Effective Numerical Computation of Parameter Dependent Problems

Lennart Jansen and Caren Tischendorf

Abstract We analyse parameter dependent differential-algebraic-equations (DAEs)

$$Ad'(x,t,p) + b(x,t,p) = 0.$$

For these systems one is interested in the relation between the numerical solutions x and some associated parameters p. The standard approach is to discretise the equations with respect to the parameters and solve the parameter independent equations afterwards. This approach forces a calculation of the differential equations multiple times (for a huge number of parameter values p). This may lead to high computational costs. By using the already computed solutions to calculate the remaining ones and thus exploiting the smoothness of the solution with respect to the parameters, it is possible to save the majority of the computational cost.

1 Introduction

Nowadays parameter dependent problems have their applications in various fields. In the electric circuit simulation a circuit is modeled by a differential-algebraic-equation obtain via a modified-nodal-analysis (MNA) like in [1]. These DAEs depend on the parameters of the many electric parts in the circuit. Since these parts are afflicted with a manufacturing error, one is interested in the relation between small variation in these parameters and the behavior of the circuit.

In modern medicine the effect of the drugs used during a chemo therapy can be described by an ordinary-differential-equation (ODE), on of the models is presented in [2]. To obtain the optimal dosing of the drugs, simulating the therapy many

L. Jansen (✉) and C. Tischendorf
Mathematical Institute, University of Cologne, Eyestalk Weyertal 86-90, 50931 Köln
e-mail: lejansen@math.uni-koeln.de; tischendorf@math.uni-koeln.de

B. Michielsen and J.-R. Poirier (eds.), *Scientific Computing in Electrical Engineering SCEE 2010*, Mathematics in Industry 16, DOI 10.1007/978-3-642-22453-9_6,

times is necessary. Therefore the associated ODE has to be solved for many sets of parameters.

In meteorology the reliability of a weather forecast suffers because of the huge amount of the needed data and the chaotic behavior of the weather. These two problems force the calculation of a big partial-differential-equation(PDE) for many different parameters during the weather forecast.

These problems can be described mathematically by a parameter dependent differential-equation and a set of parameters. The general task is to efficiently solve the equations at all parameters. The main objective of this article is to present a new approach to accelerate this calculation. We will restrict the analysis to DAEs, even so the idea can be transferred to PDEs as well. The improvement will be demonstrated through the comparison of the convergence estimate and an example in the circuit simulation.

This paper is organized as follows. First we describe parameter dependent DAE and state the well-known convergence estimate for a Backward Differentiation Formula Method (BDF). Section 3 is devoted to the presentation of the new solving approach and its numerical results. In Sect. 4 these results are exemplified by a numerical example.

2 Parameter Dependent DAEs

In this chapter we want to introduce the problem in a general setup. Therefore we need to define the structure of the DAE and its properties. Furthermore an associate set of parameters is needed. To combine these two things define a parameter dependent DAE:

Definition 1. Define a semi-linear parameter dependent DAE as followed:

$$Ad'(x,t,p) + b(x,t,p) = 0, \tag{1}$$

with

$$A \in \mathbb{R}^{n\times k},\ d(x,t,p) \in \mathbb{R}^k,\ b(x,t,p) \in \mathbb{R}^n,\ x \in \mathbb{D} \subset \mathbb{R}^n,\ t \in \mathbb{I} \subset \mathbb{R},\ p \in \mathbb{P} \subset \mathbb{R}^l.$$

Recall that $(.)'$ means the total derivative with respect to time, i.e. $\frac{\mathrm{d}}{\mathrm{d}t}(.)$. Furthermore d and b with their partial derivatives d_y, d_x, b_x and b_t are continuous. We call this DAE properly stated, if:

- $\forall\, p \in \mathbb{P}$: $\ker A$ and $\operatorname{im} d_x$ are $C^1 - subspaces$
- $\forall\, p \in \mathbb{P}$: $\ker A \oplus \operatorname{im} d_x(x,t,p) = \mathbb{R}^n\ \ \forall\ y \in \mathbb{R}^n,\ x \in \mathbb{D},\ t \in \mathbb{I}$.

In the following we assume that the DAE is properly stated. The concept of DAEs with properly stated leading terms is described in detail in [5,6]. Also the index of the DAE is limited by two. An DAE index describes the complexity of the structure

of a DAE. There are many ways to define an index, since in our applications there is often little smoothness we refer here to the tractability index.

To solve the parameter problem we need to calculate the numerical solution of the DAE at every single parameter point respectively. For the computation we use a BDF-method and achieve the well-known convergence estimate for the numerical solution of the DAE.

Theorem 1. *Discretise the DAE at a fixed parameter point p_0 with a BDF-method and achieve the following system:*

$$A(\frac{1}{h}\sum_{i=0}^{K}\alpha_i d(x_{n-i}, t_{n-i}, p_0)) + b(x_n, t_n, p_0) = 0. \tag{2}$$

Let h be the constant step size in time. Let the initial steps be sufficiently accurate. Then the error of the numerical solution obtained by solving the BDF-discretised system can be bounded by:

$$\max_{n>\mu K} ||x(t_n) - x_n|| \leq c(h^K + \frac{1}{h^\mu} \max_{0<j<\mu K} ||\delta_{n-j}||) \tag{3}$$

with K the order of the BDF-method, $\mu + 1$ the index DAE and c being a bounded constant.

The proof can be found in [3, 7]. Notice that the error depends on the index. With an index bigger than one the computational error δ of the linear solvers must be guarantied to be small enough in relation to the step size h. The order K of the

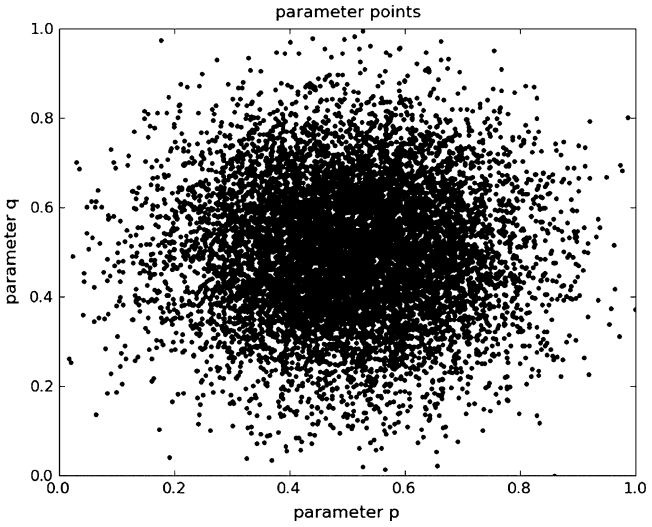

Fig. 1 A set of 10000 random parameter points

BDF-method mostly has to be chosen very small because of the stiffness of the applications.

For better understanding of the next chapter visualize a set of parameter points: Let for example $\mathbb{P} \subset \mathbb{R}^2$ be a set of 10000 random parameter points (Fig. 1).

Due to the applications we are interested in solving the DAE in every of these points to achieve the sensitivity of the solution with respect to the parameters. This may lead to high computational cost. So we need an approach that take advantage of the situation.

3 Parameter-Time-Integrator

This section will be used to present a more efficient approach to calculate the numerical solutions of the DAE at every point of the parameter set. To accelerate the computation of the solutions of the parameter dependent equation we want to take advantage of the solutions which are already calculated. Since the solutions won't be similar to other solutions for big enough differences in the parameters, one has to ensure that the change in the parameters will be sufficiently small. Therefore split the parameter points in sufficiently small packages and observe every package separately. The splitting of the points can depend on external data or it can be implemented in a adaptive way (Fig. 2). Solve the parameter dependent DAE at some parameter points in every single package to obtain interpolation nodes with respect to the parameters. With this nodes it is possible to improve the performance

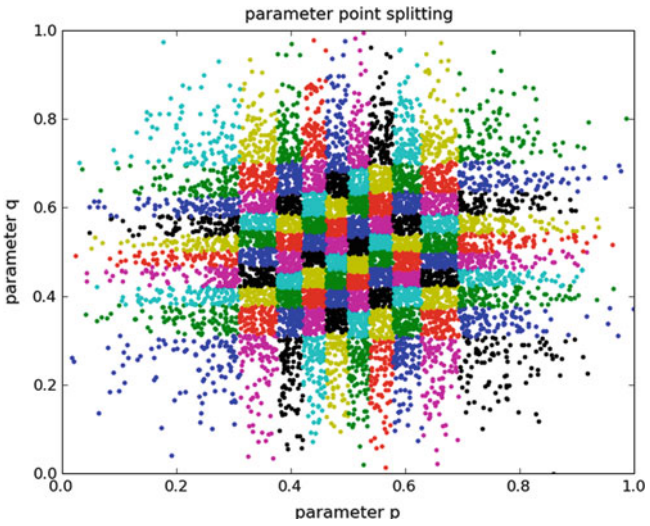

Fig. 2 A set of 10000 random parameter points splitted in parameter packages regarding their distribution

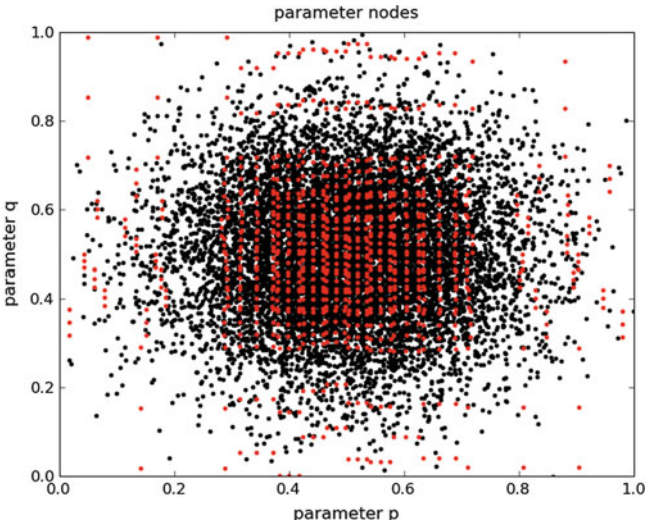

Fig. 3 A set of 10000 random parameter points and the associated set of 100 parameter nodes

of the calculation of the numerical solutions. For this reason formulate a modified system of DAEs involving the parameter interpolation nodes (Fig. 3).

Definition 2. Define a new system of DAEs by modifying the right hand side of a giving DAE with the help of some solutions x_{p_j}:

$$Ad'(x,t,p) + b(x,t,p) - \sum_{j=1}^{(m+1)^d} c_j(p)(Ad'(x_{p_j},t,p_j) + b(x_{p_j},t,p_j)) = 0 \tag{4}$$

with $m \in \mathbb{N}$ the order of the parameter interpolation, $p_j \in \mathbb{P}$ the nodes of the parameter interpolation, $x_{p_j} \in \mathbb{R}^m$ the solutions at the interpolation nodes and $c_j(p) \in \mathbb{R}$ the weighting functions of an multidimensional polynomal interpolation.

Notice that the exact solution x_{p_j} fulfills $Ad'(x_{p_j},t,p_j) + b(x_{p_j},t,p_j) = 0$, therefore the modified DAE will be solved by x_p and can be called equivalent to the original formula (1). In praxis one only has the numerical solution $x_{p_j,n}$ at a given point t_n and $Ad'(x_{p_j},t,p_j) + b(x_{p_j},t,p_j)$ will not be exactly zero. For the interpolation in the parameter space a polynomial interpolation is used. This will force the interpolation nodes p_j to be on a tensor-product grid in the parameter space. In praxis this is objectionable and can be avoided by an interpolation with radial basis functions or other interpolation methods. But assuming interpolation nodes p_j to be on a tensor-product grid and using an multidimensional polynomial interpolation does make the proofs and formulas much easier. Now again use a BDF-method to solve the modified system of DAEs.

Theorem 2. *Discretise the modified DAE for a fixed parameter point p_0 with a BDF-method. Discretise the parameter interpolation part as well with the same BDF-method and achieve:*

$$A\frac{1}{h}\sum_{i=0}^{K}\alpha_i d(x_{n-i}, t_{n-i}, p_0) + b(x_n, t_n, p_0)$$
$$-\sum_{j=1}^{(m+1)^d} c_j(p_0)(A\frac{1}{h}\sum_{i=0}^{K}\alpha_i d(x_{p_j,n-i}, t_{n-i}, p_j) + b(x_{p_j,n}, t_n, p_j)) = 0 \quad (5)$$

Let h be the constant step size in time. Let the initial steps be sufficiently accurate. Then the error of the numerical solution obtained by solving the BDF-discretised modified system can be bounded by:

$$\max_{n>\mu K} ||x(t_n) - x_n|| \leq c^*(diam(\mathbb{P})^{m+1}(h^K + \frac{1}{h^\mu}\max_{0<j<\mu K}||\delta_{n-j}||) + h_0^K) \quad (6)$$

with K the order of the BDF-method, $\mu + 1$ the index DAE and c being a bounded constant. Furthermore $\mathbb{P}$ is the parameter domain in one package and h_0 is the step size used in the calculation of the solutions at the interpolation nodes.

Compare this result with [4]. At this point notice the h_0^K term in the error estimate, because of this term the solutions computed with the help of the modified system cannot be more accurate than the solutions at the parameter nodes. Therefore the step size h_0 must be as small as the step size we would have chosen without this new approach. The source of the improvement of this error estimation is the parameter interpolation witch yields to the term $diam(\mathbb{P})^{m+1}$ in front of the normal relation of the error to the step size of the BDF-method h. You could say that we can accelerate the computation of the numerical solution because we have a good guess of the solution before the calculation itself starts. At this point there are three different cases to be observed. First the interpolation guess is as accurate as the given tolerance. In that case we don't have to calculate a new solution. In this trivial case we don't need a Parameter-Time-Integration since the parameter interpolation is already good enough. Second the interpolation guess is not as accurate as the given tolerance but accurate enough to accelerate the calculation of the new numerical solution. That means again that we can chose a bigger step size but still achieve the given tolerance. And third the accuracy of the guess is to low to improve the computation, which means we have to solve the original system. With an adaptive time step solver these three cases can switch on every timestep depending of the smoothness of the solution at the given time point regarding the parameters.

Think again of the example parameter set, so let $\mathbb{P} \subset \mathbb{R}^2$ again be a set of 10000 random parameter points. In this example one has to solve only 100 DAE of the original system, if the new Parameter-Time-Integrator is be used. The remaining solutions can be obtained by solving the modified system with the improved convergence estimate. In the example the parameter point splitting is based on external information in that case the distribution of the points, therefore we can

assume that the parameter smoothness is big enough in relation to the size of the parameter packages.

4 Example

As an example for the applications in the circuit simulation observe the following circuit (Fig. 4):

The linear time-varying DAE

$$q_1' + G_1(t)e_1 + j_V = 0, \quad q_2' + G_2(t)e_2 + \eta t j_V = 0$$
$$e_1 = -\frac{1}{1-\eta t}e_2, \quad q_1 = C_1 e_1, \quad q_2 = C_2 e_2$$

simulates the electric circuit.

Here $G_1(t) = (1-\eta t-\lambda)$ and $G_2(t) = (\frac{\eta}{1-\eta t} - \lambda - \eta t)$ are the resistor functions which can be changed by the parameters η and λ to simulate the circuit with different resistors. Let $\lambda = -5$ be constant and $p = \eta$ be the variating parameter of our system. So this is a one-dimensional parameter space.

Furthermore e_1, e_2 describe the voltages at the nodes 1 and 2 with respect to the mass node. j_V is the current through the voltage source and q_1, q_2 represent the charges of the capacitances C_1 and C_2. For simplicity $C_1 = C_2 = 1$ is assumed. With

$$x = \begin{pmatrix} q_1 \\ q_2 \\ e_1 \\ e_2 \\ j_V \end{pmatrix}, \; A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \; b(x,t,p) = \begin{pmatrix} (6-pt)x_2 + x_4 \\ (5 + \frac{p}{1-pt} - pt)x_3 + ptx_4 \\ x_2 + \frac{1}{1-pt}x_3 \\ x_0 - x_2 \\ x_1 - x_3 \end{pmatrix}$$

$$d(x,t,p) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} x$$

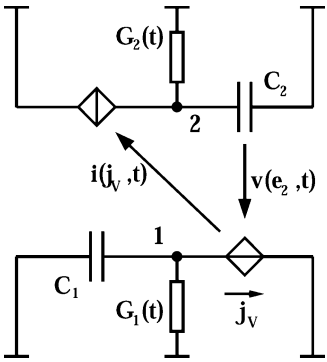

Fig. 4 The circuit diagram shows two small separated circuit whose sources are being controlled by the current or the voltage of the respective subcircuit. Apart from that the subcircuit consists of a resistor and a capacitor only

The circuit describing DAE can be written as:

$$Ad'(x,t,p) + b(x,t,p) = 0.$$

The parameter domain is defined through $p \in \mathbb{P} := [-20, -19]$. Solve the normal system at two parameter points $p_1 = -19.15$ and $p_2 = -19.85$ with an implicit Euler method with a constant step size $h = 10^{-4}$ in a time interval $T = [0, 1]$. This means 10000 steps have to be calculated with one Newton step each. At each parameter point a numerical solution x_{p_1} and x_{p_2} with an error $||x_{p_i}(t_n) - x_{p_i,n}|| \leq 2 \cdot 10^{-2}$ is calculated. Choose a random parameter in $\mathbb{P}$, for example $p_3 = -19.6$ and just solve it normally with a constant step size $h = 10^{-3}$ with an implicit Euler. Then the third component of the solution j_v is calculated with an error $||(j_v)_{p_3}(t_n) - (j_v)_{p_3,n}|| \leq 2 \cdot 10^{-1}$ (Fig. 5).

Now again solve the DAE with a constant step size $h = 10^{-3}$ and with an implicit Euler at $p_3 = -19.6$, but use the changed system. Again 1000 time and Newton steps are required, but an error $||(j_v)_{p_3}(t_n) - (j_v)_{p_3,n}|| \leq 2.5 \cdot 10^{-2}$ is achieved (Fig. 6).

5 Conclusion

In this paper we have seen a new approach to solve parameter dependent problems. The main idea was to approximate the solution at a parameter point with the already calculated solution of other points to have a good guess of the solution before calculating it. With this guess it is possible to accelerate the computation of the

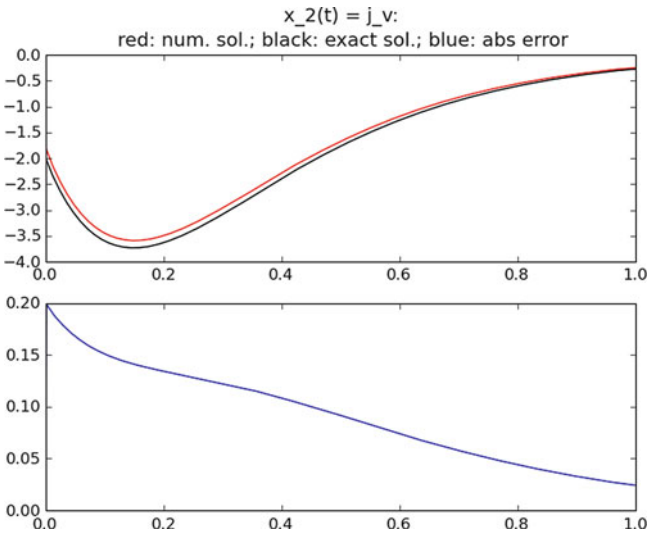

Fig. 5 Solution j_v with time on the x-axis and absolute error or solution values on the y-axis. The initial system is used

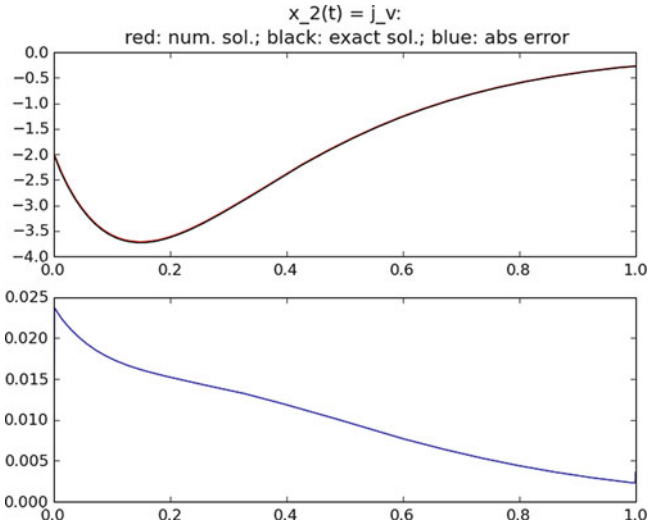

Fig. 6 Solution j_v with time on the x-axis and absolute error or solution values on the y-axis. The modified system is used

numerical solution. The degree of acceleration depends strongly on the smoothness of the solution regarding the parameter, since it wont be possible to obtain a good guess before the calculation, if there is nearly no connection between the parameters and the solution of our problem. But since we can decide at every single parameter point whether we use the parameter-time-integration or we solve the equation of the problem without it, one can exploit the smoothness of the parameter as long as there is some.

References

1. Schwarz, D.E., Tischendorf, C.: Structural analysis of electrical circuits and consequences for MNA. Int. J. Circuit Theory Appl. **28**, 131–162 (2000)
2. Scholz, M., Engel, C., Loeffler, M.: Modelling Human Granulopoiesis under Polychemotherapy with G-CSF Support. Math. Biol. **50**(4), 397–439 (2004)
3. Tischendorf, C.: Coupled Systems of Differential Algebraic and Partial Differential Equations in Circuit and Device Simulation, Habitilationsschrift, pp. 6–31. Humboldt University of Berlin (2004), http://www.mi.uni-koeln.de/~ctischen/publications.html
4. Jansen, L.: Effektive Numerische Simulation Elektrischer Schaltungen in Bezug auf Parameterschwankungen, Master's thesis, University of Cologne (2009)
5. März, R.: Nonlinear Differential-Algebraic Equations with Properly Formulated Leading Term, Preprint 3, Humboldt University of Berlin, Institute of Mathematics (2001)
6. Higueras, I., März, R.: Differential-algebraic equations with properly stated leading terms. Comput. Math. Appl. **48**, 215–235 (2004)
7. Voigtmann, S.: General Linear Methods for Integrated Circuit Design. Dissertation, pp. 111–173 (2004)