Initial Value Calculation for DAE with Higher Index

Andreas Uhlig, Torsten Blochwitz, Uwe Schnabel, Tobias Nähring
ITI GmbH
Webergasse 1, 01067 Dresden, Germany
{Uhlig, Blochwitz, Schnabel, Naehring} @iti.de

Abstract

The solution of Differential-Algebraic Equations (DAEs) requires the calculation of initial values at the beginning of the simulation as well as after discontinuities. An approach is described that allows consistent initial value calculation (IVC) for higher index DAE with structural changes. To consider rigid impacts integral equations for the conservation of momentum are automatically generated. The method includes handling of fixed initial values and the observance of feasible regions.

Keywords: Initial Value Calculation, DAE, Index calculation, Impulsive Distribution

1 Introduction

A great variety of physical-technical systems can be expressed and modeled by Differential Algebraic Equations (DAEs). System simulation tools, especially Modelica simulators, have to solve DAEs of various structures. The general form of DAEs is written as

$$F(x, \dot{x}, t) = 0 \qquad (1)$$

with the vector of state variables $x \in \mathbf{R}^n$, the vector of the time derivatives $\dot{x}$ and the time t. The integration procedure requires dim (F) = n (but it system does not necessarily depend on all components of $\dot{x}$, see example 1). The first task for the time domain simulation is to find a *consistent initial value* at the start time t_0, i.e., a point x_0 for which a solution curve $x(t)$ of (1) with $x(t_0) = x_0$ exists. If the Jacobian matrix of partial derivatives $\partial F / \partial \dot{x}$ is regular everywhere then (1) defines an implicit system of Ordinary Differential Equations (ODE). With ODE any vector of initial values x_0 is consistent since (1) can locally be resolved for $\dot{x}$. The situation is different in the case of DAE since $\partial F / \partial \dot{x}$ can be singular. The system has *algebraic constraints* if $x \in \mathbf{R}^n$ exist for which there is no algebraic solution of (1) for the unknown $\dot{x}$. Additionally, the system has hidden constraints if there are algebraic solutions $x, \dot{x}$ of (1) for which x is not a consistent initial value, i.e., $\dot{x}$ is not the time-derivative of any solution curve.

Example 1: Consider the system

$$x_1 = t, \quad x_2 = \dot{x}_1 \qquad (2)$$

At $t = t_0$ a vector $x \in \mathbf{R}^2$ must fulfill the equation $x_1 = t_0$ to correspond to an algebraic solution of (2). Therefore, the first equation imposes an algebraic constraint. Not all algebraic solutions are consistent initial values since regarding $\dot{x}_1$ as the time-derivative of x_1 further restricts x by $x_2 = \dot{x}_1 = \dot{t} = 1$. Therefore, the second equation of (2) constitutes a hidden constraint.

In complex systems user often cannot determine consistent initial values for each of the variables because some of them must meet algebraic and hidden constraints. On the other hand he must insist on (i.e. fix) the initial values of certain variables. In this situation tools provide support by allowing *fixed* and *not fixed* values, the latter being used as guesses for the initial value calculation (IVC) executed by the software. The determination of initial values is treated in [1], [5], and [7]. However, the initial values have to be calculated not only at the beginning $t = t_0$ but also after discontinuities ([6]). Now the values at/before the discontinuity play the role of the initial values. Again one has to distinguish, which of these values are fixed and which may change. In this paper we consider an approach that determines consistent initial values from this input.

Figure 1 gives an overview about the main steps of simulation. Before a Modelica model is treated by a numerical solver symbolic simplifications and transformations are carried out (second block in Figure 1). During this process symbolic index reduction is applied in order to transform the system into

DOI: 10.3384/ecp09430122

an ODE or at least into a DAE with index 1 (see [1]). Here we refer to the *differentiation index* that describes how many times a DAE must be differentiated until - after some transformation - an ODE is obtained. The symbolic index reduction carries out these differentiations as far as possible. If it leads to a system without hidden constraints the remaining algebraic constraints can be solved within implicit blocks and so the system can be handled by standard ODE solvers. In this case the computation of consistent initial values is straightforward.

After the symbolic analysis of the model the numerical time-domain simulation starts (third block in Figure 1). It includes handling of discrete-time events through event iterations and time-continuous integration of the DAE-system (1). Time-discrete events are triggered when inequalities change their logical value depending on DAE-states and/or time. Discrete variables and equations describe the behaviour of the system at such events. A change of the time-discrete state may modify the DAE-system (1). As explained below, that calls for a numerical index reduction and a calculation of new consistent initial values. Time-discrete and time-continuous states influence each other. An event iteration takes place until the time-discrete state of the system is stabilized. After that the simulation continues with the time-integration of the DAE.

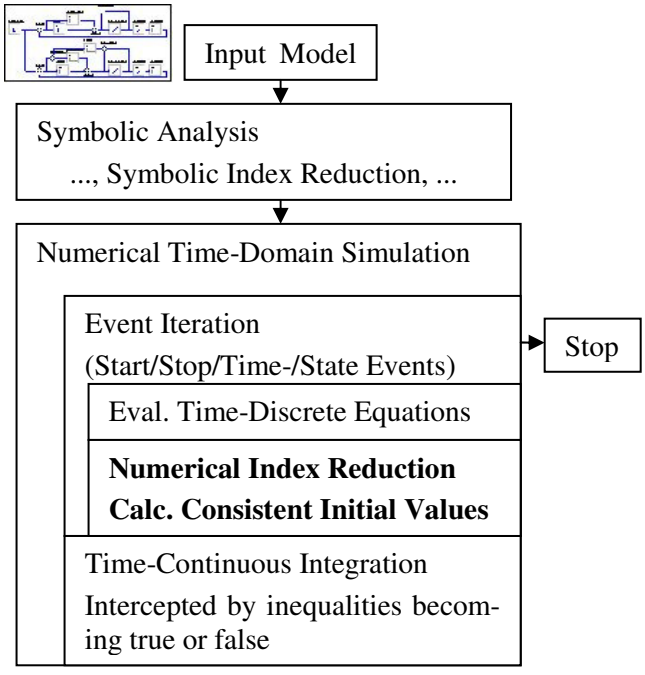


Figure 1: Course of simulation flow

There are situations where the symbolic index reduction cannot differentiate certain equations analytically and some hidden constraints remain in the system for the numerical time-domain simulation. The most important cases are the following:

- *External Functions* do not provide symbolic expressions for equations or even their derivatives.
- If the system contains structural changes (that may happen through *conditional equations* in Modelica as explained in Example 2) the solver is faced with several branches having different indexes. Here symbolic index reduction does not solve the problem. To guarantee a low index all combinations of valid branches would have to be considered separately. In the worst case the analytical effort of regarding all branches increases exponentially. This is not practicable for large models.

Example 2: Consider the system with the single conditional equation

$$1 = \text{if } t < 0.5 \text{ then } x \text{ else } \dot{x}.$$

For t<0.5 the first branch is active and the differentiation index is 1. After that the second branch causes differentiation index 0. The symbolic index reduction does not distinguish between the two branches and wrongly deduces that the equation depends always on both variables x and $\dot{x}$ and the system is classified as to be of index 0. Therefore, the equation is not symbolically differentiated but the IVC cannot compute $\dot{x}$ at $t = 0$ without differentiation of the equation.

If the symbolic index reduction cannot free the system from hidden constraints it has to be supplemented by *numerical index reduction*. For this end it is necessary to calculate derivatives numerically.

In such cases ODE solvers cannot be applied directly for the simulation of the system.

Beyond this, there are more challenges. With impact events in mechanical models (see section 3) *conservation of momentum* is expected. Since the respective equations are normally not part of the model, they have to be deduced numerically from the equations of motion and considered during re-initialization after such events.

Even after index reduction some *algebraic equations* may remain. Furthermore, higher derivative variables often occur *nonlinear*. For both reasons the solution of a nonlinear system is required for the IVC and considered briefly in this paper.

At last, we discuss approaches to meet inequality constraints, that might be defined for some states x_i.

2 Numerical Index Reduction

For the description of the numerical index reduction we assume that (1) is the active set of DAE-equations after a step of the event iteration at time $t = t_d$. As explained in section 1 the system may still have hidden constraints. To transform these for the IVC into algebraic constraints the numerical index reduction supplements the original system with time-difference quotients of selected equations from (1). Thereby, the values of x and $\dot{x}$ after the event time t_d are expressed as the result of one or two steps of the Euler-forward method.

2.1 Index Calculation

The index calculation is the first part of the numerical index reduction. It determines which equations F_i and which states x_j have to be differentiated how many times. The algorithm can be divided into two steps:

1st: *assignment of equations to variables*
2nd: *determination of the number of necessary differentiations*.

Assignment of equations to variables:
Roughly spoken the equations are assigned to variables (or their time derivatives) which 'can be calculated from their equations'. The *best assignment* is found with the help of the following discrete optimization procedure:
Every assignment of equations to variables can be described by a permutation σ of the numbers 1 to n. Here, $\sigma(j) = i$ means that F_i is assigned to x_j. We denote the set of all such permutations as $\mathrm{Perm}(n)$.
From the current numerical Jacobian matrices $\partial F / \partial x$ and $\partial F / \partial \dot{x}$ a cost matrix C is composed:

$$C_{ij} := \begin{cases} -n-2 & \text{if } \partial F_i / \partial \dot{x}_j \neq 0 \\ -n-1 & \text{if } \partial F_i / \partial x_j \neq 0 \wedge \partial F_i / \partial \dot{x}_j = 0 \\ 0 & \text{else} \end{cases}$$

An entry C_{ij} causes rather high costs if F_i does not depend on x_j. Otherwise, the caused costs are lower for F_i depending on the time derivative of x_j than only depending on x_j.

The σ that minimizes the overall costs $\sum_{j=1}^{n} C_{\sigma(j),j}$ is the chosen best assignment of equations to variables. For the minimization one can apply the algorithm from [3].

Determination of the number of necessary differentiations:

The number of necessary differentiations of equations is determined by an iterative procedure. How the equation F_i of the original system depends on the variables x_j is determined the numerical Jacobian and stored in the dependency matrix

$$D^0_{ij} := \begin{cases} 1 & \text{if } \partial F_i / \partial \dot{x}_j \neq 0 \\ 0 & \text{if } \partial F_i / \partial x_j \neq 0 \wedge \partial F_i / \partial \dot{x}_j = 0 \\ -\infty & \text{else} \end{cases}$$

for $i, j = 1,...,n$. Starting from D^0 a sequence of dependency matrices D^k for derived systems with differentiated equations is generated.

In the following we list the algorithm in 'quasi-Modelica' formulation:

```
k := 0; // Iteration index.
// D^0 already defined above.
while true loop
  for j in 1:n loop
    // Determine the highest derivative of x_j that
    // explicitly occurs within the current system:
    d_j^k := max{ D^k_{i,j} | i=1,...,n }
  end for;
  for j in 1:n loop
    if D^k_{σ(j),j} > -∞ then // x_j in eq. σ(j)?
      // Differentiate the σ(j)-th equation:
      for i in 1:n loop
        D^{k+1}_{σ(j),i} := D^k_{σ(j),i} + d_j^k - D^k_{σ(j),j};
      end for;
    end if;
  end for;
  // Stop if no further differentiations were needed:
  if D^{k+1} == D^k then break; end if;
  k := k + 1;
```

end while;
If $D^0_{\sigma(i),j} > -\infty$ then in each iteration the matrix entry $D^k_{\sigma(i),j} > -\infty$ stands for the order of the highest time-derivative $x_j^{\left(D^k_{\sigma(i),j}\right)}$ present in the $d := D^k_{\sigma(i),i} - D^0_{\sigma(i),i}$ times differentiated equation $F_{\sigma(i)}^{(d)}$. So, at the end of the algorithm one can read off the number of needed differentiations of equations and variables.
The algorithm ensures that the highest derivatives of the variables occur in the highest derivatives of their assigned equations if this is possible at all.

It determines the needed additional equations and variables for the re-formulated index-reduced DAE system.

The advantage of the above algorithm over the algorithm of Pantelides [1] is that it also handles singular systems.

2.2 Numerical Differentiation of Equations

The index calculation from section 2.1 tells us which equations of system (1) at $t = t_d$ have to be numerically differentiated for index reduction and if so, how many times. Only first and second order information is numerically generated. So, three is the highest differentiation index that can be handled by the solver itself.

If equation F_i is marked for differentiation the overall system is supplemented by the equation

$$0 = \frac{F_i(x + dt\cdot\dot{x}, \dot{x} + dt\cdot\ddot{x}, t_d + dt) - F_i(x, \dot{x}, t_d)}{dt}.$$

If second order information is needed the result of two Euler-forward steps is added, too:

$$0 = \Big[F_i\Big(x + 2dt\cdot\dot{x} + dt^2\cdot\ddot{x}, \dot{x} + 2dt\cdot\ddot{x} + dt^2\cdot\dddot{x}, t_d + 2dt\Big) - F_i(x, \dot{x}, t_d)\Big] / (2dt)$$

Here, the quantities in these equations have the following meaning:

x	…	value of the state variable at t_d
$\dot{x}$, $\ddot{x}$, $\dddot{x}$	…	1st, 2nd, and 3rd derivatives at t_d
dt	…	step size, automatically selected by the initial value solver

3 Conservation of Momentum

3.1 Motivation

To motivate the necessity of integral equations in simulation we shortly consider the equations of two centrally colliding soft elastic bodies which are only under the influence of the contact force F_C. The equations of motion of the two bodies are

$$\begin{aligned} m_1\dot{v}_1 &= F_C, \quad & \dot{x}_1 &= v_1, \\ m_2\dot{v}_2 &= -F_C, \quad & \dot{x}_2 &= v_2. \end{aligned} \tag{3}$$

And the behavioral description of the contact force is

$$F_C = \text{if } x_1 > x_2 \text{ then } k(x_2 - x_1) \text{ else } 0. \tag{4}$$

Here, we implicitly assume that body 1 with mass m_1 approaches with some start velocity v_{10} from the left hand side $(x_1 < x_2)$ while body 2 with mass m_2 from the right hand side with some start velocity $v_{20} < v_{10}$.

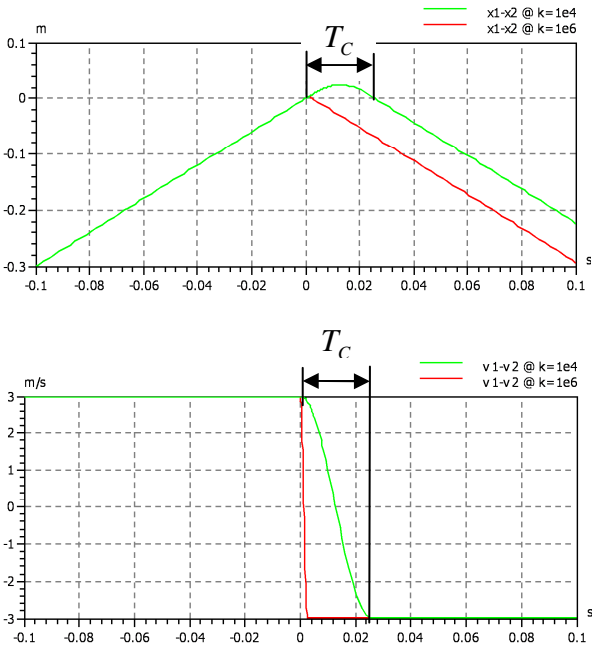


Figure 2: Displacement $x_1 - x_2$ and difference velocity $v_1 - v_2$ of the two masses for $k = 10^4$ N/m (green) and $k = 10^6$ N/m (red)

Figure 2 shows the displacement and the difference velocity for two different values of the contact stiffness. The higher the stiffness the shorter the impact time T_C and the maximal deformation becomes smaller. Furthermore, the difference velocity $v_1 - v_2$ changes sign within the contact phase.

In many practical situations the contact stiffness is very high such that the contact time span and the maximal deformation are relatively small compared to the time- and distance scale ,resp., of the interesting processes to be modeled. If, under these circumstances, the soft impact model (3), (4) is used, the solver is forced to reduce the computation time step size just for capturing the fast contact phase. This can be avoided by modeling the contact phase as one discrete impact event where the sign of the difference velocity changes, i.e., conceptually:

$$\text{when} \quad x_1 > x_2 \quad \text{at } t = t_d \text{ then}$$
$$v_1(t_d + 0) - v_2(t_d + 0) := -(v_1(t_d - 0) - v_2(t_d - 0));$$

With the jump in the difference velocity $v_1 - v_2$ also (at least one of) the velocities v_1 or v_2 must jump and they are no longer differentiable in the classical sense.

In those events the IVC algorithm reformulates the equations of motion (3) depending on $\dot{v}_1$ and $\dot{v}_2$ as integral equations:

$$0 = \lim_{\varepsilon \to 0} \int_{t_d - \varepsilon}^{t_d + \varepsilon} (m_1 \dot{v}_1(t_d) - F_C(t_d))dt$$
$$= m_1(v_1(t_d + 0) - v_1(t_d - 0)) - \hat{F}_C$$
$$0 = \lim_{\varepsilon \to 0} \int_{t_d - \varepsilon}^{t_d + \varepsilon} (m_2 \dot{v}_2(t) + F_C(t))dt$$
$$= m_2(v_2(t_d + 0) - v_2(t_d - 0)) + \hat{F}_C$$

Thereby, the new integral quantity $\hat{F}_C$ is automatically added by the solver.

One may even give the generated equations physical meaning. The new equations reflect the balance of momentum and $\hat{F}_C$ stands for the impulse exchanged by the colliding bodies at the time of impact.

In the context of impulsive distributions as introduced in [4] the force F_C is a linear combination

$$F_C(t) = \overline{F}_C(t) + \hat{F}_C \cdot \delta(t - t_d)$$

of a regular signal $\overline{F}_C$ and a Dirac-delta distribution shifted to the time of impact t_d and weighted by the integral quantity $\hat{F}_C$.

The introductory example is very simple. In practice, the simulator must be able to generate the impulse equations for much more complicated models. Special challenges are simultaneous state changes in multiple impact and end-stop elements, nonlinear transformations between masses and contact elements as well as rigid friction elements parallel to contact elements.

Figure 3 shows a SimulationX model combining all those aspects. It is a swinging pendulum (yellow) bumping against a vertically guided plate (blue) which initially lies on some end stop (gray). The nonlinear transformation of the angular position into the height of the nose of the pendulum and the friction of the vertical guide of the body are taken into account.

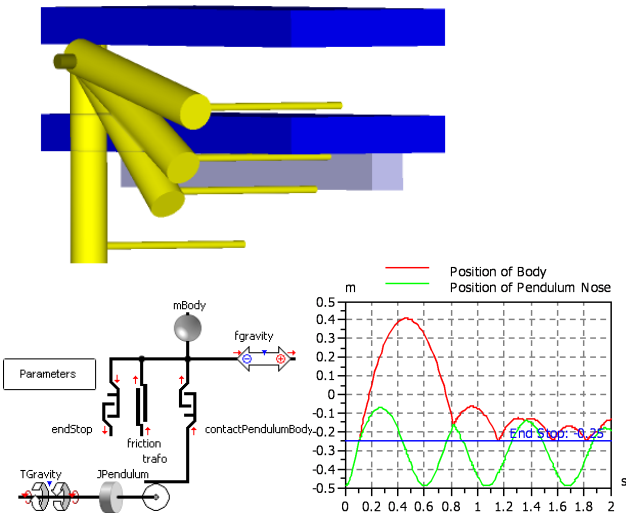


Figure 3: 3D-animation (top, multiple frames are shown), model structure (left), and results (right) of the swinging pendulum

3.2 Generation of Equations

In the transient simulation conservation of momentum is guaranteed by the equations of motion and the relation $\dot{v} = a$. In the IVC at an ideal impact this relationship is not directly used, therefore conservation of momentum must be ensured by additional conditions. For this purpose we integrate the equations of motion numerically and the Dirac-impulse in forces and accelerations resulting from the impact can be treated.

The solver needs model input to identify the impact source in the system (e.g., the contact force F_C in the example from section 3.1) and the jumping variables that determine indirectly the integral value of the impulsive variables for description of the impact (e.g., the jumping velocity difference $v_1 - v_2$ in the example). At present, this is done by a flag set by a function *SetImpact* and by an attribute *notFixed*, respectively. By evaluation of the dependencies of the system the IVC algorithm can determine the equations for which integrals are to be added:

Let $\overline{D}$ be the dependency matrix of the numerically index reduced order-one system in the current

discrete state (see section 2.1). We define the dependency matrix D with rows (corresponding to equations) permuted into the order of the assigned variables: $D_{i,j} := \overline{D}_{\sigma(i),j}$. We define the vector b^0 of initial lower bounds for the impulse order. The indexes j that correspond to variables marked with *setImpact* get the lower bound $b_j := 0$. For the other variables the bounds are initially set to $b_j := -\infty$. The impulse order is found by iteratively adapting the lower bounds of variables x_i to the impulse order of the other variables and derivatives in the assigned equation $F_{\sigma(i)}$. This can be expressed as following iterative assignment

$$b^{k+1} := \mathrm{rowmax}\left(\mathrm{ones}(n,n)\cdot b^k + D\right) - diag(D)$$

to be run for $k = 0,1,...$ until the impulse order $I := b^k$ with $b^{k+1} == b^k$ is found.

The approach can handle even multiple impacts at the same time instance.

After the impulse order of all variables (and assigned equations) has been determined the integrals of equations $F_I(...) = 0$ with impulse order 0 are generated by the solver (currently, only equations up to impulse order 0 are handled). For the discussion we repartition the set of states and time-derivatives $x, \dot{x}$ into *regular variables* v_R and *impulsive variables* v_I. Locally, at $t = t_d$ the regular variables are piecewise continuous while the impulsive variables are linear combinations

$$v_I(t) = \bar{v}_I(t) + \hat{v}_I \delta(t - t_d)$$

of a regular part $\bar{v}_I$ and an impulsive part with the integral value $\hat{v}_I$. For time derivatives $\dot{x}_k$ that are components of v_I the integral value $\hat{\dot{x}}$ is determined by the jump height

$$\begin{aligned}\hat{\dot{x}}_k &= \int_{t_d-0}^{t_d+0} \left(\bar{\dot{x}}_k + \hat{\dot{x}}_k \delta(t - t_d)\right) dt \\ &= \left(x_k(t_d + 0) - x_k(t_d - 0)\right)\end{aligned}$$

of the corresponding state x_k at t_d. Here, the variable right limit $x_k(t_d + 0)$ is the unknown in the supplemented system of equations. For the other purely algebraic components of v_I the integral quantities $\hat{v}_I$ are additional unknowns.

One precondition for the system is that all equations depend at most quasi-linearly on impulsive variables, i.e., the partial system of these equations is representable as

$$F_I(...) \equiv A(v_R, t)\cdot v_I + F_R(v_R, t) = 0 \qquad (5)$$

with a matrix function A and a function F_R both only depending on the regular variables. It is assumed that A depends continuously on its arguments. For F_R only integrability and boundedness is required.

$$\lim_{\varepsilon\to 0}\int_{t_d-\varepsilon}^{t_d+\varepsilon} \left(A(v_R,t)\cdot(\bar{v}_I(t) + \hat{v}_I\delta(t-t_d)) + \right. \\ \left. + F_R(v_R,t)\right)\cdot dt = 0$$

For brevity we do not note the time dependence of all variables explicitly.

Since the integrand of the partial integral

$$\lim_{\varepsilon\to 0}\int_{t_d-\varepsilon}^{t_d+\varepsilon}\left(A(v_R,t)\cdot\bar{v}_I(t) + F_R(v_R,t)\right)\cdot dt = 0$$

is bounded its integral vanishes identically for the limit $\varepsilon \to 0$ and only

$$\lim_{\varepsilon\to 0}\int_{t_d-\varepsilon}^{t_d+\varepsilon} A(v_R,t)\cdot\hat{v}_I\delta(t-t_d)dt = 0 \qquad (6)$$

remains to be considered. If v_R is continuous one directly obtains from the shifting-property of the Dirac-delta distribution the equation

$$A(v_R(t_d), t_d)\hat{v}_I = 0.$$

Things become more complicated when v_R is discontinuous at $t = t_d$. That happens with translational-rotational transmissions as in Figure 3.

This case is not directly covered by distribution theory for the following reason. Let $g : \mathbf{R} \to \mathbf{R}$ be jumping at 0 and otherwise continuous and let δ_ε be a family of L^1-approximations of the Dirac-delta distribution. Then the limit

$$\lim_{\varepsilon\to 0}\int_{-\infty}^{\infty} g(t)\,\delta_\varepsilon(t)dt$$

depends on the actual sequence of Dirac-approximations. Symmetric approximations $\delta_\varepsilon(t) = \delta_\varepsilon(-t)$ lead to

$$\lim_{\varepsilon\to 0}\int_{-\infty}^{\infty} g(t)\,\delta_\varepsilon(t)dt = (g_+ + g_-)/2.$$

with $g_\pm := \lim_{\varepsilon\to 0} g(\pm\varepsilon)$. The left- and right-hand approximations δ_ε^+ and δ_ε^- with $\delta_\varepsilon^\pm(t) = 0$ at $t \geq 0$ and $t \leq 0$, resp., lead to

$$\lim_{\varepsilon\to 0}\int_{-\infty}^{\infty} g(t)\,\delta_\varepsilon^\pm(t)dt = g_\pm.$$

As we show in the following, the symmetric approximation often matches the symbolic integration better.

Let g be sufficiently smooth. In constraint equations there are often additive terms of the form $g'(x_i(t))\dot{x}_i(t)$ with some state variable x_i. For smooth x_i the term has the anti-derivative $g(x_i(t))$. Therefore, one expects the 'symbolical integral' to be

$$I_S := \int_{t_d-0}^{t_d+0} g'(x_i(t))\dot{x}_i(t)dt = g(x_{i+}) - g(x_{i-})$$

even if the state x_i jumps at t_d. Expressing the integral with the help of the symmetric Dirac approximation one gets the 'numerical integral'

$$I_N := \int_{t_d-0}^{t_d+0} g'(x_i(t))(\bar{x}_i(t) + (x_{i+} - x_{i-})\delta(t - t_d))dt$$
$$= \tfrac{1}{2}(g'(x_{i+}) + g'(x_{i-}))\cdot(x_{i+} - x_{i-})$$

where $\bar{x}_i$ is the continuous part of x_i. Taylor series expansion of both integrals I_N and I_S for $\Delta x_i := x_{i+} - x_{i-}$ at $x_{im} := (x_{i+} + x_{i-})/2$ gives $g'(x_{im})\cdot\Delta x_i + O\left(|\Delta x_i|^3\right)$. Therefore, both integrals are equal up to order two. For the corresponding limit of the one-sided Dirac approximation the numerical integrals match the symbolical integral only up to order one.

The foregoing remarks motivate that for the integral equation (6) with jumping signals v_R the result

$$\tfrac{1}{2}(A(v_{R-}, t_d) + A(v_{R+}, t_d))\hat{v}_I = 0 \qquad (7)$$

of the symmetric Dirac approximation should be used. Actually, we implemented

$$A\left(\tfrac{1}{2}(v_{R-} + v_{R+}), t_d\right)\hat{v}_I = 0 \qquad (8)$$

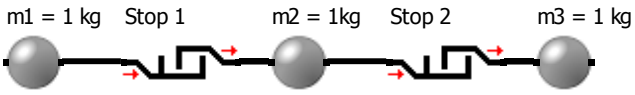
which is a second order approximation of (7) in the jump height $v_{R+} - v_{R-}$ at the midpoint $v_{Rm} := \frac{1}{2}(v_{R+} + v_{R-})$. As we have seen above the change is not crucial for the approximation and the implementation (8) leads to less function evaluations than (7).

DAE-systems with impacts result from structural changes. So the matrix A in the left-hand side of (5) cannot be easily identified through the symbolic analysis in advance. The integral equations (8) must be composed by evaluations of the original left-hand sides F_I at appropriate arguments. One way to represent the additional integral equations is:

$$F_I(v_{Rm}, \hat{v}_I, t_d) - F_I(v_{Rm}, 0, t_d) = 0.$$

Hereby (as already mentioned above), the v_{Rm} are the mean values $v_{Rm} := \frac{1}{2}(v_{R+} + v_{R-})$ of the left and the right limits of the regular variables at t_d. For continuous variables $v_{R,k}$ (e.g., non-jumping states with derivatives) $v_{Rm,k} = v_{R+,k} = v_{R-,k}$. For the other variables $v_{R,k}$ the right limits $v_{R+,k}$ are unknowns in the extended system of equations (see section 4).

Example 3 (Conservation of Momentum):
Given are two end stops and three masses



The two end stops should have an impact at the same time with the impact coefficient ci_1 and ci_2. Then the DAE for the impact is

$$0 = \dot{x}_1 - \dot{x}_2 + ci_1(\dot{x}_1(t_d - 0) - \dot{x}_2(t_d - 0))$$
$$0 = \dot{x}_2 - \dot{x}_3 + ci_2(\dot{x}_2(t_d - 0) - \dot{x}_3(t_d - 0))$$
$$0 = m_1\ddot{x}_1 + F_1$$
$$0 = -F_1 + m_2\ddot{x}_2 + F_2$$
$$0 = -F_2 + m_3\ddot{x}_3$$

Here the velocities $\dot{x}_1$, $\dot{x}_2$, and $\dot{x}_3$ have a jump and the accelerations $\ddot{x}_1$, $\ddot{x}_2$, and $\ddot{x}_3$, and the forces of the end stops F_1 and F_2 are infinite. The numerical index reduction supplements the system with the difference quotients of the first two equations:

$$0 = ((\dot{x}_1 + dt\cdot\ddot{x}_1) - (\dot{x}_2 + dt\cdot\ddot{x}_2) +$$
$$+ ci_1(\dot{x}_1(t_d - 0) - \dot{x}_2(t_d - 0)) -$$
$$- (\dot{x}_1 - \dot{x}_2 + ci_1(\dot{x}_1(t_d - 0) - \dot{x}_2(t_d - 0))))/dt$$
$$= \ddot{x}_1 - \ddot{x}_2$$

and

$$0 = ((\dot{x}_2 + dt\cdot\ddot{x}_2) - (\dot{x}_3 + dt\cdot\ddot{x}_3) +$$
$$+ ci_2(\dot{x}_2(t_d - 0) - \dot{x}_3(t_d - 0)) -$$
$$- (\dot{x}_2 - \dot{x}_3 + ci_2(\dot{x}_2(t_d - 0) - \dot{x}_3(t_d - 0))))/dt$$
$$= \ddot{x}_2 - \ddot{x}_3$$

Furthermore, the following integral equations are added to handle the velocity jumps in the end stops:

$$0 = m_1\dot{x}_1 + \hat{F}_1 - (m_1\dot{x}_1(t_d - 0) - F_1(t_d - 0)),$$
$$0 = -\hat{F}_1 + m_2\dot{x}_2 + \hat{F}_2 - (-F_1 + m_2\dot{x}_2 + F_2)(t_d - 0)$$

and

$$0 = -\hat{F}_2 + m_3\dot{x}_3 - (-F_2(t_d - 0) + m_3\dot{x}_3(t_d - 0))$$

From this we get the 10 unknowns: the velocities $\dot{x}_1$, $\dot{x}_2$, and $\dot{x}_3$, the accelerations $\ddot{x}_1$, $\ddot{x}_2$, and $\ddot{x}_3$, the

forces of the end stops F_1 and F_2 and their integrals $\hat{F}_1 - F_1(t_d - 0)$ and $\hat{F}_2 - F_2(t_d - 0)$.

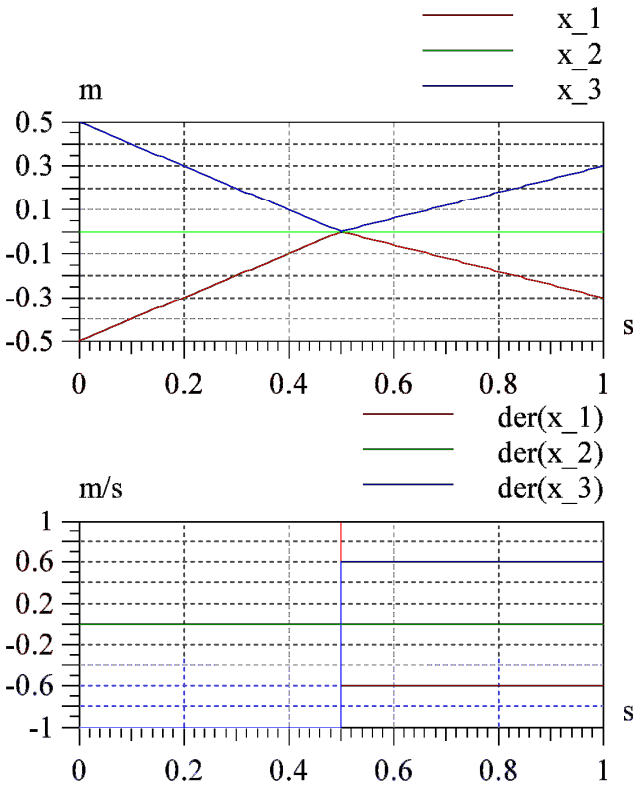


Figure 4: Simulation results for the displacements (top) and velocities (bottom) of the masses

Here it is $ci_1 = ci_2 = 0.6$. Other independent choices of ci_1 and ci_2 in the interval $[0;1]$ are possible. This shows that multiple impacts at the same time instant can be handled.

4 Fixed Initial Values

For $t = t_0$ it can be setup externally (Modelica attribute) which initial values of x and $\dot{x}$ are to be treated as *fixed*. Then they are not variables for the IVC. Moreover, there are rules for the solver to indicate and treat certain states as *fixed* or *not fixed*. Often a user would like to see the guesses for the *not fixed* variables as preferred results of the IVC at $t = t_0$. Therefore, in a first attempt the IVC algorithm can fix those ODE states which are not required for other fixations and initial equations. Only if this was not successful these experimentally fixed variables are released again.

The more interesting situation is after a discontinuity. Here a state x_j (same conditions hold for derivatives $\dot{x}_j$) is treated as *fixed* if

(i) it is specified as *fixed* externally or equation (1) depends on a higher derivative (which also can be a state or even dummy derivative; see [2]) and
(ii) the state is continuous and
(iii) the state is not externally defined as *not fixed*.

Clause (ii) is applied because variables with a discontinuity or infinite value (see section 3) cannot be treated as fixed.

If a higher derivative occurs in (1), that is not infinite, the state itself is continuous. Therefore, the second part of clause (i) fixes it. This also applies if the higher derivative is a dummy (see [2]), since in reality the relationship exists.

5 Solving the System of Equations

The final system of equations is solved by a Newton method. In the situations described in sections 2 and 3 additional equations and sometimes also variables are appended to (1). On the other hand, due to fixation of variables in the previous section such variables are omitted. Thus, as a rule the system of equation has rectangular shape.

Under-determined Systems: With an infinite number of solutions we can work with the minimum-norm solution, i.e.

$$\min\left\{\|s\|_2 : Jac * s = -res\right\}.$$

This does not need to be critical, e.g. when derivatives of certain states are not calculated in the course of IVC since they are used nowhere, as shown in the example below.

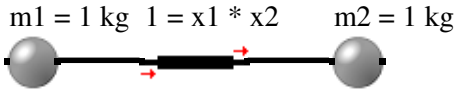
Example 4:

Let velocity v_i be a state. Its derivative $\dot{v}_i$ is the acceleration a_i. a_i is a state too, if it is a dummy-derivative, i.e. there is no equation $\dot{v}_i = a_i$. In the system of equation the derivative of v_i is always expressed by a_i. Thus $\dot{v}_i$ does not occur in the system of equation and hence cannot be calculated in the IVC.

Even over-determined systems, if correctly modeled can be solved after a discontinuity (Example 5).

Example 5:

Consider two masses linked by a mechanical constraint.



Equation before index reduction:

$$0 = m_1\ddot{x}_1 + m_2\ddot{x}_2$$

$$1 = x_1 x_2$$

Index reduction adds 1. and 2. derivatives:

$$0 = \dot{x}_1 x_2 + x_1\dot{x}_2$$

$$0 = \ddot{x}_1 x_2 + 2\dot{x}_1\dot{x}_2 + x_1\ddot{x}_2$$

If there is (caused by others parts of a model) a discontinuity at t_d then all of the states

$x_1(t_d)$, $x_2(t_d)$, $\dot{x}_1(t_d)$, $\dot{x}_2(t_d)$ are fixed, i.e. keep the value of $(t_d - 0)$. $\ddot{x}_1(t_d)$ and $\ddot{x}_2(t_d)$ are variable. The system has 4 equations and 2 variables.

$$0 = m_1\ddot{x}_1(t_d) + m_2\ddot{x}_2(t_d)$$

$$1 = x_1(t_d - 0)x_2(t_d - 0)$$

$$0 = \dot{x}_1(t_d - 0)x_2(t_d - 0) + x_1(t_d - 0)\dot{x}_2(t_d - 0)$$

$$0 = \ddot{x}_1(t_d)x_2(t_d - 0) + 2\dot{x}_1(t_d - 0)\dot{x}_2(t_d - 0) + x_1(t_d - 0)\ddot{x}_2(t_d)$$

The second and the third equation are not changed at the discontinuity. So initial values for $\ddot{x}_1$ and $\ddot{x}_2$ can be determined from the first and fourth equation. Like in this example the surplus equations are often satisfied automatically. Thus, even a rectangular system can have a solution.

6 Ensuring the Feasible Region

Derived from real world conditions inequality constraints, especially lower and upper bounds, may be attached to the variables of a DAE:

$x_i > x_{i,Min}$ and $x_i < x_{i,Max}$

(If equality shall be included (e.g. $x_i \geq x_{i,Min}$) we use the relation $x_i > x_{i,Min} - absTol * ScaleFac$, with suitable constants.)

Our first choice – if possible - is to make use of the kernel of the system matrix. If the originally calculated correction *s* would lead to

$x_i^{new} \leq x_{i,Min}$ or $x_i^{new} \geq x_{i,Max}$ with

$x_i^{new} = x_i^{old} + s_i$ and $x_{i,Min} < x_i^{old} < x_{i,Max}$

then we determine a Newton update s^N parallel to the border of the feasible region. We set $s_i^N = 0$ and calculate the rest of s^N from

*minimum norm solution + kernel vector * factor*

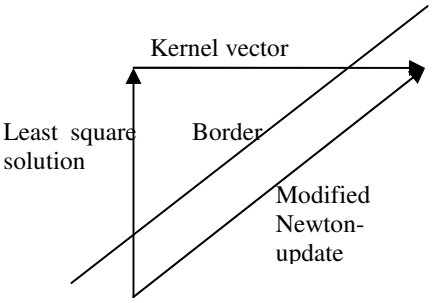


Figure 5: Newton update parallel to border of feasible region

If this approach is not applicable (no kernel) a damping factor α with $0 < \alpha < 1$ shall decrease the update vector $s^N = \alpha \cdot s$, such that for a given λ with $0 < \lambda < 1$

$$x_{i,Min} < (1-\lambda) * x_{i,Min} + \lambda * x_i^{old} \leq x_i^{old} + \alpha * s_i \leq (1-\lambda) * x_{i,Max} + \lambda * x_i^{old} < x_{i,Max}.$$

7 Conclusions

The solution of DAEs requires the calculation of initial values at the beginning of the simulation as well as after discontinuities. In this paper some measures for special situations were described. First, we can, e.g. in the case of external functions and structural changes, execute the necessary differentiation of certain equations numerically. Second, in order to enable ideal components additional equations for initial values were developed. Hereby the conservation of momentum for impacts was ensured. This approach includes assumptions, which are fulfilled in many applications. In the current implementation, the model needs to be annotated with a *SetImpact* flag. The authors propose to include a suitable description of impulsive distributions (including impulsive jumps) in the Modelica specification.

Even if the resulting system of equations is not regular, the IVC can find a solution. At last we mentioned how fixed and non-fixed initial values and feasible region for the variables in the IVC are managed.

References

[1] Pantelides C.C.: The Consistent Initialization of Differential-Algebraic Systems. SIAM J. SCI. Stat. Comput., Vol. 9, No. 2, pp. 213-231, March 1988.

[2] Mattsson S.E. and Söderlind G.: Index reduction in differential-algebraic equations using dummy derivatives. SIAM Journal on Scientific Computing, Vol. 14, No. 3, pp. 677-692, May 1993.

[3] Jonker R. and Volgenant A.: A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. Computing, Vol. 38, pp. 325-340, 1987.

[4] Kunkel P. and Mehrmann V.: Differential - Algebraic Equations. European Mathematical Society Publishing House, Zürich, 2006.

[5] Bauer I.: Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik. Dissertation, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR) der Universität Heidelberg, 1999.

[6] Wunderlich L.: Analysis and Numerical Solution of Structured and Switched Differential-Algebraic Systems, PhD Thesis, TU Berlin, 2008.

[7] Li S., Petzold L. R.: Design of New DASPK for Sensitivity Analysis, Technical Report: TRCS99-28, 1999.

