On the Solution of Differential-Algebraic Equations through Gradient Flow Embedding

Ehecatl Antonio del Rio-Chanona[a], Craig Bakker[c], Fabio Fiorelli[a], Michail Paraskevopoulos[a], Felipe Scott[d], Raúl Conejeros[b], Vassilios S. Vassiliadis[a,*]

[a] *Department of Chemical Engineering and Biotechnology, University of Cambridge, Pembroke Street, Cambridge CB2 3RA, United Kingdom*
[b] *Escuela de Ingeniería Bioquímica, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2085, Valparaiso, Chile*
[c] *Whiting School of Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, USA*
[d] *Green Technology Research Group, Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Chile, Mons. Álvaro del Portillo 12455, Las Condes, Santiago, 7620001, Chile*

Abstract

In this paper Gradient Flow methods are used to solve systems of Differential-Algebraic Equations via a novel reformulation strategy, focusing on the solution of index-1 Differential-Algebraic Equation systems. A reformulation is first effected on semi-explicit index-1 Differential-Algebraic Equation systems, which casts them as pure Ordinary Differential Equation systems subject to an embedded pointwise least-squares problem. This is then formulated as a Gradient Flow optimization problem. Rigorous proofs for this novel scheme are provided for asymptotic and epsilon convergence. The computational results validate the predictions of the effectiveness of the proposed approach, with efficient and accurate solutions obtained for the case studies considered. Beyond the theoretical and practical value for the solution of DAE systems as pure ODE ones, the methodology is expected to have an impact in similar cases where an ODE system is subjected to algebraic constraints, such as the Hamiltonian necessary conditions of optimality in Optimal Control problems.

Keywords: Differential-Algebraic Equations; Gradient Flow; semi-explicit index-1 DAE; Ordinary Differential Equations;

1. Introduction

There are many applications for which the mathematical model of a physical system can be easily represented as a system of Differential-Algebraic Equations (DAEs), *e.g.* mass/energy balances coupled with physical laws, such as physical property predictions. In chemical engineering, in particular, it was recognized that problems of this format were important for the effective simulation of dynamic distillation models and equilibrium laws [17].

DAEs have long been considered a generalization of Ordinary Differential Equations (ODEs) systems [2]; this interpretation has allowed for the resolution of DAE systems employing techniques developed for ODEs. These ideas can be first seen in [8], where the author shows that problems with both differential and algebraic equations can be solved by a rather simple implementation. This consists in finding the solution to a method relying on predictor and corrector equations, specifically a linear system derived from a Newton iteration.

*Corresponding Author
Email address: vsv20@cam.ac.uk (Vassilios S. Vassiliadis)

The direct methods that emerged apply ODE techniques directly to higher index DAE systems [1]. The problem of finding consistent initial conditions was immediately recognized, as it is necessary to satisfy strict equality conditions in the algebraic part of the DAE system. It was also seen that errors in initialization could propagate through the simulation. Implicit methods were introduced [12] to find the initial consistent initial conditions. Further investigation revealed the difficulty in providing a rigorous solution to problems at initialization [13].

A further important development comes from the ability to transform a higher index system of DAEs into an equivalent pure ODE system. This allows using more effectively numerical integration techniques with lower error. The Pantelides algorithm [16], concerned with the identification of consistent and non-redundant initial conditions, is a widespread implementation of this idea which uses bipartite graphs to find correlations between variables, therefore allowing for restructuring the problem into a form where the model Jacobian matrix has low or no nilpotency. Another way to reduce the index is the structural matrix method [19, 7], which is a symbolic-algebraic method that has been expanded to employ numerical information. By operating on incidence matrices, it accounts for cancellation of variables in linear expressions, estimating with higher reliability the correct index of the system.

The work presented in this paper introduces a new framework for the solution of DAEs, based on the concept of Gradient Flow. Instead of differentiating to obtain a full rank matrix of first derivatives of all the variables of the DAE system, the techniques introduced here allow a DAE system to be transformed into an ODE system if reformulated as an embedded least squares optimization problem, and subsequently employing a simple scaling scheme. Barrlund [3], used a least squares formulation to satisfy the algebraic constraints in a linear DAE system. This is done such that the squared equality constraints are minimized at every point in time of the ODE integration, this is then equivalent to solving the ODE system while enforcing the equality constraints. In this regard, an ODE whose solutions include the solutions of a DAE is called a *completion* of that DAE [6]. A disadvantage with the existing approaches of least squares completions is that the additional dynamics introduced by its use can affect the numerical solution of the DAE [15].

This paper is organized as follows: in section 2 an introduction to Gradient Flow is presented, in section 3 the Gradient Flow techniques are extended to the resolution of DAEs in a matrix-inversion (or factorization) free format. Convergence analysis is presented in section 4. Applicability of these techniques is presented via detailed computational case studies in section 5. Finally in section 6 the conclusions of this work are summarized.

2. Gradient Flow

The idea behind the Gradient Flow method (GF) first arose in the study of variational partial differential equations back in 1908 [9]. This approach was developed due to the necessity to solve these equations, where each of them has a function $f : X \to \mathbb{R}$ and the solution to such equation is a minimizer of f. It was a consequence of this line of research, that to solve these systems of variational partial differential equations the GF framework (also called method of gradients) was introduced [4].

A first GF algorithm is the transcription of the Steepest Descent (SD) method into ODE form. The Gradient Flow method starts with an initial point $x_0 \in X$ and seeks to find a minimizer of $f(\phi(t))$ by following a curve $\phi(t)$ defined by the ordinary differential equation

$$\begin{aligned} \frac{d\phi(t)}{dt} &= -\nabla f(\phi(t)) \\ \phi(0) &= x_0 \end{aligned} \tag{1}$$

where ∇f is the gradient of f. The solution is called an integral curve.

For the unconstrained minimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2}$$

the original method proceeds by updating the iteration point towards finding a minimizer of the objective function by the following scheme:

$$x_{k+1} = x_k - \alpha_k g_k; \quad x \in \mathbb{R}^n \tag{3}$$

where g_k is the gradient of $f(x)$ at point x_k. By considering the step-size to be taken to the limit $\alpha_k \to 0^+$, equation (3) can be transcribed into the following ODE system [5]:

$$\frac{dx(t)}{dt} = -g(x(t)); \quad x(t_0) = x_0 \tag{4}$$

This results in an algorithm that can make use of state-of-the-art integration software to solve optimization problems. The advantage of using SD is that it always has a descent direction, and the use of the GF form obviates the need for line-search.

3. Transcription of Systems of Index-1 Via Least-Squares Embedding

The work presented here is based on the idea of embedding via Gradient Flow (GF) a suitable least squares problem into the ODE part of semi-explicit index-1 DAE systems.

3.1. Implicit Index-1 System

We consider in this work only index-1 systems of DAEs. The general system is given by:

$$f(\dot{x}(t), x(t), y(t), t) = 0 \tag{5a}$$

$$B(\dot{x}(t_0), x(t_0), y(t_0)) = 0 \tag{5b}$$

$$t_0 \leq t \leq t_f$$

where x, $\dot{x} \in \mathbb{R}^{N_x}$ and $y \in \mathbb{R}^{N_y}$ have the same meaning as before.

Equation (5b) introduces generalized initial conditions (consistent and non-redundant, see *e.g.* [16]). It is noted that for index-1 systems the combined Jacobian matrix $[f_{\dot{x}}, f_y]$ must be invertible [14].

For the system in equation (5a) an equivalent pointwise least squares optimization problem will be introduced by first differentiating once the equation with respect to time, t:

$$f_{\dot{x}}\ddot{x} + f_x\dot{x} + f_y\dot{y} = F\left(\ddot{x}, \dot{x}, \dot{y}, x, y, t\right) = 0 \tag{6}$$

where $\ddot{x}$ is the second time derivative of the differential variables, and $\dot{y}$ is the first time derivative of the the algebraic variables. This allows to extract values for both $\ddot{x}$ and $\dot{y}$. System in equation (5b) can then be replaced by a least squares minimization problem, such that all the constraints in the DAE system are satisfied at every point in time. The form of the optimization problem is the following, which is to be solved for all times t in the integration interval:

$$\min_{\ddot{x},\dot{y}} \phi = \frac{1}{2}F^T F \tag{7a}$$

where

$$\begin{aligned} \phi \quad = \quad & \frac{1}{2}\ddot{x}f_{\dot{x}}^T f_{\dot{x}}\ddot{x} + \frac{1}{2}\dot{x}^T f_x^T f_x\dot{x} + \frac{1}{2}\dot{y}^T f_y^T f_y\dot{y} \\ & + \ddot{x}f_{\dot{x}}^T f_{\dot{x}}\dot{x} + \ddot{x}f_{\dot{x}}^T f_y\dot{y} + \dot{x}f_{\dot{x}}^T f_y\dot{y} \end{aligned} \tag{7b}$$

$$\forall t \in [t_0, t_f]$$

An embedded gradient flow model can be written from the equivalent ODE system, using the steepest descent method with a scaling factor $\mu > 0$, $\mu \in \mathbb{R}$:

$$\frac{dx}{dt} = \dot{x} \qquad x(t_0) = x_0 \tag{8a}$$
$$\frac{dy}{dt} = \dot{y} \qquad y(t_0) = y_0 \tag{8b}$$
$$\frac{d\dot{x}}{dt} = \ddot{x} \qquad \dot{x}(t_0) = \dot{x}_0 \tag{8c}$$
$$\frac{d\ddot{x}}{dt} = -\mu \nabla_{\ddot{x}} \phi \qquad \ddot{x}(t_0) = \ddot{x}_0 \tag{8d}$$
$$\frac{d\dot{y}}{dt} = -\mu \nabla_{\dot{y}} \phi \qquad \dot{y}(t_0) = \dot{y}_0 \tag{8e}$$

where

$$\nabla_{\ddot{x}} \phi = f_{\dot{x}}^T f_{\dot{x}} \ddot{x} + f_{\dot{x}}^T f_x \dot{x} + f_{\dot{x}}^T f_y \dot{y} \tag{8f}$$
$$\nabla_{\dot{y}} \phi = f_y^T f_{\dot{x}} \ddot{x} + f_y^T f_x \dot{x} + f_y^T f_y \dot{y} \tag{8g}$$

In all of the above x_0, y_0 and $\dot{x}_0$ are consistent initial conditions that must satisfy the following combined system

$$f\left(\dot{x}_0, x_0, y_0\right) = 0 \tag{9a}$$
$$B\left(\dot{x}_0, x_0, y_0\right) = 0 \tag{9b}$$

Once $\dot{x}_0$ is found, then $\ddot{x}_0$ and $\dot{y}_0$ are found by solving at initialization the linearised system obtained from equations (9a) and (9b), given by:

$$f_{\dot{x}} \ddot{x}_0 + f_x \dot{x}_0 + f_y \dot{y} = 0 \tag{10a}$$
$$B_{\dot{x}} \ddot{x}_0 + B_x \dot{x}_0 + B_y \dot{y}_0 = 0 \tag{10b}$$

3.2. Semi-Explicit Index-1 System

Although the presentation in the previous subsection is for the most general index-1 system, we focus in this work on the special class of semi-explicit index-1 systems. The reason for this is that for most practical applications this is the most usual form of dynamical models of interest. These systems are described by:

$$\dot{x} = f(x,y,t); \quad x(t_0) = x_0 \tag{11a}$$
$$0 = g(x,y,t) \tag{11b}$$

$$t_0 \leq t \leq t_f$$

For the system to be index-1, the Jacobian matrix g_y must be invertible. This is converted to a model with an embedded pointwise least squares optimization problem as follows:

$$\dot{x} = f(x,y,t); \quad x(t_0) = x_0 \tag{12a}$$
$$y = \arg\min \phi(x,y,t) = \frac{1}{2} g(x,y,t)^T g(x,y,t) \tag{12b}$$

$$t_0 \leq t \leq t_f$$

The embedding is transformed from a least squares problem to a pure ODE system via the steepest descent method using a scaling factor $\mu > 0,\ \mu \in \mathbb{R}$ according to:

$$\dot{x} = f(x,y,t); \quad x(t_0) = x_0 \tag{13a}$$
$$\dot{y} = -\mu \nabla_y \phi = -\mu g_y(x,y,t)^T g(x,y,t); \quad y(t_0) = y_0 \tag{13b}$$

$$t_0 \leq t \leq t_f$$

where y_0 is either solved from the least squares problem at time t_0, given the values of x_0, or equivalently by solving the original algebraic equations of the system for y_0 at time t_0, *i.e.* $g(x_0, y_0, t_0) = 0$.

4. Convergence Analysis

4.1. Convergence in the limit of $\mu \to +\infty$

Theorem 1. *Given an original linear time-invariant index-1 DAE system, the solution of a system where the algebraic equations are formulated by an embedded least squares optimization (LSQR) problem* $(x(t), y(t))$

converges to the solution of the original system $(x_{true}(t), y_{true}(t))$ *in the limit where* $\mu \to +\infty$ *for* $t_0 \leq t \leq t_f$.

Proof. Eigenvalue analysis is presented here to derive properties of when the solution of the LSQR *completion* of DAEs converges to their exact solution, as proposed in the work overall.

We consider a completion of the system:

$$\dot{x} = f(x, y) \tag{14}$$

$$0 = g(x, y) \tag{15}$$

using a steepest descent embedding as outlined earlier in equations (13a) and (13b). This leads in general to the scaled system, with arbitrary positive parameter $\mu \in \mathbb{R}^1$, $\mu > 0$:

$$\dot{x} = f \tag{16}$$

$$\dot{y} = -\mu \cdot (g_y)^T \cdot g \tag{17}$$

where arguments for the functions have been dropped, for simplicity.

We next consider the case of a linear autonomous index-1 DAE system as below:

System 1

$$\dot{x} = Ax + By + \rho \tag{18}$$

$$0 = Cx + Dy + \sigma \tag{19}$$

where according to the above, matrix $D \in \mathbb{R}^{n_y \times n_y}$ is invertible. Matrix $A \in \mathbb{R}^{n_x \times n_x}$ is square, and rectangular matrices $B \in \mathbb{R}^{n_x \times n_y}$ and $C \in \mathbb{R}^{n_y \times n_x}$ to yield a complete DAE system in the x and y state variables. Finally, ρ and σ are constant vectors of dimensions n_x and n_y, respectively.

Eigenvalue analysis requires the use of a matrix pencil (generalised eigenvalue analysis) as given below:

$$\det \left[\begin{pmatrix} A & B \\ C & D \end{pmatrix} - \lambda \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \right] = 0 \tag{20}$$

where the identity matrix has appropriate dimensions so that the overall matrices subtracted in the determinant are square. The above simplifies to solving the following generalized eigenvalue equation:

$$\det\left[\begin{pmatrix} A-\lambda I & B \\ C & D \end{pmatrix}\right] = 0 \tag{21}$$

Using the property that determinants of a (block) matrix are not influenced by linear combinations of columns or rows, the following holds true [18]:

$$\det\left[\begin{pmatrix} A-\lambda I & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} I & 0 \\ -D^{-1}C & I \end{pmatrix}\right] = 0 \quad \Rightarrow$$

$$\det\left[\begin{pmatrix} A-\lambda I - BD^{-1}C & B \\ 0 & D \end{pmatrix}\right] = 0 \tag{22}$$

By the property of triangular block matrices it holds that:

$$\det\left[\begin{pmatrix} A-\lambda I & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} I & 0 \\ -D^{-1}C & I \end{pmatrix}\right] = \det\left[A-\lambda I - BD^{-1}C\right] \cdot \det\left[D\right] \tag{23}$$

Hence, substituting into the previous equation, the eigenvalue equation becomes (by dropping the constant value $\det[D]$):

$$\det\left[A-\lambda I - BD^{-1}C\right] = 0 \tag{24}$$

The application of the scaled steepest descent least squares completion yields:

System 2

$$\dot{x} = Ax + By + \rho \tag{25}$$

$$\dot{y} = -\mu\left(D^T C\right)x - \mu\left(D^T D\right)y - \mu D^T \sigma \tag{26}$$

Eigenvalue analysis for this system yields the following eigenvalue equation:

$$\det\left[\begin{pmatrix} A & B \\ -\mu D^T C & -\mu D^T D \end{pmatrix} - \lambda \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}\right] = 0 \tag{27}$$

where the identity matrices have appropriate dimensions so that the overall matrices subtracted in the determinant are square. The above becomes:

$$\det\left[\begin{pmatrix} A-\lambda I & B \\ -\mu D^T C & -\mu D^T D-\lambda I \end{pmatrix}\right]=0 \tag{28}$$

The determinant above is equivalent, by using again block matrix determinant properties, to the following:

$$\begin{gathered}\det\left[\begin{pmatrix} A-\lambda I & B \\ -\mu\left(DC\right) & -\mu D^T D-\lambda I \end{pmatrix}\right]= \\ \det\left[\begin{pmatrix} A-\lambda I & B \\ -\mu D^T C & -\mu D^T D-\lambda I \end{pmatrix}\cdot\begin{pmatrix} I & 0 \\ -\left(-\mu D^T D-\lambda I\right)^{-1}\mu D^T C & I \end{pmatrix}\right]\end{gathered} \tag{29}$$

The lower left block in the second matrix in the determinant becomes:

$$-\left(-\mu D^T D-\lambda I\right)^{-1}\mu D^T C \quad = \quad \left(\mu D^T D+\lambda I\right)^{-1}\mu D^T C \tag{30}$$

With further algebraic manipulation the determinant equation becomes:

$$\det\left[\begin{pmatrix} A-\lambda I-B\left(\mu D^T D+\lambda I\right)^{-1}\mu D^T C & B \\ 0 & -\left(\mu D^T D+\lambda I\right) \end{pmatrix}\right]=0 \tag{31}$$

Using the properties of determinants for block matrices it is finally obtained that:

$$\det\left[A-\lambda I-B\left(\mu D^T D+\lambda I\right)^{-1}\mu D^T C\right]\cdot\det\left[-\left(\mu D^T D+\lambda I\right)\right]=0 \tag{32}$$

The product results in two polynomial terms multiplying each other, and thus the eigenvalues of the system are determined by two independent eigenvalue equations:

$$\det\left[A-\lambda I-B\left(\mu D^T D+\lambda I\right)^{-1}\mu D^T C\right]=0 \tag{33}$$

$$\det\left[-\left(\mu D^T D+\lambda I\right)\right]=0 \tag{34}$$

The second eigenvalue equation (34) determines the eigenvalues of the system due to the inclusion of the embedding of the LSQR approach via the steepest descent method. It can be seen that its dynamics can be made as fast as required by increasing the value of μ.

The first eigenvalue equation (33) is more complicated and requires some further manipulation. The

inverse matrix term in the parenthesis of the first equation, by multiplying internally (taking out a common factor $D^T D$), can be re-written as:

$$\begin{aligned}\left(\mu D^T D + \lambda I\right)^{-1} &= \left(D^T D \cdot \left(I + \frac{\lambda}{\mu} D^{-1} D^{-T}\right)\right)^{-1} \frac{1}{\mu} \\ &= \left(I + \frac{\lambda}{\mu} D^{-1} D^{-T}\right)^{-1} D^{-1} D^{-T} \frac{1}{\mu} \end{aligned} \tag{35}$$

Taking this result into the first eigenvalue equation above yields:

$$\det\left[A - \lambda I - B\left(I + \frac{\lambda}{\mu} D^{-1} D^{-T}\right)^{-1} D^{-1} C\right] = 0 \tag{36}$$

Some rearrangement yields:

$$\det\left[A - \lambda I - B\left(I + \lambda \frac{1}{\mu}\left(D^T D\right)^{-1}\right)^{-1} D^{-1} C\right] = 0 \tag{37}$$

It can be seen that if $\mu \to +\infty$, then the second eigenvalue system yields infinitely fast dynamics (adjustment of the y variables in time), while the first system is uninfluenced and yields the correct eigenvalues as for the unmodified (by embedding) case.

The above constitutes a theoretical proof that in the limit, as $\frac{1}{\mu}\left(D^T D\right)^{-1} \to 0$ for $\mu \to \infty$, the embedded problem trajectory converges to that of the true solution for the variables x and y.

It is finally noted that the least squares matrix $D^T D$ is full rank by the fact that matrix D is invertible (full rank Jacobian matrix with respect to variables y of the algebraic equations). □

4.2. Proof of ε-convergence

All proofs in this section use as norm the Euclidean norm (norm-2).

Lemma 1. *Norm of matrix-vector product of a real symmetric matrix G with a real vector ψ.*

Proof. Assume $G \in \mathbb{R}^n$ is a symmetric positive definite matrix with eigenvalues λ_i and eigenvectors ξ_i, for $i = 1, 2, \ldots, n$. By the definition of G, $\lambda_i \in \mathbb{R}$ and $\lambda_i > 0$ and $\xi_i \perp \xi_j$ for $i \neq j$. Under these conditions the eigenvectors can be chosen as orthonormal with $\|\xi_i\| = 1$. It is further assumed that the eigenvalues are ordered such that $\lambda_i \geq \lambda_{i-1}$.

We can represent a vector $\psi \in \mathbb{R}^n$ as $\psi = \sum_{i=1}^n \alpha_i \xi_i$, with $\|\psi\|^2 = \sum_{i=1}^n \alpha_i^2$. Also $G\psi = \sum_{i=1}^n \alpha_i \lambda_i \xi_i$ with $\|G\psi\|^2 = \|\sum_{i=1}^n \alpha_i \lambda_i \xi_i\|^2 = \sum_{i=1}^n \alpha_i^2 \lambda_i^2$. □

Given $\varepsilon > 0$ there exists a $\underline{\mu} > 0$ such that for any $\mu \geq \underline{\mu}$ the solution of the embedded system, $(x(t), y(t))$, for $t \in [t_0, t_f]$, converges to that of the original system $(x_{\text{true}}(t), y_{\text{true}}(t))$ according to $\|x_2(t) - x_1(t)\| \leq \varepsilon$ and $\|y_2(t) - y_1(t)\| \leq \varepsilon$.

Proof. Original system:

$$\dot{x} = Ax + By + \rho \tag{38a}$$

$$0 = Cx + Dy + \sigma \tag{38b}$$

Gradient Flow formulation:

$$\dot{x} = Ax + By + \rho \tag{39a}$$

$$\dot{y} = -\mu D^T(Cx + Dy + \sigma) \tag{39b}$$

Considering that equations (38a) and (38b) of the original system yield the true solution $(x_{\text{true}}, y_{\text{true}})$ and that equations (39a) and (39b) from the GF formulation yield a solution (x, y) (the time argument, t, is omitted from x and y for simplicity in the derivations below) by replacing the solution of x in constraint (38b) we can obtain a value $\tilde{y}$, so that:

$$0 = Cx + D\tilde{y} + \sigma \tag{40}$$

with

$$\tilde{y} = -D^{-1}(Cx + \sigma) \tag{41}$$

It can be seen that $\tilde{y}$ is the value of y that satisfies the algebraic equations (40) for any value of x. In this way, a measure w for a deviation from the solution of the GF formulation constraint can be obtained as:

$$w = y - \tilde{y} \tag{42}$$

By replacing in equation (42)

$$w = y + D^{-1}(Cx + \sigma) \tag{43}$$

and

$$y = w - D^{-1}(Cx + \sigma) \tag{44}$$

By differentiating equations (42) and (41) it is obtained

$$\dot{w} = \dot{y} - \dot{\tilde{y}} \tag{45}$$

and

$$\dot{\tilde{y}} = -D^{-1}C\dot{x} \tag{46}$$

By considering equation (39b), equation (45) yields:

$$\dot{w} = -\mu D^T(Cx + Dy + \sigma) + D^{-1}C\dot{x} \tag{47}$$

thus, by substituting $\dot{x}$ from equation (39a)

$$\begin{aligned} \dot{w} &= -\mu D^T(Cx + Dy + \sigma) + D^{-1}C(Ax + By + \rho) \\ &= (D^{-1}CA - \mu D^T C)x + (D^{-1}CB - \mu D^T D)y \\ &\quad +(D^{-1}C\rho - \mu D^T\sigma) \end{aligned} \tag{48}$$

The second term of this equation, $(D^{-1}CB - \mu D^T D)y$, can be rearranged by using equation (44) as

$$\begin{aligned} (D^{-1}CB - \mu D^T D)y &= (D^{-1}CB - \mu D^T D)(w - D^{-1}(Cx + \sigma)) \\ &= (D^{-1}CB - \mu D^T D)w - (D^{-1}CB - \mu D^T D)D^{-1}(Cx + \sigma) \\ &= (D^{-1}CB - \mu D^T D)w - (D^{-1}CBD^{-1} - \mu D^T)Cx \\ &\quad -(D^{-1}CBD^{-1} - \mu D^T)\sigma \end{aligned}$$

Thus by replacing this expression in equation (48)

$$\begin{aligned}
\dot{w} &= (D^{-1}CA - \mu D^T C)x + (D^{-1}CB - \mu D^T D)w \\
&\quad -(D^{-1}CBD^{-1} - \mu D^T)Cx - (D^{-1}CBD^{-1} - \mu D^T)\sigma \\
&\quad +(D^{-1}C\rho - \mu D^T \sigma) \\
&= (D^{-1}CA - D^{-1}CBD^{-1}C)x + (D^{-1}CB - \mu D^T D)w \\
&\quad +(D^{-1}C\rho - D^{-1}CBD^{-1}\sigma)
\end{aligned}$$

which can be written as:

$$\dot{w} = \alpha x + (D^{-1}CB - \mu D^T D)w + \beta \tag{49}$$

where $\alpha = (D^{-1}CA - D^{-1}CBD^{-1}C)$ and $\beta = (D^{-1}C\rho - D^{-1}CBD^{-1}\sigma)$.

On the other hand by replacing equation (44) in equation (39a) results in

$$\begin{aligned}
\dot{x} &= Ax + B(w - D^{-1}(Cx + \sigma)) + \rho \\
&= (A + BD^{-1}C)x + Bw + (\rho - BD^{-1}\sigma)
\end{aligned} \tag{50}$$

Now if we assume that at some $t = t_1$, $t_1 \in [t_0, t_f]$ with $w = \varepsilon_w$, $\|w\| = \|\varepsilon_w\| = \varepsilon_1 > 0$, if $\text{sign}(\varepsilon_w) \circ \dot{w}(t_1) \leq 0$, then $\frac{d}{dt}\|w(t)\| \leq 0$, and $\|w(t)\| \leq \varepsilon_1$ for $t \in [t_1, t_1 + \delta]$, $\delta > 0$ (*i.e.* if $\dot{w}$ forces w towards 0 from either side, the norm of w decreases).

In equation (49), the negative term, $-\mu D^T D w$ by positive definitiveness of $D^T D$, will always force w to zero and the terms $\alpha x + D^{-1}CBw + \beta$ may force w away from zero, so that, if $\left\|-\mu D^T D w\right\| \geq \left\|\alpha x + D^{-1}CBw + \beta\right\|$, then $\text{sign}(\varepsilon_w) \circ \dot{w}(t_1) \leq 0$.

Thus from Lemma 1:

$$\left\|-\mu D^T D w\right\| = \mu \left\|D^T D w\right\| \geq \mu \lambda_{\min}(D^T D) \|w\| = \mu \lambda_{\min}(D^T D)\varepsilon_1$$

We also have

$$\|\alpha\| \|x\| + \left\|D^{-1}CB\right\| \varepsilon_1 + \|\beta\| \geq \left\|\alpha x + D^{-1}CB\varepsilon_w + \beta\right\| \tag{51}$$

From the above if

$$\mu\lambda_{\min}(D^TD)\varepsilon_1 \geq \|\alpha\|\,\|x\| + \left\|D^{-1}CB\right\|\varepsilon_1 + \|\beta\|$$

then

$$\left\|-\mu D^TDw\right\| \geq \left\|\alpha x + D^{-1}CBw + \beta\right\|$$

where

$$\mu \geq \frac{\|\alpha\|\,\|x\| + \left\|D^{-1}CB\right\|\varepsilon_1 + \|\beta\|}{\lambda_{\min}(D^TD)\varepsilon_1}$$

Now, assume $\|x\| \leq x_{max} \quad \forall t \in [t_1, t_f]$, $\text{sign}(w) \circ \dot{w}(t) \leq 0$, $\forall t \geq t_1$, and

$$\mu \geq \frac{\|\alpha\|\,x_{max} + \left\|D^{-1}CB\right\|\varepsilon_1 + \|\beta\|}{\lambda_{\min}(D^TD)\varepsilon_1} \tag{52}$$

From this equation

$$\varepsilon_1 \leq \frac{\|\alpha\|\,x_{max} + \|\beta\|}{\mu\lambda_{\min}(D^TD) - \|D^{-1}CB\|} \tag{53}$$

which yields a minimum necessary condition for μ

$$\mu > \frac{\left\|D^{-1}CB\right\|}{\lambda_{\min}(D^TD)} \tag{54}$$

Considering the following

$$\max_{t\geq t_1} \|w(t)\| = \varepsilon_1$$

if $t_1 = 0$, then ε_1 is the maximum error in feasibility over the whole integration time span.

On the other hand, by replacing equation (41) in equation (38a) results in

$$\begin{aligned}
\dot{x}_{true} &= Ax_{true} + B\tilde{y} + \rho \\
&= Ax_{true} + B(-D^{-1}(Cx_{true} + \sigma)) + \rho \\
&= (A - BD^{-1}C)x_{true} + (\rho - BD^{-1}\sigma)
\end{aligned}$$

which by subtracting from equation (39a) yields

$$\dot{x} - \dot{x}_{true} = (A - BD^{-1}C)(x - x_{true}) + Bw$$

Let $\varepsilon_x = x - x_{true}$, and $\varepsilon_2 = \|\varepsilon_x\|$, so that

$$\dot{\varepsilon}_x = (A - BD^{-1}C)\varepsilon_x + Bw$$

$$\|\dot{\varepsilon}_x\| \leq \left\|A - BD^{-1}C\right\| \|\varepsilon_x\| + \|B\| \|w\| \leq \left\|A - BD^{-1}C\right\| \varepsilon_2 + \|B\| \varepsilon_1$$

(N.B. $\varepsilon_1 = \max_t \|w\|$)

Since

$$\frac{d}{dt} \|\varepsilon_x\| \leq \left\| \frac{d\varepsilon_x}{dt} \right\|$$

and for μ as specified in equation (52), the previous expression becomes

$$\dot{\varepsilon}_2 \quad \leq \quad \left\|A - BD^{-1}C\right\| \varepsilon_2 + \|B\| \varepsilon_1$$

by integrating both sides of the equality above, it is obtained

$$\varepsilon_2 \quad \leq \frac{\|B\|\varepsilon_1}{\|A-BD^{-1}C\|} \quad \left(e^{\left\|A-BD^{-1}C\right\|(t-t_0)} - 1\right) + \varepsilon_2(t_0)e^{\left\|A-BD^{-1}C\right\|(t-t_0)}$$

which by further expanding the upper bound on the previous inequality results in

$$\varepsilon_2 \quad \leq \frac{\|B\|\varepsilon_1}{\|A-BD^{-1}C\|} \quad \left(e^{\left\|A-BD^{-1}C\right\|(t_f-t_0)} - 1\right) + \varepsilon_2(t_0)e^{\left\|A-BD^{-1}C\right\|(t_f-t_0)}$$

Further assuming that both the original system and the embedded system (GF formulation) are started from the same initial condition, $x(t_0) = x_0$, then $\varepsilon_2(t_0) = 0$ which results in:

$$\varepsilon_2 \quad \leq \frac{\|B\|\varepsilon_1}{\|A-BD^{-1}C\|} \quad \left(e^{\left\|A-BD^{-1}C\right\|(t_f-t_0)} - 1\right)$$

Now let

$$\begin{aligned}\varepsilon_y &= y - y_{true} \\ &= w + \tilde{y} - y_{true} \\ &= w - D^{-1}(Cx + \sigma) + D^{-1}(Cx_{true} + \sigma) \\ &= -D^{-1}C(x - x_{true}) + w\end{aligned}$$

$$\varepsilon_3 = \|\varepsilon_y\| \leq \left\|D^{-1}C\right\| \|\varepsilon_x\| + \|w\| \leq \left\|D^{-1}C\right\| \varepsilon_2 + \varepsilon_1$$

$$\varepsilon_3 \leq \left\|D^{-1}C\right\| \varepsilon_2 + \varepsilon_1 = \left(\frac{\left\|D^{-1}C\right\| \|B\|}{\|A - BD^{-1}C\|} \left(e^{\|A - BD^{-1}C\|(t_f - t_0)} - 1 \right) + 1 \right) \varepsilon_1$$

From the above it can be seen that within the interval $t \in [t_0, t_f]$ the error in the x and y, respectively given by ε_2 and ε_3, are bounded above by ε_1. This in turn has been shown to be possible to be made arbitrarily small, depending on the choice of scaling factor μ, which concludes the proof for ε-convergence.

□

4.3. Implementation

The ODE systems resulting from the application of the methods proposed in this work were solved in Mathematica® using the ODE capabilities of the NDSolve® framework with default options, and using the Backward Differentiation Formula (BDF) method [11]. The resulting solutions were compared with the direct solution of the DAE problems by NDSolve® in Mathematica®, using again the BDF method and default options. CPU time (seconds), the number of integration steps and the number of evaluations, right hand side of the ODEs made during integration (RHS) and the number of residual evaluations for DAEs, were collected during runs and used as performance indicators.

5. Computational Results

In this section computational case studies are presented to demonstrate the capabilities of the novel Gradient Flow methodology for the solution of DAEs. In the following results "error" refers to the residual of the algebraic equations over the solution of the DAE system.

5.1. DAE reaction system case study

This example presents the case of simple chemical kinetics, and is formulated as a DAE in the following equations:

$$\dot{x}_1 = -r_1; \quad x_1(0) = 1. \tag{55a}$$

$$\dot{x}_2 = r_1 - r_2; \quad x_2(0) = 0. \tag{55b}$$

$$\dot{x}_3 = r_2; \quad x_3(0) = 0. \tag{55c}$$

$$r_1 = 1.0\, x_1 \tag{55d}$$

$$r_2 = 0.25\, x_2 \tag{55e}$$

$$0 \leq t \leq 30$$

Four different levels of the value of the scaling parameter μ are used so as to demonstrate convergence of the method: $\mu = 1, 10^1, 10^2, 10^5$. The numerical solution of the DAE system with transformation is presented in Figures 1–3. Higher values of μ were indistinguishable from $\mu = 10$ so they were omitted for the most part, except for visualization of the residuals. The integration tolerances for the numerical integrator NDSolve® in Mathematica® were set to 10^{-10} for the absolute precision and to the same value for the relative precision. The direct solution of the DAE system is shown in Figure 4. Table 1 summarizes the computational requirements for the five runs (including the solution of the original system).

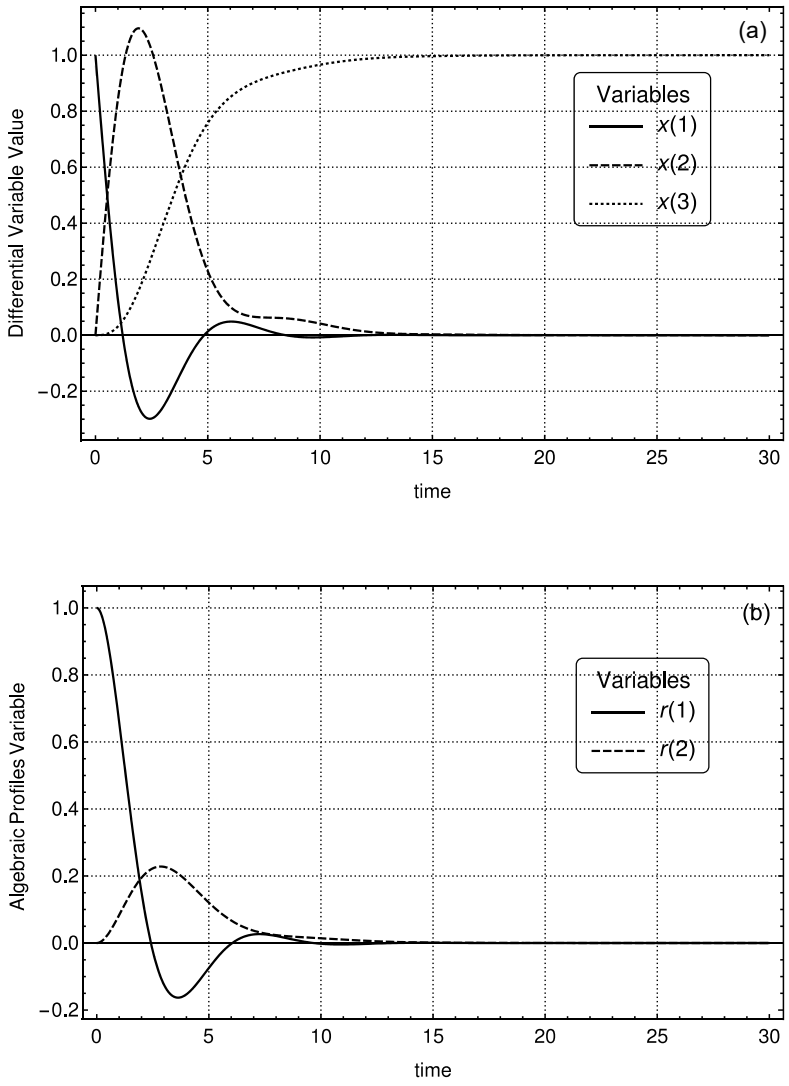

Figure 1: Trajectory for differential state variables in the DAE reaction system case study for $\mu = 1$. Figure (a) shows the molar fractions of compounds 1 to 3 while (b) shows the reaction rates r_1 and r_2.

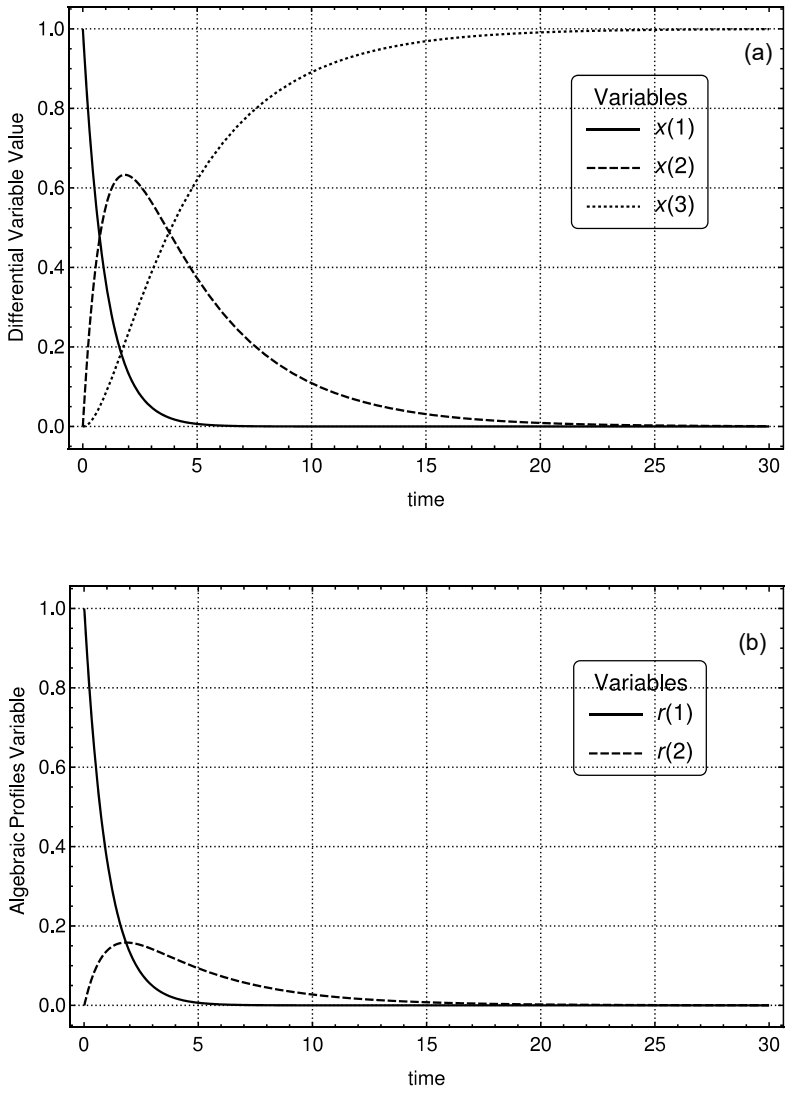

Figure 2: Trajectories for the differential variables, panel (a), and algebraic variables, panel (b), in the DAE reaction system case study when $\mu = 10$.

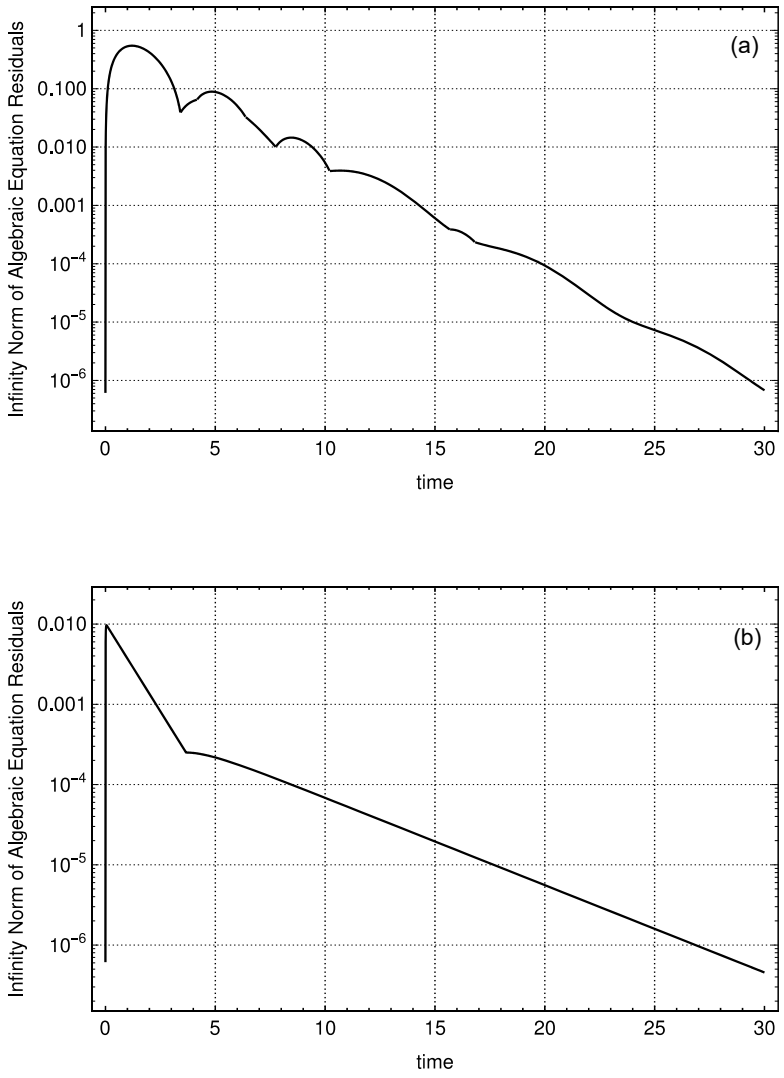

Figure 3: Residual norm of the algebraic equations when the DAE reaction system case study was solved using the method proposed in this work with values of μ set to 1.0, panel (a), and 10.0, panel (b).

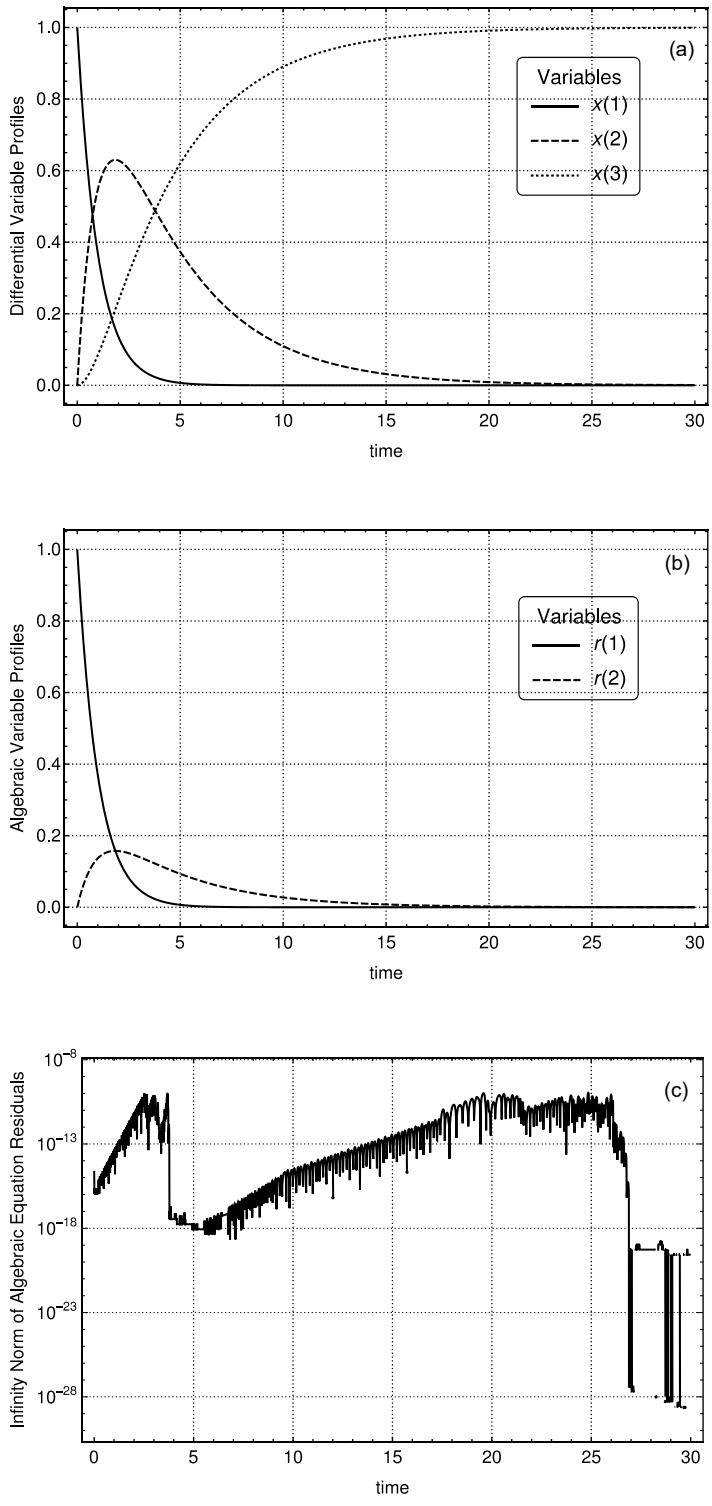

Figure 4: Solution of the problem posed in the DAE reaction system case study using NDSolve® in Mathematica®. Values of the differential and algebraic variables are shown in panels (a) and (b). Panel (c) displays the residual norm of the algebraic equations.

Table 1: DAE reaction system case study: computational run indicators for different μ values

Run	μ	CPU time (s)	# Integration steps	# RHS evaluations
1	1	0.04	432	903
2	10^1	0.06	662	1285
3	10^2	0.06	402	692
4	10^5	< 0.04	381	618
5	Direct solution of DAE system	0.3	525	694

As can be seen from the results presented in Figures 2 to 4 the strategy presented in this work is able to solve successfully this example system. In particular, the accuracy in terms of the satisfaction of the algebraic equations in the DAE system, as shown in Figure 3, is very acceptable for a value of $\mu \geq 10^1$. The size of the algebraic equations residual is larger at the beginning and is directly proportional to the value of μ. It is notable that as the size of the scaling parameter increases, the number of right hand side (RHS) evaluations and integration steps are reduced, while CPU time remains on the same approximate levels.

5.2. Discharge pressure valve case study

This example models a system consisting of a compressor with a suction-throttle valve to control discharge pressure [10]. This system is described by the following non-dimensional equations:

$$\dot{x}_1 = \frac{1}{20}\left(y_1 - x_1\right) \tag{56a}$$

$$\dot{x}_2 = -\frac{1}{75}\left(y_2 - 99.1\right) \tag{56b}$$

$$\dot{x}_3 = \left(y_4 - y_5\right) \tag{56c}$$

$$y_1 = x_2 - \frac{1}{15} y_2 \tag{56d}$$

$$y_2 = y_3\left(3.35 - 0.075\ y_5^2\right) \tag{56e}$$

$$y_3 = \frac{x_3}{20} \tag{56f}$$

$$y_4 = \sqrt{\Psi(t)} \tag{56g}$$

$$y_5 = \Big(23090.3\left(8.03494 \times 10^{10}\ \exp\left(-3t\right) + 7452.56\ \exp\left(-t\right) + 3.00391 \times 10^{-5}\right) \operatorname{sech}(10-t)^3\ x_1 + y_4\, x_1^3\, x_3 - 347.222(3 - \sqrt{\Psi(t)})^2 x_4\Big)\ \frac{1}{x_3\, x_1^3} \tag{56h}$$

$$\Psi(t) = \left(15 + 5\ \tanh\left(t - 10\right)\right)^2 \tag{56i}$$

$$0 \leq t \leq 50$$

Table 2: Initial values for discharge pressure valve system

Variable	Value	Variable	Value
x_1	0.25	x_2	6.857
x_3	734.0	y_1	0.3078
y_2	98.23	y_3	36.70
y_4	10.0	y_5	2.996

The integration horizon was $t \in [0, 50]$ with initial values presented in Table 2, including consistent values for the algebraic variables. To investigate the effect of the value of μ on the convergence of the method, four different values of this scaling parameter were used: $\mu = 10^2, 10^3, 10^4, 10^5$. Figures 5–11 show the solution obtained with the method introduced in this work. Again, most graphs with $\mu \geq 10^2$ were omitted. This time the residuals were also omitted as the only thing that changed between one graph and the other was the magnitude, which scaled down in agreement to the size of μ. The direct solution of the DAE system by the commercial solver NDSolve® in Mathematica® is shown in Figure 8. Table 3 summarizes the computational requirements for the five runs. The integration tolerances for the numerical integrator NDSolve® in Mathematica® were set to 10^{-10} for the absolute precision and to the same value for the relative precision.

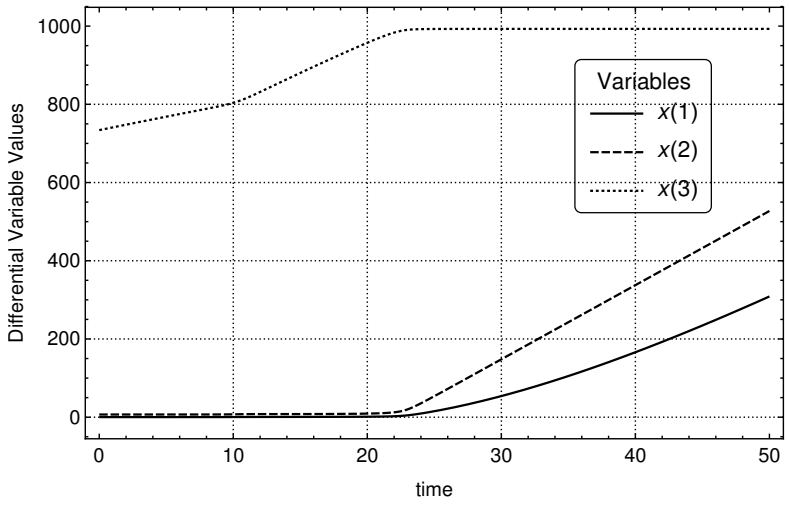

Figure 5: Trajectory for differential state variables in the discharge pressure valve case study when μ was set to 100.

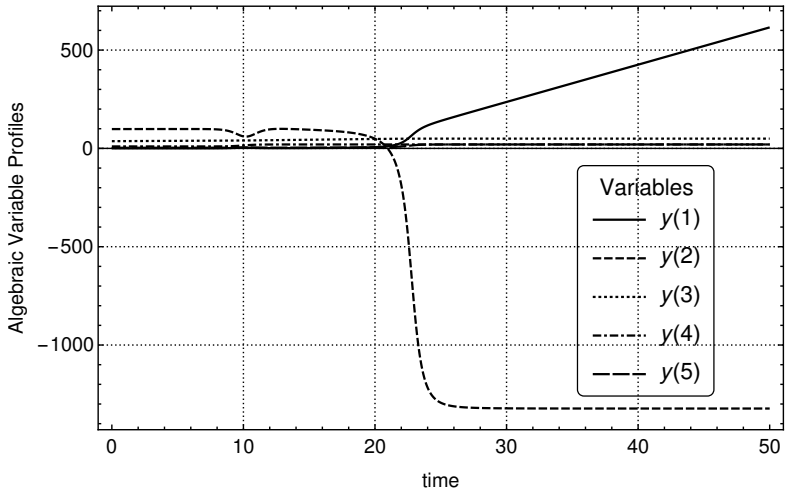

Figure 6: Values of the algebraic state variables along time for the the discharge pressure valve case study. The solution was obtained for $\mu = 100$.

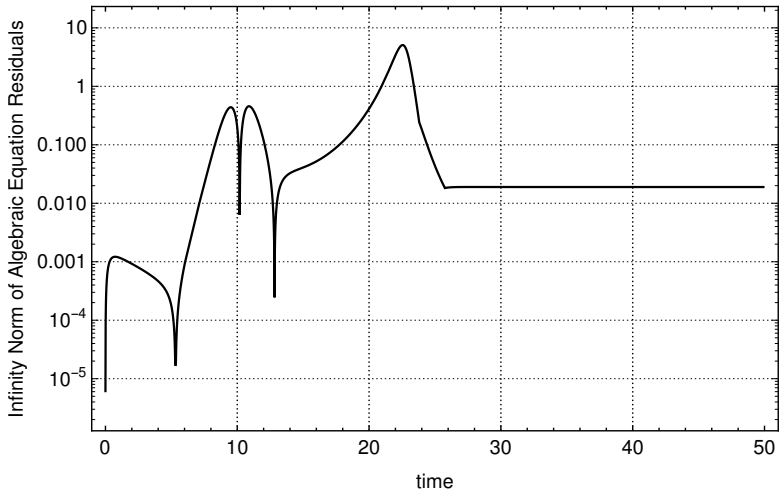

Figure 7: Residual norm of the algebraic equations in the discharge pressure valve case study for $\mu = 100$.

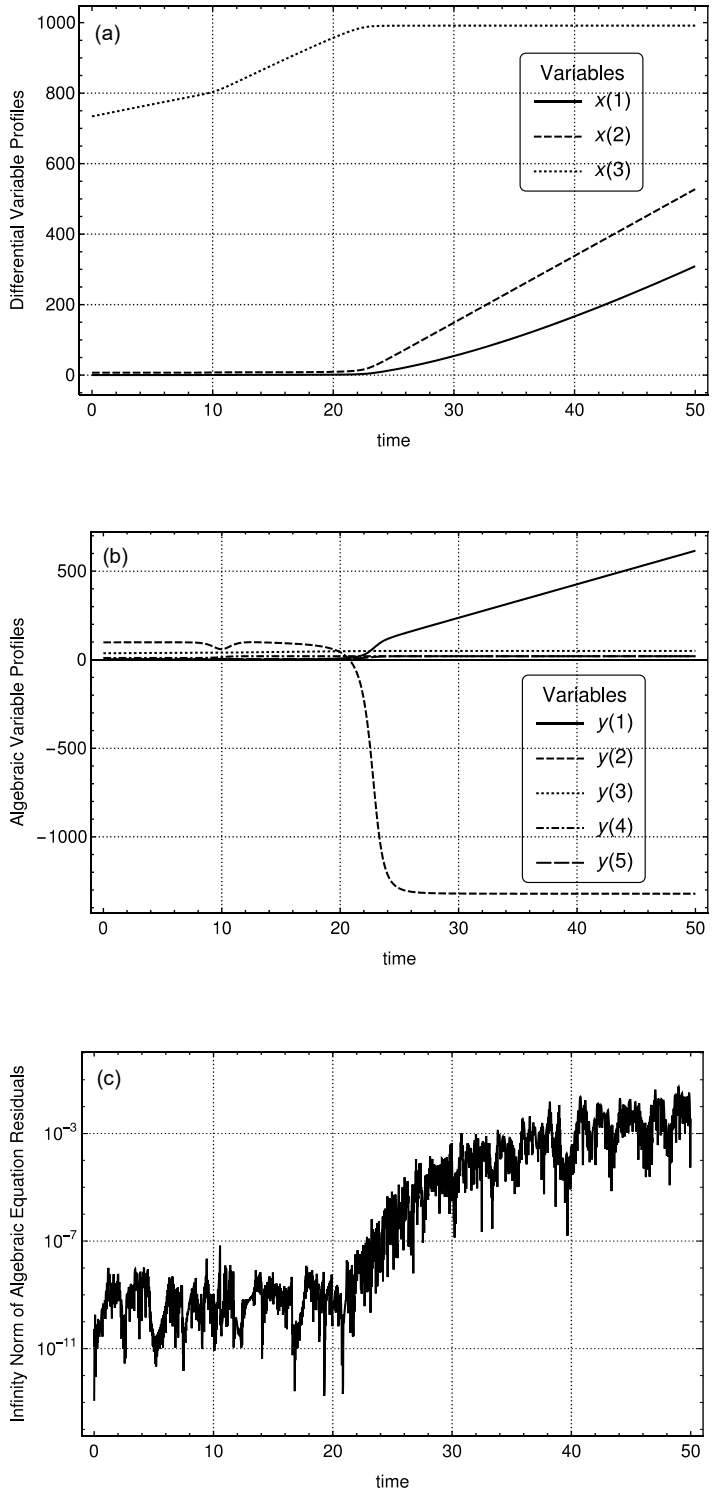

Figure 8: Solution of Example 2 obtained using NDSolve® in Mathematica®. Panels (a), (b) and (c) shows the trajectories of differential, state variables and the norm of the algebraic equations, respectively.

As shown in Figure 7, the infinity norm of the residuals of the algebraic equations is larger for $\mu = 10^2$

Table 3: Discharge pressure valve case study: computational run indicators for different μ values

Run	μ	CPU time (s)	# Integration steps	# RHS evaluations
1	10^2	1.20682	2484	3509
2	10^3	0.777882	4362	9031
3	10^4	9.26759	101514	525981
4	10^5	1.38479	7139	16547
5	Direct solution of DAE system	1.20682	3509	2484

compared to the results obtained for the direct solution of the DAE problem by NDSolve® in Mathematica®. This norm can be reduced to comparable values by increasing the value of the scaling parameter μ if more precision is needed. A trade-off exists between the precision and the time required to obtain a solution as shown in 3, although it is not a linear relationship. Thereby, an adaptive scheme to automatically choose the value of μ by approximating the integration error will be studied in future research.

5.3. Distillation column case study

The following is a DAE system representing a binary distillation column with 41 trays, a condenser and a reboiler. Each tray includes an equilibrium algebraic equation and a dynamic mass balance. The feed tray was selected to be the one in the middle and the feed is saturated liquid. The distillate and bottoms flowrates were fixed and both liquid and vapour flow are considered constant across the column. To make the problem more challenging for solution, the feed composition is changing with time. The dimensionless equations describing the system are the following:

Condenser

$$\frac{dx_C}{dt} = \frac{Vy_1 - (L + Dist)\, x_C}{M_C} \quad c \rightarrow (i = 0) \tag{57a}$$

Rectifying Section

$$\frac{dx_i}{dt} = \frac{Vy_{i+1} + Lx_{i-1} - Vy_i - Lx_i}{M_{Tr}} \quad \forall\, i \in [1, \ldots, 20] \tag{57b}$$

Feed Tray

$$\frac{dx_{21}}{dt} = \frac{Vy_{22} + Fx_{feed} + Lx_{20} - Vy_{21} - (L + F)x_{21}}{M_{Tr}} \tag{57c}$$

Stripping Section

$$\frac{dx_C}{dt} = \frac{Vy_{i+1} + (L + F)x_{i-1} - Vy_i - Lx_i}{M_{Tr}} \quad \forall\, i \in [22, \ldots, 41] \tag{57d}$$

Reboiler

$$\frac{dx_R}{dt} = \frac{Vy_1 + (L + F)x_{41} - Bx_R}{M_R} \quad R \rightarrow (i = 42) \tag{57e}$$

Feed

$$\frac{dx_{feed}}{dt} = -\frac{0.1}{t+1} \quad x_{feed}(0) = 0.8 \tag{57f}$$

Equilibrium

$$y_i \left(1 + (\alpha - 1)\, x_i\right) = \alpha x_1 \quad \forall\, i \in [1, \ldots, 42] \tag{57g}$$

Here x represents the liquid concentration of the component of interest, while y is its vapour concentration. V is the vapour flow and L the liquid flow. F is the feed flow, B the bottoms flow and $Dist$ the distillate flow. Finally α is the relative volatility parameter, which is selected to be low ($\alpha = 3.0$) to make the system more difficult to solve. The integration horizon was selected to be $t \in [0, 50]$. Parameter values are given in Table 4.

Four different levels of the value of the scaling parameter μ are used so as to demonstrate convergence of the method: $\mu = 10^4$, 10^5, 10^6, 10^7. The numerical solution of the DAE system with transformation is presented in Figures 9–11 with dimensionless values. The solution by the commercial solver NDSolve® in Mathematica® is shown in Figure 12. Table 5 summarizes the computational requirements for the five runs. The integration tolerances for the numerical integrator NDSolve® in Mathematica® were set to 10^{-10} for the absolute precision and to the same value for the relative precision.

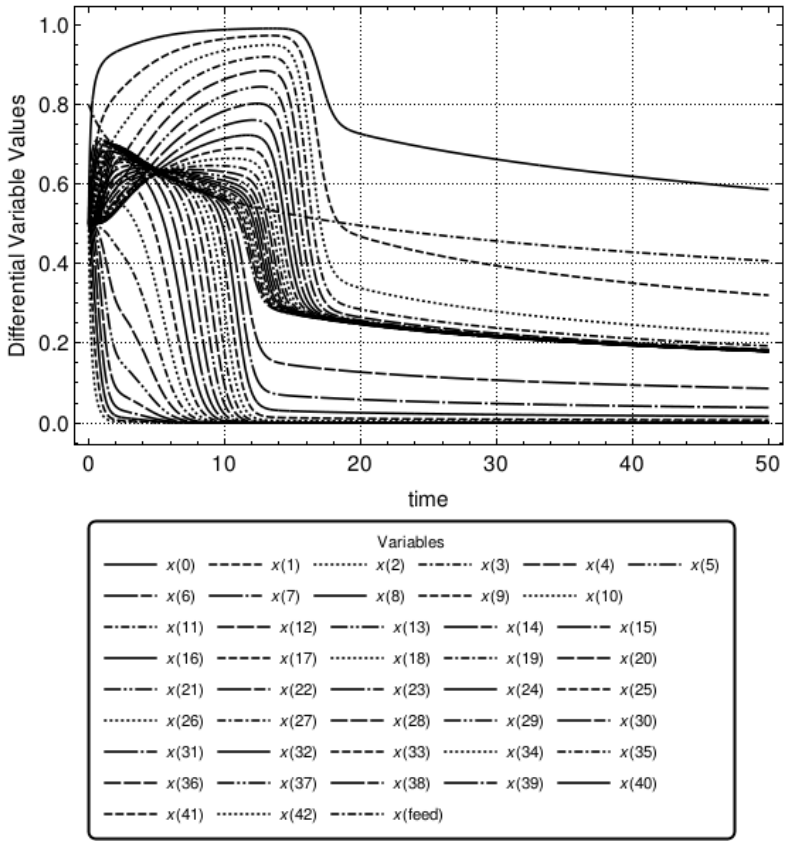

Figure 9: Trajectory for differential state variables, liquid mass fraction of the component of interest for 42 stages, including condenser and reboiler. Calculated for $\mu = 10^4$.

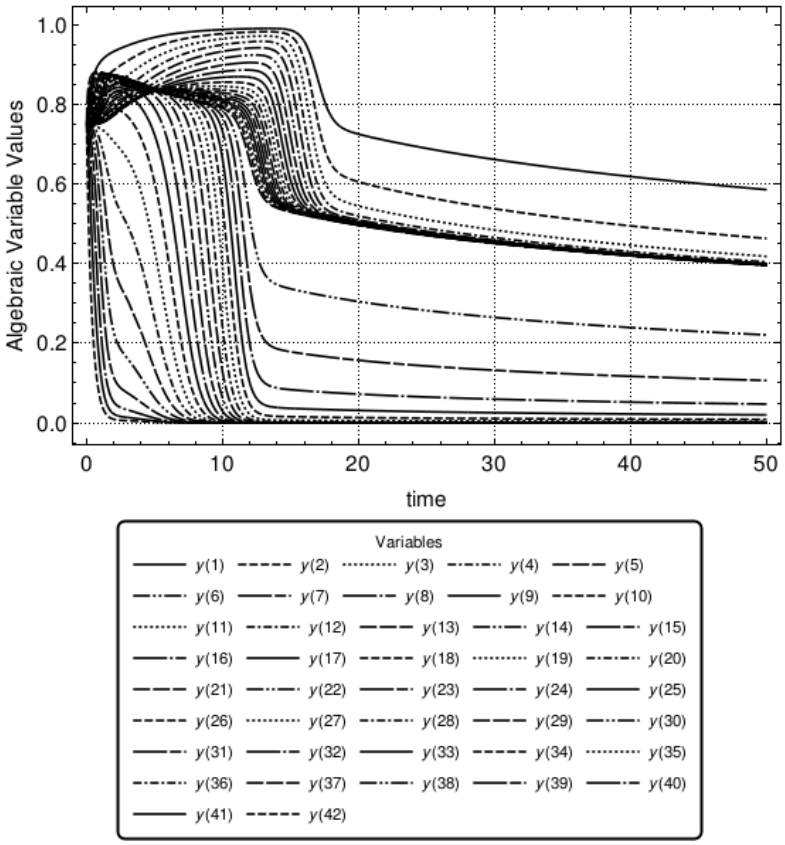

Figure 10: Trajectory for algebraic state variables, vapour mass fraction for the component of interest. Calculated for $\mu = 10^4$.

Table 4: Distillation column case study parameters and initial conditions

Parameter/Variable	Value	Parameter/Variable	Value
L	60.0	α	3.0
V	130.0	$M_{i\in\{C,T_r,R\}}$	10.0
F	100.0	x_{feed}	0.8
$Dist$	70.0	B	30.0
$y(0)_{i\in\{1,\ldots,41\}}$	0.75	$x(0)_{i\in\{0,\ldots,42\}}$	0.5

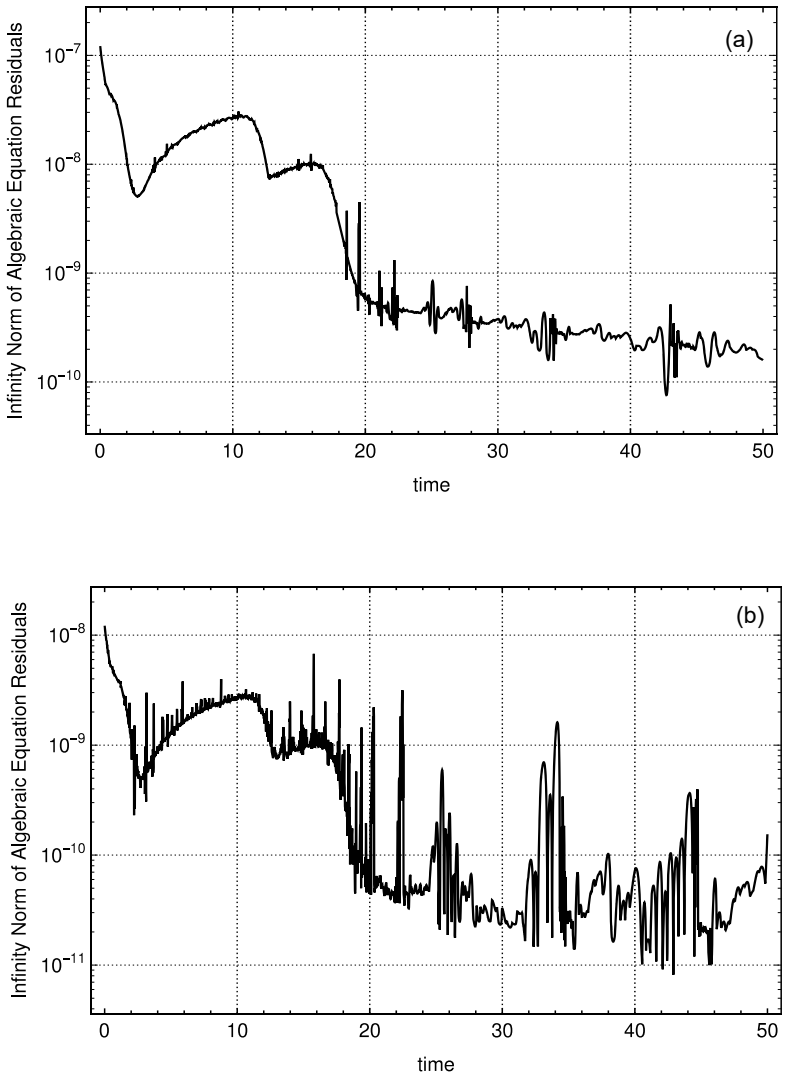

Figure 11: Residual norm of the algebraic equations in the distillation column case study for $\mu = 10^4$ (panel a) and $\mu = 10^7$ (panel b).

From the performance seen in terms of the satisfaction of the algebraic equations in the DAE system, as shown in Figure 11, the accuracy achieved is close to the scale that could be achieved by solving directly the original DAE system with the commercial solver NDSolve®. Even though a higher scaling parameter was

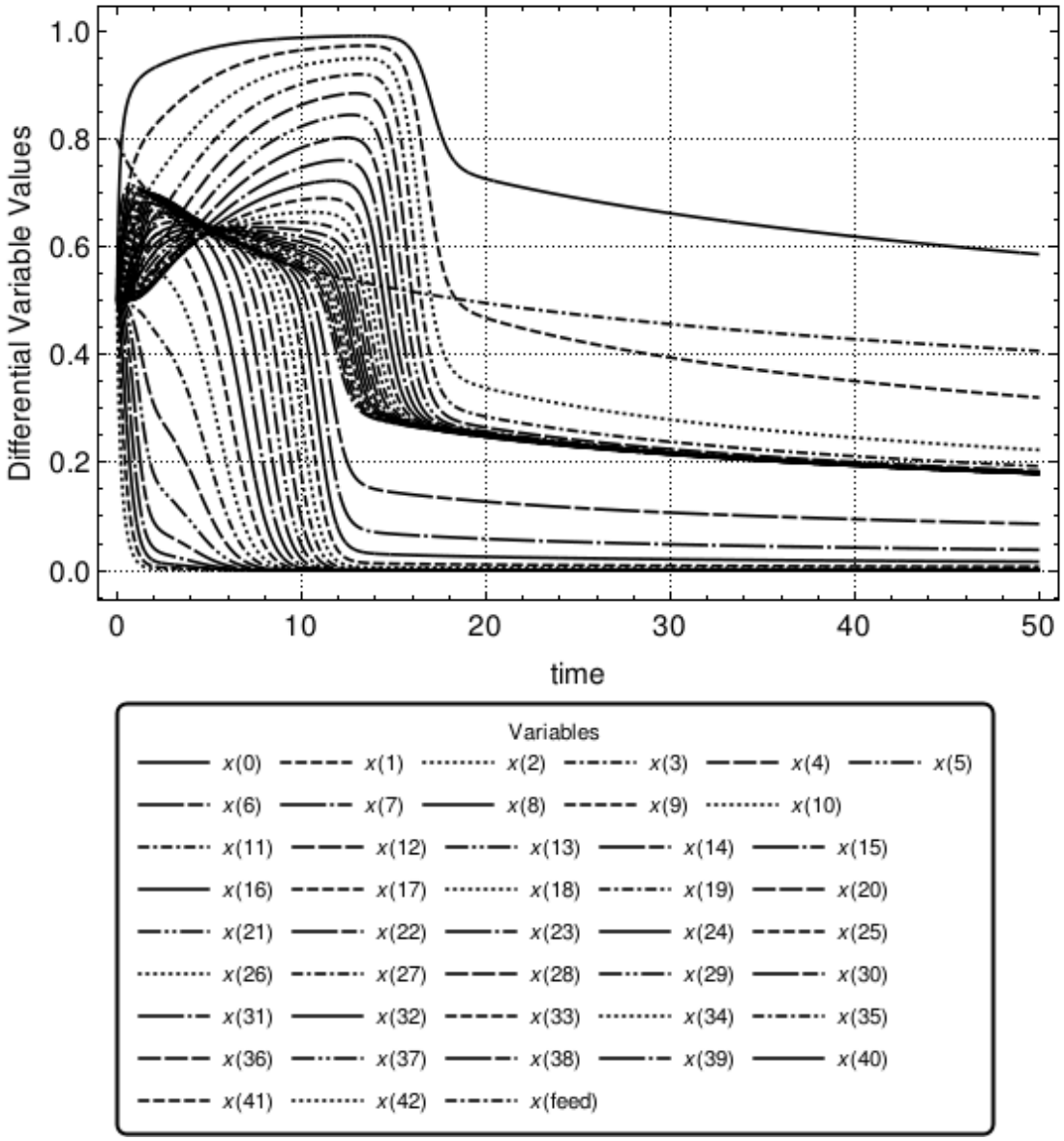

Figure 12: Solution by NDSolve® in Mathematica® of differential variables. $x(0)$ to $x(42)$ represent liquid mass fractions for the stages of the column.

Table 5: Distillation column case study: computational execution indicators for different μ values

Run	μ	CPU time (s)	# Integration steps	# RHS evaluations
1	10^4	8.66	3240	5144
2	10^5	7.48	2914	4751
3	10^6	8.78	2868	4736
4	10^7	8.5657	2957	4974
5	Direct solution of DAE system	14.87	2313	3148

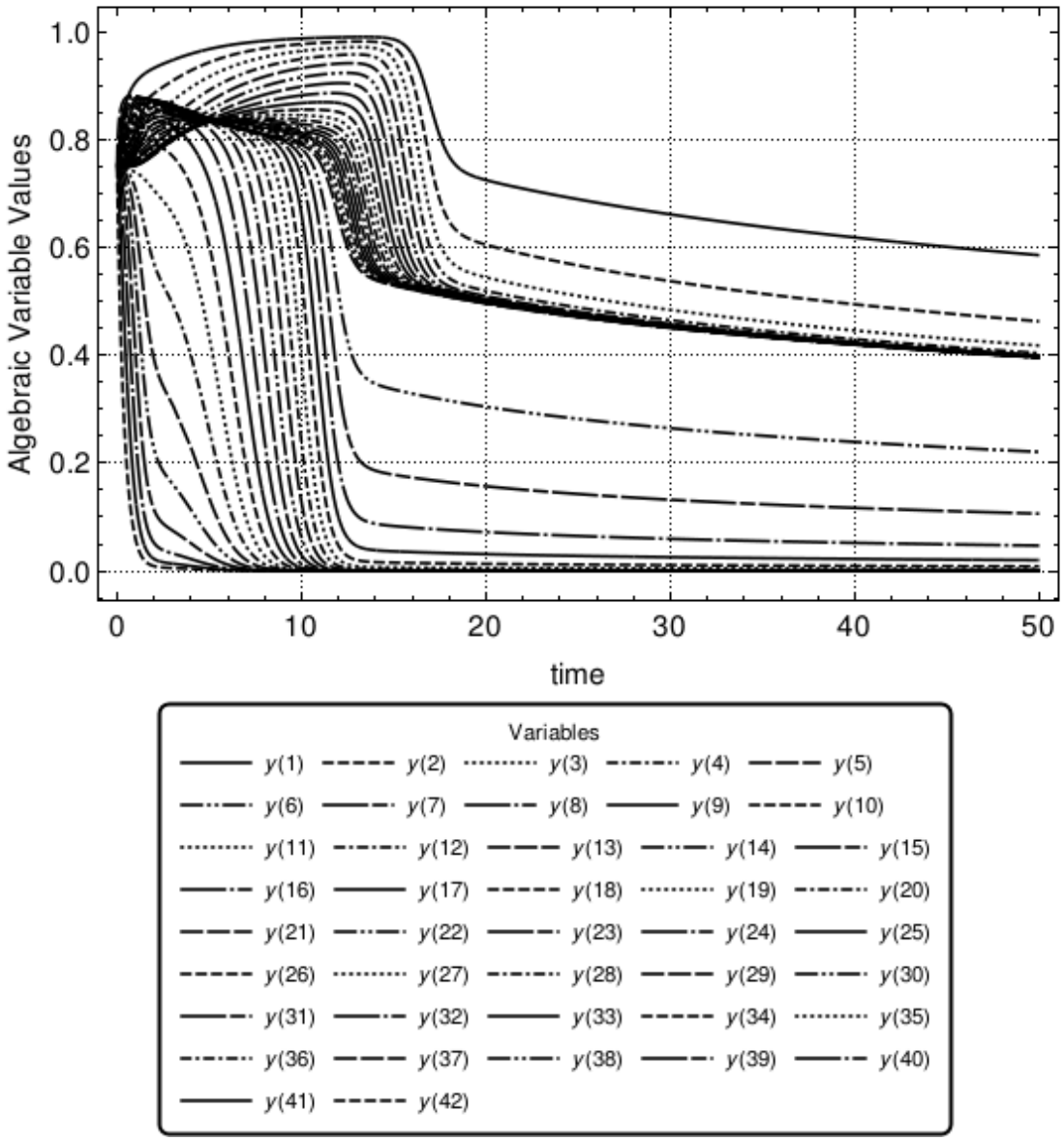

Figure 13: Solution by NDSolve® in Mathematica® of algebraic variables, vapour phase mass fractions in the distillation column case study.

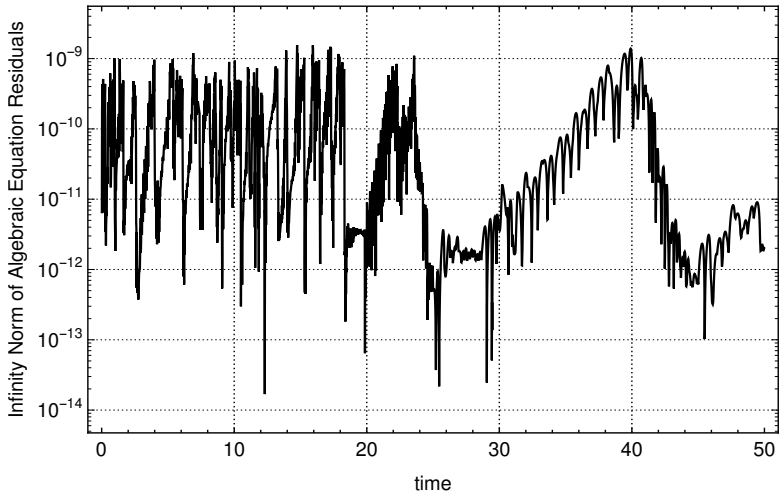

Figure 14: Solution by NDSolve® in Mathematica®: residual norm of the equations for the distillation column case study.

selected at the onset, the error computed by the presented strategy is very low overall.

Furthermore, the error is clearly sensitive to the larger values of μ, as seen in the example. In terms of performance, the system solved with the GF embedding method requires less CPU time than the one solved directly as a DAE by NDSolve®, while maintaining the same quality of solution.

6. Conclusions

In this paper, Gradient Flow methods were employed to solve DAE systems of index-1 by embedding a pointwise optimization problem within the ODE part of the system. A theoretical support of the methodology is also provided, showing both asymptotic convergence in the limit of the scaling scheme introduced, as well as ε-convergence for the case of linear DAE systems.

The computational evidence presented supports both the validity and efficacy of the proposed novel approach. The test problems solved are representative of models encountered in chemical engineering practice. The results indicate that a smaller number of integration steps was on average required for a comparable quality solution to a state-of-the-art commercial DAE solver.

Future research will focus on a number of aspects, such as (a) stepwise adaptation of the scaling parameter μ during the integration process, similar to stepsize selection as implemented in standard integrators. This will be done such that appropriate μ values are determined by the algorithm for the problem under consideration. (b) investigation of alternative optimization embedding schemes to steepest descent, chosen in this work for convenience and as a first attempt, and (c) expansion of applications to larger case studies via the production of a dedicated customized integration solver.

Acknowledgements

Author E. A. del Rio-Chanona would like to acknowledge CONACyT scholarship No. 522530 for funding this project.

References

[1] Ascher, U., Petzold, L. R., 1998. Computer methods for ordinary differential equations and differential-algebraic equations. Society For Industrial & Applied Mathematics, U.S., Philadelphia.

[2] Ascher, U. M., Petzold, L. R., 1993. Stability of Computational Methods for Constrained Dynamics Systems. SIAM Journal on Scientific Computing 14 (1), 95–120.

[3] Barrlund, A., 1991. Constrained least squares methods for linear timevarying DAE systems. Numerische Mathematik 60 (1), 145–161.

[4] Behrman, W., 1998. An Efficient Gradient Flow Method for Unconstrained Optimization. Ph.D. thesis, Stanford University.

[5] Brown, A. A., Bartholomew-Biggs, M. C., 1989. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. Journal of Optimization Theory and Applications 62 (2), 211–224.

[6] Campbell, S. L., Kunkel, P., 2009. Completions of nonlinear DAE flows based on index reduction techniques and their stabilization. Journal of Computational and Applied Mathematics 233 (4), 1021 – 1034.

[7] Chowdhry, S., Krendl, H., Linninger, A. A., 2004. Symbolic Numeric Index Analysis Algorithm for Differential Algebraic Equations. Industrial & Engineering Chemistry Research 43 (14), 3886–3894.

[8] Gear, C. W., 1971. Simultaneous Numerical Solution of Differential-Algebraic Equations. IEEE Transactions on Circuit Theory 18 (1), 89–95.

[9] Hadamard, J., 1908. Mémoire Sur Le Probleme D'analyse Relatif a L'équilibre Des Plaques Élastiques Encastrés. Imprimerie Nationale de France, Paris.

[10] Hairer, E., Roche, M., Lubich, C., 1989. The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods. Vol. 1409 of Lecture Notes in Mathematics. Springer Berlin Heidelberg.

[11] Hairer, E., Wanner, G., 1996. Solving Ordinary Differential Equations II, 2nd Edition. Vol. 14 of Springer Series in Computational Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg.

[12] Kröner, A., Marquardt, W., Gilles, E., 1992. Computing consistent initial conditions for differential-algebraic equations. Computers & Chemical Engineering 16, S131–S138.

[13] Kröner, A., Marquardt, W., Gilles, E., 1997. Getting around consistent initialization of DAE systems? Computers & Chemical Engineering 21 (2), 145–158.

[14] Navarro, A. K. W., Vassiliadis, V. S., 2014. Computer algebra systems coming of age: Dynamic simulation and optimization of DAE systems in Mathematica$^{\text{TM}}$. Computers & Chemical Engineering 62, 125–138.

[15] Okay, I., Campbell, S. L., Kunkel, P., 2007. The additional dynamics of least squares completions for linear differential algebraic equations. Linear Algebra and its Applications 425 (2), 471 – 485.

[16] Pantelides, C. C., March 1988. The Consistent Initialization of Differential-Algebraic Systems. SIAM Journal on Scientific and Statistical Computing 9 (2), 213–231.

[17] Pantelides, C. C., Gritsis, D., Morison, K., Sargent, R., 1988. The mathematical modelling of transient systems using differential-algebraic equations. Computers & Chemical Engineering 12 (5), 449–454.

[18] Powell, P. D., 2011. Calculating determinants of block matrices. arXiv preprint arXiv:1112.4379.

[19] Unger, J., Kröner, A., Marquardt, W., 1995. Structural analysis of differential-algebraic equation systems-theory and applications. Computers & Chemical Engineering 19 (8), 867–882.