Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics
(IMACM)

Preprint BUW-IMACM 16/10

Tim Jax, Gerd Steinebach

Generalized ROW-Type Methods for Solving Semi-Explicit DAEs of Index-1

March 2016

www-num.math.uni-wuppertal.de

Generalized ROW-Type Methods for Solving Semi-Explicit DAEs of Index-1

T. Jax[a,*], G. Steinebach[a]

[a]*Department EMT, Hochschule Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, 53757 Sankt Augustin, Germany*

Abstract

A new type of Rosenbrock-Wanner (ROW) methods for solving semi-explicit DAEs of index-1 is introduced. The scheme considers arbitrary approximations to Jacobian entries resulting for the differential part and thus corresponds to a first attempt of applying W methods to DAEs. Besides, it is a generalized class covering many ROW-type methods known from literature. Order conditions are derived by a consistent approach that combines theories of ROW methods with exact Jacobian for DAEs [13] and W methods with arbitrary Jacobian for ODEs [15]. In this context, rooted trees based on Butcher's theory that include a new type of vertices are used to describe non-exact differentials of the numerical solution. Resulting conditions up to order four are given explicitly, including new conditions for realizing schemes of higher order. Numerical tests emphasize the relevance of satisfying these conditions when solving DAEs together with approximations to Jacobian entries of the differential part.

Keywords: Rosenbrock methods, W methods, Differential-algebraic equations, Order conditions, Approximated Jacobian

1. Introduction

Solving ordinary differential equations (ODEs), efficient time-integration is mainly determined by given stiff and non-stiff characteristics that demand implicit or explicit strategies. In this context, W methods introduced by Steihaug and Wolfbrandt [15] are an appropriate choice for numerical computation. Belonging to the class of linearly implicit Rosenbrock-Wanner (ROW) type schemes, they realize implicit solution without having to solve non-linear equations. Besides, W methods allow for arbitrary approximations of occurring Jacobians. Thus, they also enable explicit integration by reduction to underlying explicit Runge-Kutta (RK) schemes. Due to this property, W methods can be used to reduce computational effort by combining explicit RK methods and implicit ROW methods in order to adapt solutions to requirements of a given ODE problem. In literature these combinations are implemented by strategies such as partitioning [11] or additive splitting via approximate-matrix factorization (AMF) [2, 5]. Also, as effort of ROW-type methods is mainly determined by Jacobian computations, the possibility of applying arbitrary Jacobian approximations is utilized to decrease the number of Jacobian updates [7, 18].

*Corresponding author

Email addresses: `tim.jax@h-brs.de` (T. Jax), `gerd.steinebach@h-brs.de` (G. Steinebach)

In contrast, linearly implicit ROW-type schemes solving differential algebraic equations (DAEs) seem to be completely derived just with respect to exact or special approximated Jacobians. The theory of ROW methods for semi-explicit DAEs of index-1 based on Buthcher's theory of rooted trees is introduced by Roche [13] using exact Jacobians. In [12] Rentrop, Roche and Steinebach derive new conditions realizing a ROW-type scheme for DAEs whose Jacobian entries are set to zero with respect to the differential part. Thus, computational effort is decreased by restricting implicit solution to algebraic constraints, solving the differential part assumed to be non-stiff by the underlying RK method.

Regarding methods for DAEs based on arbitrary Jacobian approximations Strehmel, Weiner and Dannehl [17] describe linearly-implicit Runge-Kutta methods that cover ROW-type schemes with special and arbitrary approximations to Jacobian entries of the algebraic part. But focusing on aspects of stability a derivation of order conditions based on Butcher's theory of rooted trees is not taken into account. Also, the scheme is restricted to Jacobian entries of the differential part set to zero, thus solving it exclusively by the underlying explicit RK method as in [12]. However, the differential part of DAEs might not always be completely stiff or non-stiff. It might include single stiff components such as source terms. For this reason, we introduced a ROW-type method in [6] that regards Jacobian entries of the differential part reduced to stiff components. Thus, given differential equations can be solved partially implicit while given algebraic constraints are solved fully implicit. The scheme corresponds to an adapted ROW-AMF method for DAEs. But focusing on aspects of application, we derived no order conditions.

In fact, in literature there seems to be no detailed theory for deriving ROW-type schemes with arbitrarily approximated Jacobian entries when solving DAEs, especially regarding aspects of order conditions. That is, realizing W methods for DAEs requires further research. In this context, regarding arbitrarily approximated Jacobians for ROW-type methods applied to DAEs should yield a generalized scheme that enables to cover different known ROW-type schemes by special approximations and to connect them by a consistent theory.

For these reasons, the purpose of this article is given by two aspects: First, we intend to introduce an appropriate generalized ROW-type method for semi-explicit DAEs of index-1 that corresponds to a first attempt of realizing W methods for DAEs. Second, we intend to formulate a consistent theory in order to realize W methods for DAEs, especially regarding derivation of order conditions based on Butcher's theory of rooted trees. In this context, we will focus on arbitrary Jacobian approximations given with respect to the differential part combining approaches introduced by Roche [13] and Hairer, Wanner [4] regarding ROW methods for DAEs as well as approaches introduced by Steihaug, Wolfbrand [15] and Hairer, Wanner [4] regarding W methods for ODEs. Extensions to additional approximations given with respect to the algebraic part as presented in [17] will be considered in a further contribution.

The paper is organized as follows: Section 2 introduces the generalized ROW-type method and its properties, Section 3 describes Taylor expansions for analytical and numerical solution using rooted trees that include a new type of vertices, Section 4 defines order conditions and their properties up to order four, Section 5 shows numerical results and Section 6 gives a conclusion and outlook.

2. A Generalized ROW-Type Method for DAEs of Index-1

Below, we regard a semi-explicit DAE system given by

$$y'(x) = f\left(y(x), z(x)\right), \quad f = f_N + f_S, \quad y(x_0) = y_0 \tag{1a}$$
$$0 = g\left(y(x), z(x)\right) \qquad\qquad z(x_0) = z_0 \tag{1b}$$

with consistent initial values, i.e. $g(y_0, z_0) = 0$. The right hand side of differential part (1a) is assumed to allow for additive splitting into a non-stiff part f_N and a stiff part f_S. By defining corresponding vectors, this strategy also covers the special case of partitioning the differential part into non-stiff and stiff equations. Both the functions f and g are supposed to be sufficiently differentiable. Moreover, we assume partial derivatives g_z to be regular, i.e. the given DAE system is assumed to be of index-1. Without loss of generality, we restrict problem formulation (1) to the autonomous case. Explicit dependencies on x can be taken into account by introducing $x' = 1$ to the system.

In order to solve the given DAE system, we apply an adapted linearly implicit ROW-type method for DAEs defined by

$$y_1 = y_0 + \sum_{i=1}^{s} b_i k_i, \quad z_1 = z_0 + \sum_{i=1}^{s} b_i k_i^{alg} \tag{2a}$$

$$\begin{pmatrix} k_i \\ 0 \end{pmatrix} = h \begin{pmatrix} f\left(v_i, w_i\right) \\ g\left(v_i, w_i\right) \end{pmatrix} + h \sum_{j=1}^{i} \gamma_{ij} \begin{bmatrix} A_y & A_z \\ (g_y)_0 & (g_z)_0 \end{bmatrix} \begin{pmatrix} k_j \\ k_j^{alg} \end{pmatrix} \tag{2b}$$

$$v_i = y_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j, \quad w_i = z_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j^{alg}. \tag{2c}$$

Here, y_1 and z_1 are numerical solutions of the differential and algebraic part, respectively, given at $x_1 = x_0 + h$, i.e. after a single step with step-size h. Weights b_i and coefficients α_{ij}, γ_{ij} with $i = 1, ..., s$, $j = 1, ..., i$ correspond to real parameters, s denotes the number of internal stages. We assume $\gamma_{ii} = \gamma$. Thus, computing stage-values k_i and k_i^{alg} by (2b) yields a linear system for each $i = 1, ..., s$ whose matrices

$$\begin{bmatrix} I - \gamma_{ii} h A_y & -\gamma_{ii} h A_z \\ -\gamma_{ii} h (g_y)_0 & -\gamma_{ii} h (g_z)_0 \end{bmatrix}$$

are all equivalent. As a consequence, just one LU-decomposition is required within every step [13, 18]. The Jacobian given in (2b) regards exact partial derivatives $(g_y)_0$ and $(g_z)_0$ of the algebraic part evaluated at initial values (y_0, z_0). However, exact partial derivatives $(f_y)_0$ and $(f_z)_0$ of the differential part are replaced by arbitrary approximations A_y and A_z, respectively.

Due to the characteristic property of using arbitrary Jacobian approximations with respect to the differential part, scheme (2) corresponds to a new ROW-type class and could be interpreted as a first extension of W methods to the DAE case. Indeed, many ROW-type methods for ODEs and DAEs defined in literature are based on Jacobians with exact or specially approximated entries regarding the differential part. Thus, choosing appropriate approximations of matrices A_y and A_z most of these schemes are covered by ROW-type method (2) already. For this reason, it could be also considered as some generalized ROW-type scheme for DAEs. A list of schemes that result when choosing the Jacobian approximations of ROW-type method (2) appropriately is shown for problem

Table 1: Schemes covered by the generalized ROW-type method (ODE case).

Problem	A_y	A_z	Method	Properties	Ref.
$y' = f(y)$	0	/	RK	expl.	[3]
$y' = f(y)$	f_y	/	ROW	impl.	[4]
$y' = f(y)$	A_y	/	W	impl.	[15]
$y' = f_N(y) + f_S(y)$	$(f_S)_y$	/	ROW-AMF	expl./impl.	[5]
$\begin{pmatrix} y'_N \\ y'_S \end{pmatrix} = \begin{pmatrix} f_N(y_N, y_S) \\ f_S(y_N, y_S) \end{pmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & (f_S)_{y_S} \end{bmatrix}$	/	part. RK	expl./impl.	[11]

Table 2: Schemes covered by the generalized ROW-type method (DAE case).

Problem	A_y	A_z	Method	Properties	Ref.
$y' = f(y,z)$ $0 = g(y,z)$	f_y	f_z	ROW	impl.	[13]
$y' = f(y,z)$ $0 = g(y,z)$	0	0	part. ROW	expl./impl.	[12]
$y' = f_N(y) + f_S(y)$ $0 = g(y,z)$	$(f_S)_y$	f_z	ROW-AMF (DAEs)	expl./impl.	
$y' = f_N(y) + f_S(y)$ $0 = g(y,z)$	$(f_S)_y$	$(f_S)_z$	ROW-AMF (DAEs) (adapted)	expl./impl.	[6]

formulations of the ODE case in Table 1 and for problem formulations of the DAE case in Table 2. Among the schemes covered are purely explicit and implicit methods as well as implicit/explicit schemes that separate given stiff and non-stiff components by partitioning or additive splitting.

3. Taylor Expansions of Exact and Numerical Solution

In this section, we derive Taylor expansions of DAE system (1) with respect to its exact and numerical solution. Regarding Butcher's theory of rooted trees, we supplement the approach by Roche [13] introducing a new type of vertices to describe all occurring elementary differentials. The resulting strategy finally provides a consistent theory that combines ROW methods for DAEs with W methods for ODEs. In this context, we proceed analogously to derivations and descriptions given by Hairer and Wanner [4] and Roche [13], however, adapted to our numerical scheme including arbitrary Jacobian approximations with respect to the differential part.

3.1. Derivatives and Taylor Expansion of Exact Solution

We start regarding derivatives required to realize Taylor expansions with respect to analytical solutions of DAE system (1). As originally derived by Roche [13] they read for the differential part

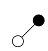
$$
\begin{aligned}
y' =& f \\
y'' =& f_y f + f_z(-g_z)^{-1} g_y f \\
y''' =& f_{yy}(f,f) + f_y f_y f + f_{yz}\left(f, (-g_z)^{-1} g_y f\right) + f_{zy}\left((-g_z)^{-1} g_y f, f\right) \\
&+ f_y(-g_z)^{-1} g_{yy}(f,f) + f_z(-g_z)^{-1} g_y f_y f + f_y f_z(-g_z)^{-1} g_y f \\
&+ f_z(-g_z)^{-1} g_y f_z(-g_z)^{-1} g_y f + f_{zz}\left((-g_z)^{-1} g_y f, (-g_z)^{-1} g_y f\right) \\
&+ f_z(-g_z)^{-1} g_{yz}\left(f, (-g_z)^{-1} g_y f\right) + f_z(-g_z)^{-1} g_{zy}\left((-g_z)^{-1} g_y f, f\right) \\
&+ f_z(-g_z)^{-1} g_{zz}\left((-g_z)^{-1} g_y f, (-g_z)^{-1} g_y f\right) \\
y'''' =& \ldots
\end{aligned}
$$

and for the algebraic part

$$
\begin{aligned}
z' =& (-g_z)^{-1} g_y f \\
z'' =& (-g_z)^{-1} g_{zy}\left((-g_z)^{-1} g_y f, f\right) + (-g_z)^{-1} g_{zz}\left((-g_z)^{-1} g_y f, (-g_z)^{-1} g_y f\right) \\
&+ (-g_z)^{-1} g_{yy}(f,f) + (-g_z)^{-1} g_{yz}\left(f, (-g_z)^{-1} g_y f\right) + (-g_z)^{-1} g_y f_y f \\
&+ (-g_z)^{-1} g_y f_z(-g_z)^{-1} g_y f \\
z''' =& \ldots
\end{aligned}
$$

All elementary differentials resulting within these derivatives of differential and algebraic part can be described by Butcher's theory of rooted trees. For this purpose, Roche [13] introduced the set of differential algebraic rooted trees.

Definition 1. Let $DAT = DAT_y \cup DAT_z$ $(DAT_y \cap DAT_z = \varnothing)$ denote the set of differential algebraic rooted trees recursively defined by

a) $\tau_y = \bullet \in DAT_y$, $\tau_z =$  $\in DAT_z$;

b) $[t_1, ..., t_m, u_1, ..., u_n]_y \in DAT_y$,
if $t_1, ..., t_m \in DAT_y$ and $u_1, ..., u_n \in DAT_z$;

c) $[t_1, ..., t_m, u_1, ..., u_n]_z \in DAT_z$,
if $t_1, ..., t_m \in DAT_y$, $u_1, ..., u_n \in DAT_z$ and $m + n \geq 2$;

d) $[t_1]_z \in DAT_z$, if $t_1 \in DAT_y$.

With $[t_1, ..., t_m, u_1, ..., u_n]_{y,z}$ unordered $(m+n)$-tuples [4].

Regarding graphical representation, elements of DAT_y are characterized by a meager root and elements of DAT_z are characterized by a fat root. In this context, $[.]_y$ denotes attaching a new meager root and $[.]_z$ denotes attaching a new fat root to all sub-trees given within brackets.

Definition 2. The order of a tree $t \in DAT_y$ or $u \in DAT_z$, denoted by $\rho(t)$ and $\rho(u)$, respectively, is the number of its meager vertices [13].

By the trees of set DAT all elementary differentials occurring in the derivatives for exact solution of DAE problem (1) can be identified distinctly [4, 13].

Definition 3. Elementary differentials $F(t)$ and $F(u)$ that correspond to trees of the set DAT are defined recursively by

a) $F(\tau_y) = f$, $F(\tau_z) = (-g_z)^{-1} g_y f$;

b) $F(t) = \frac{\partial^{m+n} f}{\partial y^m \partial z^n} (F(t_1), ..., F(t_m), F(u_1), ..., F(u_n))$,
if $t = [t_1, ..., t_m, u_1, ..., u_n]_y \in DAT_y$;

c) $F(u) = (-g_z)^{-1} \frac{\partial^{m+n} g}{\partial y^m \partial z^n} (F(t_1), ..., F(t_m), F(u_1), ..., F(u_n))$,
if $u = [t_1, ..., t_m, u_1, ..., u_n]_z \in DAT_z$, $m + n \geq 2$;

d) $F(u) = (-g_z)^{-1} g_y F(t_1)$, if $u = [t_1]_z \in DAT_z$.

Functions $F(t)$ and $F(u)$ are well defined as Definition 3 is unaffected by permutations of t_1, ..., t_m, u_1, ..., u_n due to the symmetry of partial derivatives [4]. Some of these differentials occur multiple times in derivatives of the given DAE system. In order to distinguish these elements distinctly Roche introduced the set of monotonically labeled trees $LDAT$ [13].

Definition 4. A tree $t \in DAT_y$ or $u \in DAT_z$ is called monotonically labeled if all of its meager vertices are labeled by an integer i with $1 \leq i \leq \rho(t)$ or $1 \leq i \leq \rho(u)$, respectively, that monotonically increases following any of its branches starting from the root. The set of all monotonically labeled trees is denoted by $LDAT$, where $LDAT = LDAT_y \cup LDAT_z$.

Based on these definitions, Taylor expansion of the analytical solution is given by derivatives of y and z resulting from following theorem by Roche [13].

Theorem 1. *Derivatives of an index-1 semi-explicit DAE system are given by*

$$y^{(q)}(x_0) = \sum_{\substack{t \in LDAT_y \\ \rho(t)=q}} F(t)(y_0, z_0) \quad \textit{and} \quad z^{(q)}(x_0) = \sum_{\substack{u \in LDAT_z \\ \rho(u)=q}} F(u)(y_0, z_0).$$

3.2. Derivatives of Numerical Solution

Based on elements of DAT derivatives of numerical solutions by generalized ROW-type method (2) can be determined. Their derivation is given in analogy to steps described by Hairer and Wanner [4] to obtain derivatives of ROW methods with exact Jacobian applied to DAEs. However, we will adapt their approach taking into account arbitrary Jacobian approximations with respect to the differential part.

In this context, interpreting numerical solutions given in (2a) as functions of step-size h derivatives at $h = 0$ read [4]:

$$y_1^{(q)}(0) = \sum_{i=1}^{s} b_i (k_i)^{(q)}(0) \qquad \text{and} \qquad z_1^{(q)}(0) = \sum_{i=1}^{s} b_i (k_i^{alg})^{(q)}(0). \tag{3}$$

Using Leibniz' rule as well as Faà di Bruno's formula, corresponding derivatives of stage-values k_i and k_i^{alg} defined by (2b) can be expressed by

$$(k_i)^{(q)} = q \sum_{\substack{t\in SLDAT_y \\ \rho(t)=q}} \frac{\partial^{m+n} f(y_0,z_0)}{\partial y^m \partial z^n} \left(v_i^{(\mu_1)},...,v_i^{(\mu_m)},w_i^{(\nu_1)},...,w_i^{(\nu_n)}\right)$$
$$+ qA_y \sum_{j=1}^{i} \gamma_{ij}(k_j)^{(q-1)} + qA_z \sum_{j=1}^{i} \gamma_{ij}(k_j^{alg})^{(q-1)} \tag{4}$$

and

$$(k_i^{alg})^{(q)} =$$
$$(-g_z)_0^{-1} \sum_{j=1}^{i} \omega_{ij} \sum_{\substack{u\in SLDAT_z \\ \rho(u)=q \\ m+n\geq 2}} \frac{\partial^{m+n} g(y_0,z_0)}{\partial y^m \partial z^n} \left(v_j^{(\mu_1)},...,v_j^{(\mu_m)},w_j^{(\nu_1)},...,w_j^{(\nu_n)}\right)$$
$$+ (-g_z)_0^{-1}(g_y)_0(k_i)^{(q)}. \tag{5}$$

when assuming all components computed at $h = 0$. Here, $(\omega_{ij})_{i,j=1}^{s} = B^{-1}$ with $B = (\beta_{ij})_{i,j=1}^{s}$ and $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$ [4, 9]. Elements $t \in SLDAT_y$ and $u \in SLDAT_z$ correspond to special labeled trees of DAT with orders $\rho(t)$ and $\rho(u)$. They are characterized by having no ramification except at the root and fat vertices occurring just directly connected to the root or being the root itself [4]. Parameters m and n denote the number of their sub-trees starting with a meager or fat root, respectively. Besides, occurring derivatives of v_i and w_i defined by (2c) read

$$v_i^{(\mu_\xi)} = \sum_{\kappa_\xi=1}^{i-1} \alpha_{i\kappa_\xi}(k_{\kappa_\xi})^{(\mu_\xi)} \qquad \text{and} \qquad w_i^{(\nu_\xi)} = \sum_{\kappa_{m+\xi}=1}^{i-1} \alpha_{i\kappa_{m+\xi}}(k_{\kappa_{m+\xi}}^{alg})^{(\nu_\xi)} \tag{6}$$

where μ_ξ denote orders of sub-trees t_ξ having a meager root and ν_ξ denote orders of sub-trees u_ξ having a fat root [4].

By inserting resulting formulations (4) and (5) into (3), derivatives of y_1 and z_1 can be given explicitly. For the differential part they read

$$\begin{aligned}
y_1' =& \sum b_i f \\
y_1'' =& 2\cdot \sum b_i\alpha_{ij}\cdot f_y f + 2\cdot \sum b_i\alpha_{ij}\cdot f_z(-g_z)^{-1}g_y f \\
&+ 2\cdot \sum b_i\gamma_{ij}\cdot A_y f + 2\cdot \sum b_i\gamma_{ij}A_z(-g_z)^{-1}g_y f \\
y_1''' =& 3\cdot \sum b_i\alpha_{ij}\alpha_{ik}\cdot f_{yy}(f,f) + 3\cdot \sum b_i\alpha_{ij}\alpha_{ik}\cdot f_{yz}(f,(-g_z)^{-1}g_y f) \\
&+ 3\cdot \sum b_i\alpha_{ij}\alpha_{ik}\cdot f_{zy}((-g_z)^{-1}g_y f, f) \\
&+ 3\cdot \sum b_i\alpha_{ij}\alpha_{ik} f_{zz}((-g_z)^{-1}g_y f,(-g_z)^{-1}g_y f) \\
&+ 6\cdot \sum b_i\alpha_{ij}\alpha_{jk}\cdot f_y f_y f + 6\cdot \sum b_i\alpha_{ij}\alpha_{jk}\cdot f_y f_z(-g_z)^{-1}g_y f \\
&+ 6\cdot \sum b_i\alpha_{ij}\gamma_{jk}\cdot f_y A_y f + 6\cdot \sum b_i\alpha_{ij}\gamma_{jk} f_y A_z(-g_z)^{-1}g_y f \\
&...
\end{aligned}$$

while for the algebraic part they read

$$
\begin{aligned}
z_1' =& \sum b_i(-g_z)^{-1} g_y f \\
z_1'' =& \sum b_i \omega_{ij}\alpha_{jk}\alpha_{jl} \cdot (-g_z)^{-1} g_{yy}(f,f) \\
&+ \sum b_i \omega_{ij}\alpha_{jk}\alpha_{jl} \cdot (-g_z)^{-1} g_{yz}(f,(-g_z)^{-1} g_y f) \\
&+ \sum b_i \omega_{ij}\alpha_{jk}\alpha_{jl} \cdot (-g_z)^{-1} g_{zy}((-g_z)^{-1} g_y f, f) \\
&+ \sum b_i \omega_{ij}\alpha_{jk}\alpha_{jl} \cdot (-g_z)^{-1} g_{zz}((-g_z)^{-1} g_y f, (-g_z)^{-1} g_y f) \\
&+ 2 \cdot \sum b_i \alpha_{ij} \cdot (-g_z)^{-1} g_y f_y f + 2 \cdot \sum b_i \alpha_{ij} \cdot (-g_z)^{-1} g_y f_z (-g_z)^{-1} g_y f \\
&+ 2 \cdot \sum b_i \gamma_{ij} \cdot (-g_z)^{-1} g_y A_y f + 2 \cdot \sum b_i \gamma_{ij} \cdot (-g_z)^{-1} g_y A_z (-g_z)^{-1} g_y f \\
z_1''' =& \ldots
\end{aligned}
$$

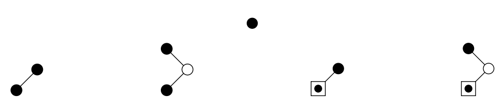
3.3. Trees and Elementary Differentials

Derivatives of the numerical solution by generalized ROW-type method (2) consider besides exact differentials known for the analytical solution additional non-exact differentials. Non-exact differentials are characterized by arbitrarily approximated Jacobian components A_y and A_z. These cannot be described by meager and fat vertices known for set DAT that correspond to exact representations of f and $(-g_z)^{-1}g$ already. Hence, a new type of vertices must be introduced to identify the non-exact differentials by appropriate trees.

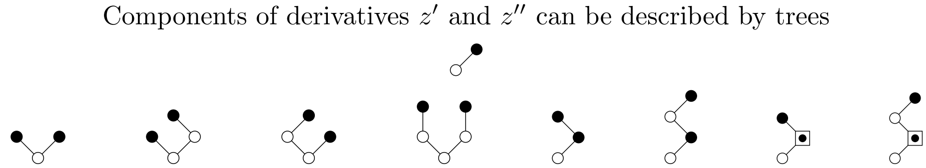
We describe this new type of vertices representing approximated differential components A using a meager vertex framed by a square. Together with meager vertices describing exact differential components f and fat vertices describing exact algebraic components $(-g_z)^{-1}g$ introduced by Roche [13] this enables to express all elementary differentials occurring for the numerical solution by distinct trees without violating definition of set DAT. Analogously to trees known for the set DAT, derivatives of f and g will be given by a branch leaving a meager or a fat vertex, respectively, while derivatives of A will be described by a branch leaving a meager vertex with square frame. In this context, any branch being followed by a meager vertex with or without a square frame now corresponds to a derivative with respect to y while any branch being followed by a fat vertex still corresponds to a derivative with respect to z.

Example.

Components of derivatives y' and y'' can be described by trees



Components of derivatives z' and z'' can be described by trees

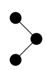
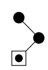
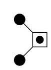
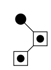


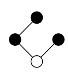
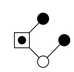
Constructing trees that consist just of meager and fat vertices is still given by rules of set DAT according to Definition 1. However, new trees including additional meager vertices framed by a square are not yet defined. In the following, we will refer to such trees representing non-exact differentials with respect to DAE system (1) as approximated differential algebraic rooted trees.

Definition 5. Let $ADAT^D = ADAT_y^D \cup ADAT_z^D$ $(ADAT_y^D \cap ADAT_z^D = \varnothing)$ denote the set of approximated differential algebraic rooted trees with respect to the differential part of DAE system (1). Elements of $ADAT^D$ include at least one meager vertex framed by a square representing an arbitrary approximation to differential function f. Elements of $ADAT_y^D$ consider a meager root with or without square frame, elements of $ADAT_z^D$ consider fat root.

For constructing elements of $ADAT^D$ properties of trees used to describe W methods for ODEs as considered in [4] can be applied. Hence, arbitrary approximations of f_y and f_z denoted by A_y and A_z can be described only by new vertices given within singly branched trees. They will never be the center of a ramification and they will never be the end of a branch. This follows from (4). Indeed, due to this property new trees can be determined by permuting a square frame over all inner meager vertices given within single branches of trees resulting for DAT, including possible singly branched meager roots.

Example.

 $\in DAT_y$ yields trees  ,  ,  $\in ADAT_y^D$

 $\in DAT_z$ yields tree  $\in ADAT_z^D$

Definition 6. Let $\Phi(t) = \{t, t_1, t_2, ...\}$ and $\Phi(u) = \{u, u_1, u_2, ...\}$ denote compilations of trees characterized by following properties:

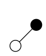
a) $\Phi(t)$ includes $t \in DAT_y$ and all elements $t_1, t_2, ... \in ADAT_y^D$ that result from permuting a square frame over all inner meager vertices given within single branches of $t \in DAT_y$ and its possible singly branched meager root.

b) $\Phi(u)$ includes $u \in DAT_z$ and all elements $u_1, u_2, ... \in ADAT_z^D$ that result from permuting a square frame over all inner meager vertices given within single branches of $u \in DAT_z$.

Remark 1. All elements of $\Phi(t)$ and $\Phi(u)$ equal $t \in DAT_y$ and $u \in DAT_z$, respectively, when using $A_y = f_y$ and $A_z = f_z$, i.e. replacing meager vertices with square frame by meager vertices without square frame.

For convenience of subsequent definitions we describe construction of all trees representing differentials of the numerical solution by a superior set that combines elements of DAT and $ADAT^D$.

Definition 7. Let $CDAT^D = CDAT_y^D \cup CDAT_z^D$ $(CDAT_y^D \cap CDAT_z^D = \varnothing)$ denote the set of combined differential algebraic rooted trees characterized by $CDAT^D = DAT \cup ADAT^D$ $(DAT \cap ADAT^D = \varnothing)$. The set is recursively defined by

a) $\tau_y = \bullet \in CDAT_y^D$, $\tau_z =$  $\in CDAT_z^D$;

b) $[t_1, ..., t_m, u_1, ..., u_n]_y \in CDAT_y^D$,
if $t_1, ..., t_m \in CDAT_y^D$ and $u_1, ..., u_n \in CDAT_z^D$;

c) $[t_1]_{\tilde{y}} \in CDAT_y^D$, if $t_1 \in CDAT_y^D$;

d) $[u_1]_{\tilde{y}} \in CDAT_y^D$, if $u_1 \in CDAT_z^D$;

e) $[t_1, ..., t_m, u_1, ..., u_n]_z \in CDAT_z^D$,
if $t_1, ..., t_m \in CDAT_y^D$ and $u_1, ..., u_n \in CDAT_z^D$, $m + n \geq 2$;

f) $[t_1]_z \in CDAT_z^D$, if $t_1 \in CDAT_y^D$

with $[t_1, ..., t_m, u_1, ..., u_n]_{y,\tilde{y},z}$ corresponding to unordered $(m+n)$-tuples.

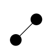
Remark 2. By Definitions 1 and 7 it holds $ADAT^D = CDAT^D \backslash DAT$.

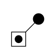
Remark 3. $CDAT_y^D = DAT_y \cup ADAT_y^D$ and $CDAT_z^D = DAT_z \cup ADAT_z^D$.

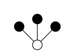
Remark 4. $\Phi(t)$ and $\Phi(u)$ are subsets of $CDAT_y^D$ and $CDAT_z^D$, respectively.

Regarding graphical representation, $[.]_y$ denotes attaching a meager root and $[.]_z$ denotes attaching a fat root to all sub-trees given within brackets as known for set DAT. Analogously, $[.]_{\tilde{y}}$ denotes attaching a meager root framed by a square. By definition, all trees t characterized by a meager root with or without square frame correspond to elements of $CDAT_y^D$ while all trees u with a fat root correspond to elements of $CDAT_z^D$.

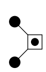
Example.

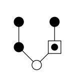

t_1

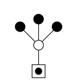

t_2


u_1

are sub-trees that enable to construct trees such as


$t = [t_2]_y \in CDAT_y^D$


$u = [t_1, t_2]_z \in CDAT_z^D$


$t = [u_1]_{\tilde{y}} \in CDAT_y^D$

Definition 8. The order of a tree $t \in CDAT_y^D$ or $u \in CDAT_z^D$, denoted by $\rho(t)$ or $\rho(u)$, respectively, corresponds to the number of all its meager vertices given with and without a square frame.

Trees of set $CDAT^D$ enable to identify all exact and non-exact differentials resulting for derivatives of DAE system (1) regarding its analytical solution and numerical computation by generalized ROW-type method (2).

Definition 9. The elementary differentials $F(t)$ and $F(u)$ corresponding to trees of the set $CDAT^D$ are defined recursively by

a) $F(\tau_y) = f$, $F(\tau_z) = (-g_z)^{-1} g_y f$;

b) $F(t) = \frac{\partial^{m+n} f}{\partial y^m \partial z^n}\,(F(t_1), ..., F(t_m), F(u_1), ..., F(u_n))$,
if $t = [t_1, ..., t_m, u_1, ..., u_n]_y \in CDAT_y^D$;

c) $F(t) = A_y F(t_1)$, if $t = [t_1]_{\tilde{y}} \in CDAT_y^D$;

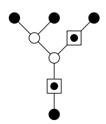
d) $F(t) = A_z F(u_1)$, if $t = [u_1]_{\tilde{y}} \in CDAT_y^D$;

e) $F(u) = (-g_z)^{-1} \frac{\partial^{m+n} g}{\partial y^m \partial z^n}\,(F(t_1), ..., F(t_m), F(u_1), ..., F(u_n))$,
if $u = [t_1, ..., t_m, u_1, ..., u_n]_z \in CDAT_z^D$, $m + n \geq 2$;

f) $F(u) = (-g_z)^{-1} g_y F(t_1)$, if $u = [t_1]_z \in CDAT_z^D$.

Remark 5. Analogously to Definition 3, the elementary differentials in Definition 9 are unaffected by permutation of $t_1, ..., t_m, u_1, ..., u_n$ due to symmetry of partial derivatives. Hence, $F(t)$ and $F(u)$ are well defined [4, 13].

Example.

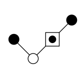
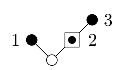
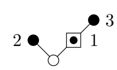
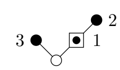


corresponds to differential
$F(t) = f_y A_z (-g_z)^{-1} g_{zy}((-g_z)^{-1} g_{yy}(f, f), A_y f)$

As for derivatives of the exact solution, elementary differentials can occur several times within a derivative of the numerical solution even with respect to non-exact differentials. In order to identify these multiple differentials the concept of labeled trees given in Definition 4 is also applied to elements of $CDAT^D$.

Definition 10. $LCDAT^D$ denotes the set of monotonically labeled rooted trees of $CDAT^D$ with $LCDAT^D = LCDAT_y^D \cup LCDAT_z^D$. Elements $t \in LCDAT_y^D$ and $u \in LCDAT_z^D$ are characterized by integer labels i with $1 \leq i \leq \rho(t)$ and $1 \leq i \leq \rho(u)$. Labels are given for each meager vertex with and without square frame and monotonically increase for every branch starting from the root.

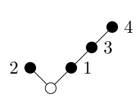
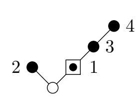
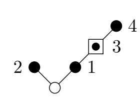
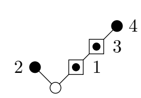
Example.

 can be monotonically labeled by:  ,  , 

Remark 6. It holds $LCDAT^D = LDAT \cup LADAT^D$ with $LADAT^D$ the set of monotonically labeled trees of $ADAT^D$.

Remark 7. Labeling of trees $t \in LDAT_y$ and $u \in LDAT_z$ is adopted by every element of corresponding subsets $\Phi(t)$ and $\Phi(u)$. Thus, for $A_y = f_y$ and $A_z = f_z$ all elements of $\Phi(t)$ and $\Phi(u)$ equal $t \in LDAT_y$ and $u \in LDAT_z$, respectively.

Example.

 yields  ,  , 

$u \in LDAT_z$ $\quad$ $u_1 \in LADAT_z^D$ $\quad$ $u_2 \in LADAT_z^D$ $\quad$ $u_3 \in LADAT_z^D$

$\Phi(u) = \{u, u_1, u_2, u_3\}$

3.4. Taylor Expansion of the Numerical Solution

In order to completely describe Taylor expansions of the numerical solution by generalized ROW-type method (2) we first define corresponding coefficients.

Definition 11. Let $\phi_i(t)$ and $\phi_i(u)$ be the coefficients in front of differentials occurring for stage-value derivatives of generalized ROW-type method (2). Regarding trees of $CDAT^D$, $\phi_i(t)$ and $\phi_i(u)$ are recursively defined by

a) $\phi_i(\tau_y) = 1$, $\phi_i(\tau_z) = 1$;

b) $\phi_i(t) = \sum \alpha_{i\kappa_1} \cdot ... \cdot \alpha_{i\kappa_m} \cdot \alpha_{i\kappa_{m+1}} \cdot ... \cdot \alpha_{i\kappa_{m+n}} \cdot$
$\phi_{\kappa_1}(t_1) \cdot ... \cdot \phi_{\kappa_m}(t_m) \cdot \phi_{\kappa_{m+1}}(u_1) \cdot ... \cdot \phi_{\kappa_{m+n}}(u_n)$,
if $t = [t_1, ..., t_m, u_1, ..., u_n]_y \in CDAT_y^D$;

c) $\phi_i(t) = \sum \gamma_{i\kappa} \phi_\kappa(t_1)$, if $t = [t_1]_{\tilde{y}} \in CDAT_y^D$;

d) $\phi_i(t) = \sum \gamma_{i\kappa} \phi_\kappa(u_1)$, if $t = [u_1]_{\tilde{y}} \in CDAT_y^D$;

e) $\phi_i(u) = \sum \omega_{ij} \cdot \alpha_{j\kappa_1} \cdot ... \cdot \alpha_{j\kappa_m} \cdot \alpha_{j\kappa_{m+1}} \cdot ... \cdot \alpha_{j\kappa_{m+n}} \cdot$
$\phi_{\kappa_1}(t_1) \cdot ... \cdot \phi_{\kappa_m}(t_m) \cdot \phi_{\kappa_{m+1}}(u_1) \cdot ... \cdot \phi_{\kappa_{m+n}}(u_n)$,
if $u = [t_1, ..., t_m, u_1, ..., u_n]_z \in CDAT_z^D$, $m + n \geq 2$;

f) $\phi_i(u) = \phi_i(t_1)$, if $u = [t_1]_z \in CDAT_z^D$.

Given sums run over all j, κ, κ_1, ..., κ_m, κ_{m+1}, ..., κ_{m+n}.

Definition 12. Let $\gamma(t)$ and $\gamma(u)$ be the integer coefficients in front of differentials occurring for stage-value derivatives of generalized ROW-type method (2). Regarding trees of $CDAT^D$, $\gamma(t)$ and $\gamma(u)$ are recursively defined by

a) $\gamma(\tau_y) = 1$, $\gamma(\tau_z) = 1$;

b) $\gamma(t) = \rho(t) \cdot \gamma(t_1) \cdot ... \cdot \gamma(t_m) \cdot \gamma(u_1) \cdot ... \cdot \gamma(u_n)$,
if $t = [t_1, ..., t_m, u_1, ..., u_n]_y \in CDAT_y^D$;

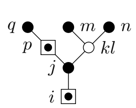
c) $\gamma(t) = \rho(t)\gamma(t_1)$, if $t = [t_1]_{\tilde{y}} \in CDAT_y^D$;

d) $\gamma(t) = \rho(t)\gamma(u_1)$, if $t = [u_1]_{\tilde{y}} \in CDAT_y^D$;

e) $\gamma(u) = \gamma(t_1) \cdot ... \cdot \gamma(t_m) \cdot \gamma(u_1) \cdot ... \cdot \gamma(u_n)$,
if $u = [t_1, ..., t_m, u_1, ..., u_n]_z \in CDAT_z^D$, $m + n \geq 2$;

f) $\gamma(u) = \gamma(t_1)$, if $u = [t_1]_z \in CDAT_z^D$.

Example.



yields

$$\phi_i(t) = \sum \gamma_{ij}\alpha_{jk}\omega_{kl}\alpha_{lm}\alpha_{ln}\alpha_{jp}\gamma_{pq}$$
and
$$\gamma(t) = 120$$

Based on Definitions 9, 11 and 12 derivatives of stage-values k_i and k_i^{alg} corresponding to (4) and (5) can be formulated using trees of $LCDAT^D$.

Theorem 2. *Stage-value derivatives for generalized ROW-type method (2) read*

$$(k_i)^{(q)} = \sum_{\substack{t \in LCDAT_y^D \\ \rho(t)=q}} \gamma(t)\phi_i(t)F(t)(y_0, z_0)$$

$$(k_i^{alg})^{(q)} = \sum_{\substack{u \in LCDAT_z^D \\ \rho(u)=q}} \gamma(u)\phi_i(u)F(u)(y_0, z_0)$$

with elementary differentials and coefficients according to Definitions 9, 11, 12.

Proof. Analogously to ROW methods for DAEs given in [4] by induction on q for (4) and (5) and rearranging resulting summations afterwards. □

Based on Theorem 2, Taylor expansions of numerical solutions by (2) can finally be constructed considering subsequent Theorem 3.

Theorem 3. *Numerical solution of generalized ROW-type method (2) satisfies*

$$y_1^{(q)}\Big|_{h=0} = \sum_{\substack{t \in LCDAT_y^D \\ \rho(t)=q}} \gamma(t) \cdot \sum_{j=1}^{s} b_i\phi_i(t)F(t)(y_0, z_0)$$

$$z_1^{(q)}\Big|_{h=0} = \sum_{\substack{u \in LCDAT_z^D \\ \rho(u)=q}} \gamma(u) \cdot \sum_{j=1}^{s} b_i\phi_i(u)F(u)(y_0, z_0)$$

with elementary differentials and coefficients according to Definitions 9, 11, 12.

4. Order Conditions

By comparing components of Taylor expansions for analytical and numerical solution according to Theorem 1 and Theorem 3 conditions for realizing generalized ROW-type method (2) up to a certain order can be determined.

Theorem 4. *A generalized ROW-type method (2) consistent of order q satisfies*

$$\begin{aligned}
&y(x_0 + h) - y_1 = \mathcal{O}(h^{q+1}) \quad \textit{iff} \\
&\qquad \sum_{i=1}^{s} b_i\phi_i(t) = 1/\gamma(t) \qquad \textit{for} \quad t \in DAT_y,\ \rho(t) \le q \\
&\qquad \sum_{i=1}^{s} b_i\phi_i(t) = 0 \qquad \textit{for} \quad t \in ADAT_y^D,\ \rho(t) \le q \\
&z(x_0 + h) - z_1 = \mathcal{O}(h^{q}) \quad \textit{iff} \\
&\qquad \sum_{i=1}^{s} b_i\phi_i(u) = 1/\gamma(u) \qquad \textit{for} \quad u \in DAT_z,\ \rho(u) \le q \\
&\qquad \sum_{i=1}^{s} b_i\phi_i(u) = 0 \qquad \textit{for} \quad u \in ADAT_z^D,\ \rho(u) \le q
\end{aligned}$$

with coefficients $\phi_i(t)$, $\phi_i(u)$ and $\gamma(t)$, $\gamma(u)$ according to Definitions 11 and 12.

Remark 8. Theorem 4 is given in analogy to descriptions by Hairer and Wanner regarding W methods for ODEs and ROW methods for DAEs [4].

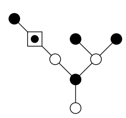
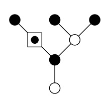
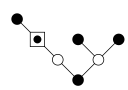
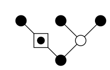
Remark 9. We assume g_z to be regular in a neighborhood of solution $y(x)$, $z(x)$ resulting from DAE system (1) with consistent initial values $y(x_0)$, $z(x_0)$. Thus, consistency of order q yields convergence of order q for both differential and algebraic part if the stability function satisfies $|R(\infty)| < 1$. This follows from the Global Convergence Theorem 1 by Deuflhard, Hairer, Zugck [1, 4, 12].

Numerous order conditions of generalized ROW-type method (2) can be saved as many trees of $CDAT^D$ yield identical coefficients.

Proposition 1. *Analogously to Proposition (4.6) by Roche [13] conditions for trees including fat vertices directly followed by a meager vertex with or without square frame are redundant. That is, order conditions for trees $u = [t_1]_z$ with $u \in CDAT_z^D$ are equal to conditions of their sub-tree $t_1 \in CDAT_y^D$.*

Proof. By Theorem 4 together with f) given in Definitions 11 and 12. □

Example.

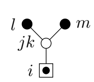
 ,  ,  yield conditions equal to 

Proposition 2. *Order conditions of trees $t = [[t_1, ..., t_m, u_1, ..., u_n]_z]_{\tilde{y}}$ regarding $m + n \geq 2$ are already satisfied by a combination of conditions resulting for trees $t^* = [t_1, ..., t_m, u_1, ..., u_n]_y$ and $t^{**} = [[t_1, ..., t_m, u_1, ..., u_n]_z]_y$ with $t, t^*, t^{**}, t_1, ..., t_m \in CDAT_y^D$ and $u_1, ..., u_n \in CDAT_z^D$. It holds:*

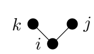
$$\gamma(t) \cdot \sum b_i\phi_i(t) = \gamma(t^*) \cdot \sum b_i\phi_i(t^*) - \gamma(t^{**}) \cdot \sum b_i\phi_i(t^{**}).$$

Proof. By Theorem 4 together with b), d), e) in Definitions 11 and 12, taking into account $B = (\beta_{ij})_{i,j=1}^s$ with $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$ and $(\omega_{ij})_{i,j=1}^s = B^{-1}$. □

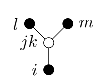
Example.

$t =$ 

yields condition
$\sum b_i\gamma_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} = 0$

$t^* =$ 

yields condition
$\sum b_i\alpha_{ij}\alpha_{ik} = 1/3$

$t^{**} =$ 

yields condition
$\sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} = 1/3$

$$\begin{aligned}\sum b_i\gamma_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} &= \sum b_i(\beta_{ij} - \alpha_{ij})\omega_{jk}\alpha_{kl}\alpha_{km} \\ &= \sum b_i\beta_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} - \sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} \\ &= \sum b_i\alpha_{ij}\alpha_{ik} - \sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} = 1/3 - 1/3 = 0\end{aligned}$$

Table 3: Order Conditions up to Order Four.

Order Conditions for Differential Part (1/2)

$\rho(t)$	Req.	No.	DAT	$ADAT^D$	Condition	Comment		
1	X	1			$\sum b_i = 1$	RK	W	Re/Ro/St
2	X	2			$\sum b_i\alpha_{ij} = 1/2$	RK	W	Re/Ro/St
	X	3			$\sum b_i\gamma_{ij} = 0$		W	
3	X	4			$\sum b_i\alpha_{ij}\alpha_{ik} = 1/3$	RK	W	Re/Ro/St
	X	5			$\sum b_i\alpha_{ij}\alpha_{jk} = 1/6$	RK	W	Re/Ro/St
	X	6			$\sum b_i\gamma_{ij}\alpha_{jk} = 0$		W	
	X	7			$\sum b_i\alpha_{ij}\gamma_{jk} = 0$		W	
	X	8			$\sum b_i\gamma_{ij}\gamma_{jk} = 0$		W	
	X	9			$\sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} = 1/3$			Re/Ro/St
		10			$\sum b_i\gamma_{ij}\omega_{jk}\alpha_{kl}\alpha_{km} = 0$			AC
4	X	11			$\sum b_i\alpha_{ij}\alpha_{ik}\alpha_{il} = 1/4$	RK	W	Re/Ro/St
	X	12			$\sum b_i\alpha_{ij}\alpha_{ik}\alpha_{kl} = 1/8$	RK	W	Re/Ro/St
	X	13			$\sum b_i\alpha_{ij}\alpha_{ik}\gamma_{kl} = 0$		W	
	X	14			$\sum b_i\alpha_{ij}\alpha_{jk}\alpha_{jl} = 1/12$	RK	W	Re/Ro/St
	X	15			$\sum b_i\gamma_{ij}\alpha_{jk}\alpha_{jl} = 0$		W	
	X	16			$\sum b_i\alpha_{ij}\alpha_{jk}\alpha_{kl} = 1/24$	RK	W	Re/Ro/St
	X	17			$\sum b_i\gamma_{ij}\alpha_{jk}\alpha_{kl} = 0$		W	
	X	18			$\sum b_i\alpha_{ij}\gamma_{jk}\alpha_{kl} = 0$		W	
	X	19			$\sum b_i\alpha_{ij}\alpha_{jk}\gamma_{kl} = 0$		W	
	X	20			$\sum b_i\gamma_{ij}\alpha_{jk}\gamma_{kl} = 0$		W	
	X	21			$\sum b_i\gamma_{ij}\gamma_{jk}\alpha_{kl} = 0$		W	
	X	22			$\sum b_i\alpha_{ij}\gamma_{jk}\gamma_{kl} = 0$		W	
	X	23			$\sum b_i\gamma_{ij}\gamma_{jk}\gamma_{kl} = 0$		W	
	X	24			$\sum b_i\alpha_{ij}\alpha_{ik}\omega_{kl}\alpha_{lm}\alpha_{ln} = 1/4$			Re/Ro/St

Order Conditions for Differential Part (2/2)

$\rho(t)$	Req.	No.	DAT	$ADAT^D$	Condition	Comment
	X	25			$\sum b_i\alpha_{ij}\alpha_{jk}\omega_{kl}\alpha_{lm}\alpha_{ln} = 1/12$	Re/Ro/St
	X	26			$\sum b_i\gamma_{ij}\alpha_{jk}\omega_{kl}\alpha_{lm}\alpha_{ln} = 0$	AC
		27			$\sum b_i\alpha_{ij}\gamma_{jk}\omega_{kl}\alpha_{lm}\alpha_{ln} = 0$	AC
		28			$\sum b_i\gamma_{ij}\gamma_{jk}\omega_{kl}\alpha_{lm}\alpha_{ln} = 0$	AC
	X	29			$\sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\alpha_{kn} = 1/4$	Re/Ro/St
		30			$\sum b_i\gamma_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\alpha_{kn} = 0$	AC
	X	31			$\sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\alpha_{mn} = 1/8$	Re/Ro/St
		32			$\sum b_i\gamma_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\alpha_{mn} = 0$	AC
	X	33			$\sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\gamma_{mn} = 0$	AC
		34			$\sum b_i\gamma_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\gamma_{mn} = 0$	AC
	X	35			$\sum b_i\alpha_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\omega_{mn}\alpha_{np}\alpha_{nq} = 1/4$	Re/Ro/St
		36			$\sum b_i\gamma_{ij}\omega_{jk}\alpha_{kl}\alpha_{km}\omega_{mn}\alpha_{np}\alpha_{nq} = 0$	AC

Order Conditions for Algebraic Part

$\rho(t)$	Req.	No.	DAT	$ADAT^D$	Condition	Comment
2	X	37			$\sum b_i\omega_{ij}\alpha_{jk}\alpha_{jl} = 1$	Re/Ro/St
3	X	38			$\sum b_i\omega_{ij}\alpha_{jk}\alpha_{jl}\alpha_{jm} = 1$	Re/Ro/St
	X	39			$\sum b_i\omega_{ij}\alpha_{jk}\alpha_{jl}\alpha_{lm} = 1/2$	Re/Ro/St
	X	40			$\sum b_i\omega_{ij}\alpha_{jk}\alpha_{jl}\gamma_{lm} = 0$	AC
	X	41			$\sum b_i\omega_{ij}\alpha_{jk}\alpha_{jl}\omega_{lm}\alpha_{mn}\alpha_{mp} = 1$	Re/Ro/St

Table 3 shows conditions that result for generalized ROW-type method (2) up to order four. In order to identify characteristic properties below, it includes elements that can be saved by Proposition 2. This enables to completely define subsets $\Phi(t)$ and $\Phi(u)$ that result for occurring trees $t \in DAT_y$ and $u \in DAT_z$. Elements belonging to such subsets are given between horizontal lines. However, conditions necessarily required for implementation are additionally marked.

The conditions listed in Table 3 directly cover order conditions known for explicit RK schemes for ODEs (RK) [3], W methods for ODEs (W) [15] and ROW-type methods for DAEs introduced by Rentrop, Roche, Steinebach (Re/Ro/St) [12]. Besides, there are up to ten additional conditions (AC).

Due to listed elements of set $ADAT^D$, order conditions of ROW methods for DAEs by Roche [13] are implicitly included. They result replacing arbitrary approximations A_y and A_z by their exact representatives f_y and f_z, i.e. replacing meager vertices with square frame by meager vertices without square frame. Doing so, trees of $ADAT^D$ following a tree of DAT in Table 3 will turn into this element of DAT. As a consequence, all elements of subsets $\Phi(t)$ and $\Phi(u)$ given between horizontal lines in Table 3 then yield same elementary differentials. Hence, their coefficients sum up to conditions introduced by Roche [13].

By this property, conditions for trees of $ADAT^D$ can be interpreted as coupling conditions. In the DAE case, they connect the ROW method by Roche [13] resulting for $A_y = f_y$, $A_z = f_z$ with the ROW-type method by Rentrop, Roche, Steinebach [12] resulting for $A_y = 0$, $A_z = 0$. Moreover, they supplement conditions by Rentrop, Roche, Steinebach [12] to regard arbitrarily approximated Jacobian entries of the differential part. In the ODE case, they connect ROW methods resulting for $A_y = f_y$ and RK methods resulting for $A_y = 0$.

By Proposition 2 seven of the ten additional conditions listed in Table 3 can be saved. Remaining additional conditions are (26), (33) and (40). Condition (40) follows directly from (39) and condition $\sum b_i \omega_{ij} \alpha_{jk} \alpha_{jl} \beta_{lm} = 1/2$ originally introduced by Roche [13] as $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$. However, conditions (26) and (33) seem to be new conditions, not considered in literature so far.

New conditions occur not before order four. Thus, a corresponding method of order three can be realized by known conditions (1) - (9) and (37) already.

Theorem 5. *Any stiffly accurate ROW-type method of order three with four internal stages that satisfies*

- *conditions of W methods for ODEs (1)-(8) [4, 15], plus*
- *conditions by Scholz [14] or, equivalently, Ostermann and Roche [8]*

enables to save conditions (9) and (37) and thus proves to be appropriate for realizing generalized ROW-type method (2) up to order three.

Remark 10. A stiffly accurate ROW-type method is characterized by [9]

$$\beta_{si} = b_i, \quad i = 1, ..., s, \quad \text{and} \quad \alpha_s = \sum_{j=1}^{s-1} \alpha_{sj} = 1.$$

Remark 11. Relevant conditions by Scholz [14] or Ostermann and Roche [8] reduce effects of order reduction. They read [9]

$$b^T B^j (2B^2 e - \alpha^2) = 0, \quad 1 \leq j \leq 2$$

with $B = (\beta_{ij})_{i,j=1}^{s}$, $\alpha^2 = (\alpha_1^2, ..., \alpha_s^2)^T$, $\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}$, $e = (1, ..., 1)^T \in \mathbb{R}^s$ [9]. For third order schemes with four stages satisfying (1) - (8) they simplify to [9]

$$b_4\beta_{32}\beta_{43}\alpha_2^2 = 2\gamma^4 - 2\gamma^3 + \frac{1}{3}\gamma^2 \tag{38a}$$

$$b_3\beta_{32}\alpha_2^2 + b_4(\beta_{42}\alpha_2^2 + \beta_{43}\alpha_3^2) = 2\gamma^3 - 3\gamma^2 + \frac{2}{3}\gamma \tag{39a}$$

$$b_4\beta_{43}\beta_{32}\beta_{21} = 0. \tag{40a}$$

For stiffly accurate schemes (38a) and (39a) are equal. Thus, it follows [9]

$$b_3\beta_{32}\alpha_2^2 = 2\gamma^3 - 2\gamma^2 + \frac{1}{3}\gamma \tag{38b}$$

$$b_3\beta_{32}\beta_2 = 0. \tag{40b}$$

Remark 12. As is known, ROW-type methods of order three with four stages satisfying (1) - (8) plus (38a) - (40a) also fulfill (37) [9]. Hence, proving Theorem 5 just requires to show that (9) is satisfied by (1) - (8) and (38b), (40b).

PROOF. We assume the stiffly accurate case and four internal stages. By using $(\omega_{ij})_{i,j=1}^{s} = B^{-1}$ with $B = (\beta_{ij})_{i,j=1}^{s}$, $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$ condition (9) thus reads

$$b_3\alpha_{32}\alpha_2^2/\gamma + b_4\alpha_{42}\alpha_2^2/\gamma + b_4\alpha_{43}\alpha_3^2/\gamma - b_4\alpha_{43}\beta_{32}\alpha_2^2/\gamma^2 = 1/3.$$

Multiplying by γ, considering $b_4 = \gamma$ and $b_3\alpha_{32}\alpha_2^2 + b_4\alpha_{42}\alpha_2^2 = \frac{1}{6}\alpha_2 - b_4\alpha_{43}\alpha_3\alpha_2$ resulting from condition (5) after multiplying by α_2, condition (9) reads

$$\tfrac{1}{6}\alpha_2 - b_4\alpha_{43}\alpha_3\alpha_2 + b_4\alpha_{43}\alpha_3^2 = \tfrac{1}{3}\gamma + \alpha_{43}\beta_{32}\alpha_2^2.$$

Using $b_4 = \gamma$ and $\alpha_2 = 2\gamma$ that results from (38b) together with summation of (5) and (6) including (2), above equation reads after division by γ and $\alpha_{43} \neq 0$

$$-2\gamma\alpha_3 + \alpha_3^2 = 4\gamma\beta_{32}.$$

Replacing the right hand side by $4\gamma\beta_{32} = (\frac{1}{3} - 2\gamma + 2\gamma^2)/b_3$ that results from two times the sum of (5) and (6) including (2) and $\alpha_2 = 2\gamma$ we get

$$-2\gamma b_3\alpha_3 + b_3\alpha_3^2 = \tfrac{1}{3} - 2\gamma + 2\gamma^2.$$

Replacing $b_3\alpha_3^2$ given on left hand side by $b_3\alpha_3^2 = \frac{1}{3} - \gamma - b_2\alpha_2^2$ that results from condition (4), reformulating after division by $\alpha_2 = 2\gamma$ finally yields

$$b_2\alpha_2 + b_3\alpha_3 = \tfrac{1}{2} - \gamma$$

that is condition (2) in the stiffly accurate case regarding four internal stages. Hence, (9) can be completely described by (1)-(8) and (38b), (40b). □

Table 4: Properties for different sets of coefficients.

Set	p	s	W	Ro	Re/Ro/St	Sc	Stiffly Acc.	$R(\infty)$	Ref.
ROS34PRW	3	4	•	•	•	•	Yes	0	[10]
ROS34PW2	3	4	•	•	•	•	Yes	0	[9]
ROS34PW1a	3	4	•	•		•	No	0	[9]
ROS34PW1b	3	4	•	•		•	No	0	[9]
RKF4DA	3	6			•		No	0	[12]
RODASP	4	6		•		•	Yes	0	[16]

W: Conditions by Steihaug/Wolfbrandt [15], Ro: Conditions by Roche [13], Re/Ro/St: Conditions by Rentrop/Roche/Steinebach [12],
Sc: Conditions by Scholz [14] (see Remark 11)

5. Numerical Results

We investigate orders of convergence by comparing sets of coefficients listed in Table 4. For this purpose, we apply an artificial DAE test problem given by

$$y_1' = -\frac{z_1^3}{y_3^2}\left(3(y_2 - y_1 + y_3^{-1} - \frac{z_1}{10})^2 + \frac{1}{5}(y_2 - y_1 + y_3^{-1} - \frac{z_1}{10})\right) - y_4 \tag{7a}$$

$$y_2' = \frac{1}{10}z_1 - y_4 \tag{7b}$$

$$y_3' = z_1^3\left(3(y_2 - y_1 + y_3^{-1} - \frac{z_1}{10})^2 + \frac{1}{5}(y_2 - y_1 + y_3^{-1} - \frac{z_1}{10})\right) \tag{7c}$$

$$y_4' = y_1 - y_3^{-1} \tag{7d}$$

$$0 = (y_1 - y_3^{-1})^2 + y_4^2 - \frac{1}{10}z_1 \tag{7e}$$

with consistent initial values $y_1(0) = 2$, $y_2(0) = 2$, $y_3(0) = 1$, $y_4(0) = 0$, $z_1(0) = 10$ and exact solution

$$y_1(x) = (100x^2(10x+1)+1)^{-1} + \cos(x), \quad y_2(x) = 1 + x + \cos(x),$$
$$y_3(x) = 100x^2(10x+1)+1, \quad y_4(x) = \sin(x), \quad z_1(x) = 10.$$

DAE system (7) satisfies the index-1 assumption as $g_z = -1/10$. It will be solved up to $x_{end} = 1.5$ applying four test cases. These tests correspond to computations by generalized ROW-type method (2) using different approximations for Jacobian entries of the differential part A_y and A_z. Each test case is solved applying constant step-sizes $h = 1/(1000 \cdot 2^k)$ with $k = 0, ..., 6$ and analytically determined Jacobians. For determining the order of convergence, we compute global errors in discrete L_2-norm by $err = \|\tilde{y}_{num}(x_{end}) - \tilde{y}_{ana}(x_{end})\|_2$ where $\tilde{y} = (y, z)^T$. Terms $\tilde{y}_{num}$ and $\tilde{y}_{ana}$ denote the numerical and analytical solution, respectively. Order of convergence is then given by $q = \log_2(err_{2h}/err_h)$ [9].

Among the sets of coefficients in Table 4, just ROW34PW2 and ROS34PRW satisfy all conditions required for realizing a generalized ROW-type method (2) up to order three. Hence, we expect ROS34PW2 and ROS34PRW to preserve order for any choice of A_y and A_z given below. In contrast, remaining coefficient sets should cause order reduction except for special choices of A_y and A_z.

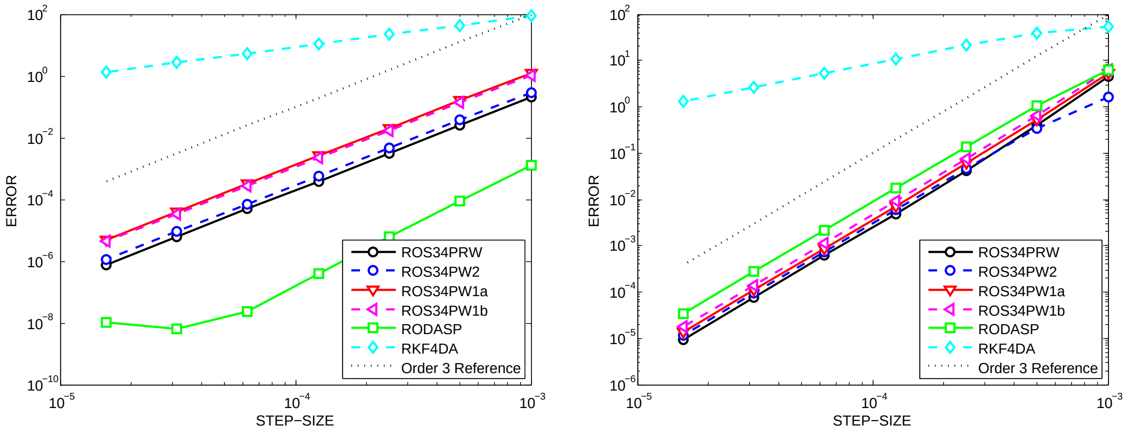


Figure 1: *Results for Test Case 1 (left) and Test Case 2 (right).*

5.1. Test Case 1: Applying Standard ROW Method

We solve DAE system (7) regarding $A_y = f_y$ and $A_z = f_z$, so we compute the full Jacobian with every time-step. Hence, generalized ROW-type method (2) corresponds to a ROW method for DAEs as defined by Roche [13].

Results for the different coefficient sets are shown in Figure 1 (left). They all preserve order of convergence except for RKF4DA that reduces order to one. Order reduction of RKF4DA is due to its assignment solving differential equations of a DAE system by its underlying RK method exclusively. Thus, it does not satisfy all conditions required to solve the differential part implicitly.

5.2. Test Case 2: Applying Constant Jacobian Regarding Differential Part

We choose $A_y = f_y + \mathcal{O}(h)$ and $A_z = f_z + \mathcal{O}(h)$ by keeping Jacobian entries of the differential part constant for several time-steps. This approach corresponds to an usual strategy of W methods for ODEs [4, 18], however, now regarding a first attempt to the DAE case. The full Jacobian of (7) is updated just with every tenth step. For all other steps updates are reduced to algebraic part (7e).

Results are given in Figure 1 (right). Except for RODASP and RKF4DA all sets of coefficients preserve order. Order of RODASP reduces to three and order of RKF4DA reduces to one. ROS34PW1a and ROS34PW1b preserve order three although they do not satisfy condition (9) that is required to realize generalized ROW-type method (2). This is due to the given special choice of Jacobian approximation. Choosing $A_z = f_z + \mathcal{O}(h)$ conditions (9) and (10) combine to satisfied condition (4). Condition (10) itself switches to higher order.

5.3. Test Case 3: Applying Explicit Integration of Differential Part

We compute DAE system (7) with $A_y = 0$ and $A_z = 0$, that is all Jacobian entries of the differential part are neglected. Remaining components are updated with every time-step. Hence, the strategy by Rentrop, Roche, Steinebach [12] is realized, using the underlying explicit RK method to solve differential equations while applying the implicit ROW method to compute algebraic constraints.

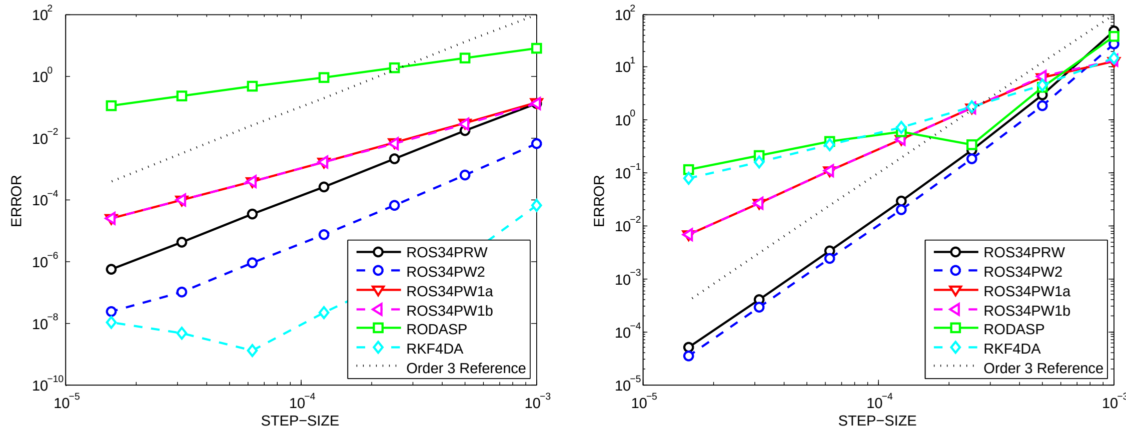


Figure 2: *Results for Test Case 3 (left) and Test Case 4 (right).*

Results are given in Figure 2 (left). ROS34PRW, ROS34PW2 and RKF4DA reach full order. The order for remaining coefficient sets drops down: Order of ROS34PW1a and ROS34PW1b reduces to two, order of RODASP reduces to one. ROS34PW1a and ROS34PW1b attain higher order than RODASP because they satisfy all W method conditions up to order three. Thus, condition (9) is the first violated. However, RODASP violates condition (2) already.

5.4. Test Case 4: Applying Partial Explicit Integration of Differential Part

DAE system (7) is solved using $A_y = (f_S)_y$ and $A_z = (f_S)_z$. So, Jacobian entries with respect to the differential part are reduced to components assumed to be stiff. Explicit solution can thus be applied partially to single elements of the differential part. Hence, an adapted ROW-AMF scheme for DAEs as presented in [6] is used that allows for additive splitting and partitioning of given differential equations. Regarding (7), we neglect Jacobian entries that result for y_4 in (7a) and (7b). Also, we neglect all entries resulting for (7c). So, additive splitting and partitioning are combined, solving these components explicitly. Remaining elements are solved implicitly using updates with every time-step.

Results are shown in Figure 2 (right). ROS34PRW and ROS34PW2 are the only coefficient sets that preserve full order. ROS34PW1a and ROS34PW1b reduce order to two. RODASP reduces order to three for first steps, afterwards to values smaller than one. RKF4DA reduces order to one.

6. Conclusion and Outlook

We introduced a generalized ROW-type method for index-1 DAEs that allows for arbitrarily approximated Jacobian entries with respect to the differential part. Hence, it is a first attempt of realizing W methods for DAEs. Order conditions were derived using a consistent approach based on Butcher's theory of rooted trees that includes a new type of vertices to express occurring non-exact differentials. Resulting order conditions include known conditions of different schemes covered and introduce new conditions for realizing schemes of higher order. Numerical tests showed the relevance of satisfying all conditions predicted when using arbitrarily approximated Jacobian entries with respect to the differential part of DAE problems, especially when realizing additive splitting and partitioning. Only ROS34PRW and ROS34PW2 that satisfy all conditions for a generalized ROW-type method of order three preserved order for each Jacobian approximation applied. Other sets of coefficient showed order reduction unless special approximations were considered.

Continuative works will focus on trying to realize a set of coefficients for generalized ROW-type methods up to order four. As well, we will try to extend the approach introduced to additional approximations of Jacobian entries given with respect to the algebraic part.

Acknowledgements

This article is dedicated to P. Rentrop who introduced us to the fascinating world of ROW-type methods and DAEs. Also, we thank the members of the Chair of Applied Mathematics / Numerical Analysis at Bergische Universität Wuppertal for all their fruitful comments and discussions. Special thanks go to J. Rang whose coefficients derived in [9, 10] helped a lot motivating and verifying some of our investigations. Finally, the corresponding author is indebted to the Graduate-Institute of the Hochschule Bonn-Rhein-Sieg University of Applied Sciences for financial support by PHD scholarship.

References

[1] P. Deuflhard, E. Hairer, J. Zugck, One-step and Extrapolation Methods for Differential-Algebraic Systems, Numer. Math. 51 (1987) 501-516.

[2] S. González-Pinto, Some families of W-methods to stabilize standard explicit Runge-Kutta methods for stiff problems, to be published in Proc. of NUMDIFF 2015.

[3] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations. I: Nonstiff Problems, in: Springer Series in Computational Mathematics, vol. 8, 2. rev. ed., Springer-Verlag, Berlin, 2008.

[4] E. Hairer, G. Wanner, Solving Ordinary Differential Equations. II: Stiff and Differential-Algebraic Problems, in: Springer Series in Computational Mathematics, vol. 14, 2. rev. ed., Springer-Verlag, Berlin, 2002.

[5] W. Hundsdorfer, J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, in: Springer Series in Computational Mathematics, vol. 33, Springer-Verlag, Berlin, 2003.

[6] T. Jax, G. Steinebach, ROW Methods Adapted to Network Simulation, to be published in Proc. of ECMI 2014.

[7] P. Kaps, A. Ostermann, Rosenbrock Methods using few LU-Decompositions, IMA J. Numer. Anal. 9 (1989) 15-27.

[8] A. Ostermann, M. Roche, Rosenbrock Methods for Partial Differential Equations and Fractional Orders of Convergence, SIAM J. Numer. Anal. 30 (1993) 1084-1098.

[9] J. Rang, L. Angermann, New Rosenbrock W-Methods of Order 3 for Partial Differential Algebraic Equations of Index 1, BIT 45 (2005) 761-787.

[10] J. Rang, An analysis of the Prothero-Robinson example for constructing new DIRK and ROW methods, J. Comput. Appl. Math. 262 (2014) 105-114.

[11] P. Rentrop, Partitioned Runge-Kutta Methods with Stiffness Detection and Stepsize Control, Numer. Math. 47 (1985) 545-564.

[12] P. Rentrop, M. Roche, G. Steinebach, The Application of Rosenbrock-Wanner Type Methods with Stepsize Control in Differential-Algebraic Equations, Numer. Math. 55 (1989) 545-563.

[13] M. Roche, Rosenbrock Methods for Differential Algebraic Equations, Numer. Math. 52 (1988) 45-63.

[14] S. Scholz, Order Barriers for the B-Convergence of ROW Methods, Computing 41 (1989) 219-235.

[15] T. Steihaug, A. Wolfbrandt, An Attempt to Avoid Exact Jacobian and Nonlinear Equations in the Numerical Solution of Stiff Differential Equations, Math. Comp. 33 (1979) 521-534.

[16] G. Steinebach, Order-reduction of ROW-methods for DAEs and Method of Lines Applications, Preprint 1741, Technische Universität Darmstadt, Darmstadt, 1995

[17] K. Strehmel, R. Weiner, I. Dannehl, On Error Behaviour of Partitioned Linearly Implicit Runge-Kutta Methods for Stiff and Differential Algebraic Systems, BIT 30 (1990) 358-375.

[18] R. Weiner, K. Strehmel, Linear-implizite Runge-Kutta Methoden und ihre Anwendung, in: Teubner-Texte zur Mathematik, vol. 127, Springer Fachmedien, Wiesbaden, 1992.