Derivation of a New Method for Derivative Estimation by Linear Combinations

Simon Genser

VIRTUAL VEHICLE Research Center
Inffeldgasse 21a, 8010 Graz, Austria
Telephone: +43/316/873-9600
Email: Simon.Genser@v2c2.at

Martin Benedikt

VIRTUAL VEHICLE Research Center
Inffeldgasse 21a, 8010 Graz, Austria
Telephone: +43/316/873-9048
Email: Martin.Benedikt@v2c2.at

***Abstract*—This paper introduces a new method to estimate derivatives of a noiseless time-domain signal, the so called Derivative Estimation by Linear Combinations method. This method can be considered as a generalization of the Finite Difference Method, for the case of backward differentiation. The advantages in computational effort and accuracy of the proposed method compared, to widely known methods of derivation approximation, are demonstrated based on an analytical reference solution. Especially the output-input sensitivity of systems can be determined more robust, with significant fields of application in control, simulation and sensitivity analysis. It is shown that the method is well-suited for real-time applications due to the low computational effort and low time-delay.**

Keywords–derivative estimation; generalized finite difference method; sensitivity analysis; numerical derivation.

I. INTRODUCTION

Many algorithms which are used in control engineering [1], signal processing [2] or numerical simulations need derivatives of differentiable time signals. As computer-implemented algorithms are processing discrete quantities by nature, the computation of derivatives is based on these samples exclusively. Furthermore, for ensuring a proper working of the related and derivative utilizing algorithms, a high approximation of the derivatives is crucial and has a significant impact on the overall performance. Commonly used to compute such derivatives are Finite Difference Methods (FDM) [3], [4], the Complex-Step method [5] or the algebraic method (AM) [6]. Typical drawbacks of those methods are in accuracy, especially for AM, in the computational effort and in time-delay.

The aim of this paper is to introduce a new method to compute derivatives of a discrete signal, using only linear combinations of samples. This proposed method is called Derivative Estimation by Linear Combinations (DELC). The requirements for the newly introduced DELC method are a high accuracy, a low computational effort and a small time delay.

The first target is to compute derivatives of the signal by exclusively using past time samples, which makes the method applicable for real-time applications, as therefore a causal method is strictly mandatory. The second focus on the derivation of the DELC method is that, the computation time is sufficiently low, so it is usable for real-time applications.

The usage of the DELC Method is pointed out by an output-input sensitivity analysis of a control system, so one is intrested in

$$\frac{dy(t)}{du(t)}.$$

From the control system are only the input $u(t)$ and output $y(t)$ known, there is no information about the structure or the dynamics of the system, it is a so called black box.

The key idea of the method is to use linear combinations consisting of the past discrete-time samples y_k

$$y_k^{(j)} = \sum_{i=0}^{N} \hat{c}_{i_j} y_{k-i}, \tag{1}$$

whereby $y_k^{(j)}$ denotes the j-th derivative of the signal $y(t)$ at the actual time instant k, N stands for the number of used samples and $\hat{c}_{i_j}$ represents the appropriate scalars depending on the order of the derivative j. Instead of interpreting (1) as a linear combination of one dimensional vectors, it is possible to identify it as a weighted sum.
To reduce the computational effort it is advantageous, if the scalars $\hat{c}_{i_j}$ are fixed and time invariant. It will be shown that, this is the case if the step size is chosen constant over the whole computation, as it is typically the case in real-time applications. It will also be pointed out that the accuracy depends on the number of samples, which makes it possible to improve accuracy by increasing the number of used samples N. The proposed method is designed for differentiable signals, excluding status signals, discontinuities and events.

The paper is organized as follows: in Section II, the DELC method is derived from two different points of view, first by a Taylor series and then by an approach which is based on the first derivation. It also contains important properties of the method, especially regarding the computational effort, an error estimation and the time delay. The advantages of the DELC method compared to the FDM and the AM and the fulfilment of the requirements will be shown by various examples.

 ISBN: 978-1-61208-594-4

II. DERIVATION OF THE DELC METHOD

This section establishes the method to approximate the derivatives of a differentiable time signal based on discrete samples, i.e., discrete time signal. The following notation is used:
y_k stands for the evaluation of a function $y(t) : \mathbb{R} \to \mathbb{R}$ at the sample time t_k. For simplicity, we consider only constant step size h, this means

$$t_{k+1} = t_k + h. \tag{2}$$

The general statement of the problem, in detail, is that

$$\boldsymbol{y} = \{y_r : r = 1, \ldots, k\}$$

is given, thereby stands k for the index of t_k, so data points at sample times $t_r \leq t_k$ are known. The aim is to compute the j-th derivatives of y at t_k:

$$y_k^{(j)} \quad \forall j = 1, \ldots, N.$$

This states the fact, that the method is causal.

A. Derivation from Taylor-Series

The origin of the derivation is the Taylor series expansion of the function $y(t)$

$$y(t) \approx \sum_{i=0}^{N} \frac{y^{(i)}(t_0)}{i!}(t - t_0)^i, \tag{3}$$

where N denotes the order of the approximation and t_0 the center of the series. It should be mentioned that the order of the approximation N is equivalent to the highest computable derivative.
Note: The idea is to set $t_0 = t_k$ and evaluate the series expansion (3) at $t = t_{k-i}$ for $i = 1, \ldots, N$. This leads to

$$\begin{aligned} y_{k-1} &= \sum_{i=0}^{N} \frac{y_k^{(i)}}{i!}(t_{k-1} - t_k)^i, \\ y_{k-2} &= \sum_{i=0}^{N} \frac{y_k^{(i)}}{i!}(t_{k-2} - t_k)^i, \\ &\vdots \\ y_{k-N} &= \sum_{i=0}^{N} \frac{y_k^{(i)}}{i!}(t_{k-N} - t_k)^i. \end{aligned}$$

Considering, the assumption that the step size h is constant, which is outlined in (2), it follows

$$y_{k-l} = \sum_{i=0}^{N} \frac{y_k^{(i)}}{i!}(-lh)^i \quad \forall l = 1, \ldots, N,$$

leading to

$$y_{k-l} - y_k = \sum_{i=1}^{N} \frac{y_k^{(i)}}{i!}(-lh)^i \quad \forall l = 1, \ldots, N.$$

After rewriting, in matrix vector representation, it results in

$$\hat{A}\boldsymbol{y} = \boldsymbol{b}, \tag{4}$$

whereby

$$\begin{aligned} \hat{A}[l, i] &:= (-1)^i \frac{(lh)^i}{i!}, \\ \boldsymbol{y}[i] &:= y_k^{(i)}, \\ \boldsymbol{b}[l] &:= y_{k-l} - y_k \end{aligned}$$

for $l, i = 0, \ldots, N$. This linear system of equations should be solved symbolically, because the formular should be able to handle arbitrary but constant step size h.

For $N = 2$ the computation of (4) results in the following calculation formulas, for the first and second derivative:

$$y_k^{(1)} = \frac{1}{h}\left[1.5y_k - 2y_{k-1} + 0.5y_{k-2}\right] \tag{5}$$

$$y_k^{(2)} = \frac{1}{h^2}\left[1y_k - 2y_{k-1} + 1y_{k-2}\right] \tag{6}$$

B. Derivation from an approach

The formulas (5) and (6) suggest the approach

$$y_k^{(j)} = \frac{1}{h^j} \sum_{i=0}^{N} c_{i_j} y_{k-i}, \tag{7}$$

where $c_{i_j} \in \mathbb{R}$ are constants which have to be determined. The key idea is, to demand that the derivatives has to be exact for

$$p(t) = \sum_{i=0}^{N+1-j} a_i t^i,$$

whereby $a_i \in \mathbb{R}$. So p stands for polynomials up to degree $N + 1 - j$. The j-th derivative of p at t_k is

$$p_k^{(j)} = \sum_{i=j}^{N+1-j} a_i \frac{i!}{(i-j)!} t_k^{i-j}.$$

Using the approach from (7) with respect to (2) leads to

$$\sum_{i=j}^{N+1-j} a_i \frac{i!}{(i-j)!}(kh)^{i-j} = \frac{1}{h^j} \sum_{l=0}^{N} c_{l_j} \sum_{i=0}^{N+1-j} a_i((k-l)h)^i$$

Using the method of equating the coefficients for h^i leads to

$$a_i \frac{i!}{(i-j)!} k^{i-j} = \sum_{l=0}^{N} c_{l_j} a_i (k-l)^i \quad \forall i = 0, \ldots, N+1-j,$$

under the assumption that,

$$\frac{i!}{(i-j)!} = 0 \quad \text{for } i - j < 0.$$

In matrix and vector representation, the system of equations results in

$$A\boldsymbol{c} = \boldsymbol{b} \tag{8}$$

whereby

$$\begin{aligned} A[i, l] &:= (k - l)^i, \\ \boldsymbol{c}[l] &:= c_{l_j}, \\ \boldsymbol{b}[i] &:= \frac{i!}{(i-j)!} k^{i-j} \end{aligned} \tag{9}$$

 ISBN: 978-1-61208-594-4

for $i, l = 0, \ldots, N$ and for $\frac{i!}{(i-j)!} = 0$ for $i - j < 0$. It should be mentioned that $k \in \mathbb{N}$ is arbitrary and it is possible to set

$$k = 0,$$

because the solution of the system is independent of the choice of k.

To calculate the constants for the linear combination via the system resulting from the derivation with the approach is easier because, the system in (8) is independent from h and so can be solved numerically more efficient, instead of the system (4). But the derivation due to the Taylor-series is fundamental to get the correct approach (7), whereas both ways are important.

C. Properties of the DELC method

The following section discusses important properties of the DELC method, including implementation aspects.

1) Computational Effort: For the computation of the derivative of a signal

$$y_k^{(j)} = \frac{1}{h^j} \sum_{i=0}^{N} c_{i_j} y_{k-i}, \tag{10}$$

has to be evaluated, which is a linear combination of known samples. For the taken assumption that the step size h is arbitrary but constant, the c_{i_j} are constant for the whole computation, so the determination of the c_{i_j} has not to be done in every time step. It is convenient to store them in a map and use the appropriate constants in the algorithm. It does not matter which system (4) or (8) is solved, the solution is equal, it might only make a difference for numerical issues. The remaining computation in every time step is represented by (10), which needs $(N+1)+1$ multiplications and $N+1$ additions. The calculation is non-iterative and therefore the method is applicable for real-time applications.

2) Generalization of the FDM: When choosing $j = 1, N = 1$ and have a close look on (7), or just compute the constants, it follows

$$c_0 = 1,$$
$$c_1 = -1$$

and (7) turns into

$$y_k^{(1)} = \frac{y_k - y_{k-1}}{h},$$

which matches exactly with backward difference quotient. Therefore the DELC method can be interpreted as an generalization of the FDM.

3) Error Estimation: An error estimation of the method can be derived from the Taylor-series expansion (3) of y, whereby y needs to be sufficiently smooth. It holds

$$y(t) = \sum_{i=0}^{\infty} \frac{y^{(i)}(t_0)}{i!}(t-t_0)^i \tag{11}$$

$$\tilde{y}_N(t) = \sum_{i=0}^{N} \frac{y^{(i)}(t_0)}{i!}(t-t_0)^i,$$

whereby $\tilde{y}_N$ is an approximation of y from degree N, leading to

$$|y - \tilde{y}_N| = \left| \sum_{i=N+1}^{\infty} \frac{y^{(i)}(t_0)}{i!}(t-t_0)^i \right|.$$

As outlined in the derivation of the method, the center of the series t_0 is set to t_k and y is evaluated in the discrete points $t = t_{k-i}$ for $i = 1, \ldots, N$. For simplicity, the assumption of constant step size h is here also applied. This results in

$$|y - \tilde{y}_N| \leq ch^{N+1}$$

where $c > 0$ represents an appropriate constant. This error estimation, which is only valid for y and not its derivatives, has to be generalized for derivatives up to degree N. From (11) follows

$$y^{(j)} = \sum_{i=j}^{\infty} \tilde{c}_i y^{(i)}(t_0)(t-t_0)^{i-j}$$

$$\tilde{y}_N^{(j)} = \sum_{i=j}^{N} \tilde{c}_i y^{(i)}(t_0)(t-t_0)^{i-j},$$

for appropriate $\tilde{c}_i$ and $j = 1, \ldots, N$. Again $t_0 = t_k$ and y is evaluated in $t = t_{k-i}$ for $i = 1, \ldots, N$ and this leads to

$$\left| y^{(j)} - \tilde{y}_N^{(j)} \right| = \left| \sum_{i=N+1}^{\infty} \tilde{c}_i y_k^{(i)} h^{i-j} \right|.$$

That results in the error estimation

$$\left| y^{(j)} - \tilde{y}_N^{(j)} \right| \leq ch^{N-j+1}, \tag{12}$$

for $c \in \mathbb{R}^+$, $j = 0, \ldots, N$ and $N \in \mathbb{N}$.

4) Preconditioning: It should be mentioned that the condition-number of the matrix (9) depends on the size of the matrix N, and for high order approximation ($N \geq 11$), a preconditioning strategy should be implemented because the matrix nearly gets singular, for more details on preconditioning see [7]–[10].

5) Time-Delay: Typically methods like the FDM or the AM underlie the problem of the so called time-delay t_{delay}, as the phase shift in Figure 1 shows. Especially for control systems applications time-delays represents significant limitations of performance.

To analyze the introduced time-delay of the different methods, the following example is chosen

$$y(t) = sin(t),$$

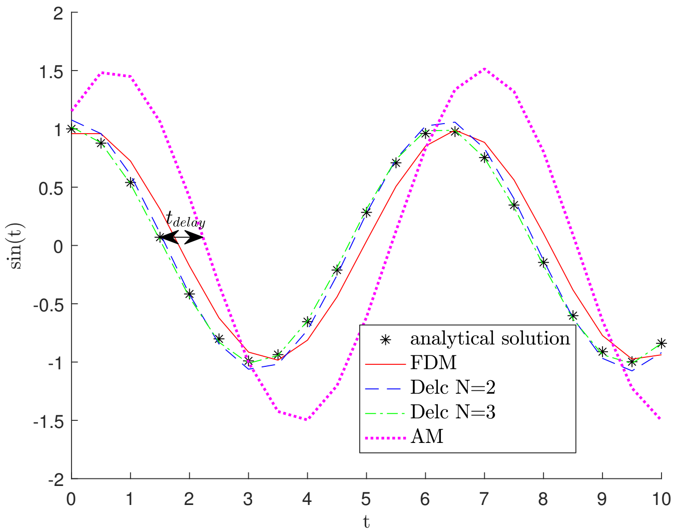

Figure 1. Time delay of the derivative estimation of $y(t) = sin(t)$ by the DELC method, the FDM and AM, parameters for the AM are $N = 2, v = 1, M = 10$, notation based on [6], the chosen step size is $h = 0.5$.

TABLE I. TIME DELAY OF THE DERIVATIVE ESTIMATION OF $y(t) = sin(t)$ BY THE DELC METHOD FOR AN DEGREE OF APPROXIMATION OF $N = 2, \ldots, 16$ AND A CHOSEN STEP SIZE $h = 0.5$.

N	t_{delay}	N	t_{delay}	N	t_{delay}
2	0.0280198	7	-0.0000147	12	-0.0000058
3	-0.0250456	8	-0.0003928	13	-0.0000076
4	-0.0090735	9	-0.0000819	14	-0.0000005
5	0.0024091	10	0.0000607	15	0.0000015
6	0.0020479	11	0.0000301	16	0.0000005

sampled with a step size $h = 0.5$. For the FDM and the AM the following time delay is determined

$$t_{delay,AM} = 0.7030257,$$
$$t_{delay,FDM} = 0.2467885.$$

Table I shows, that the time delay of the DELC method gets quick smaller by increasing the degree of the approximation N. So this feature of the DELC method represents a significant benefit compared to the FDM or the AM.

III. EXAMPLES

In this section, the advantage of the DELC method compared to the FDM and the AM is demonstrated. Error measures are defined as

$$e = \frac{1}{M} \sum_{i=1}^{M} |\boldsymbol{y}[i] - \tilde{\boldsymbol{y}}[i]|,$$

where $|a|$ stands for the absolute value of $a \in \mathbb{R}$ and M denotes the size of the vector of the analytical solution $\boldsymbol{y}$ and of the vector of the approximation $\tilde{\boldsymbol{y}}$.

A. Differentiable Time Signal

For analysis purpose a time-domain signal, a differentiable function, is defined as follows

$$y(t) := sin(t)exp(t)$$

TABLE II. ERRORS AND NUMBER OF OPERATIONS OF THE DELC METHOD, FOR A DIFFERENTIABLE SIGNAL, FOR AN APPROXIMATION ORDER $N = 1, \ldots, 12$ AND FOR THE STEP SIZE $h = 0.1$.

N	e	# operations
1	$4.2703 \cdot 10^{-1}$	5
2	$4.6332 \cdot 10^{-2}$	7
3	$3.7491 \cdot 10^{-3}$	9
4	$2.7147 \cdot 10^{-4}$	11
5	$4.3495 \cdot 10^{-5}$	13
6	$6.5061 \cdot 10^{-6}$	15
7	$6.8851 \cdot 10^{-7}$	17
8	$5.0015 \cdot 10^{-8}$	19
9	$7.7212 \cdot 10^{-9}$	21
10	$1.3139 \cdot 10^{-9}$	23
11	$1.6258 \cdot 10^{-10}$	25
12	$1.4103 \cdot 10^{-11}$	27

and discretized using a constant step size h. The theoretical results are proved by comparison to the analytical derivative

$$y'(t) = (cos(t) + sin(t))exp(t).$$

The first example approximates the first derivative of $y(t)$, compares the different methods and points out the performance of the DELC method by varying the order of the approximation, i.e., increasing the number of used samples (7). For this example the following parameters are chosen

$$t \in [0, 3], \quad h = 0.1.$$

For the FDM or the AM the errors are

$$e_{FDM} = 0.4270308, \quad e_{AM} = 0.4270573,$$

the parameters for the AM are $N = 2, v = 1, M = 10$ whereby the notation is based on [6]. In Table II the errors from the DELC method are given, there the error estimation (12) is clearly valid. It is worthy to point out that the errors of the introduced DELC method are much smaller, for $N > 1$. For $N = 1$ the error coincides with e_{FDM}, this shows the mentioned fact of section II.C.2. It is also obvious that the computational effort, also for high order approximations, is low, so the computation of derivatives can be made nearly instantly. In Figure 2, the errors depending on the order of the approximation N are depicted.

To check the error estimation (12) for decreasing step size and different orders of approximations, the same function and parameters are chosen. In Table III, the order of convergence is shown, if the absolute error gets small, about 10^{-12}, the convergence stops, that comes from numerical errors - this can be neglected. Table III shows that (12) is satisfied, concluding with a match of theory and practice.

To verify the error estimation (12) the function and the parameters are the same as above but the step size and the degree of the derivative is varied. For this purpose the order of the approximation with $N = 6$ is fixed. Table IV shows that the error estimation is satisfied, also for high order derivatives.

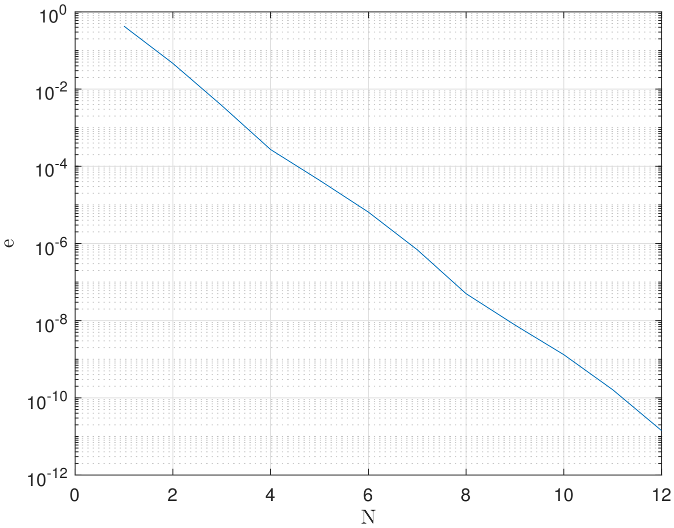

Figure 2. Plot of the errors from Table II, depending on the approximation order N.

TABLE III. ORDER OF CONVERGENCE OF THE DELC METHOD, FOR A DIFFERENTIABLE SIGNAL, FOR THE FIRST DERIVATIVE AND AN APPROXIMATION ORDER $N = 1, \ldots, 6$.

h	N=1	N=2	N=3	N=4	N=5	N=6
1/2	0.93	1.65	2.41	3.54	3.80	3.36
1/4	1.00	1.82	2.75	3.63	3.99	4.83
1/8	1.00	1.90	2.89	3.71	4.50	5.49
1/16	1.00	1.95	2.95	3.85	4.75	5.76
1/32	1.00	1.98	2.98	3.92	4.88	5.89
1/64	1.00	1.99	2.99	3.96	4.94	5.95
1/128	1.00	2.00	2.99	3.98	4.97	5.93
1/256	1.00	2.00	3.00	3.99	4.94	1.03
1/512	1.00	2.00	3.00	4.00	1.98	-0.95

TABLE IV. ORDER OF CONVERGENCE OF DELC METHOD, FOR A DIFFERENTIABLE SIGNAL, FOR AN APPROXIMATION ORDER 6 AND FOR AN ORDER OF DERIVATIVE $j = 1, \ldots, 6$.

h	j=1	j=2	j=3	j=4	j=5	j=6
1/2	3.36	2.55	1.80	1.13	0.50	0.26
1/4	4.82	3.88	2.95	2.06	1.22	0.51
1/8	5.49	4.50	3.52	2.56	1.62	0.75
1/16	5.76	4.77	3.78	2.79	1.82	0.87
1/32	5.89	4.89	3.89	2.90	1.91	0.94
1/64	5.95	4.95	3.95	2.95	1.96	0.97
1/128	5.93	5.10	3.87	2.95	1.99	0.99

B. Black Box System

The task is to compute the output-input sensitivity of a black box system. Using $u(t)$ the input and $y(u(t))$ the output of the system, the chain role leads to

$$\frac{dy}{dt} = \frac{dy}{du}\frac{du}{dt}$$

which is equivalent to

$$\frac{dy}{du} = \frac{dy}{dt} / \frac{du}{dt}.$$

By computing dy/dt and du/dt the output-input sensitivity dy/du can be determined. This has to be performed by using the discrete signals of u and y exclusively.
To simulate the black box the following linear, time invariant system is used

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + \boldsymbol{b}u$$
$$\boldsymbol{y} = \boldsymbol{c}^T\boldsymbol{x}$$

whereby $u(t) := cos(t)$ and

$$A = \begin{bmatrix} -0.1 & 0 \\ 1 & -1 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{c} = \begin{bmatrix} 2 \\ 1 \end{bmatrix},$$
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \boldsymbol{x}(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Note: The simple linear system is chosen for the derivation of the analytical solution for comparison purpose. For the calculation of dy/dt and du/dt it does not matter if the system is linear, non-linear or high-dimensional in general, unless the signal is differentiable, the computation exclusively depends on the discrete input and output signals.

To compute dy/du three different methods are applied, the FDM, the AM and the DELC. The chosen parameters of the simulation are

$$t \in [5, 10], \quad h = 0.1,$$

leading to following errors

$$e_{FDM} = 10.039617, \quad e_{AM} = 15.289524.$$

The errors of the DELC method are stated in Table V. As pointed out in Figure 3, there are two singularities in the derivative, especially in their vicinity the solution from the FDM and the AM is poor. To prove that, this is not the only reason why the DELC method is outperforming, the time interval for a second simulation is changed to

$$t \in [7, 9],$$

the solutions in this case are depicted in Figure 4. For this the errors are

$$e_{FDM} = 0.2657728 \quad e_{AM} = 0.4568813.$$

So the errors, in case of no singularity, are much smaller, but still huge compared to the DELC method. Also the errors of the DELC decreases a lot, comparing to the cases with and without singularities, see Table V.

IV. CONCLUSION AND FUTURE WORK

A. Conclusion

The DELC method has some major advantages in accuracy, computational effort and time-delay compared to the FDM or the AM, the examples show that the approximation works well. Especially the low computational effort, the high accuracy and the small time delay, makes the introduced DELC method attractive and well suited for control systems and real-time applications.

 ISBN: 978-1-61208-594-4

TABLE V. ERRORS OF THE DELC METHOD, FOR THE BLACK BOX EXAMPLE, FOR AN APPROXIMATION ORDER $N = 1, \dots, 12$, FOR DIFFERENT TIME INTERVALS AND FOR STEP SIZE $h = 0.1$.

N	e for $t \in [5, 10]$	e for $t \in [7, 9]$
2	$8.7236 \cdot 10^{-2}$	$1.3019 \cdot 10^{-3}$
3	$8.5957 \cdot 10^{-2}$	$1.3543 \cdot 10^{-3}$
4	$1.1549 \cdot 10^{-3}$	$1.7732 \cdot 10^{-5}$
5	$5.6694 \cdot 10^{-4}$	$8.8775 \cdot 10^{-6}$
6	$1.2895 \cdot 10^{-5}$	$1.9959 \cdot 10^{-7}$
7	$4.1388 \cdot 10^{-6}$	$6.4848 \cdot 10^{-8}$
8	$1.3604 \cdot 10^{-7}$	$2.1108 \cdot 10^{-9}$
9	$3.1904 \cdot 10^{-8}$	$4.9962 \cdot 10^{-10}$
10	$1.4402 \cdot 10^{-9}$	$2.1866 \cdot 10^{-11}$
11	$1.5474 \cdot 10^{-10}$	$3.7318 \cdot 10^{-12}$
12	$1.8636 \cdot 10^{-10}$	$2.9004 \cdot 10^{-12}$

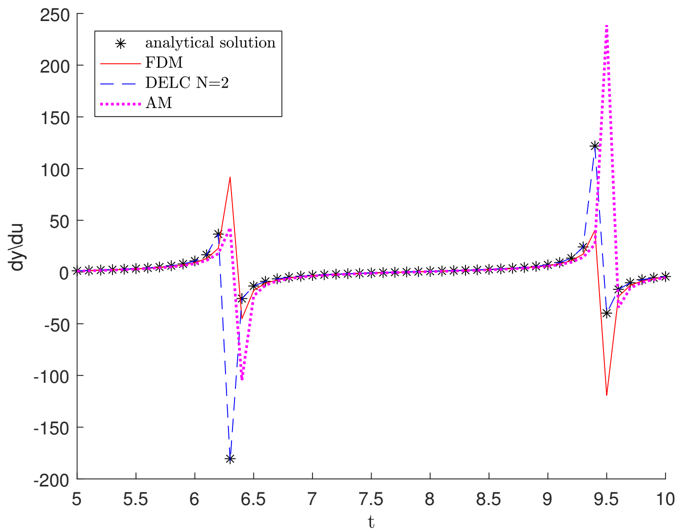

Figure 3. dy/du for $t \in [5, 10]$, with two singularities at $t_1 = 6.3$ and $t_2 = 9.4$, parameters for the AM are $N = 2, v = 1, M = 20$, notation based on [6].

B. Future Work

Future work will focus on implementation aspects like:

1) adaptive step size: The assumption that the step size is constant might represent a significant restriction for industrial applications, so the method has to be generalized for completely arbitrary step size. Only then the method will be completely applicable for simulations in the industrial context, where typically the step size is chosen in an adaptive manner.

2) noise handling: The DELC method presented in this paper is developed and tested for noiseless signals, which is fine e.g., simulations. In case of noisy signals the proposed method has to be improved due to the fact that polynomial interpolation, see Section II.B, is not useful for noisy signals.

3) solvability of the matrix equation: From a theoretical point of view, the solvability of the matrix equations should be proofed. It would also be interesting to verify if the coefficients are the optimal choice and how big the influence of round off errors for the computation is.

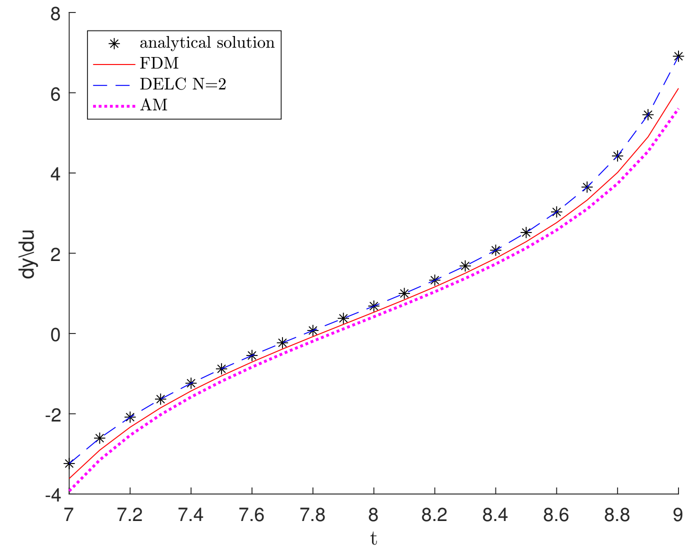

Figure 4. dy/du for $t \in [7, 9]$, without a singularity, parameters for the AM are $N = 2, v = 1, M = 20$, notation based on [6].

ACKNOWLEDGMENT

This work was accomplished at the VIRTUAL VEHICLE Research Center in Graz, Austria. The authors would like to acknowledge the financial support of the COMET K2 - Competence Centers for Excellent Technologies Programme of the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit), the Austrian Federal Ministry of Science, Research and Economy (bmwfw), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG).

REFERENCES

[1] S. P. Banks, Ed., *Control systems engineering: modelling and simulation, control theory and microprocessor implementation*. Prentice-Hall, 1986.

[2] P. Sircar, Ed., *Mathematical aspects of signal processing*. Cambridge University Press, 2016.

[3] A. Kaw and E. Kalu, “A stable finite difference ansatz for higher order differentation of non exact data,” *Bull.Austral. Math. Soc.*, vol. 58, pp. 223–232, 1998.

[4] J. W. Thomas, Ed., *Finite difference methods*. Springer, 1995.

[5] D. De Pauw and P. Vanrolleghem, “Using the complex-step derivative approximation method to calculate local sensitivity functions of highly nonlinear bioprocess models.”

[6] J. Zehetner, J. Reger, and M. Horn, “Realtime implementation of an algebraic derivative estimation scheme,” *Oldenbourg Wissenschaftsverlag*, pp. 553–560, 11/2007.

[7] A. Owe, Ed., *Iterative Solution Methods*. Cambridge University Press, 1994.

[8] M. Benzi, “Preconditioning techniques for large linear systems: A survey,” *Mathematics and Computer Science Department, Emory University, Atlanta, Georgia*, 2002.

[9] A. J. Wathen, “Preconditioning,” *Cambridge University Press*, vol. 48, pp. 329 – 376, 2015.

[10] K. Chen, Ed., *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, 2005, vol. 1.