Numerische Mathematik, 59:55-69, 1991

Numerical Solution of Differential-Algebraic Equations for Constrained Mechanical Motion

C. Führer
DLR - Laboratory for Flight System Dynamics
German Aerospace Research Establishment
Oberpfaffenhofen
D-8031 Wessling (Germany)

B.J. Leimkuhler
Department of Mathematics
University of Kansas
Lawrence, KS 66045 (USA)

Abstract

The two most popular formulations of the equations of constrained mechanical motion, the *descriptor* and *state-space* forms, each have severe practical limitations. In this paper, we discuss and relate some proposed reformulations of the equations which have improved numerical properties.

Subject Classifications: AMS(MOS): 65L05; CR: G1.7

1 Introduction

We are concerned with the equations of constrained mechanical motion. The equations are typically developed from variational principles, and the resulting system, which can be considered a differential equation on a manifold, is mapped to an ordinary differential equation [2]. The mapping is not easily automated, nor necessarily desirable in the computational setting, while the original constrained statement of the problem is not solvable by current numerical software.

Two alternative formulations have recently been discussed in the literature, one based on an overdetermined system of equations including time derivatives of the constraints, and the other based on stabilization with respect to those differentiated constraints via additional Lagrange multipliers. These formulations lead to systems of equations which are more amenable to numerical computation than either of the two traditional models. In the sequel we discuss and relate the reformulations, in the context of numerical discretization.

Historically, mechanical systems have been formulated as Lagrange equations of the first or second kind. The Lagrangian equations of the first kind, derived directly from variational principles, are a constrained differential equation or a differential-algebraic equation (DAE):

$$\begin{aligned} \dot{p} &= v && (1)\\ M(p)\dot{v} &= f(p,v,t) - G^T(p)\lambda && (2)\\ 0 &= g(p) && (3) \end{aligned}$$

where $p, v \in \mathbf{R}^{n_p}$, $M(p)$ is a $n_p \times n_p$ regular (symmetric, positive definite) mass matrix, f is a vector of applied forces, and λ represents the n_λ Lagrange multipliers or constraint forces coupled to the system by the $n_\lambda \times n_p$ constraint matrix $G := \partial g/\partial p$. Such a system can be thought of as a differential equation on a manifold [24].

It is possible to rewrite the equations as an ordinary differential equation. First, we twice differentiate the constraint (3), yielding the n_λ constraint equations on *velocity level*

$$0 = G(p)v \tag{4}$$

and the n_λ constraint equations on the *acceleration level*

$$0 = G(p)\dot{v} + g''(p)(v,v) =: G(p)\dot{v} + z(p,v) \tag{5}$$

Equation (2) can be solved for $\dot{v}$. If the result is introduced into (5), this gives n_λ constraint equations involving the Lagrange multipliers λ:

$$0 = G(p)M^{-1}(p)f(p,v,t) - G(p)M^{-1}(p)G^T(p)\lambda + z(p,v) \tag{6}$$

It will be assumed that the systems considered have no redundant constraints, thus

$$H(p) := G(p)M^{-1}(p)G^T(p) \tag{7}$$

is a nonsingular $n_\lambda \times n_\lambda$ - matrix and (6) can be solved for the algebraic variable λ:

$$\begin{aligned} \dot{p} &= v && (8)\\ M(p)\dot{v} &= \tilde{f}(p,v,t) && (9) \end{aligned}$$

with

$$\tilde{f}(p,v,t) := f(p,v,t) - G(p)^T H(p)^{-1}\left[G(p)M^{-1}(p)f(p,v,t) + z(p,v)\right]$$

In this paper, we call the equation (9) the *underlying ODE* of the DAE, although technically that equation should involve a differential equation for λ as well. Given initial values lying on the position and velocity constraints, the solutions to this ODE are solutions of the original DAE. However, the difficulty in obtaining accurate initial values, and the presence of discretization errors in the numerical solution of the problem mean that the position constraints are not in general satisfied by the numerical solution of (9). The theoretical and practical behavior of the drift from the position constraints is unpredictable, and most authors recommend some sort of stabilization against the undifferentiated constraints [5, 20].

Alternatively, it may be possible to solve the constraints, (3), (4), and (5), together with the differential relations, for an ODE in *minimal* coordinates, whose solutions then "automatically" satisfy the various constraints. The resulting formulation is called, variously, the *state-space form* or the *Langrange equations of the second kind* and is derived in Section 2. Until recently, mathematical and numerical techniques for studying the solution of such problems depended on the transformation to state-space form [18, 2]. The practical determination of a global state-space form may be an exceptionally difficult problem, since the reduction depends on the solution of nonlinear functional equations and on the differentiation of those solutions.

Yet another approach is to attempt to discretize the equations of the descriptor form, *directly*; that is to say, to treat the problem as a DAE. Unfortunately, this class of so called "index-3" DAEs, is not solvable by state-of-the-art numerical software, although some theoretical work on variable step-size discretization schemes has recently appeared [14, 21].

In this paper we consider some alternatives to the traditional formulations of the equations of constrained motion. One approach, taken in [11] is to rewrite the equations to improve properties of numerical methods. A multiplier is introduced whose purpose is to insure that the velocity level is satisfied. This has the effect of reducing the index, which in turn leads to a computationally tractable DAE. At the same time, the position level constraints are maintained, so that the problem retains the stability features of the original equations. Another approach is taken by [7]. In this work, the first and second derivatives of the position constraints (the velocity and acceleration levels) are taken together with the original equations and constraints, yielding an overdetermined nonlinear system at each stage of the numerical integration. These formulations are mathematically equivalent in this sense: the exact solution of any of them solves the others when no discretization is involved.

We relate the behaviour of the various formulations under discretization. It is shown that the methods of [11] and [7] are mathematically equivalent, if the acceleration level is neglected in [7], and a certain pseudo-inverse is selected for the solution of the overdetermined system.

A numerical experiment is then explored which illustrates the ideas of the paper.

2 Differentiated Constraint Formulations and the State-Space Form

2.1 Differentiation of Constraints

A usual way to discuss the influence of using the time derivative of a constraint instead of the original one is to look at the residual of the original constraint. For the problem (1)-(2), (4) it can easily be seen that the residual of the position constraint is $g(p(t)) \equiv \epsilon_0$ when the initial conditions satisfy

$$0 = G(p_0)v_0 \qquad \text{and} \qquad \epsilon_0 = g(p_0).$$

Furthermore, if the problem has been transformed into an index-1 DAE by replacing the constraint (3) by (5) or (6) we will see a growth in the residual which is linear in time. This

observation has motivated many methods for numerically solving the equations of motion by integrating the corresponding reduced index system together with an additional correcting step, which is performed if the residual in the position level constraint exceeds a certain level, see e.g. [22].

Here we want to give an example, which demonstrates that the growth of the residual of the original constraint is qualitatively different from the growth of the error in the solution components.

Example 2.1 *The index-1 DAE*

$$\begin{aligned} x' &= y - ax^2 + u(t) \\ y &= ax^2 \end{aligned} \tag{10}$$

has the state-space form $x' = u(t)$ *and has the solution* $x(t) = x_0 + U(t)$ *with* $U(t) := \int_0^t u(\tau)d\tau$ *and* $x_0 = x(0)$.

If, on the other hand, the index is reduced by differentiating the constraint, we obtain

$$\begin{aligned} x' &= y - ax^2 + u(t) \\ y' &= 2ax(y - ax^2 + u(t)) \end{aligned} \tag{11}$$

with the solution

$$\begin{aligned} x(t) &= \epsilon_0 t + U(t) + x_0 \\ y(t) &= \epsilon_0 + a\left(\epsilon_0 t + U(t) + x_0\right)^2 \end{aligned} \tag{12}$$

with $\epsilon_0 := y_0 - ax_0^2$.

The two problems, while formally having the same solutions when initial data are consistent, have quite different numerical properties. For example, the spectrum of the linearized underlying ODE contains extraneous positive and negative eigenvalues which may have an impact on the behavior of numerical methods; and the error constants will be different for the two problems. In [1] and [5], more detailed discussion of these issues can be found. Here we simply demonstrate how the numerical solution depends on the parameter a when a popular computational scheme is employed.

In Table 1, we have summarized the behavior of DASSL when applied (i) directly to the DAE (10), (ii) to the state-space form $x' = u(t)$ (SSF) and (iii) to the index-reduced form (11); all tests were on $[0, 10\pi]$, with $a = 200$, $u(t) = \cos t$, and the tolerances were chosen to roughly equalize the error at the right endpoint.

2.2 Reduction to State-Space Form

For general nonlinear mechanical systems the state-space form can only be established locally and the reduction process must be carried out in every integration step. It requires the following computations:

Table 1: DASSL Applied to the Problem of Example 2.1

Formulation	Error at 10π		Number of Steps	Number of Jacobians
	x	y		
(10)	9.6×10^{-6}	4.4×10^{-3}	253	20
SSF	8.9×10^{-6}	3.6×10^{-3}	284	24
(11)	9.0×10^{-6}	3.6×10^{-3}	3718	201

- Elimination of constraint forces λ.
- Specification of independent position variables y and independent velocity variables $\dot{y}$.
- Elimination of the corresponding dependent variables w and $\dot{w}$.

The first step makes use of constraint (6) and results in the ODE (9).

In order to describe the second step we assume that p is partitioned into $n_y := n_p - n_\lambda$ variable y and n_λ variables w: $p^T = [y^T, w^T]^T$ and that for this partition $\det \frac{\partial g}{\partial w}(p) \neq 0$ holds in a neighbourhood of a particular solution point p_0. Then, by the implicit function theorem there exists locally a function h with $w = h(y)$ and $g([y^T, h(y)^T]^T) = 0$. Furthermore h is continuously differentiable and

$$\dot{p} = \begin{bmatrix} I \\ h_y(y) \end{bmatrix} \dot{y} =: V_1(y)\dot{y} \tag{13}$$

is a solution of (4) in a neighbourhood of p_0. A further differentiation with respect to time yields a similar expression for the accelerations:

$$\ddot{p} = V_1(y)\ddot{y} + \frac{\partial}{\partial y}(V_1(y)\dot{y})\dot{y} =: V_1(y)\ddot{y} + \zeta(y, \dot{y}) \tag{14}$$

Premultiplying (9) by V_1^T, using the notation $y_1 := y$ and $y_2 := \dot{y}$ and replacing p, v by the relations derived above gives:

$$\dot{y_1} = y_2 \tag{15}$$

$$V_1(y_1)^T \tilde{M}(y_1) V_1(y_1)\dot{y_2} = V_1(y_1)^T \tilde{f}(y_1, y_2, t) \tag{16}$$

with $\tilde{M}(y_1) := M([y_1^T, h(y_1)^T]^T)$ and $\tilde{f}$ being defined analoguously.

This is the *local state space form* of system (1),(2),(3) which is sometimes refered to as the Lagrange equations of the second kind.

In some papers, a technique has been proposed for solving the equations of motion which simply neglects the position constraints and solves the reduced index (and sometimes, reduced complexity) problems which result from combining the acceleration or velocity levels with the differential relations. The justification presented in those papers is that the drift from the position constraints is only linear or perhaps quadratic. However, in light of the comments of the previous subsection and Example 2.1, this strategy is probably ill-considered. *Any sound numerical method for solving the equations of motion in the constrained form should be stabilized by insuring that the position constraints are satisfied.* In the following sections, we describe and relate two such methods.

3 Stabilized Descriptor Form Approach

The state space form (15), (16) depends on the simultaneous solution of the constraint equation (3) and its first and second derivatives with respect to time.

In contrast, the 'classical' descriptor form approaches use only one type of constraint equation resulting in an index-3, index-2 or index-1 DAE. For the determination of consistent initial values, all three types of constraint equations are used in [16].

In [25, 30] it was proposed to use the constraints and also one or more derivatives in the DAE formulation. This approach results in an overdetermined system of differential-algebraic equations:

$$\begin{aligned} \dot{p} &= v \\ M(p)\dot{v} &= f(p,v,t) - G^T(p)\lambda \\ 0 &= g(p) \\ 0 &= G(p)v \\ 0 &= G(p)M^{-1}(p)f(p,v,t) - H(p)\lambda + z(p,v) \end{aligned} \tag{17}$$

(cf. eqs. (1) - (4), (6), (7)).

For consistent initial values $x(t_0)^T := \left[p(t_0)^T, v(t_0)^T, \lambda(t_0)^T\right]$ and with the usual smoothness requirements the overdetermined system has a unique solution which is identical with the solution of the system (1) - (3). We are interested in the behavior of such an approach under discretization.

Let us consider a typical time step from t_{n-1} to $t_n := t_{n-1} + h$. Applying a multistep method replaces $\dot{p}(t_n)$ by its approximation $\dot{p}_n$ which is defined by

$$\dot{p}_n = \frac{\alpha_0}{h} p_n + \frac{1}{h} \underbrace{\sum_{i=1}^{k} \alpha_i p_{n-i}}_{=:\bar{\rho}_n(p)} - \underbrace{\sum_{i=1}^{k} \beta_i \dot{p}_{n-i}}_{=:\bar{\sigma}_n(\dot{p})} \tag{18}$$

This and the corresponding expression for $\dot{v}(t_n)$ transforms the overdetermined DAE (17) to the following overdetermined problem of nonlinear algebraic equations:

$$\varphi(x_n) = 0$$

with

$$\varphi(x_n) := \begin{bmatrix} \frac{\alpha_0}{h} p_n + \frac{1}{h}\bar{\rho}_n(p) - \bar{\sigma}_n(\dot{p}) - v_n \\ M(p_n)(\frac{\alpha_0}{h} v_n + \frac{1}{h}\bar{\rho}_n(v) - \bar{\sigma}_n(\dot{v})) - f(p_n, v_n, t_n) + G^T(p_n)\lambda_n \\ g(p_b n) \\ G(p_n)v_n \\ G(p_n)M^{-1}(p_n)f(p_n, v_n, t_n) - H(p_n)\lambda_n + z(p_n, v_n) \end{bmatrix} \tag{19}$$

and $x_n = \left[p_n^T, v_n^T, \lambda_n^T\right]^T$. This system defines the corrector step of an integration method for overdetermined DAEs.

Due to the approximation error involved when applying formula (18) this system may become inconsistent and has in general no (classical) solution. A Newton-like method can be defined based on iteration of the linear systems:

$$\varphi'(x_n^{(m)})x_\Delta = -\varphi(x_n^{(m)}) \text{ with } x_n^{(m+1)} := x_n^{(m)} + x_\Delta \tag{20}$$

A traditional way to solve this system is to define a solution in terms of the Moore-Penrose pseudoinverse. This leads to a Gauss-Newton iteration. In the context of overdetermined differential-algebraic equations this approach has been discussed in [30, 7]. It makes no use of the special structure of the problem and nothing is known about the propagation of the errors.

A related approach can be motivated by the projection technique for ODEs with invariants [29] (see also [10]). The constraints (3) and (4) may be considered as invariants to the index-1 problem (1),(2),(5). The projection technique then leads to a *constrained least squares formulation* for the solution of (20), see [7].

In this section a different approach will be described and investigated. This method defines a solution of (20) by exploiting the special structure of the problem. In [7] it was shown that for linearly constrained problems this approach is equivalent to numerically integrating the state space form with the same implicit multistep scheme.

In matrix form, the system (20) becomes:

$$\begin{bmatrix} \frac{\alpha_0}{h}I & -I & 0 \\ K & \frac{\alpha_0}{h}M + D & G^T \\ G & 0 & 0 \\ \Gamma & G & 0 \\ \Gamma_1 & \Gamma_2 & -H \end{bmatrix} \begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \end{bmatrix} = -\varphi(x_n) \tag{21}$$

with $K := \dot{v}_n^T M_p(p_n) - f_p(p_n, v_n, t_n) + \lambda_n^T G_p(p_n)$, $M := M(p_n)$, $G := G(p_n)$, $D := -f_v(p_n, v_n, t_n)$, $\Gamma := v_n^T g'(p_n)$, $\Gamma_1 := \frac{\partial}{\partial p}\left(G(p_n)M^{-1}(p_n)f(p_n, v_n, t_n) - H(p_n)\lambda_n + z(p_n, v_n)\right)$, $\Gamma_2 := \frac{\partial}{\partial v}\left(G(p_n)M^{-1}(p_n)f(p_n, v_n, t_n) - H(p_n)\lambda_n + z(p_n, v_n)\right)$ and the iteration counter m being suppressed for notational simplicity.

This system can be partitioned into a square, regular problem

$$A_1 \begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \end{bmatrix} = a_1 \tag{22}$$

with

$$A_1 := \begin{bmatrix} \frac{\alpha_0}{h}I & -I & 0 \\ K & \frac{\alpha_0}{h}M + D & G^T \\ \Gamma_1 & \Gamma_2 & -H \end{bmatrix} \text{ and } a_1 := -\begin{bmatrix} \varphi_1(x_n) \\ \varphi_2(x_n) \\ \varphi_5(x_n) \end{bmatrix}, \tag{23}$$

and an underdetermined system:

$$A_2 \begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \end{bmatrix} = a_2 \tag{24}$$

$$A_2 := \begin{bmatrix} G & 0 & 0 \\ \Gamma & G & 0 \end{bmatrix} \text{ and } a_2 := -\begin{bmatrix} \varphi_3(x_n) \\ \varphi_4(x_n) \end{bmatrix} \tag{25}$$

Let $A_2^\dagger$ denote the Moore-Penrose pseudoinverse of A_2, and let V be a matrix whose columns span the null space of A_2, then a general solution of (24) can be expressed by

$$\hat{x} = A_2^\dagger a_2 + Vy \tag{26}$$

with y being an arbitrary vector of length $\dim \ker(A_2) = 2n_p - n_\lambda$.

Definition 3.1 *A ssf-generalized solution of (21) is defined by*

$$x_{ssf} = \begin{bmatrix} \hat{p}_n \\ \hat{v}_n \\ \hat{\lambda}_n \end{bmatrix} := A_2^\dagger a_2 + V(V^T A_1 V)^{-1} V^T \left(a_1 - A_1 A_2^\dagger a_2\right), \tag{27}$$

*whenever $V^T A_1 V$ is nonsingular. The matrix $(A_1 V)^{ssf} := (V^T A_1 V)^{-1} V^T$ is called a ssf-pseudoinverse of $(A_1 V)$. (*ssf *stands for* state space form *due to the relation between this solution and the solution of the state space form shown in [7]).*

Remarks:

- The generalized solution of (21) defined by the constrained least-squares problem

$$\min_x ||A_1 x - a_1|| \text{ subject to } A_2 x = a_2 \tag{28}$$

  has a form similar to (27):

$$\tilde{x} = A_2^\dagger a_2 + V(A_1 V)^\dagger (a_1 - A_1 A_2^\dagger a_2)$$

  (see [6]).

- The ssf-pseudoinverse is a (1,2,4) - pseudoinverse in the sense of [4]. It has especially the "equation solving property", i.e.

$$A(A^{ssf} b) = b \text{ if } b \in \text{Range}(A)$$

- The ssf-generalized solution $\hat{x}$ is independent of the particular choice of V. In particular, V can be defined by

$$V = \begin{bmatrix} V_1 & 0 & 0 \\ -G^\dagger \Gamma V_1 & V_1 & 0 \\ 0 & 0 & I \end{bmatrix} \tag{29}$$

  with the columns of V_1 spanning the nullspace of G, c.f. (13).

- The matrix $V^T A_1 V$ is always nonsingular in the linear case. More generally, $V^T A_1 V$ is nonsingular if and only if $Q := A_2 A_1^{-1} A_2^T$ is nonsingular, and it can easily be shown that the ssf-generalized solution x_{ssf} may then be written as

$$x_{ssf} = A_1^{-1}(I - A_2^T Q^{-1} A_2 A_1^{-1}) a_1 + A_1^{-1} A_2^T Q^{-1} a_2 \tag{30}$$

  This representation, where V is not involved, may be advantageous.

The numerical properties of a method based on solving the ODAE with the ssf-generalized solution are shown by the following theorem:

Theorem 3.1 *Using the ssf-generalized inverse for solving the overdetermined DAE (17) is equivalent to numerically solving the extended problem*

$$\begin{aligned}
\dot{p} &= v - G^T(p)\mu - \left[v^T G_p(p)\right]^T \eta && (31)\\
M(p)\dot{v} &= f(p,v,t) - G^T(p)\lambda - G^T(p)\eta && (32)\\
0 &= g(p) && (33)\\
0 &= G(p)v && (34)\\
0 &= G(p)M^{-1}(p)f(p,v,t) - H(p)\lambda + z(p,v) && (35)
\end{aligned}$$

for the variables p, v and λ with the same multistep formula, provided a modified Newton iteration which neglects terms in the Jacobian involving μ and η is used to solve the nonlinear equations.

Proof:

For notational simplicity the proof will be shown for the BDF-case only ($\bar{\sigma}_n = 0$).

Clearly, system (31) - (35) has the same exact solution as (17) with the additional Lagrange multipliers being zero. For the corresponding discretized system these multipliers are in general no longer zero. At the nth integration step it becomes:

$$\begin{bmatrix}
\frac{\alpha_0}{h}p_n + \frac{1}{h}\bar{\rho}_n(p) - v_n + G^T(p_n)\mu_n + \left[v_n^T G_p(p_n)\right]^T \eta_n \\
M(p_n)(\frac{\alpha_0}{h}v_n + \frac{1}{h}\bar{\rho}_n(v)) - f(p_n,v_n,t_n) + G^T(p_n)\lambda_n + G^T(p_n)\eta_n \\
g(p_n) \\
G(p_n)v_n \\
G(p_n)M^{-1}(p_n)f(p_n,v_n,t_n) - H(p_n)\lambda_n + z(p_n,v_n)
\end{bmatrix} = 0 \qquad (36)$$

If a modified Newton scheme neglecting all terms involving μ and η in the iteration matrix is applied, we observe after some row permutations that the following linear system must be solved at any iteration step:

$$\begin{bmatrix}
\frac{\alpha_0}{h}I & -I & 0 & G^T & \Gamma^T \\
K & \frac{\alpha_0}{h}M + D & G^T & 0 & G^T \\
\Gamma_1 & \Gamma_2 & -H & 0 & 0 \\
G & 0 & 0 & 0 & 0 \\
\Gamma & G & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \\ \mu_\Delta \\ \eta_\Delta \end{bmatrix} = \qquad (37)$$

$$- \begin{bmatrix}
\frac{\alpha_0}{h}p_n + \frac{1}{h}\bar{\rho}_n(p) - v_n + G^T(p_n)\mu_n + \Gamma^T\eta_n \\
M(p_n)(\frac{\alpha_0}{h}v_n + \frac{1}{h}\bar{\rho}_n(v)) - f(p_n,v_n,t_n) + G^T(p_n)\lambda_n + G^T(p_n)\eta_n \\
G(p_n)M^{-1}(p_n)f(p_n,v_n,t_n) - H(p_n)\lambda_n + z(p_n,v_n) \\
g(p_n) \\
G(p_n)v_n
\end{bmatrix}$$

With A_1,A_2,a_1 and a_2 defined as above this linear equation can be partitioned as follows

$$\begin{bmatrix} A_1 & A_2^T \\ A_2 & 0 \end{bmatrix} \begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \\ \mu_\Delta \\ \eta_\Delta \end{bmatrix} = \begin{bmatrix} a_1 - \begin{bmatrix} G^T(p_n)\mu_n + \Gamma^T\eta_n \\ G^T(p_n)\eta_n \\ 0 \end{bmatrix} \\ a_2 \end{bmatrix} = \begin{bmatrix} a_1 - A_2^T \begin{bmatrix} \mu_n \\ \eta_n \end{bmatrix} \\ a_2 \end{bmatrix} \tag{38}$$

Solving the lower part for p_Δ, v_Δ and λ_Δ results in

$$\begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \end{bmatrix} = A_2^\dagger a_2 + Vy \tag{39}$$

with y being an arbitrary vector, which is determined by introducing this into the upper part. This gives:

$$A_1 A_2^\dagger a_2 + A_1 V y + A_2^T \begin{bmatrix} \mu_\Delta \\ \eta_\Delta \end{bmatrix} = \begin{bmatrix} a_1 - A_2^T \begin{bmatrix} \mu_n \\ \eta_n \end{bmatrix} \\ a_2 \end{bmatrix}$$

Multiplying this by V^T, solving for y and introducing the result in (39) gives

$$\begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \end{bmatrix} = A_2^\dagger a_2 + V(V^T A_1 V)^{-1} V^T \left(a_1 - A_1 A_2^\dagger a_2\right) \tag{40}$$

This is the ssf-generalized solution of (21).

The individual iteration steps of the two formulations have the same solution, so the methods are equivalent.□

If we scale all h- dependent equations in (36) by $\gamma = h/\alpha_0$ it can be shown by applying Theorem 12.6.4 from [19], that for sufficiently small stepsizes h the convergence of the method is not affected by neglecting the terms involving μ and η in the iteration matrix. Thus, also the ssf-iteration converges if A_1 is scaled in the same way.

The proof of the above theorem shows that the interesting components of the solutions of (31) - (35) can be numerically determined without computing the stabilizing Lagrange multipliers. It is also apparent that the only algebraic variable which needs to be computed is λ, which is an index-1 variable.

Finally, we would like to point out that the special choice of pseudo-inverse has a useful and interesting geometric interpretation.

Corollary 3.1 *The ssf-generalized solution of (21) is the solution of the constrained least squares problem*

$$\min_{A_1 x - a_1 \in Range(A_2^T)} ||A_1 x - a_1|| \quad \textit{subject to} \quad A_2 x = a_2 \tag{41}$$

Proof:

With z, which will be fixed later, y has the form

$$y = A_1 x - a_1 = A_2^T z.$$

Solving for x gives

$$x = A_1{}^{-1}(A_2^T z + a_1) \tag{42}$$

Inserting this into the constraint $A_2 x = a_2$ gives

$$A_2 A_1{}^{-1} A_2^T z = a_2 - A_2 A_1{}^{-1} a_1.$$

This can be solved for z. Introducing this solution in eq. (42) gives x_{ssf} in the representation (30). □

The space of admissible solutions of this variational problem actually contains only one point. Thus, if we had values p, v, λ satisfying $A_2 x - a_2 = 0$ and $A_1 x - a_1 \in Range(A_2^T)$, these values are the ssf-solution of the ODAE.

4 Stabilization by Additional Lagrange Multipliers

Reducing the index of a problem by differentiating and introducing additional Lagrange multipliers was suggested in [11]. When differentiating the constraint (3) only once, this leads to the following index-2 problem:

$$\begin{aligned} \dot{p} &= v - G^T(p)\mu && (43)\\ M(p)\dot{v} &= f(p,v,t) - G^T(p)\lambda && (44)\\ 0 &= g(p) && (45)\\ 0 &= G(p)v && (46) \end{aligned}$$

Again, this system has the solution p, v, λ, μ if and only if p, v, λ is a solution of (1) - (3) and $\mu = 0$.

Along the lines of the proof of Theorem 3.1 it is also possible to show an equivalence of this formulation with the overdetermined system

$$\begin{aligned} \dot{p} &= v \\ M(p)\dot{v} &= f(p,v,t) - G^T(p)\lambda \\ 0 &= g(p) \\ 0 &= G(p)v \end{aligned} \tag{47}$$

when the latter system is solved by BDF together with an ssf-iteration scheme.

Considering the linear equation in a typical corrector iteration step corresponding to (21), the partitioning leading to an (index-2) ssf-generalized inverse is given by the square, regular subsystem

$$\underbrace{\begin{bmatrix} \frac{\alpha_0}{h} I & -I & 0 \\ K & \frac{\alpha_0}{h} M + D & G^T \\ 0 & G & 0 \end{bmatrix}}_{=:A_1} \begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \end{bmatrix} = \underbrace{- \begin{bmatrix} \varphi_1(x_n) \\ \varphi_2(x_n) \\ \varphi_4(x_n) \end{bmatrix}}_{=:a_2} \tag{48}$$

and the underdetermined linear system

$$\underbrace{\begin{bmatrix} G & 0 & 0 \end{bmatrix}}_{=:A_2} \begin{bmatrix} p_\Delta \\ v_\Delta \\ \lambda_\Delta \end{bmatrix} = \underbrace{-\begin{bmatrix} \varphi_3(x_n) \end{bmatrix}}_{=:a_2} \tag{49}$$

With this the ssf-iteration can be defined for problem (47) as in the previous section and we can prove:

Theorem 4.1 *Using the ssf-generalized inverse for solving the overdetermined DAE (47) is equivalent to numerically solving the extended problem (43) - (46) for the variables p, v and λ if the same multistep formula are used.*

The proof is similar to the proof of Theorem 3.1 and is omitted.

Again, the system can be solved for the "interesting" solution components without computing the stabilizing Lagrange multipliers μ.

5 Computational Remarks and Numerical Example

Both stabilization techniques, the overdetermined DAE approach with the ssf-iteration and the introduction of additional Lagrange Multipliers require additional computational effort that depends on the number of algebraic constraints. Fortunately, modern mechanical formalisms for generating the equations of motion of complex mechanical systems eliminate most constraints (globally) during the generating procedure due to a special choice of coordinates [9]. These so-called multibody formalisms result in equations of the form (1) - (2) with a small number of additional algebraic constraints (3),(4) and(6).

For test and demonstration purposes we coupled a modified implementation, ODASSL, of DASSL (the "Differential-Algebraic Systems SoLver" [23]) which solves overdetermined systems with a mechanical multibody code, SIMPACK [27]. In ODASSL, the standard Newton corrector step in DASSL was replaced by the ssf-iteration scheme. The computational scheme for computing the ssf-generalized solution (27) is similar to the null space technique for solving constrained least squares problems, as suggested in [6]. For algorithmic details the reader is referred to [7].

It should be noted that alternatively the index-2 formulation (31)-(34) which uses additional Lagrange multipliers can be treated efficiently by modifying the usual corrector iteration schemes. Due to the large amount of structure available in this stabilized formulation the equations can be arranged into relatively decoupled subsystems (e.g. according to position and velocity levels). In [8], techniques are discussed and analyzed for exploiting this decoupling in (31)-(34) at the level of either the nonlinear equations or at the linear system level.

As a test example we use here the model of a mechanism consisting in seven rigid bodies interconnected by springs and dampers and Hooke's joints as shown in Fig. 1.

The motion of this system can be described by 14 first order nonlinear ODEs and six constraint equations. For a complete description of the mechanical model the reader is refered to [28] where this problem is used as a benchmark problem for multibody codes. Here we only give the results obtained, when integrating this model with SIMPACK together with the overdetermined DAE solver ODASSL over 0.03 secs (approx. 3 periods) with a requested (local) error tolerance bound RTOL=ATOL=10^{-5} on position and velocity variables and no error tolerance restrictions on the algebraic variables.

Table 2: Relative Error for a Simulation of the Mechanism

Method	Maximum Relative Error after 0.03 secs in		
	Position Variables	Velocity Variables	Lagrange Multipliers
Stab. Index 1 Form	$1.71\ 10^{-5}$	$1.92\ 10^{-3}$	$3.08\ 10^{-4}$
Stab. Index 2 Form	$1.38\ 10^{-4}$	$1.54\ 10^{-2}$	$1.45\ 10^{-3}$
Index 2 Form	$1.82\ 10^{-4}$	$2.95\ 10^{-1}$	$2.20\ 10^{-3}$

In Tab. 3 we show the computational effort of the different approaches for this example, when run on an APOLLO DOMAIN DN 590. In the table, NSTEPS = number of steps, NFE = number of function evaluations, NJE = number of evaluations of the Jacobian, NEF = number of error test failures. It should be noted that the Jacobians are of different sizes for the various formulations.

Table 3: Computational Effort of the Different Methods

Method	CPU (secs)	NSTEPS	NFE	NJE	NEF
Stab. Index 1 Form	401	382	978	78	20
Stab. Index 2 Form	348	434	1058	60	28
Index 2 Form	402	469	1145	77	30

6 Conclusion

The stabilized schemes advocated in this paper increase the size of the problem by a factor dependent on the number of constraints. On the other hand, they lead to numerically tractable DAEs with the same space of solutions as the original problem. A critical evaluation must weigh the increased computational effort for the stabilized approaches against the numerical and practical weaknesses of the traditional alternatives.

Acknowledgements

The authors would like to thank Professors Dahlquist, Nevanlinna and Rentrop and Dr. Schwertassek for their interest in this work and for many useful comments which have been incorporated in the text. We thank Professors Bulirsch and Grübel for their support during the formulation of these ideas. Also, we are indebted to Dipl. Ing. W. Rulka (MAN Neue Technologie) for his assistance with the mechanical example.

References

[1] Alishenas,T., Dahlquist, G.: personal communication. KTH Stockholm 1989

[2] Arnold, V.I.: Mathematical methods of classical mechanics. New York, Berlin, Heidelberg: Springer 1978

[3] Brenan, K.B., Campbell, S.L., Petzold, L.R.: Numerical solution of initial value problems in ordinary differential-algebraic equations. New York, Amsterdam, London: North Holland Publishing Co. 1989

[4] Campbell, S.L., Meyer, C.D.: Generalized inverses of linear transformations. Montreal: Pitman 1979

[5] Campbell, S.L., Leimkuhler, B.: Differentiation of constraints in differential-algebraic equations. Mechanics of Structures and Machines, to appear 1990

[6] Lawson, C.L., Hanson, R.J.: Solving Least squares problems. Englewood Cliffs: Prentice-Hall 1974

[7] Führer, C.: Differential-algebraische Gleichungssysteme in mechanischen Mehrkörpersystemen. TU München, Thesis, Mathematisches Institut, TU München, TUM-M 8807, August 1988

[8] Führer, C., Leimkuhler, B.J.: Formulation and numerical solution of the equations of constrained mechanical motion. Technical Report DFVLR-FB 89-08, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR), Köln 1989

[9] Führer, C., Schwertassek, R.: Generation and solution of multibody system equations. Int. Journal of Nonl. Mechanics **25** 127–141 (1990)

[10] Gear, C.W.: Maintaining solution invariants in the numerical solution of ode. SIAM J. Sci. Stat. Comput. **7** 734–743 (1986).

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

[12] Gear, C.W., Petzold, L.R.: ODE Methods for the solution of differential/ algebraic systems. Technical Report UIUCDCS-R-82-1103, University of Illinois, Urbana 1982

[13] Griepentrog, E., März, R.: Differential-algebraic equations and their numerical treatment. Teubner-Texte zur Mathematik No. 88, Leipzig: BSB B.G. Teubner Verlagsgesellschaft 1986

[14] Hairer, E., Lubich, Ch., Roche, M.: The numerical solution of differential-algebraic equations by Runge-Kutta methods. Lecture Notes in Mathematics, 1409 , New York, Berlin, Heidelberg: Springer 1989

[15] Leimkuhler, B.J: Some notes on perturbations of differential-algebraic equations. Technical Report, Institute of Mathematics, Helsinki University of Technology, 1989

[16] Leimkuhler, B.J.,Petzold, L.R., Gear, C.W.: Approximation methods for the consistent initialization of differential-algebraic equations. SIAM J. Num. Anal., submitted for publication.

[17] Lubich, Ch.: h^2 *Extrapolation methods for differential-algebraic equations of index-2.* Technical Report, Universität Innnsbruck, Institut für Mathematik und Geometrie, Innsbruck, 1988

[18] Nikravesh, P.E: Some methods for dynamical analysis of constrained mechanical systems: a survey. In: Haug, E.J (ed.) Computer Aided Analysis and Optimization of Mechanical System Dynamics, pp. 223–259. New York, Berlin, Heidelberg: Springer 1984

[19] Ortega, J.M., Rheinboldt, W.C.: Iterative solutions of nonlinear equations in several variables. New York: Academic Press 1970

[20] Ostermeyer, G.: Die Stabilisierung von Bindungen und ersten Integralen als Regelungsproblem und ihre Konsequenzen. ZAMM Z.Angew.Math. u. Mech. **65** 185–187 (1985)

[21] Ostermann, A.: A half-explicit extrapolation method for differential-algebraic systems of index 3. Technical Report, Université de Genève, Dept. de mathématiques, 1989.

[22] Park, C.T., Haug, E.J.: Numerical methods for mixed differential-algebraic equations in kinematics and dynamics. PhD thesis, College of Engineering, University of Iowa, Iowa City, 1985

[23] Petzold, L.R.: A description of DASSL: a differential/ algebraic system solver. In: Stepleman, R.S. (ed.) IMACS Trans. Sci. Comput. Vol. 1, p. 65. Amsterdam: North Holland 1983

[24] Reich, S.: Beitrag zur Theorie der Algebrodifferentialgleichungen. Thesis, Fakultät für Elektrotechnik/Elektronik, Technische Universität Dresden, 1989

[25] Rheinboldt, W.C.: Differential - algebraic systems as differential equations on manifolds. Math. Comp. **43** 473–482 (1984).

[26] Roberson, R.E., Schwertassek, R.: Dynamics of multibody systems. New York, Berlin, Heidelberg: Springer 1988

[27] Rulka, W.: SIMPACK, ein Rechenprogramm zur Simulation von Mehrkörpersystemen mit grossen Bewegungen. T-PROGRAMM GMBH, P.O. Box 1444 D-7410 Reutlingen, 1989

[28] Schiehlen, W.: Multibody handbook. New York, Berlin, Heidelberg: Springer 1990

[29] Shampine, L.F.: Conservation laws and the numerical solution of odes. Technical Report 84-1241, Sandia National Laboratories, Livermore, 1984

[30] Starner, J.W.: A numerical algorithm for the solution of implicit algebraic-differential systems of equations. PhD thesis, University of New Mexico, 1976