

Parareal for index two differential algebraic equations

Citation for published version (APA):

Cortes Garcia, I., Kulchytska-Ruchka, I., & Schöps, S. (2022). Parareal for index two differential algebraic equations. *Numerical Algorithms*, 91, 389-412. <https://doi.org/10.1007/s11075-022-01267-1>

Document license:

CC BY

DOI:

[10.1007/s11075-022-01267-1](https://doi.org/10.1007/s11075-022-01267-1)

Document status and date:

Published: 01/09/2022

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.



Parareal for index two differential algebraic equations

Idoia Cortes Garcia¹ · Iryna Kulchytska-Ruchka² · Sebastian Schöps²

Received: 24 February 2021 / Accepted: 24 January 2022 / Published online: 11 March 2022
© The Author(s) 2022

Abstract

This article proposes modifications of the Parareal algorithm for its application to higher index differential algebraic equations (DAEs). It is based on the idea of applying the algorithm to only the differential components of the equation and the computation of corresponding consistent initial conditions later on. For differential algebraic equations with a special structure as, e.g. given in flux-charge modified nodal analysis, it is shown that the usage of the implicit Euler method as a time integrator suffices for the Parareal algorithm to converge. Both versions of the Parareal method are applied to numerical examples of nonlinear index 2 differential algebraic equations.

Keywords Parareal · Differential algebraic equations · Index 2 · Modified nodal analysis

1 Introduction

The time-domain simulation of models from physics, finance or social sciences often leads to systems with differential and algebraic equations. These systems can be both time dependent differential equations, such as, e.g. in circuit models for the simulation of microchips and energy networks [28], or space and time dependent partial differential equations (PDEs), as is the case for the simulation of the electromagnetic behaviour of electric machines [25]. In the latter one, typically the method of lines is used. Here, first, a spatial discretisation method is applied to the PDE to obtain an

✉ Idoia Cortes Garcia
i.cortes.garcia@tue.nl

Iryna Kulchytska-Ruchka
iryna.kulchytska-ruchka@tu-darmstadt.de

Sebastian Schöps
sebastian.schoeps@tu-darmstadt.de

¹ Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

² Technische Universität Darmstadt, 64289, Darmstadt, Germany

only time dependent system of differential equations which is then solved in time as an initial value problem (IVP).

The time domain simulation of large systems of equations, e.g. obtained from fine meshes or systems with excitations having fast dynamics, which require small time step sizes, together with large time windows, considerably increase simulation time. In these cases, parallelisation methods allow reducing computation time. When spatial parallelisation through domain decomposition methods is used up to saturation or whenever the time domain dynamics is the bottleneck of the simulation time, parallel-in-time methods [12, 21, 24, 29] can be used. Parareal is such an algorithm that is based on the same idea as multiple shooting methods [12, 21].

Many initial value problems arising from physical systems such as, e.g. electric networks, spatial discretisation of some approximations to Maxwell's equations or constrained mechanical systems such as the pendulum are systems of differential algebraic equations (DAEs). These are systems that contain both ordinary differential equations (ODEs) and algebraic constraints. They convey analytical and numerical difficulties that do not arise when handling ODEs. This includes their potentially large sensitivity towards small, high frequent perturbations [3, 20] as well as their nontrivial selection of appropriate initial conditions [7, 20]. Their index, a natural number ≥ 0 , gives one way of classifying them according to the difficulties they pose. Especially when using less standard algorithms on DAEs with higher index, such as, e.g. Parareal, a correct handling of the equations is of utmost importance. Otherwise, their convergence may not be ensured, deteriorated, or they may even not be applicable. Low index problems have already been numerically solved by Parareal, e.g. [4]. The relatively straightforward index-1 case was already discussed in [11, 27, 28]. This work focuses on the application of the Parareal algorithm to index 2 DAEs and how they can be handled appropriately. In particular, the fulfilment of the hidden and explicit algebraic constraints and the extraction of the dynamic components is studied in the context of the Parareal algorithm.

The paper is structured as follows: Sections 1.1 and 1.2 introduce the classic Parareal algorithm and the concept of differential algebraic equations, their tractability index as well as the idea of consistent initial conditions for DAEs. A modification of the Parareal algorithm is proposed in Section 2 for its application to index 2 tractable DAEs, and it is applied to a nonlinear toy example. In Section 3, the behaviour of the implicit Euler scheme is shown on DAEs with a specific structure and its significance for the Parareal algorithm is explained. The section ends with the application of the Parareal algorithm to a nonlinear index 2 DAE arising from circuit simulation. The paper concludes in Section 5 with a summary.

1.1 Parareal

In the following section, we will introduce the parallel-in-time method Parareal for ordinary differential equations [21]. Let us consider the initial value problem of an ODE

$$\mathbf{Ax}' + \mathbf{b}(\mathbf{x}, t) = 0 \quad (1)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (2)$$

with constant matrix \mathbf{A} , $\det \mathbf{A} \neq 0$, $\mathbf{x} : \mathcal{I} \rightarrow \mathbb{R}^{n_{\text{dof}}}$ and $\mathbf{x}' = d\mathbf{x}/dt$ on time interval $\mathcal{I} = [t_0, t_{\text{end}}] \subset \mathbb{R}$. To apply the Parareal algorithm, first the time interval \mathcal{I} is partitioned into N smaller time windows $\mathcal{I}_n = (T_{n-1}, T_n]$ of size $\Delta T = (t_{\text{end}} - t_0)/N$, with $T_0 = t_0$ and $T_N = t_{\text{end}}$. In each iteration k , Parareal solves in parallel the N initial value problems

$$\mathbf{A} \frac{d}{dt} \mathbf{x}^{(n,k)} + \mathbf{b}(\mathbf{x}^{(n,k)}, t) = 0, \mathbf{x}^{(n,k)}(T_{n-1}) = \mathbf{X}_{n-1}^k \quad \text{for } t \in \mathcal{I}_n, \quad (3)$$

with $\mathbf{X}_0^k = \mathbf{x}_0$ and $n = 1, \dots, N$. This, however requires initial conditions for each subwindow \mathcal{I}_n , which are a priori unknown. Therefore, the algorithm has to start with incorrect initial conditions at the interface points T_n . This yields jumps in the solution across subwindows, which Parareal tries to iteratively eliminate by updating the initial conditions. Thus, in addition to the parallel computation of the N initial value problems described previously, Parareal performs an update formula for the initial conditions which for the $(k + 1)$ th iteration reads [14, 21]

$$\mathbf{X}_n^{k+1} = \mathcal{F}(T_n, T_{n-1}, \mathbf{X}_{n-1}^k) + \mathcal{G}(T_n, T_{n-1}, \mathbf{X}_{n-1}^{k+1}) - \mathcal{G}(T_n, T_{n-1}, \mathbf{X}_{n-1}^k), \quad (4)$$

for $n = 1, \dots, N$. Here, \mathcal{F} and \mathcal{G} are solution operators of the initial value problem called the fine and coarse propagator, respectively.

The fine propagator $\mathcal{F}(T_n, T_{n-1}, \mathbf{X}_{n-1}^k)$ returns the solution of (3) at time step T_n . It solves the problem in a very accurate way and is thus computationally expensive to apply. However, as for the $(k + 1)$ th iteration this solution only requires the initial condition computed at the previous Parareal iteration \mathbf{X}_{n-1}^k , it is performed in parallel on all subwindows \mathcal{I}_n . Therefore, simulation time is still reduced with respect to a sequential computation.

The second operator, $\mathcal{G}(T_n, T_{n-1}, \star)$ gives the solution of the initial value problem at time point T_n with initial condition \star at T_{n-1} . However, \mathbf{X}_{n-1}^{k+1} is a solution of the current Parareal iteration and thus cannot be computed in parallel. Therefore, the coarse solver has to be applied sequentially. To ensure computation time is reduced as much as possible, this operator has to be cheap to compute and as a consequence less accurate. Here, for example a larger time step size can be employed (e.g. one time step per window) or a reduced system can be solved.

Combining the parallel computation of the expensive and accurate fine propagator with the sequential computation of the cheap coarse propagator yields each Parareal iteration to require less computation time than the sequential simulation approach. This establishes a link to other parallelisation methods such as multiple shooting methods (see [12] for an historical overview) or multigrid approaches [10].

1.2 Differential algebraic equations

In this section, we introduce the concept of differential algebraic equations. For that, we consider initial value problems consisting of quasilinear DAEs of the form

$$\mathbf{A}(\mathbf{x}, t)\mathbf{x}' + \mathbf{b}(\mathbf{x}, t) = 0, \quad (5)$$

on time interval $\mathcal{I} = [t_0, t_{\text{end}}]$ with $\mathbf{x} : \mathcal{I} \rightarrow \mathbb{R}^{n_{\text{dof}}}$, n_{dof} the number of degrees of freedom and initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. Here, $\det \mathbf{A}(\mathbf{x}, t)$ can be zero, which yields

a system of equations containing both differential equations as well as algebraic constraints.

Differential algebraic equations are typically classified according to their index [3, 17, 20], which allows evaluating the analytical and numerical difficulties the system may convey. Higher index systems (index ≥ 2) require a special numerical handling. There are different types of index definitions that essentially coincide for linear systems [23]. In this paper, we use the projector-based tractability index [20] as it allows a separation of the degrees of freedom and equations into the purely differential and the algebraic ones.

Following [5, 7], we consider DAEs with a special structure and therefore take the following assumption.

Assumption 1 *We assume the spaces $\ker \mathbf{A}(\mathbf{x}, t)$ and $\text{Im } \mathbf{A}(\mathbf{x}, t)$ are independent of the degrees of freedom \mathbf{x} and depend smoothly on t .*

Remark 1 This assumption is mild, as many DAEs arising from physical systems fulfil those requirements. Later on, it is shown that, e.g. RLC circuits described with modified nodal analysis have this property (see [7, 9]) or even space discretised partial differential algebraic equations such as, e.g. the eddy current problem [26]. Similar assumptions are taken, e.g. in [19] and ensure that the BDF method integrates index 2 problems well.

Following [7], we consider DAEs fulfilling Assumption 1 and their corresponding projectors $\mathbf{Q}(t)$ onto $\ker \mathbf{A}(\mathbf{x}, t)$ as well as its complementary $\mathbf{P}(t) = \mathbf{I} - \mathbf{Q}(t)$. We introduce the matrices

$$\mathbf{B}(\mathbf{y}, \mathbf{x}, t) := \frac{\partial}{\partial \mathbf{x}} (\mathbf{A}(\mathbf{x}, t)\mathbf{y}) + \frac{\partial}{\partial \mathbf{x}} \mathbf{b}(\mathbf{x}, t) \quad (6)$$

$$\mathbf{A}_1(\mathbf{y}, \mathbf{x}, t) := (\mathbf{A}(\mathbf{x}, t) + \mathbf{B}(\mathbf{y}, \mathbf{x}, t)\mathbf{Q}(t)) (\mathbf{I} - \mathbf{P}(t)\mathbf{P}'(t)\mathbf{Q}(t)), \quad (7)$$

and the projectors $\mathbf{Q}_1(\mathbf{y}, \mathbf{x}, t)$ onto $\ker \mathbf{A}_1(\mathbf{y}, \mathbf{x}, t)$ and $\mathbf{P}_1(\mathbf{y}, \mathbf{x}, t) = \mathbf{I} - \mathbf{Q}_1(\mathbf{y}, \mathbf{x}, t)$. Finally, the matrix

$$\mathbf{G}_2(\mathbf{y}, \mathbf{x}, t) := \mathbf{A}_1(\mathbf{y}, \mathbf{x}, t) + \mathbf{B}(\mathbf{y}, \mathbf{x}, t)\mathbf{P}(t)\mathbf{Q}_1(\mathbf{y}, \mathbf{x}, t) \quad (8)$$

is defined. As we only focus on index 2 systems, we present the tractability index definition accordingly. However, it can be generalised to systems with index > 2 (see [20]).

Definition 1 (Tractability index [20]) A quasilinear DAE (5) fulfilling Assumption 1 with sufficiently smooth solution vector $\mathbf{x} : \mathcal{I} \rightarrow \mathbb{R}^{n_{\text{dof}}}$ has tractability index

- 0, if $\mathbf{A}(\mathbf{x}, t)$ is nonsingular,
- 1, if $\mathbf{A}(\mathbf{x}, t)$ is singular and $\mathbf{A}_1(\mathbf{y}, \mathbf{x}, t)$ is nonsingular,
- 2, if $\mathbf{A}(\mathbf{x}, t)$ and $\mathbf{A}_1(\mathbf{y}, \mathbf{x}, t)$ are singular and $\mathbf{G}_2(\mathbf{y}, \mathbf{x}, t)$ is nonsingular.

Example 1 We present a simple example of a linear DAE system to exemplify the concept of tractability index and the projectors that are involved. For that, we consider the DAE

$$x'_1 = x_1 + x_2 \tag{9}$$

$$0 = x_1 - \sin(t) , \tag{10}$$

with degrees of freedom $\mathbf{x}^\top = (x_1, x_2)$. Note that here the DAE is defined by the matrices

$$\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{b}(\mathbf{x}, t) = \begin{pmatrix} -x_1 - x_2 \\ -x_1 + \sin(t) \end{pmatrix} .$$

Projectors onto $\ker \mathbf{A}$ and respectively along $\ker \mathbf{A}$ are

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} .$$

Here, $\mathbf{Q}\mathbf{x}$ projects onto x_2 and thus the variable that does not appear differentiated in time in the original DAE, whereas $\mathbf{P}\mathbf{x}$ projects onto x_1 , which corresponds to the degree of freedom that is time differentiated in (9). Furthermore,

$$\mathbf{B} = \begin{pmatrix} -1 & -1 \\ -1 & 0 \end{pmatrix} \quad \text{and therefore} \quad \mathbf{A}_1 = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix} .$$

The projectors onto $\ker \mathbf{A}_1$ and respectively along $\ker \mathbf{A}_1$ are

$$\mathbf{Q}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{P}_1 = \begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix} .$$

The DAE (9)–(10) has index 2 as \mathbf{A}_1 is singular and

$$\mathbf{G}_2 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

is nonsingular.

For the rest of the paper, we will choose a specific projector \mathbf{Q}_1 , which fulfils a property that is especially helpful in the setting of the implicit Euler method. This particular choice can be taken without the loss of generality (see [7]).

Assumption 2 (Canonical projector [7]) *We define \mathbf{Q}_1 to be the canonical projector, that is for a projector $\tilde{\mathbf{Q}}_1(\mathbf{y}, \mathbf{x}, t)$ onto $\ker \mathbf{A}_1(\mathbf{y}, \mathbf{x}, t)$, we choose*

$$\mathbf{Q}_1(\mathbf{y}, \mathbf{x}, t) = \tilde{\mathbf{Q}}_1(\mathbf{y}, \mathbf{x}, t)\mathbf{G}_2(\mathbf{y}, \mathbf{x}, t)^{-1}\mathbf{B}(\mathbf{y}, \mathbf{x}, t)\mathbf{P}(t) . \tag{11}$$

Note that here the projector fulfills the following property [7]

$$\mathbf{Q}_1(\mathbf{y}, \mathbf{x}, t)\mathbf{Q}(t) = 0 . \tag{12}$$

Remark 2 The projector \mathbf{Q}_1 of Example 1 is already the canonical projector and fulfills $\mathbf{Q}_1\mathbf{Q} = 0$.

1.3 Consistent initialisation

The aim is to apply the Parareal algorithm to an initial value problem arising from a system of differential algebraic equations. For that, we introduce the concept of consistent initialisation for DAEs.

In contrast to initial value problems arising from ODEs, DAEs cannot be initialised with arbitrary initial conditions \mathbf{x}_0 , as the algebraic constraints imposed by the system have to be fulfilled by the solution at every time point $t \in \mathcal{I}$. In this context, we introduce the concept of consistent initial conditions.

Definition 2 (Consistent initial condition c.f. [3]) Let us consider an initial value problem consisting of the DAE (5) on the time interval $t \in \mathcal{I}$. Then, an initial condition \mathbf{x}_0 at t_0 is called consistent, if there exists a solution $\mathbf{x}^* : \mathcal{I} \rightarrow \mathbb{R}^n$ of (5) fulfilling $\mathbf{x}^*(t_0) = \mathbf{x}_0$.

Remark 3 Higher index DAEs (≥ 2) present hidden constraints that appear only after time differentiation of the original system and thus are not explicitly accessible. This complicates the choice of appropriate initial conditions, as they have to fulfil both the explicit as well as the hidden constraints.

Example 2 To illustrate the concept of consistent initialisation and hidden constraints in high index DAEs, we consider again the linear index 2 system of the previous section (9)–(10):

$$\begin{aligned}x_1' &= x_1 + x_2 \\0 &= x_1 - \sin(t) .\end{aligned}$$

To choose an initial condition at time t_0 , clearly the explicit algebraic constraint in (10) has to be fulfilled and thus $x_1(t_0) = \sin(t_0)$.

Differentiation of the explicit constraint and inserting it into the differential equation leads to a second, hidden, constraint which also needs to be fulfilled by the solution, that is $x_2(t_0) = \cos(t_0) - \sin(t_0)$. In particular, in this example the DAE is a purely algebraic equation with no dynamics, as all degrees of freedom are prescribed by either explicit or hidden constraints.

Example 2 illustrates that initial conditions for DAEs cannot be chosen entirely free. This is bad news for Parareal, since it is based on the idea of iteratively correcting initial values (see (4)).

Following [7, 20, 22], we make usage of the projectors defined for the tractability index to separate the degrees of freedom of the DAE system and extract the purely differential components. Their ICs are freely choosable as they are not characterised by any type of algebraic constraint. Furthermore, when they are prescribed together with the DAE (5), a uniquely solvable initial value problem arises [22]. They can be extracted with

$$\mathbf{x}_{\text{diff}}(t) := \mathbf{PP}_1(\mathbf{y}, \mathbf{x}, t)\mathbf{x}(t)$$

and, if they are fixed, the rest of the components, all of them algebraic, are uniquely determined by the values of \mathbf{x}_{diff} and time t [7]. Note that, for index 2 systems, two projectors are required to extract the differential components. Here a further reduction within the explicitly differentiated $\mathbf{P}\mathbf{x}$ components is made by further applying the $\mathbf{P}_1(\mathbf{y}, \mathbf{x}, t)$ projector.

With a second projector $\mathbf{T}(\mathbf{y}, \mathbf{x}, t)$ onto $\text{Im } \mathbf{Q}(t)\mathbf{Q}_1(\mathbf{y}, \mathbf{x}, t)$, the index 2 variables of the DAE can be extracted. For the differential index, the index 2 variables are the parts of \mathbf{x} that require one time differentiation of the original system to be characterised by (hidden) algebraic constraints [3]. Its complementary projector $\mathbf{U}(\mathbf{y}, \mathbf{x}, t) = \mathbf{I} - \mathbf{T}(\mathbf{y}, \mathbf{x}, t)$ allows to extract the purely differential components together with the index 1 variables, that is the ones prescribed by the explicit constraints of the system. Thus, we can separate the degrees of freedom into three types (see [7])

$$\mathbf{x} = \underbrace{\mathbf{P}\mathbf{P}_1(\mathbf{y}, \mathbf{x}, t)\mathbf{x}}_{\text{index 0}} + \underbrace{\mathbf{P}\mathbf{Q}_1(\mathbf{y}, \mathbf{x}, t)\mathbf{x}}_{\text{index 1}} + \underbrace{\mathbf{Q}\mathbf{U}(\mathbf{y}, \mathbf{x}, t)\mathbf{x}}_{\text{index 2}} + \mathbf{T}(\mathbf{y}, \mathbf{x}, t)\mathbf{x} . \tag{13}$$

Example 3 Again for illustration proposes we consider the index 2 DAE (9)–(10)

$$x_1' = x_1 + x_2 \tag{14}$$

$$0 = x_1 - \sin(t) . \tag{15}$$

As computed in Example 1, the index 2 variables are characterised by $\text{Im}\mathbf{Q}\mathbf{Q}_1$ and

$$\mathbf{Q}\mathbf{Q}_1 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{with projector} \quad \mathbf{T} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} .$$

As $\mathbf{T}\mathbf{x}$ projects onto the index 2 components whose algebraic behaviour is characterised by hidden constraints, we obtain that x_2 is an index 2 variable. This is consistent to the results obtained in Example 2, where x_2 was described by the hidden constraint $x_2 = \cos(t) - \sin(t)$.

Furthermore, even though

$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

(see Example 1) projects onto x_1 as it appears differentiated in time in (14), from Example 2 we know that all variables are algebraic. As

$$\mathbf{P}\mathbf{P}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{P}\mathbf{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} ,$$

the DAE has no differential components and x_1 is an index 1 variable.

The degrees of freedom $\mathbf{x}^\top = (x_1 \ x_2)$ in (9)–(10) can be decomposed as

$$\mathbf{x} = \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{x}}_{\text{index 0}} + \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{x}}_{\text{index 1}} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \mathbf{x}}_{\text{index 2}} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{x}}_{\text{index 2}} . \tag{16}$$

We have summarised the projectors introduced to separate the degrees of freedom of index 2 DAEs in Table 1.

Table 1 Summary of important projectors for the (theoretical) decomposition of the degrees of freedom of index 2 DAEs

P	Projector onto the degrees of freedom that appear explicitly differentiated in time within the original DAE.
Q	Projector onto the degrees of freedom that appear only algebraically within the original DAE.
U	Projector onto the index 0 and index 1 degrees of freedom.
T	Projector onto the index 2 degrees of freedom.
PP₁	Projector onto the purely differential components (index 0) of the DAE.

2 Parareal for DAEs

The application of the Parareal algorithm for nonlinear ordinary differential equations and its convergence is already studied [13]. However, for the case of higher index DAEs, its convergence has not been studied yet and its applicability is not ensured.

Example 4 In this example, we illustrate the consistency problem that arises when applying the Parareal update formula (4) to compute initial conditions for nonlinear DAEs. Let us consider the nonlinear DAE

$$x'_1 = f(\mathbf{x}, t) \quad (17)$$

$$0 = g(\mathbf{x}, t) \quad (18)$$

with $\mathbf{x} = (x_1, x_2)$. If $\partial g/\partial x_2 \neq 0$ then the DAE is index 1. If $\partial g/\partial x_2 = 0$ but the hidden constraint

$$h(\mathbf{x}, t) := \frac{\partial g(\mathbf{x}, t)}{\partial x_1} f(\mathbf{x}, t) = 0 \quad (19)$$

fulfills $\partial h/\partial x_2 \neq 0$ then it is index 2. Now, two consistent initial values \mathbf{x}_c and \mathbf{x}_f at time t_n , e.g. obtained by a coarse and fine propagator, may fulfill the constraints

$$g(\mathbf{x}_c, t_n) = 0 \quad \text{and} \quad g(\mathbf{x}_f, t_n) = 0 \quad (20)$$

or

$$h(\mathbf{x}_c, t_n) = 0 \quad \text{and} \quad h(\mathbf{x}_f, t_n) = 0 \quad (21)$$

separately but their linear combination $\mathbf{x}_c + \mathbf{x}_f$ can in general violate the constraints and is therefore not a consistent initial condition.

As it can be seen in Example 4, especially for nonlinear DAEs, it can happen that the update formula (4) yields an inconsistent initial condition for the fine solver. This may lead to divergence of the algorithm, slower convergence or an incorrect solution.

Parareal has been successfully applied to DAEs previously. For example, in [4], the algorithm is applied to a system of DAEs without a special handling. In [27] it is applied to index 1 DAEs with a special structure and in [11] with a modified Parareal algorithm. Both approaches are special cases and are covered by the theory given in this paper.

First, a modification of the Parareal algorithm is presented here that works for general index 2 DAEs as (5). The method is similar to the one given in [18, 19] for multiple shooting methods. There it is ensured that the algorithm works by means of extending the Jacobian that is computed to update the initial conditions to avoid it being singular. This is achieved by including an equation for the calculation of consistent initial conditions. The idea behind the method here is to only apply the Parareal algorithm on the purely differential components of the DAE and compute, if necessary, the rest of the degrees of freedom accordingly. For that, the update formula (4) is restricted to

$$\hat{\mathbf{X}}_n^{k+1} = \mathbf{PP}_1(\tilde{\mathbf{X}}_n^k, T_n)\tilde{\mathbf{X}}_n^k + \mathbf{PP}_1(\bar{\mathbf{X}}_n^{k+1}, T_n)\bar{\mathbf{X}}_n^{k+1} - \mathbf{PP}_1(\bar{\mathbf{X}}_n^k, T_n)\bar{\mathbf{X}}_n^k, \tag{22}$$

where $\tilde{\mathbf{X}}_n^k := \mathcal{F}(T_n, T_{n-1}, \mathbf{X}_{n-1}^k)$ and $\bar{\mathbf{X}}_n^k := \mathcal{G}(T_n, T_{n-1}, \mathbf{X}_{n-1}^k)$. That is, only the purely differential components (index 0 components) of the DAE are updated within the Parareal algorithm. This corresponds to the freely choosable components of the degrees of freedom that are not constrained by algebraic equations. The resulting solution $\hat{\mathbf{X}}_n^{k+1}$ is then used to compute the consistent initial condition \mathbf{X}_n^{k+1} such that

$$\mathbf{PP}_1(\mathbf{X}_n^{k+1}, T_n) (\mathbf{X}_n^{k+1} - \hat{\mathbf{X}}_n^{k+1}) = 0. \tag{23}$$

That is, a consistent initial condition is computed that has the same differential components as obtained from the Parareal update. This can either be done analytically for simple DAEs or with numerical techniques as proposed, e.g. in [7, 8]. The algorithm of [8] is implemented in a Python package called *InitDAE*¹, which is able to numerically compute consistent initial conditions for DAEs.

In a nutshell, to properly handle the algebraic constraints imposed by DAEs, we propose a modified Parareal algorithm for differential algebraic equations. Here, the Parareal update on each iteration k is only to be performed on the differential components of the degrees of freedom as in (22). The new initial condition at the corresponding time points $t = T_n$ is obtained by means of computing a consistent initial condition with the same differential components than the solution of the modified Parareal update (see (23)).

Remark 4 Note that the modified Parareal algorithm for DAEs proposed within this section requires, for each iteration, the computation of the projectors $\mathbf{P}(t)$ and $\mathbf{P}_1(\mathbf{x}, t)$ explicitly for the modified update (22). Furthermore, for each window \mathcal{I}_n and iteration, consistent initial conditions need to be computed.

This approach can also be performed on DAEs with index higher than two. In that case, the projectors to extract the purely differential components in (22) and (22) have to be adapted accordingly (see [20]).

¹https://www2.mathematik.hu-berlin.de/~lamour/software/python/InitDAE/html/InitDAE_Integration2020_3_7/

2.1 Numerical example

In the following, the modified Parareal algorithm for DAEs from the previous section is applied to an index 2 toy example

$$x'_0 + g(x_2) = 0 \tag{24a}$$

$$x'_1 - x_2 = 0 \tag{24b}$$

$$x_1 - 0.015 \sin(2\pi 10t) = 0, \tag{24c}$$

with degrees of freedom $\mathbf{x}^\top = (x_0, x_1, x_2)$ and nonlinear function

$$g(x) = \begin{cases} 0 & \text{if } x \leq 1, \\ e^{-(x-1)^{-2}} & \text{if } 1 < x \leq 2, \\ e^{-(x-1)^{-2}} - \frac{1}{8}e^{\frac{3}{4}}e^{-(x-2)^{-2}} & \text{otherwise.} \end{cases} \tag{25}$$

Here, the projector matrices $\mathbf{P}, \mathbf{P}_1(\mathbf{x}, t)$ are

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{P}_1(\mathbf{x}, t) = \begin{pmatrix} 1 & \frac{\partial}{\partial x_2}g(x_2) & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 1 \end{pmatrix}. \tag{26}$$

Remark 5 Note that this example is an artificially created DAE without dynamics. Due to the definition of the nonlinear function $g(x)$ and the fact that $x_2 = x'_1 = 0.3\pi \cos(2\pi 10t) \leq 1$, we have that $x'_0 = 0$. Therefore, x_0 is a constant function that equals its initial condition.

To study the proposed modification of the Parareal algorithm, we apply the variants:

– **PR Trapez**

In the first algorithm, no special handling is implemented, that is classic Parareal with the implicit trapezoidal rule as time integrator is applied with a small time step size δt for the fine solver and a larger one ΔT for the coarse propagator.

– **PR Init T**

For the second simulation, the time integration is still performed with the trapezoidal rule. However, in the update formula, we only consider the $\mathbf{PP}_1(\mathbf{x}, t)$ components as in (22). Afterwards, the corresponding consistent initial condition with the same $\mathbf{PP}_1(\mathbf{x}, t)$ components as in (23) is computed.

For this particular example, the Parareal update (22), with \mathbf{PP}_1 as shown in (26), is

$$\begin{aligned} (\hat{x}_0)_n^{k+1} &= (\bar{x}_0)_n^k + \frac{\partial}{\partial x}g(x)\Big|_{(\bar{x}_2)_n^k} (\tilde{x}_1)_n^k + (\bar{x}_0)_n^{k+1} + \frac{\partial}{\partial x}g(x)\Big|_{(\bar{x}_2)_n^{k+1}} (\bar{x}_1)_n^{k+1} \\ &\quad - (\bar{x}_0)_n^k - \frac{\partial}{\partial x}g(x)\Big|_{(\bar{x}_2)_n^k} (\bar{x}_1)_n^k. \end{aligned}$$

Once the updated value $(\hat{x}_0)_n^{k+1}$ is obtained, the consistent initial condition is computed analytically with

$$\begin{aligned} (x_2)_n^{k+1} &= 0.3\pi \cos(20\pi T_n) \\ (x_1)_n^{k+1} &= 0.015 \sin(20\pi T_n), \end{aligned}$$

and using (23) with projector matrices (26) the differential variable is obtained

$$(x_0)_n^{k+1} = (\hat{x}_0)_n^{k+1} + \frac{\partial}{\partial x} g(x) \Big|_{(x_2)_n^{k+1}} (0 - (x_1)_n^{k+1}).$$

For both simulations, $N = 25$ processors are chosen and the simulation time window $\mathcal{I} = [0 \ 1)$. The time step size of the fine trapezoidal rule propagator is set to $\delta t = 10^{-5}$ and the coarse solver is chosen to perform one time step per window and thus has time step size $\Delta T = 1/N$. The Parareal algorithm is iterated until the l_2 norm of the difference between the \mathbf{PP}_1 components of the solution at the end of interval \mathcal{I}_{n-1} and the initial condition of \mathcal{I}_n for all T_n is below a relative tolerance of $5 \cdot 10^{-8}$ and an absolute tolerance of 10^{-15} (see error norm of [16]).

The second algorithm (PR Init T) requires 2 Parareal iterations to converge. However, the first algorithm (PR Trapez), which does not handle the DAE in a special manner, does not manage to converge before having used $N = 25$ iterations, which corresponds to the amount of windows that are employed. The obtained solutions for the purely differential component x_0 and the index two variable x_2 are depicted in Fig. 1.

Even in this simple case without real dynamics, the classic algorithm requires 25 iterations in contrast to the 2 iterations employed by the modified Parareal algorithm for DAEs.

3 Parareal and implicit Euler

As the update (22) in the Parareal algorithm is performed sequentially, the computation of the initial conditions \mathbf{X}_n^{k+1} out of the obtained solution after the update $\hat{\mathbf{X}}_n^{k+1}$ must be done in a sequential manner. This requires the explicit computation of the projector matrices $\mathbf{PP}_1(\mathbf{x}, t)$ as well as of consistent initial conditions. If this operation is computationally expensive, then it can considerably increase the simulation time of the Parareal algorithm. Furthermore, one of the advantages of the Parareal method is that, as it is not intrusive, it can even be applied to black box simulators (as long as you can prescribe initial values). In such cases, obtaining or modifying the matrices of the DAE system that is solved might not be possible.

These two inconveniences can be overcome for DAEs with a specific structure by means of using the implicit Euler method as a time integrator. For that we first make an introduction into these types of DAEs as well as a special property of the implicit Euler method.

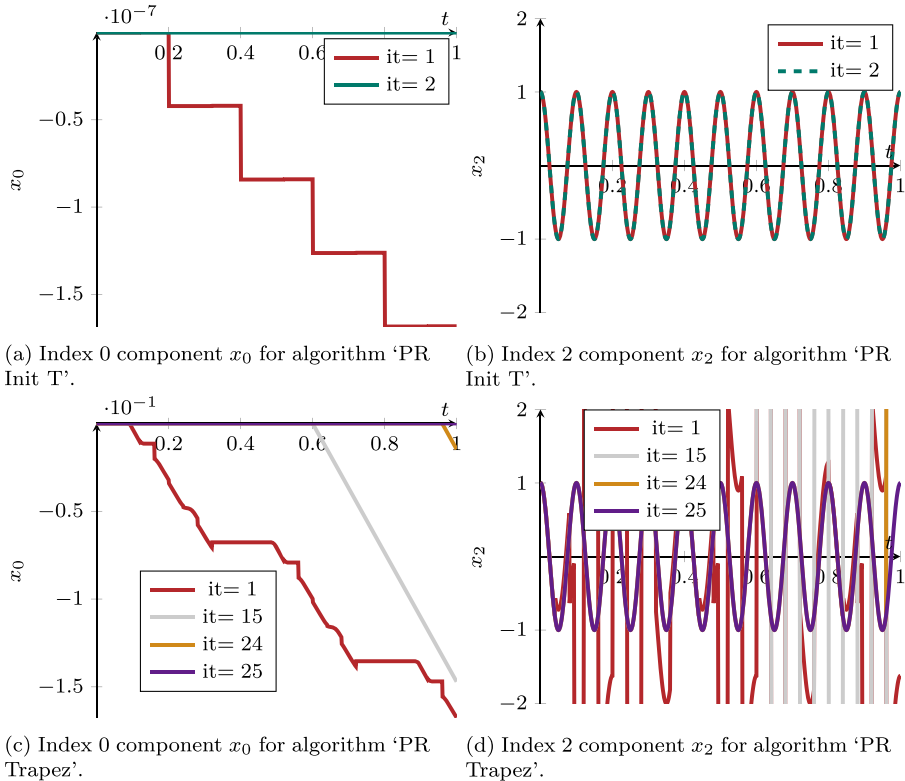


Fig. 1 Solution of the Parareal algorithm after different iterations for the classic algorithm ‘PR Trapez’ and the modified algorithm for DAEs ‘PR Init T’

3.1 Simplified structure

In the following, we consider DAEs with a simplified structure. For that, we impose some additional requirements on the index 2 components.

Assumption 3 (Simplified structure [7]) *We assume that our DAE is index 2 tractable with linear index 2 components and constant matrix \mathbf{A} , that is the system can be re-written as*

$$\mathbf{Ax}' + \underbrace{\mathbf{b}_1(\mathbf{Ux}, t) + \mathbf{B}_2\mathbf{T}\mathbf{x}}_{=: \mathbf{b}(\mathbf{x}, t)} = 0. \tag{27}$$

Furthermore, the projectors \mathbf{P} , \mathbf{Q} , \mathbf{T} and \mathbf{U} are constant. Without loss of generality (see [7]), we consider the projector \mathbf{T} to fulfill $\mathbf{TP} = 0$.

The previous assumption as well as the structure in (27) implies that the index 2 components $\mathbf{T}\mathbf{x}$ appear linearly within the original DAE system. However, this structure is only required for the derivation of theoretical results within the following

section and the system does not need to be explicitly rewritten as (27). Therefore, neither $\mathbf{b}_1(\mathbf{Ux}, t)$, \mathbf{B}_2 nor \mathbf{U} and \mathbf{T} need to be computed.

Assumption 4 (Constant \mathbf{Q}_1) *We assume the index 2 tractable DAE in (27) has a constant projector \mathbf{Q}_1^* onto $\text{Im}\mathbf{Q}_1(\mathbf{Ux}, t)$ and define $\mathbf{P}_1^* = \mathbf{I} - \mathbf{Q}_1^*$.*

Remark 6 In particular, systems of DAEs, e.g. arising from the description of nonlinear RLC circuits fulfil the structural assumptions presented here. This example will be detailed later to demonstrate that the assumptions cover relevant problems.

Before discussing several special properties of the implicit Euler scheme, some important characteristics of the projectors are presented.

Proposition 1 (Projectors c.f. [7]) *We consider a DAE as in (27) with its corresponding projectors fulfilling Assumptions 2, 3 and 4. Then it holds that*

- (i) $\mathbf{P}\mathbf{P}_1(\mathbf{Ux}, t)\mathbf{P} = \mathbf{P}\mathbf{P}_1(\mathbf{Ux}, t)$,
- (ii) $\mathbf{U}\mathbf{Q}\mathbf{P}_1(\mathbf{Ux}, t)\mathbf{P} = 0$,
- (iii) $\mathbf{G}_2(\mathbf{Ux}, t)^{-1}\mathbf{A} = \mathbf{P}_1(\mathbf{Ux}, t)\mathbf{P}$,
- (iv) $\mathbf{G}_2(\mathbf{Ux}, t)^{-1}\mathbf{B}_2\mathbf{T} = \mathbf{T}$,
- (v) $\mathbf{P}\mathbf{P}_1(\mathbf{Ux}, t) = \mathbf{P}\mathbf{P}_1(\mathbf{Ux}, t)\mathbf{P}\mathbf{P}_1^*$,

where the matrix $\mathbf{G}_2(\mathbf{Ux}, t)$ is defined as (8).

Proof Property (i) follows from the feature of the canonical projector shown in Assumption 2 and (ii) from the definition of the projector \mathbf{T} . In [7, Chapter 2.3], Property (iii) is shown and proven and for (iv) the equivalent expression $\mathbf{B}_2\mathbf{T} = \mathbf{G}_2(\mathbf{Ux}, t)\mathbf{T}$ is derived by applying the definitions of the projectors, Assumption 3 and (i). Finally, (v) follows from the definition of \mathbf{Q}_1^* that implies $\mathbf{P}_1(\mathbf{Ux}, t)\mathbf{P}_1^* = \mathbf{P}_1(\mathbf{Ux}, t)$ (see [1, Appendix A.1]). □

Proposition 2 (IERODE, [20]) *A regular DAE (27) fulfilling Assumption 3 with tractability index $v \leq 2$ has an inherent explicit regular ODE (IERODE) which can be expressed only in its purely differential components $\mathbf{x}_{\text{diff}}(t)$.*

Following the results of Proposition 2, Parareal can be applied to the inherent ODE which is formulated only in the differential variables. In this case, convergence is guaranteed by [13] and the index 1 and index 2 components can be post-processed. While this gives us a theoretical justification to apply Parareal to index 2 DAEs, it is commonly not helpful in practice since the inherent ODE is not computable or at prohibitively high cost. Therefore, we study the application of the implicit Euler method directly to the original DAE system (27).

3.2 Implicit Euler

Given a DAE (27) defined on the time interval \mathcal{I} with consistent initial condition \mathbf{x}_0 at initial time t_0 , it can numerically be integrated with the implicit Euler method. For

time steps t_0, t_1, \dots, t_n with $t_n = t_{\text{end}}$, step size $t_{i+1} - t_i = h$ and approximated solutions $\mathbf{x}_0, \dots, \mathbf{x}_i$, the implicit Euler method performs for the $(i + 1)$ th time step the approximation

$$\mathbf{A} \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h} + \mathbf{b}(\mathbf{x}_{i+1}, t_{i+1}) = 0 .$$

Proposition 3 (Implicit Euler with inconsistent initial condition) *We consider a DAE that can be written as (27), with the corresponding projectors fulfilling Assumptions 2, 3 and 4 and two initial conditions at t_0 , the first one, \mathbf{x}^0 , being inconsistent*

$$\mathbf{x}(t_0) = \mathbf{x}^0 \tag{28}$$

and the second one, \mathbf{x}_0 , being consistent

$$\mathbf{x}(t_0) = \mathbf{x}_0 . \tag{29}$$

Let \mathbf{x}^2 be the solution obtained at time point t_2 after two implicit Euler steps of the IVP (27) with inconsistent IC (28) and \mathbf{x}_2 the one obtained with consistent IC (29) both with the same time step sizes. If both initial conditions are such that

$$\mathbf{P}\mathbf{P}_1^* \mathbf{x}^0 = \mathbf{P}\mathbf{P}_1^* \mathbf{x}_0 , \tag{30}$$

then $\mathbf{x}^2 = \mathbf{x}_2$.

Proof The proof uses the splitting procedure employed in [7, Chapter 2.5] for the illustration of how the different components of the degrees of freedom within linear index 2 DAEs affect the computation of consistent initial conditions for index 2 DAEs. The superscript i is used to denote the solutions of the implicit Euler method for time step t_i starting with inconsistent initial condition \mathbf{x}^0 and subscript i for the solutions with consistent initial condition \mathbf{x}_0 .

Application of the implicit Euler scheme yields for the first time step with step size h

$$\mathbf{A} \frac{\mathbf{x}^1 - \mathbf{x}^0}{h} + \mathbf{b}_1(\mathbf{U}\mathbf{x}^1, t_1) + \mathbf{B}_2\mathbf{T}\mathbf{x}^1 = 0 \tag{31}$$

$$\mathbf{A} \frac{\mathbf{x}_1 - \mathbf{x}_0}{h} + \mathbf{b}_1(\mathbf{U}\mathbf{x}_1, t_1) + \mathbf{B}_2\mathbf{T}\mathbf{x}_1 = 0 . \tag{32}$$

The equations are then multiplied by $\mathbf{G}_2(\mathbf{U}\mathbf{x}^1, t_1)^{-1}$ and $\mathbf{G}_2(\mathbf{U}\mathbf{x}_1, t_1)^{-1}$, respectively, which leads to

$$\mathbf{P}_1(\mathbf{U}\mathbf{x}^1, t_1)\mathbf{P} \frac{\mathbf{x}^1 - \mathbf{x}^0}{h} + \mathbf{G}_2(\mathbf{U}\mathbf{x}^1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}^1, t_1) + \mathbf{T}\mathbf{x}^1 = 0 \tag{33}$$

$$\mathbf{P}_1(\mathbf{U}\mathbf{x}_1, t_1)\mathbf{P} \frac{\mathbf{x}_1 - \mathbf{x}_0}{h} + \mathbf{G}_2(\mathbf{U}\mathbf{x}_1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}_1, t_1) + \mathbf{T}\mathbf{x}_1 = 0 \tag{34}$$

due to Proposition 1 (iii) and (iv).

Now, we multiply both equations by \mathbf{T} and \mathbf{U} to split them into the parts defining $\mathbf{T}\mathbf{x}^1$ (respectively $\mathbf{T}\mathbf{x}_1$) and $\mathbf{U}\mathbf{x}^1$ (respectively $\mathbf{U}\mathbf{x}_1$). Thus, we have

$$\mathbf{TP}_1(\mathbf{U}\mathbf{x}^1, t_1)\mathbf{P}\frac{\mathbf{x}^1 - \mathbf{x}^0}{h} + \mathbf{TG}_2(\mathbf{U}\mathbf{x}^1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}^1, t_1) + \mathbf{T}\mathbf{x}^1 = 0 \tag{35}$$

$$\mathbf{UP}_1(\mathbf{U}\mathbf{x}^1, t_1)\mathbf{P}\frac{\mathbf{x}^1 - \mathbf{x}^0}{h} + \mathbf{UG}_2(\mathbf{U}\mathbf{x}^1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}^1, t_1) = 0 \tag{36}$$

and

$$\mathbf{TP}_1(\mathbf{U}\mathbf{x}_1, t_1)\mathbf{P}\frac{\mathbf{x}_1 - \mathbf{x}_0}{h} + \mathbf{TG}_2(\mathbf{U}\mathbf{x}_1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}_1, t_1) + \mathbf{T}\mathbf{x}_1 = 0 \tag{37}$$

$$\mathbf{UP}_1(\mathbf{U}\mathbf{x}_1, t_1)\mathbf{P}\frac{\mathbf{x}_1 - \mathbf{x}_0}{h} + \mathbf{UG}_2(\mathbf{U}\mathbf{x}_1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}_1, t_1) = 0. \tag{38}$$

Application of Proposition 1 (ii), (i) and (v) together with equality $\mathbf{PP}_1^*\mathbf{x}^0 = \mathbf{PP}_1^*\mathbf{x}_0$ to the equations defining the \mathbf{U} components, and $\mathbf{PT} = 0$ yields

$$\mathbf{TP}_1(\mathbf{U}\mathbf{x}^1, t_1)\mathbf{PU}\frac{\mathbf{x}^1 - \mathbf{x}^0}{h} + \mathbf{TG}_2(\mathbf{U}\mathbf{x}^1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}^1, t_1) + \mathbf{T}\mathbf{x}^1 = 0 \tag{39}$$

$$\mathbf{UPP}_1(\mathbf{U}\mathbf{x}^1, t_1)\frac{\mathbf{x}^1 - \mathbf{x}_0}{h} + \mathbf{UG}_2(\mathbf{U}\mathbf{x}^1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}^1, t_1) = 0 \tag{40}$$

and

$$\mathbf{TP}_1(\mathbf{U}\mathbf{x}_1, t_1)\mathbf{PU}\frac{\mathbf{x}_1 - \mathbf{x}_0}{h} + \mathbf{TG}_2(\mathbf{U}\mathbf{x}_1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}_1, t_1) + \mathbf{T}\mathbf{x}_1 = 0 \tag{41}$$

$$\mathbf{UPP}_1(\mathbf{U}\mathbf{x}_1, t_1)\frac{\mathbf{x}_1 - \mathbf{x}_0}{h} + \mathbf{UG}_2(\mathbf{U}\mathbf{x}_1, t_1)^{-1}\mathbf{b}_1(\mathbf{U}\mathbf{x}_1, t_1) = 0. \tag{42}$$

Here, (40) and (42) are to be solved to obtain the solutions for $\mathbf{U}\mathbf{x}^1$ and $\mathbf{U}\mathbf{x}_1$, respectively. It can be seen that both equations are equivalent and their solution only depends on $\mathbf{PP}_1^*\mathbf{x}_0$. Therefore, they yield the same solution, i.e. $\mathbf{U}\mathbf{x}^1 = \mathbf{U}\mathbf{x}_1$.

To obtain $\mathbf{T}\mathbf{x}^1$ and $\mathbf{T}\mathbf{x}_1$, (39) and (41) have to be solved. Again, both equations are equivalent; however, whereas the first one depends on $\mathbf{U}\mathbf{x}^1$ and $\mathbf{U}\mathbf{x}^0$, the second one depends on $\mathbf{U}\mathbf{x}_1$ and $\mathbf{U}\mathbf{x}_0$. Due to the previous step, $\mathbf{U}\mathbf{x}^1 = \mathbf{U}\mathbf{x}_1$, but, $\mathbf{U}\mathbf{x}^0$ is not necessarily equal to $\mathbf{U}\mathbf{x}_0$. Thus, if $\mathbf{U}\mathbf{x}^0 = \mathbf{U}\mathbf{x}_0$, then $\mathbf{T}\mathbf{x}^1 = \mathbf{T}\mathbf{x}_1$ and only one implicit Euler step is required to obtain the same solution with both initial conditions \mathbf{x}^0 and \mathbf{x}_0 .

If $\mathbf{U}\mathbf{x}^0 \neq \mathbf{U}\mathbf{x}_0$, then the analogous procedure is repeated to obtain the solution for t_2 . This time, the index 2 components $\mathbf{T}\mathbf{x}^2$ and $\mathbf{T}\mathbf{x}_2$ are again defined by the same equation and depend on $\mathbf{U}\mathbf{x}^1$ and $\mathbf{U}\mathbf{x}^2$ or $\mathbf{U}\mathbf{x}_1$ and $\mathbf{U}\mathbf{x}_2$, respectively. Due to the previous step, $\mathbf{U}\mathbf{x}^1 = \mathbf{U}\mathbf{x}_1$ and, analogously as before, we obtain $\mathbf{U}\mathbf{x}^2 = \mathbf{U}\mathbf{x}_2$ and thus $\mathbf{x}_2 = \mathbf{x}^2$. □

Remark 7 Note that, as the implicit Euler scheme starting with a consistent initial condition yields solutions that are consistent for semiexplicit index 2 DAEs, \mathbf{x}_2 is consistent [3]. Therefore, as $\mathbf{x}^2 = \mathbf{x}_2$, \mathbf{x}^2 is a consistent solution that is obtained after two implicit Euler steps with an inconsistent initial condition. In practice, this consistency is only obtained up to a certain tolerance which depends, e.g. on the accuracy of the Newton scheme.

In [7], a similar result is shown for the implicit Euler scheme. However, for schemes starting with only inconsistent index 2 variables, that is schemes where $\mathbf{U}\mathbf{x}^0 = \mathbf{U}\mathbf{x}_0$ (in fact, only $\mathbf{P}\mathbf{x}^0 = \mathbf{P}\mathbf{x}_0$ is required). We have extended this result to consider DAEs, where also the $\mathbf{P}\mathbf{Q}_1^*$ components might be inconsistent. Furthermore, we have shown that the $\mathbf{P}\mathbf{P}_1^*$ components of the initial condition are “remembered” by the time integration scheme.

3.3 Implicit Euler as propagator

In the following, we present an alternative to the modified Parareal for DAEs of Section 2 for DAEs of type (27).

Proposition 4 (Parareal with implicit Euler) *We consider a DAE that can be written as (27) and the corresponding projectors fulfilling Assumptions 2, 3 and 4 and apply Parareal with the implicit Euler scheme on the coarse and the fine level. If*

- for index 1 at least one time step is used on the fine level,
- for index 2 at least two time steps are used on the fine level,

Parareal is applicable without the requirement of explicitly making the initial conditions consistent.

Let us consider the DAE (27) fulfilling Assumptions 3 and 4, as required in Proposition 3. Here, a constant projector $\mathbf{P}\mathbf{P}_1^*$ onto the differential components exists, and thus the classic Parareal update

$$\hat{\mathbf{X}}_n^{k+1} = \tilde{\mathbf{X}}_n^k + \bar{\mathbf{X}}_n^{k+1} - \bar{\mathbf{X}}_n^k \tag{43}$$

implies

$$\mathbf{P}\mathbf{P}_1^* \hat{\mathbf{X}}_n^{k+1} = \mathbf{P}\mathbf{P}_1^* \tilde{\mathbf{X}}_n^k + \mathbf{P}\mathbf{P}_1^* \bar{\mathbf{X}}_n^{k+1} - \mathbf{P}\mathbf{P}_1^* \bar{\mathbf{X}}_n^k . \tag{44}$$

Implicit Euler yields a consistent solution after at most two time steps (see Proposition 3). Furthermore, its solution corresponds to the one obtained with a consistent initial condition where the $\mathbf{P}\mathbf{P}_1^*$ coincide, that is

$$\mathbf{P}\mathbf{P}_1^* \left(\mathbf{X}_n^{k+1} - \hat{\mathbf{X}}_n^{k+1} \right) = 0 . \tag{45}$$

Thus, if on the fine level at least two Euler steps are performed, the solution at the end of the interval is equivalent to first computing the consistent initial condition \mathbf{X}_n^{k+1} and then starting the simulation with it. On the coarse level, already one implicit Euler step suffices, as only the $\mathbf{P}\mathbf{P}_1^*$ components are relevant for the update and those are remembered by the time integration method also in the first time step, as shown in the proof of Proposition 3.

Therefore, for DAEs fulfilling the requirements of Proposition 3 it is not necessary to use the modified Parareal update (22) and the subsequent computation of a consistent initial condition. Here, the usage of the implicit Euler as coarse propagator and performing at least two implicit Euler steps on the fine level are sufficient for the Parareal algorithm to converge.

Remark 8 If we know that a DAE fulfils Assumptions 3 and 4, then no projectors need to be computed nor any special handling is required. It suffices to use the implicit Euler method as a time integrator within the classic Parareal algorithm to ensure that the algorithm handles the algebraic constraints appropriately.

In [27], the implicit projection property of the implicit Euler method is observed for an index 1 DAE with constant mass matrix in the context of Parareal. This is a special case of the theory presented in Section 3.3 and is extended here to index 2 systems including hidden constraints. On the other hand, the approach taken in [11], is a special case of the algorithm proposed in Section 2 for index 1 DAEs. There, all the components are added in the Parareal update formula (4), which is equivalent to (22) if the projector \mathbf{P} is constant and the algebraic variables are made consistent afterwards.

3.3.1 DAE with nonlinear index 2 components

Let us come back to the DAE of the first numerical example in Section 2.1

$$x'_0 + g(x_2) = 0 \tag{46a}$$

$$x'_1 - x_2 = 0 \tag{46b}$$

$$x_1 - 0.015 \sin(2\pi 10t) = 0, \tag{46c}$$

note that the index 2 component x_2 appears nonlinearly in the original system. Therefore, this example cannot be rewritten as (27) and Proposition 4 does not hold. This implies that starting the Euler scheme with an inconsistent initial value \mathbf{x}^0 does not yield the same solution after two time steps than starting with the consistent initial condition \mathbf{x}_0 with the same $\mathbf{PP}_1(\mathbf{x}_0, t)$ components.

To exemplify this behaviour, a counterexample is presented. Let us consider the inconsistent initial condition $\mathbf{x}^0 = (0, -1, 0)^\top$ at $t_0 = 0$. Its corresponding consistent initial condition \mathbf{x}_0 with

$$\mathbf{PP}_1(\mathbf{x}_0, t)(\mathbf{x}_0 - \mathbf{x}^0) = 0$$

is $\mathbf{x}_0 = (0, 0, 0.3\pi)^\top$. After two implicit Euler steps with, e.g. time step $h = 1/3$ and starting with the inconsistent initial condition \mathbf{x}^0 ,

$$x^2 = -g(0.045 \sin(20\pi/3) + 3)/3 \neq 0$$

is obtained. The same scheme for consistent initial value \mathbf{x}_0 , however, yields the different solution

$$x_2 = 0$$

which differs from x^2 , i.e. $x^2 \neq x_2$.

Therefore, for this DAE, the implicit Euler scheme does not ‘remember’ the $\mathbf{PP}_1(\mathbf{x}, t)$ (or differential) components of inconsistent initial conditions and the classic Parareal algorithm without special handling for DAEs leads in this case to a slower convergence.

4 Application

To exemplify the theoretical results presented in Sections 3.2 and 3.3 as well as demonstrate that the assumptions that are taken hold for DAEs arising from real-life applications, we further present an example describing an electric network. Before presenting the simulation results of applying classic as well as the DAE-modified Parareal algorithm, we introduce a system of DAEs that describes the dynamic behaviour of RLC-circuits.

4.1 Modified nodal analysis

We consider electric networks containing capacitors (C), inductors (L), resistors (R) and independent voltage (V) and current (I) sources. In modified nodal analysis (MNA), networks are described by means of incidence matrices \mathbf{A}_\star , $\star \in \{C, L, R, V, I\}$ that characterise the branch-to-node relation of the underlying graph for the corresponding elements. Applying Kirchhoff’s current law and the lumped parameter models of the different components, we obtain the following system of DAEs [9, 15]

$$\mathbf{A}_C \mathbf{q}' + \mathbf{A}_R \mathbf{g}_R(\mathbf{A}_R^\top \mathbf{e}, t) + \mathbf{A}_L \mathbf{i}_L + \mathbf{A}_V \mathbf{i}_V + \mathbf{A}_I \mathbf{i}_s(t) = 0 \tag{48a}$$

$$\mathbf{q} - \mathbf{q}_C(\mathbf{A}_C^\top \mathbf{e}, t) = 0 \tag{48b}$$

$$\boldsymbol{\phi}' - \mathbf{A}_L^\top \mathbf{e} = 0 \tag{48c}$$

$$\boldsymbol{\phi} - \boldsymbol{\phi}_L(\mathbf{i}_L, t) = 0 \tag{48d}$$

$$\mathbf{A}_V^\top \mathbf{e} - \mathbf{v}_s(t) = 0. \tag{48e}$$

Here, the system of DAEs is given in the flux-charge formalism. In this formulation, the degrees of freedom are $\mathbf{e} : \mathcal{I} \rightarrow \mathbb{R}^{n_e}$, the vector of node potentials, $\mathbf{i}_\star : \mathcal{I} \rightarrow \mathbb{R}^{n_\star}$, the vector of currents through branches containing the element \star , $\mathbf{q} : \mathcal{I} \rightarrow \mathbb{R}^{n_c}$, the vector of charges in capacitances and $\boldsymbol{\phi} : \mathcal{I} \rightarrow \mathbb{R}^{n_l}$, the vector of fluxes in inductances. Finally, $\mathbf{q}_C(\cdot)$, $\boldsymbol{\phi}_L(\cdot)$, $\mathbf{g}_R(\cdot)$, $\mathbf{i}_s(\cdot)$ and $\mathbf{v}_s(\cdot)$ are (nonlinear) functions describing the lumped parameter relations for the different elements. The vector of node potentials and the incidence matrices allow extracting the voltages across the branches containing a given element \star with the relation $\mathbf{v}_\star = \mathbf{A}_\star^\top \mathbf{e}$.

The tractability index of this system has already been analysed (see, e.g. [9]) and is in the worst case 2. This result is given by only topological properties of the underlying graph.

The index analysis in [9] shows that the possible index 2 components of the system are currents through voltage sources \mathbf{i}_V and voltages across inductances $\mathbf{A}_L^\top \mathbf{e}$. These two degrees of freedom appear linearly in the original system (48) and thus flux-charge MNA has linear index 2 components [2]. Therefore, system (48) is, in the worst case, an index 2 tractable DAE with linear index 2 components and constant matrix \mathbf{A} and thus can be written as (27). Finally, in [9], it is also shown that Assumption 4 is fulfilled, as $\text{Im} \mathbf{Q}_1(\mathbf{U}\mathbf{x}, t)$ is constant. This allows the application of Proposition 3 to the system of DAEs obtained from flux-charge MNA and thus the

implicit Euler scheme returns a consistent solution after at most two time steps even if an inconsistent initial condition is given.

4.2 Numerical example

We consider a nonlinear index 2 DAE arising from a circuit described with flux-charge modified nodal analysis. It's a nonlinear index 2 circuit shown in Fig. 2 with parameters $R_{1,1} = 10^{-2} \Omega$, $R_{1,2} = 10^{-2} \Omega$, $L_1 = 10^{-4} \text{ H}$ and current source

$$i_1(t) = (100 \sin(100\pi t) + 50 \sin(400\pi t)) \text{ A} .$$

For the nonlinear inductance, the model of [6] is used with nominal inductance $L_{\text{nom}} = 10^{-3} \text{ H}$, deep saturation inductance $L_{\text{deepsat}} = 8 \cdot 10^{-4}$, smoothness factor $\sigma = 5 \cdot 10^{-2}$ and current $I_L^* = 90 \text{ A}$.

Again, we apply to Parareal variants:

- **PR Euler**

In the first variant, again, the classic Parareal method without any special handling is implemented; however, this time, the implicit Euler scheme is used as a time integrator with a small time step size δt for the fine solver and a larger one ΔT for the coarse propagator.

- **PR Init E**

The second simulation is performed again with the implicit Euler method as time integrator. However, the modified Parareal algorithm for DAEs is used, where the update formula is computed only on the the $\mathbf{PP}_1(\mathbf{x}, t)$ components as in (22). Afterwards, the corresponding consistent initial condition with the same $\mathbf{PP}_1(\mathbf{x}, t)$ components as in (23) is computed.

The consistent initial conditions are not computed manually, but numerically with the Python package InitDAE. For the ‘PR Init’ algorithm, the projectors \mathbf{PP}_1^* are used for the update formula. In both simulations $N = 15$ processors are chosen and

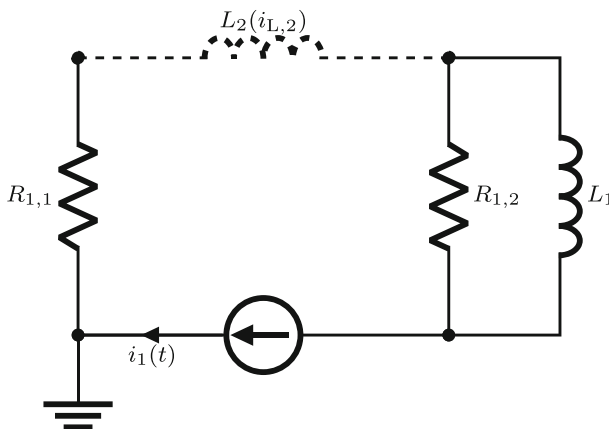
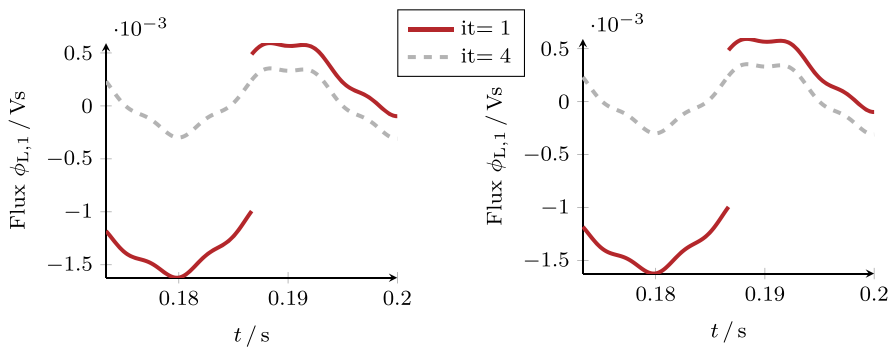


Fig. 2 Index 2 circuit with nonlinear inductance $L_2(i_{L,2})$ as described in [6]. Figure based on [5]

the simulation time window is set to $\mathcal{I} = [0, 0.2)$. For the fine solution, implicit Euler with a time step size of $\delta t = 10^{-5}$ is used and the coarse implicit Euler solver performs one time step per window. The initial condition \mathbf{x}_0 is computed by starting an implicit Euler scheme with an inconsistent value $\mathbf{x}^{-2} = 0$ at time step $t_0 - 2\delta t$ and performing two steps until t_0 . Exploiting Proposition 3, the obtained solution \mathbf{x}_0 for the rest of the simulation is a consistent initial condition. The error is computed as in the previous example and the relative tolerance is set to 10^{-4} , whereas the absolute tolerance is chosen to be 10^{-8} .

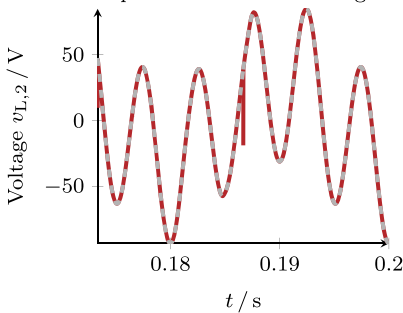
Both algorithms require 4 Parareal iterations to reach the required tolerance. The result is not surprising, as the DAE has the special structure presented in Section 3.2 such that the implicit Euler method is able to handle the algebraic constraints appropriately. Therefore, both algorithms perform the same update on the differential \mathbf{PP}_1 components (and thus the transient part) of the system. Whereas the modified Parareal algorithm for DAEs does this explicitly, the classic Parareal algorithm does the same update implicitly by means of using the implicit Euler method as the time integrator. Furthermore, both algorithms use the same step sizes and time integrators.

In Fig. 3, the purely differential component $\phi_{L,1}$ and the index 2 component $v_{L,2}$ of the fine solutions on the last two windows $\mathcal{I}_n, n = 14, 15$ are depicted for the first

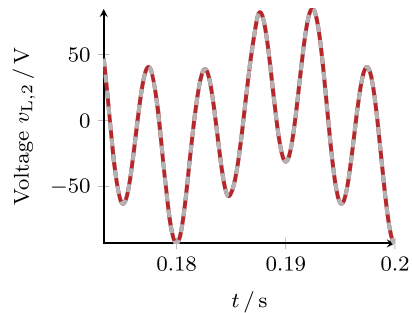


(a) Solution with the classic Parareal algorithm and implicit Euler as time integrator.

(b) Solution with the modified Parareal algorithm for DAEs.



(c) Solution with the classic Parareal algorithm and implicit Euler as time integrator.



(d) Solution with the modified Parareal algorithm for DAEs.

Fig. 3 Solution of the Parareal algorithm for the last two windows \mathcal{I}_{14} and \mathcal{I}_{15}

and the last Parareal iterations. It can be seen that in both cases the purely differential component has a jump at the first iteration and becomes continuous at the last one, which is a typical behaviour observed when applying the Parareal algorithm. As in the previous example, the index 2 component has no jumps and is converged at the first iteration for the ‘PR Init E’ algorithm, whereas ‘PR Euler’ starts with an inconsistent solution at the first step in the first iteration. Here, the behaviour of the implicit Euler scheme when starting with inconsistent initial conditions can be observed: the solution jumps to the correct value due to Property 3. This, however, does not negatively affect the convergence of ‘PR Euler’, as the purely differential components are handled equally in both algorithms.

Remark 9 Note that, even for a DAE fulfilling the requirements of Proposition 3, it can happen that ‘PR Euler’ may reach the required tolerance for the differential components, while the algebraic ones have not reached the required accuracy yet. This inconvenience, however, can easily be overcome without requiring additional Parareal iterations. Let us consider the solution at \mathcal{I}_n . For index 1 components, it suffices to take the result of the last time step of the previous window \mathcal{I}_{n-1} instead of the initial value at the beginning of \mathcal{I}_n to ensure a consistent value at time T_{n-1} . The index 2 components would in addition require to either ignore the first (possibly) inconsistent time step of \mathcal{I}_n at $T_{n-1} + \delta t$ (with δt being the time step size of the fine propagator), or perform one extra time step of the fine solution of \mathcal{I}_{n-1} to arrive at the time $T_{n-1} + \delta t$.

In Fig. 4, the convergence of the Parareal algorithm is plotted for the purely differential component $\phi_{L,1}$. Here, the error after the first Parareal iteration $k = 1$ with respect to the sequential solution both evaluated at time step T_2 , $\phi_{L,1}^{2,1}(T_2)$ and $\phi_{L,1}^{seq}(T_2)$, respectively, is shown for different window sizes H . Both algorithms use

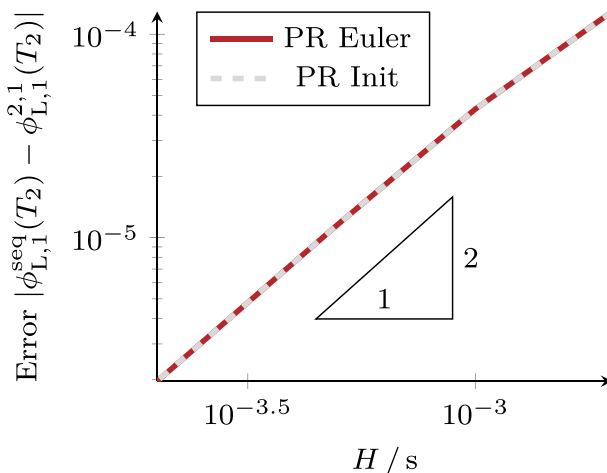


Fig. 4 Convergence plot for Parareal at iteration $k = 1$ and time T_2 applied with different window sizes H

the implicit Euler method as coarse propagator, whose local truncation error is of order 2. The convergence rate is 2, which corresponds to the one expected from the classic Parareal theory for ODEs (see [13]). This is not surprising, as the plot is illustrating the convergence order of the purely differential components (index 0 component) of the DAE.

5 Conclusions

This article has presented a modification of the Parareal algorithm for its application to quasilinear index 2 tractable DAEs. Its extension to higher index systems requires extra care, however it follows analogously from the projector-based decoupling of differential algebraic equations. For a large class of DAEs, i.e. linear index 2 components and constant matrix \mathbf{A} as given in flux-charge formulated modified nodal analysis, a new property of the implicit Euler scheme is proven. This property allows the usage of the classic Parareal algorithm, as long as the implicit Euler scheme is used as the time integrator of both the first two time steps of the fine as well as for the coarse propagator.

The theoretical results are backed up by numerical simulations of two DAEs, one toy example with nonlinear index 2 components and the other one arising from a flux-charge modified nodal analysis formulated circuit. As theoretically expected, the modified Parareal algorithm speeds up the convergence when applied to a DAE with nonlinear index 2 components.

In future works, the study of other time integration methods with similar projection properties as the implicit Euler scheme could be performed. This would enable the application of Parareal to DAEs without special handling for further, possibly higher order, time stepping schemes.

Acknowledgements We would like to thank Diana Estévez Schwarz and Lennart Jansen for the fruitful discussions as well as Pia Callmer for her assistance in implementing the algorithm. This work is based on Chapter 5 of the PhD thesis of Idoia Cortes Garcia [5].

Funding This work is supported by the Graduate School CE within the Centre for Computational Engineering at Technische Universität Darmstadt and DFG Grant SCHO1562/1-2 and BMBF Grant 05M2018RDA (PASIROM). This research was partially funded from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955701. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, Switzerland.

Availability statement The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as

you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Baumanns, S.: Coupled electromagnetic field/circuit simulation: modeling and numerical analysis. Dissertation, Universität zu Köln (2012)
2. Baumanns, S., Selva Soto, M., Tischendorf, C.: Consistent initialization for coupled circuit-device simulation. In: Roos, J., Costa, L.R.J. (eds.) *Scientific Computing in Electrical Engineering SCEE 2008, Mathematics in Industry*, vol. 14, pp. 297–304. Springer (2010). https://doi.org/10.1007/978-3-642-12294-1_38
3. Brenan, K.E., Campbell, S.L., Petzold, L.R.: Numerical solution of initial-value problems in differential-algebraic equations. Society for industrial and applied mathematics. <https://doi.org/10.1137/1.9781611971224> (1995)
4. Cadeau, T., Magoules, F.: Coupling the Parareal algorithm with the waveform relaxation method for the solution of differential algebraic equations. In: *10Th International symposium on distributed computing and applications to business, engineering and science*, pp. 15–19 (2011). <https://doi.org/10.1109/DCABES.2011.34>
5. Cortes Garcia, I.: Mathematical analysis and simulation of field models in accelerator circuits. Dissertation, Technische Universität Darmstadt. <https://doi.org/10.1007/978-3-030-63273-1>. Springer Theses: Recognizing Outstanding Ph.D. Research (2021)
6. Di Capua, G., Femia, N.: A novel method to predict the real operation of ferrite inductors with moderate saturation in switching power supply applications. *IEEE Transactions on Power Electronics* **31**(3), 2456–2464 (2016). <https://doi.org/10.1109/TPEL.2015.2438952>
7. Estévez Schwarz, D.: Consistent initialization for index-2 differential algebraic equations and its application to circuit simulation. Dissertation. Humboldt-Universität Berlin (2000)
8. Estévez, S.schwarz, D., L.amour, R.: A new approach for computing consistent initial values and taylor coefficients for daes using projector-based constrained optimization. *Numerical Algorithms* **78**(2), 355–377 (2018). <https://doi.org/10.1007/s11075-017-0379-9>
9. Estévez Schwarz, D., Tischendorf, C.: Structural analysis of electric circuits and consequences for MNA. *International Journal of Circuit Theory and Applications* **28**(2), 131–162 (2000). [https://doi.org/10.1002/\(SICI\)1097-007X\(200003/04\)28:2;<131::AID-CTA100>3.0.CO;2-W](https://doi.org/10.1002/(SICI)1097-007X(200003/04)28:2;<131::AID-CTA100>3.0.CO;2-W)
10. Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. *SIAM J. Sci. Comput.* **36**(6), C635–C661 (2014). <https://doi.org/10.1137/130944230>
11. Falgout, R.D., Lecouvez, M., Woodward, C.S.: A parallel-in-time algorithm for variable step multistep methods. *J. Comput. Sci.* **37**, 101029 (2019). <https://doi.org/10.1016/j.jocs.2019.101029>
12. Gander, M.J.: 50 years of time parallel time integration. In: Carraro, T., Geiger, M., Körkel, S., Rannacher, R. (eds.) *Multiple shooting and time domain decomposition methods, contributions in mathematical and computational sciences*, vol. 9, pp. 69–113. Springer (2015). https://doi.org/10.1007/978-3-319-23321-5_3
13. Gander, M.J., Hairer, E.: Nonlinear convergence analysis for the Parareal algorithm. In: Langer, U., Discacciati, M., Keyes, D.E., Widlund, O.B., Zulehner, W. (eds.) *Domain decomposition methods in science and engineering XVII*, pp. 45–56. Springer (2008). https://doi.org/10.1007/978-3-540-75199-1_4
14. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Stat. Comput.* **29**(2), 556–578 (2007). <https://doi.org/10.1137/05064607X>
15. Günther, M., Feldmann, U., ter Maten, E.J.W.: Modelling and discretization of circuit problems. In: Schilders, W.H.A., ter Maten, E.J.W. (eds.) *Numerical methods in electromagnetics, handbook of numerical analysis*, vol. 13, pp. 523–659. Elsevier (2005). [https://doi.org/10.1016/s1570-8659\(04\)13006-8](https://doi.org/10.1016/s1570-8659(04)13006-8)
16. Hairer, E., Nørsett, S.P., Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics, 2nd edn. Springer, Berlin, Heidelberg (2000)

17. Hairer, E., Nørsett, S.P., Wanner, G. Solving ordinary differential equations II: stiff and differential-algebraic problems. Springer Series in Computational Mathematics, 2nd edn. Springer, Berlin, Heidelberg (2002)
18. Lamour, R.: A shooting method for fully implicit index-2 differential-algebraic equations. <https://doi.org/10.1137/S1064827595287274>. Humboldt-Universität Berlin (1994)
19. Lamour, R.: A shooting method for fully implicit index-2 differential algebraic equations. *SIAM J. Sci. Comput.* **18**(1), 94–114 (1997). <https://doi.org/10.1137/s1064827595287274>
20. Lamour, R., März, R., Tischendorf, C.: Differential-algebraic equations: a projector based analysis. *Differential-algebraic equations forum springer*. <https://doi.org/10.1007/978-3-642-27555-5> (2013)
21. Lions, J.L., Maday, Y., Turinici, G.: A Parareal in time discretization of PDEs. *Comptes Rendus de l'Académie des Sciences – Series I – Mathematics* **332**(7), 661–668 (2001). [https://doi.org/10.1016/S0764-4442\(00\)01793-6](https://doi.org/10.1016/S0764-4442(00)01793-6)
22. März, R., Tischendorf, C.: Solving more general index-2 differential-algebraic equations. *Computers and Mathematics with Applications* **28**(10-12), 77–105 (1994). [https://doi.org/10.1016/0898-1221\(94\)00187-1](https://doi.org/10.1016/0898-1221(94)00187-1)
23. Mehrmann, V.: Index concepts for differential-algebraic equations, pp. 676–681. Springer. https://doi.org/10.1007/978-3-540-70529-1_120 (2015)
24. Nievergelt, J.: Parallel methods for integrating ordinary differential equations. *Communications of the Association for Computing Machinery* **7**, 731–733 (1964)
25. Salon, S.J.: *Finite element analysis of electrical machines* kluwer (1995)
26. Schöps, S.: Multiscale modeling and multirate time-integration of field/circuit coupled problems. Dissertation, Bergische Universität Wuppertal & Katholieke Universiteit Leuven. <http://elpub.bib.uni-wuppertal.de/servlets/DocumentServlet?id=2132>. VDI Verlag. Fortschritt-Berichte VDI, Reihe 21.
27. Schöps, S., Niyonzima, I., Clemens, M.: Parallel-in-time simulation of eddy current problems using Parareal. *IEEE Trans. Magn.* **54**(3), 1–4 (2018). <https://doi.org/10.1109/TMAG.2017.2763090>
28. Schroder, J.B., Falgout, R.D., Woodward, C.S., Top, P., Lecouvez, M.: Parallel-in-time solution of power systems with scheduled events. In: 2018 IEEE power energy society general meeting (PESGM), pp. 1–5 (2018). <https://doi.org/10.1109/PESGM.2018.8586435>
29. Takahashi, Y., Fujiwara, K., Iwashita, T., Nakashima, H.: Parallel finite-element method based on space-time domain decomposition for magnetic field analysis of electric machines. *IEEE Trans. Magn.* **55**(6), 1–4 (2019). <https://doi.org/10.1109/tmag.2019.2895590>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.