Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 639–642

Inexact Newton method as a tool for solving differential-algebraic systems

Paweł Drąg
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Janiszewskiego 11/17, 50-372, Wrocław, Poland
Email: pawel.drag@pwr.wroc.pl

Krystyn Styczeń
Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Janiszewskiego 11/17, 50-372, Wrocław, Poland
Email: krystyn.styczen@pwr.wroc.pl

***Abstract*—The inexact Newton method is commonly known from its ability to solve large-scale systems of nonlinear equations. In the paper the classical inexact Newton method is presented as a tool for solving differential-algebraic equations (dae) in fully-implicit form $F(\dot{y}, y, t) = 0$. The appropriate statement of dae using the backward Euler method makes the possiblity to see the differential-algebraic system as a large-scale system of nonlinear equations. Because a choice of the forcing terms in the inexact Newton method significantly affects the convergence of the algorithm, in the paper new variants of the inexact Newton method were presented and tested. The simulations were executed in Matlab environment using Wroclaw Centre for Networking and Supercomputing.**

***Index Terms*—differential-algebraic equations, systems of nonlinear equations, inexact Newton method.**

I. Introduction

DIFFERENTIAL-algebraic equations (dae) play a key role in control science and engineering [16], [17]. Describing the system with equations that incorporate dynamics and conservation laws, creates new opportunities for the development of the numerical methods and has a direct application in the industry [2], [3]. Design and control of chemical reactors and motor vehicle requires precise knowledge of the links between the system and the signals flowing from the environment, as well as between the internal elements of the system. Needs arising from the control of the large complex installations always outweight the modern computing capabilities, and are becoming a cause for the progress of both the hardware as well as the algorithms and the numerical methods.

The question raised in the article refers to the situation when the considered system is described by differential-algebraic equations in a general way possible. This approach has a chance to wide and common use in industry. The presented method is part of a widely used approach, which reduces infinite dimensional task to the large-scale finite-dimensional problem.

The paper is constructed as follows. In the next section the backward differential formula (bdf) is presented as the tool for solving dae systems. New aspects of the inexact Newton method were presented in $3rd$ and $4th$ sections. The presented algorithms were tested on the kinetic batch reactor model. The results were presented in $5th$ section.

II. The backward Euler method

The codes for solving dae in the $fully-implicit$ form are based on a technique which was introduced by Gear [12]. The backward differential formula is the first general technique for the numerical solution of dae and have emerged as the most popular. The idea of this technique is that the derivative $\frac{dy(t)}{dt}$ can be approximated by a linear combination of the solution $y(t)$ at the current mesh point and at several previous mesh points ([14]).

Bdf was initially defined for the systems of differential equations coupled to algebraic equations. This method was soon extended to apply to any fully-implicit system of differential-algebraic equations

$$G\Big(\frac{dy(t)}{dt}, y(t), z(t), t\Big) = 0. \tag{1}$$

The simplest method for solving differential-algebraic systems is the first order bdf, or the backward Euler method, which consists of replacing the derivative in (1) by a backward difference

$$F\Big(\frac{y_n - y_{n-1}}{h}, y_n, z_n, t_n\Big) = 0. \tag{2}$$

where $h = t_n - t_{n-1}$.

The resulting system of nonlinear equations for y_n at each step is then usually solved by the Newton method [4]. In this way, the solution is advanced from time t_n to time t_{n+1}. It is assumed, that $y(t_0)$ is known. Assume too, that t (time) is the independent variable. In practical applications in chemical enginnering, as the independent variable is used usually the lenght of the reactor. If the time interval, in which the system has to be considered, is known, it can be scaled to the interval $[0, 1]$.

III. The inexact Newton method

The methodology presented in the previous paragraph leads to the following equation

$$F(x) = 0. \tag{3}$$

This equation is very general and is often found in scientific and engineering computing areas. We assume that the function F is considered, where $F : \mathcal{R}^n \to \mathcal{R}^n$ is a nonlinear mapping with the following properties

978-1-4673-4471-5/$25.00

(1) There exists an $x^* \in \mathcal{R}$ with $F(x^*) = 0$.
(2) F is continously differentiable in a neighborhood of x^*.
(3) $F'(x^*)$ is nonsingular.

There are a lot of methods for solving this nonlinear equation. One of the most popular and important is the Newton method. The Newton's method is attractive because it converges rapidly (quadratically) from any sufficiently good initial point. Its computational cost can be expensive, particularly, when the size of the problem is very large, because in each iteration step the Newton equations

$$F(x_k) + F'(x_k)s_k = 0 \tag{4}$$

should be solved. Here x_k denotes the current iterate, and $F'(x_k)$ is the Jacobian matrix of $F(x)$ at point x_k. The solution s_k^N of the Newton equation is the Newton step. Once the Newton step is obtained, the next iterate is given by

$$x_{k+1} = x_k + s_k^N. \tag{5}$$

In 1982 Dembo, Eisenstat and Steihaug proposed the inexact Newton method, which is a generalization of the Newton method [8]. The inexact Newton method is any method which, given an initial guess x_0, generates a sequence x_k of approximations to x^* as in Algorithm 1.

ALGORITHM 1. The inexact Newton method
1. Given $x_0 \in \mathcal{R}^n$
2. For $k = 0, 1, 2, \cdots$ until x_k convergence
  2.1 Choose some $\eta_k \in [0, 1)$
  2.2 Inexactly solve the Newton equations and obtain a step s_k, such that $\|F(x_k) + F'(x_k)s_k\| \le \eta_k \|F(x_k)\|$. $(\star)$
  2.3 Let $x_{k+1} = x_k + s_k$.

In the Algorithm 1, η_k is the forcing term in the k-th iteration, s_k is the inexact Newton step and $(\star)$ is the inexact Newton condition.

In each iteration step of the inexact Neton method a real number $\eta_k \in [0, 1)$ should be chosen. Then the inexact Newton step s_k is obtained by solving the Newton equation approximately with an iteration solver for systems of nonlinear equation. Since $F(x_k) + F'(x_k)s_k$ is both residual of the Newton equations and the local linear model of $F(x)$ at x_k, the inexact Newton condition $(\star)$ reflects both the reduction in the norm of the local linear model and certain accuarcy in solving the Newton equations. Thus the role of forcing terms is control the degree of accuracy of solving the Newton equations. In particular, if $\eta_k = 0$ for all k, then the inexact Newton method is reduced into the Newton method.

The inexact Newton method, like the Newton method, is locally convergent.

Theorem 1 ([8]): Assume that $F : \mathcal{R}^n \to \mathcal{R}^n$ is continously differentiable, $x^* \in \mathcal{R}^n$ such that $F'(x^*)$ is nonsingular. Let $0 < \eta_{max} < \beta < 1$ be the given constants. If the forcing terms η_k in the inexact Newton method satisfy $\eta_k \le \eta_{max} < \beta < 1$ for all k, then there exists $\varepsilon > 0$, such that for any $x_0 \in N_\varepsilon(x^*) \equiv \{x : \|x - x^*\| < \varepsilon\}$, the sequence $\{x_k\}$ generated by the inexact Newton method converges to x^*, and

$$\|x_{k+1} - x^*\|_* \le \beta \|x_k - x^*\|_*, \tag{6}$$

where $\|y\|_* = \|F'(x^*)y\|$.

If the forcing terms $\{\eta_k\}$ in the inexact Newton method are uniformly strict less than 1, then by Theorem 1, the method is locally convergent. The following result states the convergence rate of the inexact Newton method.

Theorem 2 ([8]): Assume that $F : \mathcal{R}^n \to \mathcal{R}^n$ is continously differentiable, $x^* \in \mathcal{R}^n$ such that $F'(x^*)$ is nonsingular. If the sequence $\{x_k\}$ generated by the inexact Newton method converges to x^*, then

(1) x_k converges to x^* superlinearly when $\eta_k \to 0$;
(2) x_k converges to x^* quadratically if $\eta_k = \mathcal{O}(\|F(x_k)\|)$ and $F'(x)$ is Lipschitz continuous at x^*.

Theorem 2 indicates, that the convergence rate of the inexact Newton method is determined by the choice of the forcing terms.

IV. A CHOICE OF FORCING TERMS

In the literature, researchers proposed some strategies to determine a good sequence of forcing terms. Here, four representatives strategies were selected.

(1) The choice of Dembo and Steihaug [9]:

$$\eta_k = \min\left\{\frac{1}{k+2}, \|F(x_k)\|\right\}. \tag{7}$$

The two strategies given by Eisenstat and Walker are more popular [11]. Among this two strategies, choice (2a) reflects the agreement between $F(x)$ and its local linear model at the previous step. Choice (2b) reflects the reduction rate of $\|F(x)\|$ from x_{k-1} to x_k.

For computational purposes of preventing the forcing terms from becoming quickly too small, some safeguards were added. The following strategies were obtained.

(2a) Given $\eta_0 \in [0, 1)$, choose

$$\eta_k = \begin{cases} \xi_k, & \eta_{k-1}^{(1+\sqrt{5})/2} \le 0.1, \\ \max\{\xi_k, \eta_{k-1}^{(1+\sqrt{5})/2}\}, & \eta_{k-1}^{(1+\sqrt{5})/2} > 0.1, \end{cases} \tag{8}$$

where

$$\xi_k = \frac{\|F(x_k) - F(x_{k-1}) - F'(x_{k-1})s_{k-1}\|}{\|F(x_{k-1})\|}, \tag{9}$$

$k = 1, 2, \ldots,$ or

$$\xi_k = \frac{|\,\|F(x_k)\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\|\,|}{\|F(x_{k-1})\|}, \tag{10}$$

$k = 1, 2, \ldots.$

(2b) Given $\gamma \in (0, 1]$, $\omega \in (1, 2]$, $\eta_0 \in [0, 1)$, choose

$$\eta_k = \begin{cases} \xi_k, & \gamma(\eta_{k-1})^\omega \le 0.1, \\ \max\{\xi_k, \gamma(\eta_{k-1})^\omega\}, & \gamma(\eta_{k-1})^\omega > 0.1, \end{cases} \tag{11}$$

where

$$\xi_k = \gamma\left(\frac{\|F(x_k)\|}{\|F(x_{k-1})\|}\right)^\omega, \tag{12}$$

$k = 1, 2, \ldots$.

(3) Choice of H.-B. An et al. [1]. Assume, that x_k is the current iterate and s_k is the step from x_k. The actual reduction $Ared_k(s_k)$ and predicted reduction $Pred_k(s_k)$ of $F(x)$ at x_k with step s_k are definned as follows

$$Ared_k(s_k) = \|F(x_k)\| - \|F(x_k + s_k)\|, \tag{13}$$

$$Pred_k(s_k) = \|F(x_k)\| - \|F(x_k) + F'(x_k)s_k\|. \tag{14}$$

Furthermore, let

$$r_k = \frac{Ared_k(s_k)}{Pred_k(s_k)}. \tag{15}$$

In this approach, r_k is used to adjust the forcing term η_k. Considering the value of r_k, one can distinguish four situation, which can have a place in computations.

(a) If $r_k \approx 1$, the the local linear model and nonlinear model will agree well on their scale and $\|F(x)\|$ usually will be reduced.

(b) If r_k nears 0, but $r_k > 0$, then the local linear model and nonlinear model disagree and $\|F(x)\|$ can be reduced very little.

(c) If $r_k < 0$, then the local linear model and nonlinear model disagree and $\|F(x)\|$ will be enlarged.

(d) If $r_k \gg 1$, then the local linear model and nonlinear model also disagree, but $\|F(x)\|$ will be reduced greatly.

The acceptable situations are, when $r_k \approx 1$ or $r_k \gg 1$, because in this cases the local linear model and nonlinear model agree well or at least leads to a great reduction point.

According to the property of r_k, one can choose forcing terms as follows.

$$\eta_k = \begin{cases} 1 - 2p_1, & r_{k-1} < p_1, \\ \eta_{k-1}, & p_1 \le r_{k-1} < p_2, \\ 0.8\eta_{k-1}, & p_2 \le r_{k-1} < p_3, \\ 0.5\eta_{k-1} & r_{k-1} \ge p_3, \end{cases} \tag{16}$$

where $0 < p_1 < p_2 < p_3 < 1$ are prescribed at first and $p_1 \in (0, \frac{1}{2})$. Assume, that η_0 is given.

The choice of forcing terms proposed in [1] is to determine η_k by the magnitude of r_{k-1}.

It is worth to note, that the current forcing term η_k is determined by the previous value r_{k-1} and η_k determines the value r_k through solving the Newton equations approximately.

V. Numerical results

As an example the kinetic batch reactor was choosed. This example is known from the literature [2], [6], [7] The concentrations are modeled by the system of differential and algebraic equations. The desired product AB is formed in the reaction

$$HA + 2BM \rightarrow AM + MBMH. \tag{17}$$

For the formulation given here the differential and algebraic variables are denoted by y_j and z_j respectively (Table 1).

The kinetic model is stated in terms of six differential mass balance equations

$$\dot{y}_1 = -k_2 y_2(t) z_8(t), \tag{18}$$

TABLE I
Batch reactor dynamic variables.

y_1	Differential State	$[HA] + [A^-]$
y_2	Differential State	$[BM]$
y_3	Differential State	$[HABM] + [ABM^-]$
y_4	Differential State	$[AB]$
y_5	Differential State	$[MBMH] + [MBM^-]$
y_6	Differential State	$[M^-]$
z_7	Algebraic State	$[H^+]$
z_8	Algebraic State	$[A^-]$
z_9	Algebraic State	$[ABM^-]$
z_{10}	Algebraic State	$[MBM^-]$

$$\dot{y}_2 = -k_1 y_2(t) y_6(t) + k_{-1} z_{10}(t) - k_2 y_2(t) z_8(t), \tag{19}$$

$$\dot{y}_3 = k_2 y_2(t) z_8(t) + k_3 y_4(t) y_6(t) - k_{-3} z_9(t), \tag{20}$$

$$\dot{y}_4 = -k_3 y_4(t) y_6(t) + k_{-3} z_9(t), \tag{21}$$

$$\dot{y}_5 = k_1 y_2(t) y_6(t) + k_{-1} z_{10}(t), \tag{22}$$

$$\dot{y}_6 = -k_1 y_2(t) y_6(t) - k_3 y_4(t) y_6(t) + k_{-1} z_{10}(t) + k_{-3} z_9(t), \tag{23}$$

an electroneutrality condition

$$z_7(t) = -0.0131 + y_6(t) + z_8(t) + z_9(t) + z_{10}(t) \tag{24}$$

and three equlibirum conditions

$$z_8(t) = \frac{K_2 y_1(t)}{K_2 + z_7(t)}, \tag{25}$$

$$z_9(t) = \frac{K_3 y_3(t)}{K_3 + z_7(t)}, \tag{26}$$

$$z_{10}(t) = \frac{K_1 y_5(t)}{K_1 + z_7(t)}. \tag{27}$$

with initial conditions $y_1(0) = 1.5776$, $y_2(0) = 8.32$, $y_j(0) = 0.0, j = 3, 4, 5$, $y_6(0) = 0.0131$, $z_7(0) = 0.5(-K_2 + \sqrt{K_2^2 + 4K_2 y_1(0)})$, $z_8(0) = z_7(0)$, $z_j(0) = 0.0, j = 9, 10$.

The following values of rate and equlibirum constants were used $k_1 = 21.893(hr^{-1} \cdot Kg \cdot gmole^{-1}), k_{-1} = 2.14E09(hr^{-1}), k_2 = 32.318(hr^{-1} \cdot Kg \cdot gmole^{-1}), k_3 = 21.893(hr^{-1} \cdot Kg \cdot gmole^{-1}), k_{-3} = 1.07E09(hr^{-1}), K_1 = 7.65E-18(gmole \cdot Kg^{-1}), K_2 = 4.03E-11(gmole \cdot Kg^{-1}), K_3 = 5.32E-18(gmole \cdot Kg^{-1})$.

The equations were considered in the time domain $t \in [0, 2.5]$. Then the equations were discretized into equidistant points with distnace 0.025. It resulted in 600 differential and 400 algebriac state variables. Then, 1000 equality constraints from the backward Euler method were imposed. The Jacobian matrix was obtained analitically and stored as the 1000×1000 sparse matrix.

This large-scale system of the linear equations was solved using GMRES algorithm [15]. The inexact Newton backtracking method [10] was used with four presented approaches for adjusting the forcing terms.

The results in Table 2 indicate, that the considered problem is difficult to solve. Iterations quickly converge to the locally optimal solution. The parameter r_k gives an answer, what

TABLE II
RESULTS FOR CHOICE OF H.-B. AN ET $al.$ [1].

iter	η_3	$\|F_3(x_k)\|$	r_k
1	0.4375	5.8635e3	1.4670
2	0.4297	4.0018e3	1.7745
3	0.4287	2.7828e3	2.1213
4	0.4286	2.0803e3	3.4620
5	0.4286	1.4856e3	2.2609
6	0.4286	1.2529e3	1.5051
7	0.5000	1.1420e3	5.3385e-4
8	0.5000	1.1420e3	NaN
9	0.5000	1.1420e3	NaN
10	0.5000	1.1420e3	NaN

TABLE III
RESULTS FOR OTHER SEQUENCES OF FORCING TERMS.

iter	η_1	$\|F_1(x_k)\|$	η_{2a}	η_{2b}	$\|F_{2a,2b}(x_k)\|$
1	0.3333	5.8635e3	0.5000	0.5000	5.8635e3
2	0.2500	4.0018e3	0.8057	0.4677	4.0018e3
3	0.2000	2.7828e3	0.9226	0.4655	2.7828e3
4	0.1667	2.0803e3	0.9689	0.4654	2.0803e3
5	0.1429	1.4856e3	0.9874	0.4654	1.4856e3
6	0.1250	1.2529e3	0.9949	0.4654	1.2529e3
7	0.1111	1.2143e3	0.9979	0.4773	1.1420e3
8	0.1000	1.1986e3	1.0000	0.5000	1.1420e3
9	0.0909	1.1972e3	1.0000	0.5000	1.1420e3
10	0.0833	1.1966e3	1.0000	0.5000	1.1420e3

is the relation between linear model and the whole system. The linear model agrees with the nonlinear model only at the beginning of the solution process. It is worth to note, that only the approach presented in [1] idicates, that after 7 iterations some difficulties can occur.

The simulations were executed with the parameters: $\gamma = 0.5$, $\omega = 1.5$ for proposition $2b$ and $p_1 = 0.25$, $p_2 = 0.6$, $p_3 = 0.8$ for the choice proposed in [1].

There are results for forcing terms adjusted in other manners in Table 3. The forcing terms adjusted as presented in [9] were decreased monotonically, but there is no information about agreement between $F(x)$ and its local linear model.

The forcing terms, adjusted as presented in [11], did not decrease monotonically to 0. Its main drawback is, that either the agreement between $F(x)$ and its local linear model at the previous step or the reduction rate of $\|F(x)\|$ are reflected in adapatation of forcing terms.

The simulations were executed in Matlab environment using Wroclaw Centre for Networking and Supercomputing.

As one can see, the results presented in the Table 2 and 3 are not the optimal solutions. If the initial guess for the inexact Newton method is close enough to the desired solution, then the convergence is very fast provided that the forcing terms are sufficiently small. But a good initial guess is generally very diffcult to obtain, especially for nonlinear equations that have unbalanced nonlinearities. Then the step length is often determined by the components with the strongest nonlinearities [5]. The nonlinearities are "unbalanded" when the step length is determined by a subset of the overall degrees of freedom.

VI. CONCLUSION

In the paper the new apsects of the inexact Newton method for solving differential-algebraic equations were presented, then the dae systems in the fully implicit form were considered. The methods for the choice of forcing terms for the inexact Newton method were presented and tested on the difficult and highly nonlinear kinetic batch reactor.

The authors would like to indicate, that the choice of forcing terms, which reflects both the agreement between $F(x)$ and its local linear model and the reduction rate of $\|F(x)\|$ are especially usefull for solving the large scale differential-algebraic equations. As the next step, the new preconditioned Jacobian-free optimization algorithm, which could solve the large-scale optimization tasks, will be studied and adjusted for new challenges in solving the optimal control problems [13].

REFERENCES

[1] H.-B. An, Z.-Y. Mo, X.-P. Liu, "A choice of forcing terms in inexact Newton method", *Journal of Computational and Applied Mathematics*, vol. 200, 2007, pp. 47-60.

[2] J.T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming. Second edition*, SIAM, Philadelphia 2010.

[3] L.T. Biegler, *Nonlinear Programming. Concepts, Algorithms, and Applications to Chemical Processes*, SIAM, Philadelphia 2010.

[4] K.E. Brenan, S.L. Campbell, L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM, Philadelphia, 1996.

[5] X.-C. Cai, D.E. Keyes, "Nonlinearly Preconditioned Inexact Newton Algorithms", *SIAM Journal on Scientific Computing*, vol. 24, 2002, pp. 183-200.

[6] M. Caracotsis, W. E. Stewart, "Sensitivity analysis of Initial Value Problems with mixed ODEs and algebraic equations", *Computers and Chemical Engineering*, vol. 9, 1985, pp. 359-365.

[7] P. Drąg, K. Styczeń, "A Two-Step Approach for Optimal Control of Kinetic Batch Reactor with electroneutrality condition", *Przegląd Elektrotechniczny (Electrical Review)*, vol. 6, 2012, pp. 176-180.

[8] R.S. Dembo, S.C. Eisenstat, T. Steihaug, "Inexact Newton Methods", *SIAM Journal on Numerical Analysis*, vol. 19, 1982, pp. 400-408.

[9] R.S. Dembo, T. Steihaug, "Truncated-Newton algorithm for large-scale unconstrained optimization", *Mathematical Programming*, vol. 26, 1983, pp. 190-212.

[10] S.C. Eisenstat, H.F. Walker, Globally convergent inexact Newton methods, *SIAM Journal on Optimization*, vol. 4, 1994, pp. 393-422.

[11] S.C. Eisenstat, H.F. Walker, "Choosing the forcing terms in an inexact Newton method", *SIAM Journal on Scientific Computing*, vol. 17, 1996, pp. 16-32.

[12] C.W. Gear, "The simultaneous numerical solution of differential-algebraic equations", *IEEE Transactions on Circuit Theory*, vol. 18, 1971, pp. 89-95.

[13] D.A. Knoll, D.E. Keyes, "Jacobian-free Newton-Krylov methods: a survey of approaches and applications", *Journal of Computational Physics*, vol. 193, 2004, pp. 357-397.

[14] L. Petzold, "Differential/Algebraic Equations are not ODEs", *SIAM Journal on Scientific Computing*, vol. 3, 1982, pp. 367-384.

[15] Y. Saad, M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems", *SIAM J. Sci. Stat. Comput.*, vol. 7, 1986, pp. 856-869.

[16] V.S. Vassiliadis, R.W.H. Sargent, C.C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems without Path Constraints", *Ind. Eng. Chem. Res.*, vol. 33, 1994, pp. 2111-2122.

[17] V.S. Vassiliadis, R.W.H. Sargent, C.C. Pantelides, "Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems with Path Constraints", *Ind. Eng. Chem. Res.*, vol. 33, 1994, pp. 2123-2133.