.4) [16,17] (where the parameterization is updated continuously) is equivalent to the A-ssf solution of the ODAE (3.9) and the method of projected invariants (5.10) [1] for a certain projection. For general G these methods are not equivalent to solving the state-space form, however they do define a local state-space form in the sense of Section 4. Since the method based on projected invariants has been shown [1] under reasonable assumptions to overcome the stability limita-

tions of methods based on previously defined extended DAEs [5–7], these equivalent methods seem to be among the best potential candidates for the foundation of a robust solution method. Efficient solution techniques for these types of systems in some of the above formulations are described in [15].

References

[1] U.M. Ascher and L.R. Petzold, Stability of computational methods for constrained dynamics systems, Tech. Rept. 91-3, Department of Computer Science, University of British Columbia, Vancouver, BC (1991); also *SIAM J. Sci. Statist. Comput.* (to appear).

[2] U.M. Ascher and L.R. Petzold, Projected collocation for higher-order higher-index differential-algebraic equations, *J. Comput. Appl. Math* 43 (1992).

[3] K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations* (Elsevier Science Publishers, Amsterdam, 1989).

[4] J. Drosdol and F. Panik, The Daimler-Benz driving simulator, a tool for vehicle development, SAE Tech. Paper Series 550334, Warrendale, PA (1985).

[5] C. Führer and B.J. Leimkuhler, Formulation and numerical solution of the equations of constrained mechanical motion, DFVLR-FB 89-09, Munich (1989).

[6] C. Führer and B.J. Leimkuhler, A new class of generalized inverses for the solution of discretized Euler–Lagrange equations, in: R. Deyo and E.J. Haug, eds., *NATO Advanced Research Workshop in Real-Time Integration Methods for Mechanical System Simulation* (Springer, Heidelberg, 1991) 143–154.

[7] C. Führer and B.J. Leimkuhler, Numerical solution of differential-algebraic equations for constrained mechanical motion, *Numer. Math.* 59 (1991) 55–69.

[8] C.W. Gear, Differential-algebraic equation index transformations, *SIAM J. Sci. Statist. Comput.* 9 (1988) 39–47.

[9] C.W. Gear, G.K. Gupta and B.J. Leimkuhler, Automatic integration of the Euler–Lagrange equations with constraints, *J. Comput. Appl. Math.* 12–13 (1985) 77–90.

[10] E.J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems, Vol. I: Basic Methods* (Allyn and Bacon, Boston, MA, 1989).

[11] E.J. Haug et al., *DADS User's Manual, Rev. 4.0* (Computer Aided Design Software, Inc., Oakdale, IA, 1987).

[12] E.J. Haug and J. Yen, Implicit numerical integration of constrained equations of motion via generalized coordinate partitioning, University of Iowa Center for Simulation and Design Optimization of Mechanical Systems, Tech. Rept. R-39, Iowa City, IA (1989).

[13] B.J. Leimkuhler, Some notes on perturbations of differential-algebraic equations, Helsinki University of Technology, Research Rept. A266 (1989).

[14] C. Lubich, Extrapolation integrators for constrained multibody systems, *IMPACT Comput. Sci. Engrg.* 3 (1991) 213–234.

[15] F.A. Potra, Multistep method for solving constrained equations of motion, University of Iowa, Department of Mathematics Report No. 6, Iowa City, IA (1991).

[16] F.A. Potra and W.C. Rheinboldt, Differential-geometric techniques for solving differential algebraic equations, in: R. Deyo and E.J. Haug, eds., *NATO Advanced Research Workshop in Real-Time Integration Methods for Mechanical System Simulation* (Springer, Heidelberg, 1991) 155–192.

[17] F.A. Potra and W.C. Rheinboldt, On the numerical solution of the Euler–Lagrange equations, *J. Mech. Structures and Machines* 19 (1) (1991) 155–192.

[18] F.A. Potra and J. Yen, Implicit integration for Euler–Lagrange equations via tangent space parameterization, *J. Mech. Structures and Machines* 19 (1) (1991) 77–98.

[19] W.C. Rheinboldt, On the existence and uniqueness of solutions of nonlinear semi-implicit algebraic equations, *Nonlinear Anal. Theory Methods Appl.* 10 (1991) 647–661.

[20] D.E. Rosenthal, An order n formulation for robotic systems, *J. Astronaut. Sci.* 38 (1990) 511–529.

[21] D.E. Rosenthal, Personal communication, RASNA Corp. (1991).

[22] R.A. Wehage and E.J. Haug, Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, *J. Mech. Design* 104 (1982) 247–255.