28 Mathcad[1] Worksheets for Differential Algebraic Equations

Address: Prof. Dr. K. Schittkowski
Department of Mathematics
University of Bayreuth
D - 95440 Bayreuth

Phone: +921 553278 (office)
+921 32887 (home)

Fax: +921 35557

E-mail: klaus.schittkowski@uni-bayreuth.de
Web: http://www.klaus-schittkowski.de
Date: March 15, 2004

Abstract

The purpose of the paper is to introduce a set of Mathcad worksheets containing differential algebraic (DAE) equations. They can be used to become familiar with the Mathcad implementation of differential equations and with the behavior of dynamical systems in general. The problems are taken from a collection of test examples for data fitting in dynamical systems, see Schittkowski [17]. The report contains a summary of 28 differential algebraic equations that have been transferred to Mathcad and a detailed example. All worksheets can be downloaded from the home page of the author[2]. A particular advantage of executing these problems from Mathcad is the possibility to plot corresponding solutions very easily.

[2]http://www.klaus-schittkowski.de

1 Introduction

We consider systems of differential algebraic equations, which can be considered as extensions of ordinary differential equations by adding algebraic equations and state variables. For example, they describe certain equilibrium or steady-state conditions. The system depends on m_d differential variables $y(t)$ and m_a algebraic variables $z(t)$, and the dynamical system is given in the form

$$\begin{array}{rcll} \dot{y}_1 & = & F_1(y,z,t) \ , & y_1(0) = y_1^0 \ , \\ & \cdots & & \\ \dot{y}_{m_d} & = & F_{m_d}(y,z,t) \ , & y_{m_d}(0) = y_{m_d}^0 \ , \\ 0 & = & G_1(y,z,t) \ , & z_1(0) = z_1^0 \ , \\ & \cdots & & \\ 0 & = & G_{m_a}(y,z,t) \ , & z_{m_a}(0) = z_{m_a}^0 \ . \end{array} \tag{1}$$

Without loss of generality, we assume again that the initial time is zero. If $m_d = 0$, we get a steady-state system, and for $m_a = 0$ a system of ordinary differential equations. The solution depends on the time variable t and is denoted by $y(t)$ and $z(t)$. It is assumed that the right-hand side of (1) is defined by continuous functions $F(y,z,t) = (F_1(y,z,t),\ldots,F_{m_d}(y,z,t))^T$ and $G(y,z,t) = (G_1(y,z,t),\ldots,G_{m_a}(y,z,t))^T$.

We need initial values for the differential equations $y_0 = (y_1^0,\ldots,y_{m_d}^0)^T$ and also for the algebraic equations $z_0 = (z_1^0,\ldots,z_{m_a}^0)^T$. $y(t)$ and $z(t)$ represent the solution of a joint system of $m_d + m_a$ differential and algebraic equations (DAE). The system is called an index-1-problem or an index-1-DAE, if the algebraic equations can be solved with respect to z. A necessary condition is that the Jacobian matrix

$$\nabla_z G(y,z,t) \tag{2}$$

possesses full rank. In all other cases, we obtain DAEs with a higher index, see Hairer and Wanner [9] for a suitable definition and more details. Note that systems with higher index can be transformed to systems of index 1 by successive differentiation of the algebraic equations, and that the number of these differentiations defines the index. Now we consider only problems up to index 3.

We have to be very careful when defining the initial values of the DAE model, since they must satisfy the consistency equation

$$G_1(y_0,z_0,t) = 0 \ , \ \ldots \ , G_{m_a}(y_0,z_0,t) = 0 \ . \tag{3}$$

Otherwise, we have to check whether the consistency condition is satisfied before starting the integration. If not, consistent initial values must be computed by solving the above system of nonlinear equations subject to z, where the initial values of the differential equations are inserted. If this is not possible as in case of an index greater than 1, we have to consider alternative equations obtained by one or two differentiations of (3) as discussed in the subsequent section.

Example 1.1 *We consider a unipolar hydrodynamic model for semiconductors in the isotropic case discussed by Asher and Petzold [1] with two differential and one algebraic equation of index 1,*

$$
\begin{array}{lcll}
\frac{d}{dt}\phi(t) & = & \rho(t)E(t) - \alpha J \ , & \phi(0) = 3.08 \ , \\
\frac{d}{dt}E(t) & = & \rho(t) - 1 \ , & E(0) = -1.14 \ , \\
0 & = & J^2 + \rho(t)^2 - \phi(t)\rho(t) \ , & \rho(0) = \frac{1}{2}\phi(0) + \sqrt{\frac{1}{4}\phi(0)^2 - J^2} \ ,
\end{array}
\tag{4}
$$

$\phi(t)$ and $E(t)$ are the differential variables and $\rho(t)$ is the algebraic variable. The initial value for $\rho(t)$ at $t = 0$ is consistent. Moreover, there are two constants $J = 0.5$ and $\alpha = 0.1$.

The software system EASY-FIT, see Schittkowski [17], comes with a collection of 1,000 test examples for data fitting in dynamical systems. Among them are 34 systems of differential algebraic equations, where some parameters of the right-hand side or the initial conditions are to be fitted. Most problems have some practical background.

However, the basic structure of these problems is more general and adopted to data fitting. For example, some of the test problems possess additional constraints, there are break or switching points where the system changes its structure, and some of the data fitting test problems only differ in the data, not the dynamical system. Moreover, some of the problems are too complex for the purpose of this collection. Thus, a subset of 28 test problems is selected and re-implemented in Mathcad. The mcd-files can be downloaded from the home page of the author,

`http://www.uni-bayreuth.de/departments/math/~kschittkowski/home.htm`

The report is one out of a series of Mathcad test problem collections by which numerical routines are tested and the implementation of optimization problems and dynamical systems is outlined, i.e.,

1. nonlinear programming [19],
2. data fitting [20],
3. ordinary differential equations [21],
4. partial differential equations [22],

5. partial differential algebraic equations [23].

Section 2 contains a brief outline of the implicit integration routine called *Odesolve* in Mathcad, which is used for all test cases. A simple example is shown in Section 3 to illustrate the numerical solution of differential algebraic equations. A list of the Mathcad worksheet files and some further details about problem structure, background, and source is given in Section 4.

2 Implicit Solution Methods

A characteristic property of explicit integration methods for differential equations is that a new approximation of the solution is evaluated explicitly from the known one at a previous time, and from some intermediate function values of the right-hand side of the differential equation. This iterative integration process breaks down in case of numerical instability of the underlying differential equation called stiffness, or in case of algebraic equations. In these situations, we need more powerful algorithms that are more stable and can satisfy additional equations.

Typically, differential algebraic equations are solved by implicit solution methods, since the internal solution of a system of nonlinear equations allows us to add the algebraic constraints and to satisfy them in each iteration step. To analyze the situation in more detail, we assume that a DAE of index 3 is given in explicit formulation

$$
\begin{array}{rclcrcl}
\dot{y}_1 & = & F_1(y_1, y_2, t) \ , & & y_1(0) & = & y_1^0 \ , \\
\dot{y}_2 & = & F_2(y_1, y_2, z, t) \ , & & y_2(0) & = & y_2^0 \ , \\
0 & = & G(y_1, t) \ , & & z(0) & = & z_0 \ ,
\end{array}
\tag{5}
$$

where $F_1 = (F_1^1, \ldots, F_{m_{d1}}^1)^T$, $F_2 = (F_1^2, \ldots, F_{m_{d2}}^2)^T$ with $m_{d1} + m_{d2} = m_d$, and $G = (G_1, \ldots, G_{m_a})^T$. Consistent initial values have to satisfy the equations

$$
\begin{array}{lcl}
G(y_1, t) & = & 0 \ , \\
\nabla_{y_1} G(y_1, t)^T F_1(y_1, y_2, t) & = & 0 \ , \\
\nabla_{y_1 y_1}(G(y_1, t), F_1(y_1, y_2, t)) & & \\
\quad + \nabla_{y_1} G(y_1, t)^T \nabla_{y_1} F_1(y_1, y_2, t)^T F_1(y_1, y_2, t) & & \\
\quad + \nabla_{y_1} G(y_1, t)^T \nabla_{y_2} F_1(y_1, y_2, t)^T F_2(y_1, y_2, z, t) & = & 0
\end{array}
\tag{6}
$$

at $t = 0$. Here $\nabla_{y_1 y_1}(G(y_1, t), F_1(y_1, y_2, t))$ denotes the partial derivatives of $\nabla_{y_1} G(y_1, t)$ with respect to y_1 applied to $F_1(y_1, y_2, t)$. Moreover, the index-3-assumption requires that the algebraic constraints in the reduced equivalent index-1-formulation can be eliminated. The matrix

$$\nabla_{y_1} G(y_1,t)^T \nabla_{y_2} F_1(y_1,y_2,t)^T \nabla_z F_2(y_1,y_2,z,t)^T$$

is non-singular in a neighborhood of a solution. We call y_1 the vector of index-1-variables, y_2 the vector of index-2-variables and z the vector of algebraic or index-3-variables.

Now we apply an implicit solution method for ordinary differential equations as discussed in Schittkowski [17, 21] defined by a so-called Butcher array. Let h_j be a stepsize of the j-th integration step, $t_{j+1} = t_j + h_j$ a new trial point with $t_0 = 0$ and η_j^1, η_j^2, and ζ_j known approximations of the solution $y_1(t_j)$, $y_2(t_j)$, and $z(t_j)$. It is also assumed that consistent initial values (6) are given for $t = 0$. Then a new approximation of the solution is obtained from

$$\begin{aligned} \eta_{j+1}^1 &= \eta_j^1 + h_j \textstyle\sum_{i=1}^r b_i k_i^1 \ , \\ \eta_{j+1}^2 &= \eta_j^2 + h_j \textstyle\sum_{i=1}^r b_i k_i^2 \ , \\ \zeta_{j+1} &= \zeta_j + h_j \textstyle\sum_{i=1}^r b_i l_i \ , \end{aligned} \tag{7}$$

where the coefficients k_i^1, k_i^2 and l_i depend on previous approximations and the current one. They are computed by solving a system of nonlinear equations

$$\begin{aligned} k_i^1 &= F_1(\Phi_j^1, \Phi_j^2, t_j + h_j c_i) \ , \\ k_i^2 &= F_2(\Phi_j^1, \Phi_j^2, \Psi_j, t_j + h_j c_i) \ , \\ l_i &= G(\Phi_j^1, t_j + h_j c_i) \ , \end{aligned} \tag{8}$$

for $i = 1, \ldots, r$ with

$$\begin{aligned} \Phi_j^1 &= \eta_j^1 + h_j \sum_{m=1}^r a_{im} k_m^1 \ , \\ \Phi_j^2 &= \eta_j^2 + h_j \sum_{m=1}^r a_{im} k_m^2 \ , \\ \Psi_j &= \zeta_j + h_j \sum_{m=1}^r a_{im} l_m \ . \end{aligned} \tag{9}$$

The constant coefficients a_{im}, c_i and b_i are chosen by any appropriate tableau, for example by the Butcher array of type Radau IIA with three stages and order five,

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array} \tag{10}$$

For the modified implicit solution method, it is possible to prove that the system of nonlinear equations (9) is always solvable and that a solution of the given equations (1) is approximated, see Hairer, Lubich and Roche [8]. However, the local and global convergence errors are somewhat different. The local error behaves on the order $o(h^{q+1})$, $o(h^q)$ and $o(h^{q-1})$ with respect to Φ^1, Φ^2 and Ψ, respectively, where $q = 3$ for the Radau-type algorithm defined by (10). Thus, it is recommended to adapt the error tolerances by multiplying the index-2-variables by h_j and the index-3-variables by ${h_j}^2$.

3 A Mathcad Worksheet Example

Mathcad (`http://www.mathcad.com`) is an interactive GUI with a large number of built-in mathematical functions. Special commands allow to solve systems of ordinary differential equations, especially stiff and differential algebraic equations by the implicit Radau method introduced in the previous section. The subsequent lines describe the usage of *Odesolve* for solving DAEs, see also the Mathcad documentation

The Odesolve function is used within solve blocks, allowing for natural notation, and are the easiest to use and interpret. *Odesolve(vector,x,b,[step])* returns a vector of functions of x which is a solution to the system of ordinary differential algebraic equations, subject to initial values provided in the solve block. The DAEs must be of first order, of index 1,2, or 3, and explicitly given, see (1). Odesolve allows algebraic equations which adds an extra unknown function for each equation to the system, which must be specified as one of the output functions in the Odesolve call.

Arguments:

- *vector* is the explicit vector of function names as they appear within the solve block.
- x is the variable of integration and must be real.
- b is the terminal point of the integration interval.
- *step* is the optional number of steps used internally when calculating the solution.

A *solve block* refers to a group of steps involved when solving a system of differential equations. Needed are initial the key word *Given*, a set of equations, and the solving function *Odesolve*. Collectively, these steps are known as a solve block.

To give an impression how a test problem is implemented, we consider problem HYDRODYN, see Example 1.1. Subsequently, the Mathcad implementation is listed, see Figures 1 and 2 . The mcd-file contains also an error plot of the algebraic equation subject to the termination tolerance 0.001.

HYDRODYN

Differential Algebraic Equation

Description: Unipolar hydrodynamic model for semiconductors in the isotropic case

Parameters: $J := 0.5$ $\quad alpha := 0.1$ $\quad tend := 2$

$phi0 := 3.08$ $\quad E0 := -1.14$

Consistent Initial Values: $rho0 := 0.5 \cdot phi0 + \sqrt{0.25 phi0^2 - J^2}$

Initial Values: Given $\quad phi(0) = phi0 \quad E(0) = E0 \quad rho(0) = rho0$

Differential Equation:

$$\frac{d}{dt} phi(t) = rho(t) \cdot E(t) - alpha \cdot J$$

$$\frac{d}{dt} E(t) = rho(t) - 1$$

Algebraic Equation:

$$J^2 + rho(t)^2 - phi(t) \cdot rho(t) = 0$$

Integration:

$$\begin{pmatrix} phi \\ rho \\ E \end{pmatrix} := Odesolve\left[\begin{pmatrix} phi \\ rho \\ E \end{pmatrix}, t, tend\right]$$

Figure 1: Mathcad Implementation: Equations

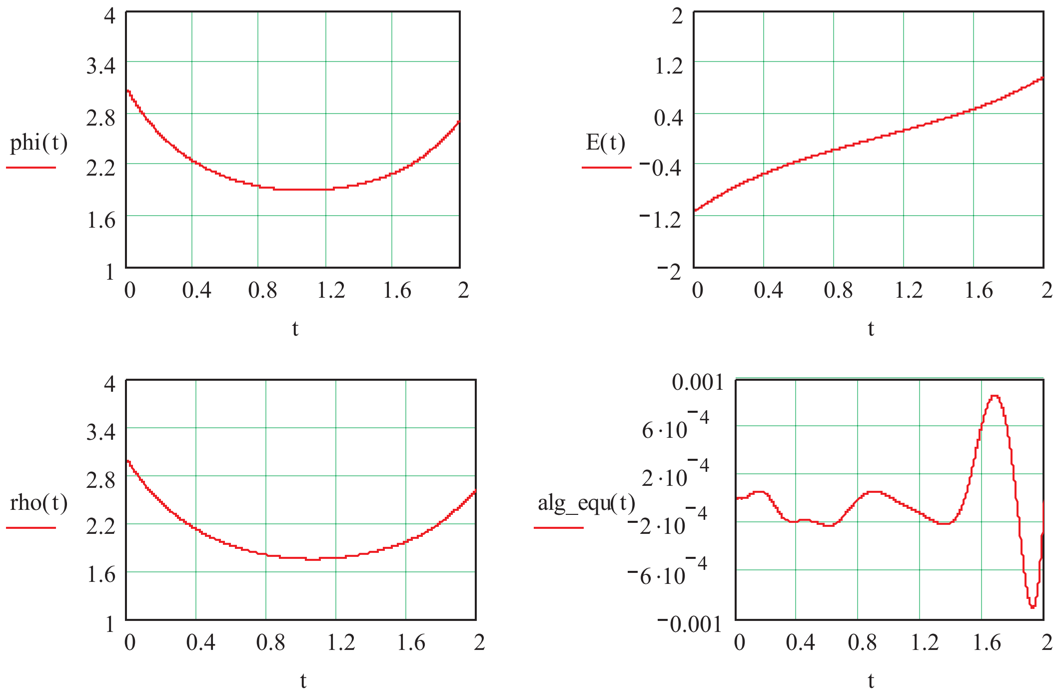

Figure 2: Mathcad Implementation: Plots

4 List of All Test Problems

The subsequent table contains a list of all test problems together with the number of differential equations m_d, the number of algebraic equations m_a, a brief description of the practical or mathematical background, and some references. The differential equations have first been implemented in the modelling language PCOMP, see Dobmann et al. [6] or Schittkowski [17, 16, 18]. The transformation into Mathcad worksheets follows a unified format based on the PCOMP equations. Thus, the implementations do not exploit all possible features of Mathcad to get the most elegant and compact description. All mcd-files can be downloaded from the home page of the author[3].

Differential Algebraic Equations

name	m_d	m_a	*background*	*ref*
AEROSOL	2	2	Substrate concentration in two-phase aerosol devices	
BATCH	6	3	Isothermal batch reactor, slow and fast reactions	[3]
BATCH_E	6	3	Isothermal batch reactor, slow and fast reactions, two data sets	[3]
BATCHREA	6	1	Batch reactor	[5]
BUBBLEC	8	5	Bubble point calculation for a batch distillation column	[10]
CELLS	3	2	Cultivation of isolated plant cells in suspension culture	[14]
CONDENS	1	5	Condensation of methanol with constant volume	[15]
DAE_I1	4	1	Academic example, index-1-formulation	
DAE_I2	4	1	Academic example, index-2-formulation	
DAE_I3	4	1	Academic example, index-3-formulation	
DAE_IN2	2	1	System of three differential algebraic equations of index 2	[1]
DAE_SYS	1	2	Particle diffusion and reaction (2nd order BVP)	[1]
EVAPOR	3	10	Evaporation of benzol with constant volume	[11]
EXOBATCH	5	6	Batch reactor with strongly exothermic reactions and cooling jacket	[24]
HYDRODYN	2	1	Unipolar hydrodynamic model for semiconductors in the isotropic case	[1]
MEM_WIRE	3	1	Optimal form of shape memory wires	
P_IDENT	2	1	Identification of parameters, academic example	[12]

(continued)

[3] http://www.klaus-schittkowski.de

name	m_d	m_a	*background*	*ref*
PENDULUM1	4	1	Plain pendulum, index 3	
PENDULUM2	4	1	Plain pendulum, index 2	
PENDULUM3	4	1	Plain pendulum, index 1	
PENDULUM4	4	1	Plain pendulum, index 0	
PHOSPH_A	3	2	Chemical reaction, phosphorescence	
RESPIR	1	1	Human respiratory system	[25]
SHOCK	6	3	Reaction zone in detonating explosives	[7]
TRANSIST	3	2	Transistor amplifier, highly oscillating	
TUBULAR	2	2	Stationary tubular reactor with cooling wall	[13]
URETHAN	3	10	Urethan reaction in a semi batch reactor with two feed vessels	[2]
VDPOL	1	1	Van der Pol equation, electrical circuit	

References

[1] Ascher U.M., Petzold L.R. (1998): *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations,* SIAM, Philadelphia

[2] Bauer I., Bock H.G., Koerkel S., Schloeder J. (1999): *Numerical methods for optimum experimental design*, Report, IWR, University of Heidelberg

[3] Biegler L.T., Damiano J.J., Blau G.E. (1986): *Nonlinear parameter estimation: a case study comparison*, AIChE Journal, Vol. 32, No. 1, 29-45

[4] Butcher J.C. (1964): *Integration processes based on Radau quadrature formulas,* Mathematics of Computations, Vol. 18, 233-244

[5] Caracotsios M., Stewart W.E. (1985): *Sensitivity analysis of initial value problems with mixed ODE's and algebraic equations*, Computers and Chemical Engineering, Vol. 9, 359-365

[6] Dobmann M., Liepelt M., Schittkowski K. (1995): *Algorithm 746: PCOMP: A Fortran code for automatic differentiation,* ACM Transactions on Mathematical Software, Vol. 21, No. 3, 233-266

[7] Edsberg L., Wedin P.A. (1995): *Numerical tools for parameter estimation in ODE-systems*, Optimization Methods and Software, Vol. 6, 193-218

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

[9] Hairer E., Wanner G. (1991): *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems,* Springer Series Computational Mathematics, Vol. 14, Springer, Berlin

[10] Ingham J., Dunn I.J., Heinzle E., Prenosil J.E. (1994): *Chemical Engineering Dynamics*, VCH, Weinheim

[11] Majer C., Marquardt W., Gilles E.D. (1995): *Reinitialization of DAE's after discontinouities*, Proceedings of the Fifth European Symposium on Conputer-Aided Process Engineering, 507-512

[12] Majer C. (1998): *Parameterschätzung, Versuchsplanung und Trajektorienoptimierung für verfahrenstechnische Prozesse*, Fortschrittberichte VDI, Reihe 3, Nr. 538, VDI, Düsseldorf

[13] Mayer U. (1993): *Untersuchungen zur Anwendung eines Einschritt-Polynom-Verfahrens zur Integration von Differentialgleichungen und DA-Systemen*, Ph.D. Thesis, Dept. of Chemical Engineering, University of Stuttgart

[14] Munack A., Posten C. (1989): *Design of optimal dynamical experiments for parameter estimation*, Proceedings of the Americal Control Conference, Vol. 4, 2010-2016

[15] Pantelides C.C., Gritsis D., Morison K.R., Sargent R.W.H. (1988): *The mathematical modeling of transient systems using differential-algebraic equations*, Computers and Chemical Engineering, Vol. 12, 440-454

[16] Schittkowski K. (2001): *EASY-FIT: A software system for data fitting in dynamic systems*, Structural and Multidisciplinary Optimization, Vol. 23, No. 2, 153-169

[17] Schittkowski K. (2002): *Numerical Data Fitting in Dynamical Systems - A Practical Introduction with Applications and Software*, Kluwer Academic Publishers

[18] Schittkowski K. (2004): *PCOMP: A modeling language for nonlinear programs with automatic differentiation*, in: *Modeling Languages in Mathematical Optimization*, J. Kallrath ed., Kluwer, Norwell, MA, 349-367

[19] Schittkowski K. (2004): *110 Mathcad worksheets for nonlinear programming*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[20] Schittkowski K. (2004): *178 Mathcad worksheets for data fitting*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[21] Schittkowski K. (2004): *295 Mathcad worksheets for differential equations*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[22] Schittkowski K. (2004): *131 Mathcad worksheets for partial differential equations*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[23] Schittkowski K. (2004): *17 Mathcad worksheets for partial differential algebraic equations*, Report, Dept. of Computer Science, University of Bayreuth, Germany

[24] Vassiliadis V.S., Sargent R.W.H., Pantelides C.C. (1994): *Solution of a class of multistage dynamic optimization problems, 2. Problems with path constraints*, Industrial Engineering and Chemical Research, Vol. 33, No. 9, 2123-2133

[25] Vlassenbeck J., van Dooren R. (1983): *Estimation of the mechanical parameters of the human respiratory system*, Mathematical Biosciences, Vol. 69, 31-55