Comparison of Regularisation Approaches for High Index DAEs

Carina Pöll[1], Irene Hafner[2], Bernhard Heinzl[3], Felix Breitenecker[1]
[1]Vienna University of Technology, Institute for Analysis and Scientific Computing
[2]dwh GmbH, Simulation Services, Vienna
[3]Vienna University of Technology, Institute of Computer Aided Automation
carina.poell@tuwien.ac.at

The use of object–oriented modelling approaches for modelling physical or mechanical systems leads to differential–algebraic equations which cannot be directly transformed into ordinary differential equations, because they have a differential index greater than zero. The transformation into an odrinary differential equation is important for the numerical solving. This paper is intended to provide an overview of common regularisation approaches for differential–algebraic equations with high differential index. First there are some mathematical definitions and basic findings. After a short overview of the discussed methods, including a classification of the techniques in the areas differentiation, projection and transformation, the different methods are demonstrated using the commonly known example of a rotational pendulum described in Cartesian coordinates. With respect to this example a comparison of the numerical solutions of the used methods is possible.

1 Introduction

A component–based acausal model description for physical or mechanical systems, such as Modelica or MATLAB/Simscape, usually leads to differential–algebraic equations (DAEs) with a non–trivial differential index. Solving DAEs with a high index using methods designed for ordinary differential equations (ODEs) is generally very complex and therefore numerically extensive or may even be impossible. This problem leads to the so–called index reduction, in which the given DAE is reformulated as a DAE with lower index or an ODE. Due to the large differences (structure, properties, etc.) of DAEs a lot of different index reduction methods can be found in the literature, see [1] and [2]. The different techniques discussed in this paper can be classified into the following areas:

- methods using differentiation
- methods using projection
- methods using transformation of the state space

On the following pages these methods are discussed and a short example is given to illustrate their functionality.

2 Basics

In this section there are some basic definitions which are used later on. A differential–algebraic equation (DAE), see [1], is given by an implicit equation

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

with a function $F\colon I \times D_x \times D_{\dot{x}} \to \mathbb{R}^n$, where $I \subseteq \mathbb{R}$ is a real interval and $D_x, D_{\dot{x}} \subseteq \mathbb{R}^n$ are open sets, $n \in \mathbb{N}$ and $x\colon I \to \mathbb{R}^n$ is a differentiable function, where $\dot{x}$ is the derivative of x with respect to t. According to the implicit function theorem F can be solved for $\dot{x}$ if the matrix $\frac{\partial F}{\partial \dot{x}}$ is regular. The algebraic equations of the differential–algebraic equation system $F(t,x,\dot{x}) = 0$ are of the form

$$g(x) = 0, \tag{2}$$

where $g\colon \mathbb{R}^n \to \mathbb{R}^m$ is a function with $m < n$, and they are called constraints. A differential–algebraic equation has differential index $k \in \mathbb{N}_0$ (see [2]) if k is the minimal number of derivatives, so that an ODE can be extracted from the system

$$F(t,x,\dot{x}) = 0,\ \frac{\mathrm{d}F(t,x,\dot{x})}{\mathrm{d}t} = 0,\ \ldots,\ \frac{\mathrm{d}^k F(t,x,\dot{x})}{\mathrm{d}t^k} = 0. \tag{3}$$

This generated ODE can (through algebraic transformations) be written in the form $\dot{x} = \varphi(t,x)$ with a function $\varphi\colon I \times D_x \to \mathbb{R}^n$.

3 Regularisation Approaches

The regularisation approaches are split into three different parts like in [2].

3.1 Regularisation by Differentiation

There are various approaches which are using differentiation.

3.1.1 Differentiation of the Constraint

The procedure of this appoach is to differentiate the constraint $g(x) = 0$ and substitute the constraint by its derivative until the system has differential index 1. The problem of this approach is that due to the differentiation there is a loss of information. Therefore the necessary initial values for the integration are unknown and numerical "drift–off" occurs, i.e. the numerical solution departs from the exact solution.

3.1.2 Baumgarte–Method

This method can only be used for DAEs of index 3. The initial point of this approach is the index–1–formulation of the index–3–system. The constraint $\ddot{g}(x) = 0$ is substituted by a linear combination of g, $\dot{g}$ and $\ddot{g}$ of the form (see [3])

$$\ddot{g} + 2\alpha\dot{g} + \beta^2 g = 0. \tag{4}$$

Because of the consideration of the original constraint there is no loss of information. α and β have to be chosen, so that the differential equation is asymptotically stable. Therefore follows $\alpha > 0$. The problem of this appoach is the exact choice of the constants α and β.

3.1.3 Pantelides–Algorithm

For each equation of the constraints the following procedure has to be used.

1. Each constraint has to be differentiated.
2. The differentiated constraint has to be added to the DAE. If there is an algebraic variable in the constraint, then the derivative of this variable is a so–called dummy derivative.
3. An integrator which has a connection to the constraint and the derivative of the constraint respectively is eliminated, i.e. for example $\dot{x}$ is eliminated and instead of $\dot{x}$ a new variable called dx is used.
4. Through differentiation of the constraint it can occur that a new variable is generated, i.e. for example through differentiation y (algebraic variable) becomes dy and there is an equation where y can be computed in the system.
5. Therefore this equation also has to be differentiated.
6. The proceeding of the points 4–5 continues until no new variables are created.

A disadvantage of this algorithm is that during the procedure a lot of variables and equations may be created and therefore the system of the resulting equations can be confusing.

3.2 Regularisation by Projection

A DAE with differential index $k > 1$ is given. If the numerical solution does not fulfill the constraint, the numerical solution is projected onto a manifold, which is given by the constraint $g(x) = 0$ and the $1^{st}, \ldots, (k-2)^{th}$ derivatives with respect to t of the constraint. The solution manifold is

$$M = \left\{ x \in \mathbb{R}^n\colon\ g(x) = 0, \frac{\mathrm{d}^i g(x)}{\mathrm{d}t^i} = 0, \forall i \in I \right\}, \tag{5}$$

with $I = \{1, \ldots, k-2\}$. The algebraic variables can be expressed by the $(k-1)^{th}$ derivative of the constraint and are inserted into the differential equations of the DAE. This procedure leads to a system of differential equations $\dot{y} = f(t, y)$ on the manifold M.

3.2.1 Standard Projection Method

One step $y_n \mapsto y_{n+1}$ of the Standard Projection Method is calculated in the following way, see [4]:

- $\hat{y}_{n+1} = \Phi_h(t_n, y_n)$ is calculated, where Φ_h is a numerical integrator applied to $\dot{y} = f(t, y)$ (for example a Runge–Kutta method).
- For getting $y_{n+1} \in M$, $\hat{y}_{n+1}$ is projected orthogonally onto the manifold M.

In figure 1 the Standard Projection Method is represented.

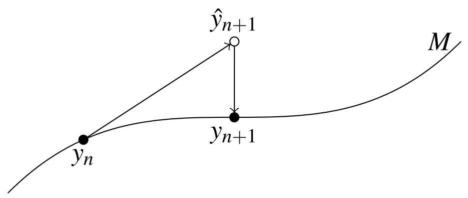

Figure 1: Illustration of the Standard Projection Method

This method is also called Orthogonal Projection Method.

3.2.2 Symmetric Projection Method

A one–step–method Φ_h is called symmetric, if $\Phi_h = \Phi_{-h}^{-1}$.

One step $y_n \mapsto y_{n+1}$ of the Symmetric Projection Method is calculated in the following way, see [5]:

- $\hat{y}_n = y_n + \frac{\partial g}{\partial y}^T(y_n)\mu$ with $g(y_n) = 0$.
- $\hat{y}_{n+1} = \Phi_h(\hat{y}_n)$ is calculated, where Φ_h is a symmetric one–step–method applied to $\dot{y} = f(y)$.
- $y_{n+1} = \hat{y}_{n+1} + \frac{\partial g}{\partial y}^T(y_{n+1})\mu$ with μ such that $g(y_{n+1}) = 0$.

It is important that the same μ is used in the step $y_n \to y_{n+1}$. In figure 2 the Symmetric Projection Method is shown.

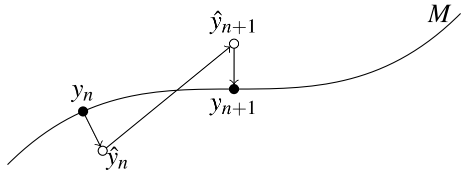

Figure 2: Illustration of the Symmetric Projection Method

3.3 Regularisation by Transformation

The DAE with differential index k is not solved on the whole state space, but on a manifold, see [4]. The manifold is implicitly given by the constraint $g(x) = 0$ and the $1^{st}, \ldots, (k-2)^{th}$ derivatives with respect to t of the constraint. The manifold M is given by equation (5). The algebraic variables can be expressed by the $(k-1)^{th}$ derivative of the constraint and are inserted into the differential equations of the DAE. This procedure leads to a system of differential equations $\dot{y} = f(t, y)$ on the manifold M. This ODE on the manifold M is solved through the introduction of local coordinate transformations.
Let a local coordinate function $\psi \colon U \to \mathbb{R}^n$, $U \subset \mathbb{R}^{n-m}$ open, $\psi(U) \subset M$ be given on the m–dimensional manifold M. The transformation $y = \psi(z)$ transforms the differential equation $\dot{y} = f(t, y)$ into

$$\frac{\partial \psi(z)}{\partial z}\dot{z} = f(t, \psi(z)). \tag{6}$$

With the assumption $f(t, y) \in T_yM$, where T_yM is the tangent space in a fixed point $y \in M$ and has dimension m, the differential equation (6) is equivalent to

$$\dot{z} = F(t, z). \tag{7}$$

One step $y_n \mapsto y_{n+1}$ using local coordinates is calculated in the following way:

- Local coordinates are chosen and z_n is calculated with $y_n = \psi(z_n)$.
- $\hat{z}_{n+1} = \Phi_h(t_n, z_n)$ is calculated with a numerical method Φ_h applied to (7).

- $y_{n+1} = \psi(\hat{z}_{n+1})$

The coordinates $y = \psi(z)$ can be changed in each step. The difficulty of this method is to find suitable coordinates.

4 Case Study

The circular motion of a pendulum in Cartesian coordinates is used as illustrative model. The equations (see [6]) of this model are given by

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v}_x &= -Fx \\ \dot{v}_y &= g - Fy \\ x^2 + y^2 &= 1, \end{aligned} \tag{8}$$

where g is the gravitational acceleration on earth and F the Lagrange multiplier. The constraint is given by the equation $x^2 + y^2 = 1$. In figure 3 the motion is schematically represented.

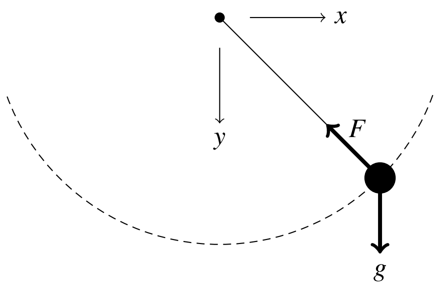

Figure 3: Illustration of the Circular Motion of a Pendulum

4.1 Analysis of the DAE

The constraint is used to determine the differential index of the DAE. The constraint and the first, second and third derivative with respect to t are given by

$$\begin{aligned} x^2 + y^2 - 1 &= 0 \\ xv_x + yv_y &= 0 \\ v_x^2 + v_y^2 - F(x^2 + y^2) + gy &= 0 \\ 4F(xv_x + yv_y) - 3gv_y + (x^2 + y^2)\dot{F} &= 0. \end{aligned} \tag{9}$$

From the third derivative with respect to t, $\dot{F}$ can be expressed. Therefore the DAE has index 3. A remarkable fact is that F can be expressend from the second derivative with respect to t. Thus F can be inserted in the differential equations. In table 1 the generated ODEs of the given DAE are shown.

index–0–system	index–1–system
$\dot{x} = v_x$	$\dot{x} = v_x$
$\dot{y} = v_y$	$\dot{y} = v_y$
$\dot{v_x} = -Fx$	$\dot{v_x} = -\frac{v_x^2+v_y^2+gy}{x^2+y^2}x$
$\dot{v_y} = g - Fy$	$\dot{v_y} = g - \frac{v_x^2+v_y^2+gy}{x^2+y^2}y$
$\dot{F} = \frac{-4F(xv_x+yv_y)+3gv_y}{x^2+y^2}$	

Table 1: Different ODEs for the Motion of the Pendulum

4.2 Regularisation by Differentiation

4.2.1 Differentiation of the Constraint

The constraint has to be differntiated twice which results in

$$v_x^2 + v_y^2 - F(x^2 + y^2) + gy = 0. \tag{10}$$

Therefore this approach leads to a DAE with index 1. This DAE is solved with the ode–solver ode15s of MATLAB. In figure 4 the result of this method is shown where the "drift–off" phenomenon is visible, whereby the simulation is calculated till 100 seconds.

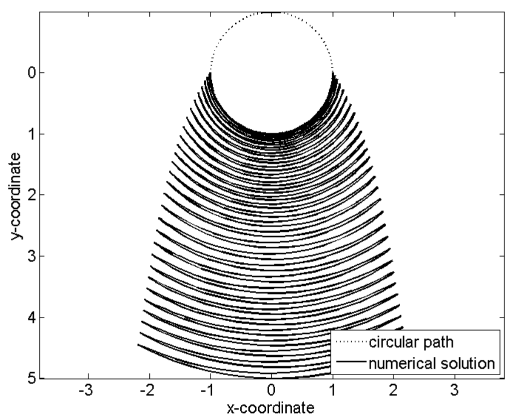

Figure 4: Result of the Method with Differentiation of the Constraint

In figure 5 the increasing error from the numerical solution to the circular path is shown.

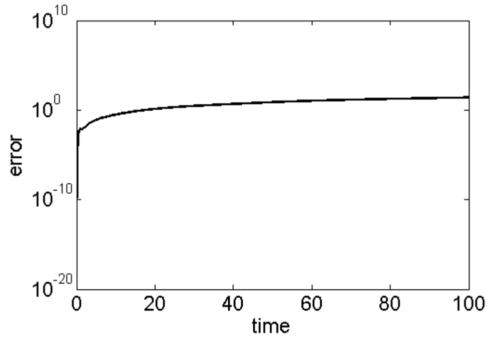

Figure 5: Error of the Method with Differentiation of the Constraint

The observed "drift–off" shows that this method is not suitable for the solving of the given DAE. Therefore other approaches for solving this DAE are necessary. The results of this other approaches are discussed in the following.

4.2.2 Baumgarte–Method

The Baumgarte–Method computes a linear combination of $\ddot{g}$, $\dot{g}$ and g and substitutes g by this linear combination. The variable F can be expressed of this linear combination and can be inserted in the other equations. This results in four differential equations

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v_x} &= -\frac{2(v_x^2+v_y^2+gy)+4\alpha(xv_x+y_vy)+\beta^2(x^2+y^2-1)}{2(x^2+y^2)}x \\ \dot{v_y} &= g-\frac{2(v_x^2+v_y^2+gy)+4\alpha(xv_x+y_vy)+\beta^2(x^2+y^2-1)}{2(x^2+y^2)}y. \end{aligned} \tag{11}$$

This equations are solved with the ode–solver ode45 of MATLAB.
The numerical solution calculated with the Baumgarte–Method stays close to the cirular path, which can be explained by the choice of the parameters α and β. The parameters have to be chosen so that equation (4) is asymptotically stable. The numerical solution of the method, which uses the differentiation of the constraint, shows "drift–off", whereas the result of the Baumgarte–Method shows no "drift–off".
In figure 6 the result of this method is shown for $\alpha = 10$ and $\beta = 100$.

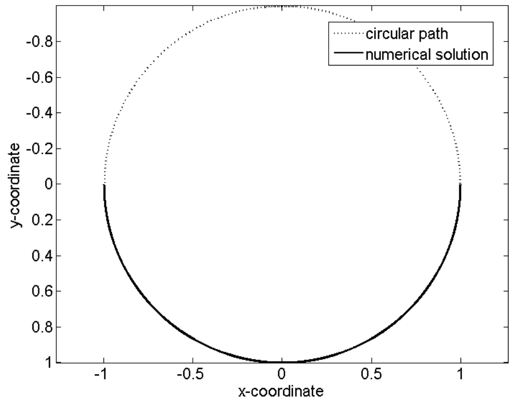

Figure 6: Result of the Baumgarte–Method

The error of the numerical solution to the circular path is shown in figure 7.

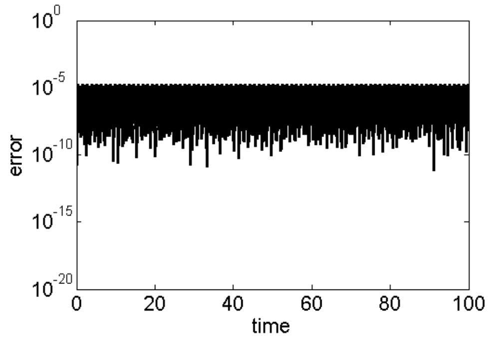

Figure 7: Error of the Baumgarte–Method

4.2.3 Pantelides–Algorithm

Application of the Pantelides–Algorithm with introduction of the dummy derivative dx leads to a system of nine equations and nine unknows (x, dx, d2x, d2y, $\dot{y}$, $\dot{v}_x$, dv_x, $\dot{v}_y$ and F). These equations are

$$\begin{aligned} \mathrm{d}x &= v_x \\ \mathrm{d}2x &= \mathrm{d}v_x \\ \dot{y} &= v_y \\ \mathrm{d}2y &= \dot{v}_y \\ \mathrm{d}v_x &= -Fx \\ \dot{v}_y &= g-Fy \\ x^2+y^2 &= 1 \\ x\mathrm{d}x+y\dot{y} &= 0 \\ (\mathrm{d}x)^2+x\mathrm{d}2x+(\dot{y})^2+y\mathrm{d}2y &= 0. \end{aligned} \tag{12}$$

For solving this problem it is necessary to use the Pantelides–Algorithm twice. Therefore it is necessary to introduce the dummy derivative dy instead of dx. This leads to the equation system

$$\begin{aligned} \mathrm{d}y &= v_y \\ \mathrm{d}2y &= \mathrm{d}v_y \\ \dot{x} &= v_x \\ \mathrm{d}2x &= \dot{v}_x \\ \mathrm{d}v_y &= g - Fy \\ \dot{v}_x &= -Fx \\ x^2 + y^2 &= 1 \\ x\dot{x} + y\mathrm{d}y &= 0 \\ (\dot{x})^2 + x\,\mathrm{d}2x + (\mathrm{d}y)^2 + y\,\mathrm{d}2y &= 0. \end{aligned} \tag{13}$$

With this two equation system four cases are considered:

- equation system (12) and $x = \sqrt{1-y^2}$
- equation system (12) and $x = -\sqrt{1-y^2}$
- equation system (13) and $y = \sqrt{1-x^2}$
- equation system (13) and $y = -\sqrt{1-x^2}$.

Each of this equations is solved with the ode–solver ode15i of MATLAB. The result of the simulation using the Pantelides–Algorithm stays close to the circular path for the chosen initial values. The result looks similar to figure 6.

4.3 Regularisation by Projection

In the following two methods using projection are considered.

4.3.1 Standard Projection Method

After each solving step, the numerical solution is projected orthogonally onto the manifold M which is given by

$$M = \{(x, y, v_x, v_y) \in \mathbb{R}^4 : x^2 + y^2 - 1 = 0, xv_x + yv_y = 0\}. \tag{14}$$

The used projection $p : \mathbb{R}^4 \to M$ onto the manifold M is defined by

$$p\begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} p_1(x) \\ p_1(y) \\ p_2(v_x) \\ p_2(v_y) \end{pmatrix} \tag{15}$$

with the mapping p_1

$$\begin{aligned} x &\mapsto \frac{x}{\sqrt{x^2+y^2}} \\ y &\mapsto \frac{y}{\sqrt{x^2+y^2}} \end{aligned} \tag{16}$$

and the mapping p_2

$$\begin{aligned} v_x &\mapsto (-p_1(y)v_x + p_1(x)v_y)(-p_1(y)) \\ v_y &\mapsto (-p_1(y)v_x + p_1(x)v_y)p_1(x). \end{aligned} \tag{17}$$

The procedure first projects the position and then the velocity. For the numerical solving the explicit Euler method is used. In figure 8 the orthogonal projection is shown.

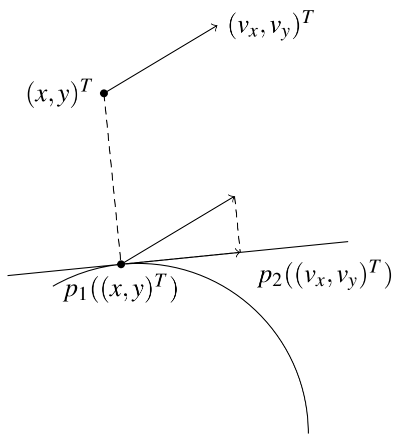

Figure 8: Orthogonal Projection

In figure 9 the result of this method is shown where the incorrect positions caused by the increasing speed (Euclidean norm of the velocity) are visible.

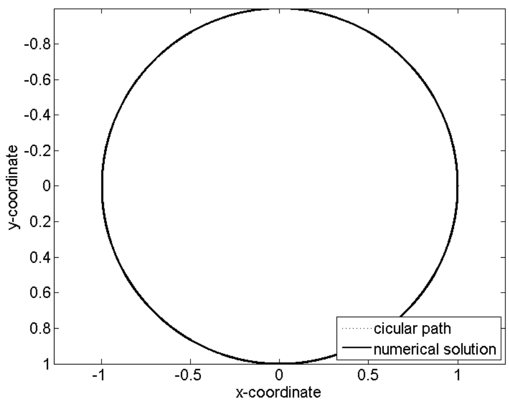

Figure 9: Result of the Orthogonal Projection Method

The Orthogonal Projection Method stays close to the cicular path because the method is designed that way but the postion is not correct because of the increasing speed. In figure 10 the speed is shown.

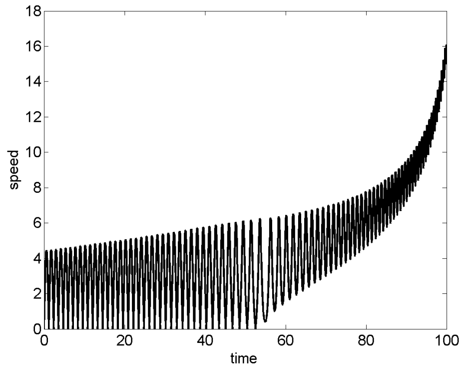

Figure 10: Speed of the Orthogonal Projection Method

4.3.2 Symmetric Projection Method

The procedure of this method leads to an equation system with x_{n+1}, y_{n+1}, $v_{x_{n+1}}$, $v_{y_{n+1}}$, μ_1 and μ_2 as unknowns. This equation system is solved by using Newton's method. In figure 11 the result is shown.

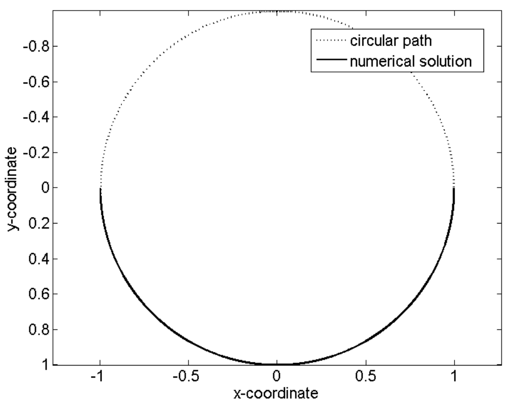

Figure 11: Result of the Symmetric Projection Method

4.4 Regularisation by Transformation

The DAE is transformed by a local state space transformation to an ODE on the manifold M (see equation (14)). The transformation $(x,y,v_x,v_y) = \psi(\varphi,\eta)$ is given by

$$\begin{aligned} x &= \cos\varphi \\ y &= \sin\varphi \\ v_x &= -\eta\sin\varphi \\ v_y &= \eta\cos\varphi. \end{aligned} \tag{18}$$

This transformation leads to the ODE

$$\begin{aligned} \dot{\varphi} &= \eta \\ \dot{\eta} &= g\cos\varphi. \end{aligned} \tag{19}$$

The state space can be transformed globally and therefore an ODE can be generated, which is an advantage for the numerical solution. The obtained ODE is solved with the ode–solver ode45 of MATLAB. Through transformation to polar coordinates the numerical solution stays on the circular path. This method leads to the most simple equations. For results of this approach see section 4.5.

4.5 Comparison

In table 2 the error $e := \max_i(x_i^2 + y_i^2 - 1)$ and elapsed time, where x_i and y_i are the numerical solutions for the positions at each solver time step t_i, are shown for the chosen initial values and till time 100. In the table below DC means differentiation of the constraint, B is the Baumgarte–Method, P the Pantelides–Algorithm, OP the Orthogonal Projection Method, SP the Symmetric Projection Method and T the transformation of the state space. The error and the elapsed time for the Baumgarte–method are calculated for $\alpha \neq \beta$ and $\alpha = \beta$.

Method	Error	Time (s)
DC	24.731	1.095
B ($\alpha \neq \beta$)	$1.909 \cdot 10^{-5}$	1.696
B ($\alpha = \beta$)	$2.463 \cdot 10^{-4}$	0.723
P	$3.345 \cdot 10^{-4}$	8.539
OP	$5.551 \cdot 10^{-16}$	76.175
SP	$2.701 \cdot 10^{-8}$	71.729
T	$2.220 \cdot 10^{-16}$	0.550

Table 2: Error e and Elapsed Time of the Approaches

The error of the method with differentiation of the constraint shows "drift–off" and therefore the error is the biggest of all approaches. The other methods have small errors, whereas the transformation to polar coordinates has the smallest value. The Orthogonal Projection Method also has a small error, but the result is not correct due to the falsified velocity therefore this method is not interesting. The choice of α and β of the Baumgarte–Method result in different values for the error e. The state space transformation also has a very small error and the position is correct. Therefore this method would be one of the best for this problem. The elapsed times of the two projection methods are bigger than the of the other methods because the Orthogonal Projection Method uses a self written explicit Euler–method and the algorithm of the Symmetric Projection Method solves a linear equation system in each step. All the other approaches use for the solving of the obtained equations ode–solvers from MATLAB.

5 Conclusion

Finally it has been shown that only differentiating and substituting the constraint equations by their derivatives is no suitable method for solving differential–algebraic equations. Therefore other approaches are necessary. These other methods are using different backgrounds like differentiation or projection.
For the chosen example and initial values the methods show mostly results which are close to the circular path. One approach has problems with the correct positions due to the increasing Euclidean norm of the velocity. Another fact is that for other initial values the method can work worse regarding the distance to the circular path. The state space transformation is one of the best appoaches, if the transformation is global.
At last it is possible that one of the mentioned approaches works for the chosen case study but is not suitable for another differential–algebraic equation system. Therefore in the future it is necessary to test the methods also for another example.

References

[1] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, Finnland, 2006.

[2] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer, Deutschland, 2002.

[3] E. Eich and M. Hanke. *Regularization Methods for Constrained Mechanical Multibody Systems*. ZAMM - Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, pp 761-773, 1995.

[4] E. Hairer. *Geometric Integration of Ordinary Differential Equations on Manifolds*. BIT Numerical Mathematics, Volume 41, Issue 5, pp 996-1007, 2001.

[5] E. Hairer. *Symmetric Projection Methods for Ordinary Differential Equations on Manifolds*. BIT Numerical Mathematics, Volume 40, issue 4, pp 726-734, 2000.

[6] F. E. Cellier and E. Kofman. *Continuous System Simulation*. Springer, USA, 2006.