TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
AUTOMATIC CONTROL | AUTOMATYKA
4-AC/2013

PAWEŁ DRĄG, KRYSTYN STYCZEŃ*

THE INEXACT NEWTON BACKTRACKING METHOD AS A TOOL FOR SOLVING DIFFERENTIAL-ALGEBRAIC SYSTEMS

NIEDOKŁADNA METODA NEWTONA Z NAWROTAMI JAKO NARZĘDZIE DO ROZWIĄZYWANIA UKŁADÓW RÓŻNICZKOWO-ALGEBRAICZNYCH

Abstract

The classical inexact Newton method was presented as a tool for solving nonlinear differential-algebraic equations (DAEs) in a fully implicit form $F(y, y, t) = 0$. This is especially in chemical engineering where describing the DAE system in a different form can be difficult or even impossible to realize. The appropriate rewriting of the DAEs using the backward Euler method makes it possible to present the differential-algebraic system as a large-scale system of nonlinear equations. To solve the obtained system of nonlinear equations, the inexact Newton backtracking method was proposed. Because the convergence of the inexact Newton algorithm is strongly affected by the choice of the forcing terms, new variants of the inexact Newton method were presented and tested on the catalyst mixing problem.

Keywords: *differential-algebraic equations, systems of nonlinear equations, inexact Newton method*

Streszczenie

Klasyczna niedokładna metoda Newtona została przedstawiona jako narzędzie do rozwiązywania równań różniczkowo-algebraicznych zapisanych w formie niejawnej $F(y, y, t) = 0$. Zapisanie układu równań różniczkowo-algebraicznych w innej postaci w różnych zastosowaniach może być trudne lub niewskazane. Odpowiednie przekształcenie układów różniczkowo-algebraicznych z wykorzystaniem wstecznej metody Eulera umożliwia przedstawienie układu równań różniczkowo-algebraicznych jako układu równań nieliniowych dużej skali. W celu rozwiązania otrzymanego układu równań zaproponowano niedokładną metodę Newtona z nawrotami. Na zbieżność niedokładnej metody Newtona znacząco wpływa wybór czynnika wymuszającego. Nowe warianty niedokładnej metody Newtona zostały zastosowane do rozwiązania układu opisującego proces mieszania w obecności katalizatora.

Słowa kluczowe: *równania różniczkowo-algebraiczne, układy równań nieliniowych, niedokładna metoda Newtona*

* M.Sc. Paweł Drąg, e-mail: pawel.drag@pwr.wroc.pl, Prof. D.Sc. Ph.D. Krystyn Styczeń, Institute of Computer Engineering, Control and Robotics, Faculty of Electronics, Wrocław University of Technology.

Symbols

G, E, F	–	definition of functions,
h	–	length of the interval,
k	–	number of a current iteration,
n	–	number of mesh points,
p_1, p_2, p_3	–	parameters used in the new algorithm,
r	–	variable in the algorithm,
s	–	step of the algorithm,
t	–	independent variable, for example time,
u	–	control variable,
x	–	vector of decision variables,
y	–	differential variables,
z	–	algebraic variables,
$\beta, \varepsilon, \theta, \nu$	–	constants used in theorem,
$\alpha, \gamma, \vartheta, \omega$	–	parameters used in classical algorithms,
η	–	forcing term,
R	–	set of real numbers.

1. Introduction

Solving the differential-algebraic equations (DAEs) stands out as an important task in the scientific computing. Thus, methods for solving systems described by both differential and algebraic relations have gained increasing commercial importance over the last 30 years [19, 20]. With the increased academic research, potential applications of DAE systems have also been identified. Chemical engineering stands out as one of the most important application fields for these equations. In particular the control, modeling and identification of the chemical processes, which can take place in reactors, especially in the presence of a catalyst [3, 8].

These equations constitute the proper way to describe dynamic systems with slowly variable dynamics, as well as dynamic systems with conservation laws. The additional algebraic relations can model connections between the considered model and both the environment and the internal elements of the system. In this way, large-scale complex systems with dynamics and conservation laws can be designed and controlled [2].

In the article, attention was focused on the situation when the chemical process was described by the general differential-algebraic equations. It is important because in some cases, describing the system in a different form can be difficult or even impossible to realize [17].

The paper is constructed as follows. In the next section, the backward differentiation formula (BDF) is presented as an approach for discretization of the DAE systems. New aspects of the inexact Newton method were presented the in 3rd and 4th sections. The presented algorithms were tested on the catalyst mixing problem. The results are discussed in 5th section.

2. The backward Euler method

The backward differentiation formula was the first general technique for solving the differential-algebraic equations. The algorithm presented in [14] was the beginning of the specialized solvers for DAE systems.

The idea of this technique was 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.

At the beginning, the backward differentiation formula was defined for the systems of the differential equations coupled to algebraic equations. Then, this method was extended to apply any fully-implicit differential-algebraic systems:

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

where $G(\cdot)$ is the vector-valued DAE system.

The first order backward differentiation formula, known as the backward Euler method, is the simplest method for solving differential-algebraic systems. It consists of replacing the derivative in (1) by a backward difference:

$$E\left(\frac{y_n - y_{n-1}}{h}, y_n, z_n, t_n\right) = 0 \tag{2}$$

where $h = t_n - t_{n-1}$ and $E(\cdot)$ is a large-scale vector-valued system of nonlinear algebraic equations.

The simultaneous approach for solving algebraic systems is one of the main direction in modern optimization algorithms [3]. The system (2) can be solved by the Newton or inexact Newton methods [4]. Additionally, it was assumed, that $y(t_0)$ is known and t (for example time) is the independent variable. In the practical applications in chemical engineering, the length of the reactor can be used as the independent variable. If the time interval, in which the systems has to be considered is known, for numerical purposes, it can be scaled to the interval [0, 1].

3. The inexact Newton method

The approach given in the previous section and presented in eq. (2) can be expressed as:

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

System of algebraic equations presented in eq. (3) is often found in the scientific computing and stands a common point of a lot of the real-life engineering problems. One was assumed, that a nonlinear mapping F: $\mathrm{R}^n \rightarrow \mathrm{R}^n$ has the following properties:

(i) there exists an $x^* \in \mathrm{R}$, for which $F(x^*) = 0$,
(ii) F is continuously differentiable in a neighborhood of x^*,
(iii) $\det F'(x^*) \neq 0$.

The Newton method is one of the best known methods for solving system of nonlinear algebraic equations (3). This method converges quadratically from any sufficiently good initial solution. It can be computationally very expensive, especially, when the size of the problem is very large. Then, in each iteration of the algorithm, the Newton equation (4) should be solved:

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

where x_k is a solution at a current iterate and $F'(x_k)$ denotes the Jacobian matrix of $F(x_k)$ at point x_k.

The solution of the Newton equation (4) computed as $s_k = (F'(x_k))^{-1}F(x_k)$ is the Newton step and denoted s_k^N. In this way, the next iterate can be obtained by:

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

The system (4) can be solved by the inexact Newton method. This inexact method is any method which for given an initial guess x_0, generates a sequence x_k as presented in the Algorithm 1.

Algorithm 1. The inexact Newton method [10]
Begin
1. Given $x_0 \in \mathrm{R}^n$
2. For $k = 1, 2, ...,$ until x_k convergence
2.1. Choose some $\eta_k \in [0, 1)$
2.2. Inexactly solve the Newton equation (4) and obtain a step s_k such that:

$$F(x_k) + F'(x_k)s_k \leq \eta_k F(x_k) \tag{6}$$

2.3. Let $x_{k+1} = x_k + s_k$.
End

One can see, that η_k in the Algorithm 1 is the forcing term in the k-th iteration. Depending on the forcing term, the inexact Newton step s_k, which satisfies the inexact Newton condition (6), should be obtained.

The role of the forcing terms is to control the degree of accuracy of solving the Newton equation (4). Therefore, at each iteration step of the inexact Newton method, a value of $\eta_k \in [0, 1)$ has to be chosen. Then, the inexact Newton step s_k can be obtained by solving the Newton equation approximately.

The inexact Newton condition (6) reflects two important features of the iteration process:
(i) reflects the reduction in the norm of the local linear model,
(ii) accuracy in solving the Newton equation (4).

Theorem 1 ([10]). Assume that F: $R^n \to R^n$ is continuously differentiable, $x^* \in R^n$ such that $F'(x^*)$ is nonsingular. Let $0 < \eta_{max} < \beta < 1$ be the given constants. If the forcing terms $\{\eta_k\}$ in the inexact Newton method satisfy $\eta_k < \eta_{max} < \beta < 1$ 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 is converged to x^*, and:

$$\|x_{k+1} - x^*\|_* \le \beta \|x_k - x^*\|_* \tag{7}$$

where $\|\upsilon\|_* \le \|F'(x^*)\upsilon\|_*$.

By Theorem 1 the inexact Newton method is locally convergent, if the forcing terms $\{\eta_k\}$ are uniformly less than 1. The convergence rate is stated by the following theorem.

Theorem 2 ([10]). Assume that F: $R^n \to R^n$ is continuously differentiable, $x^* \in R^n$ such that $F'(x^*)$ is nonsingular. If the sequence $\{x_k\}$ generated by the inexact Newton method is converged to x^*, then $\{x_k\}$ is converged to x^*, superlinearly if $\eta_k \to 0$, or quadratically if $\eta_k = 0(\|F(x_k)\|)$.

From Theorem 2 one can see, that the convergence rate of the Algorithm 1 is determined by the appropriate choice of the sequence of the forcing terms.

Since the initial point for the inexact Newton method cannot be guaranteed to be near a solution of the nonlinear system, the inexact Newton method was globalized by backtracking strategy [12]. Inexact Newton backtracking method was presented as Algorithm 2.

Algorithm 2. The inexact Newton backtracking method [12]
Begin
1. Given $x_0 \in R^n$, $\eta_{max} \in [0, 1)$, $\alpha \in (0, 1)$, and $0 < \theta_{min} < \theta_{max} < 1$
2. For $k = 0, 1, 2, ...,$ until $\{x_k\}$ convergence
2.1. Choose some $\tilde{\eta}_k \in [0, \eta_{max}]$
2.2. Inexactly solve Newton equation (4) and obtain a step $\tilde{s}_k$, such that:

$$F(x_k) + F'(x_k)\tilde{s}_k \le \tilde{\eta}_k F(x_k) \tag{8}$$

2.3. Backtracking loop:
2.3.1. Let $s_k = \tilde{s}_k$, $\eta_k = \tilde{\eta}_k$
2.3.2. While $\|F(x_k + s_k)\| > [1 - \alpha(1 - \eta_k)]\|F(x_k)\|$
(i) Choose $\theta \in [\theta_{min}, \theta_{max}]$
(ii) Update $s_k \leftarrow \theta s_k$ and $\eta_k \leftarrow 1 - \theta(1 - \eta_k)$
2.4. Let $x_{k+1} = x_k + s_k$
End

In each iteration of the inexact Newton backtracking method, the backtracking loop along $\tilde{s}_k$ is implemented until the condition:

$$\|F(x_k + s_k)\| > [1 - \alpha(1 - \eta_k)]\|F(x_k)\| \tag{9}$$

is satisfied [12]. Equation (9) has been called the sufficient decrease condition. It has been used to guarantee that $||F(x_{k+1})||$ has a certain decrease in each iteration. In practical applications a positive integer has been given in advance to control the maximal backtracking loop number along $\tilde{s}_k$.

4. A choice of forcing terms

There are some strategies to determine preferable and effective sequences for forcing terms. Four of them were selected and presented. The first strategy was proposed in [11].

1st Given $\eta_0 \in [0, 1)$, then:

$$\eta_k = \min\left\{\frac{1}{k+2}, F\left(x_k\right)\right\} \tag{10}$$

Two others strategies presented in [13] can be expressed as.

2nd For given $\eta_0 \in [0, 1)$, then:

$$\eta_k = \begin{cases} \vartheta_k, & \eta_{k-1}^{(1+\sqrt{5})/2} \le 0.1, \\ \max\left\{\vartheta_k, \eta_{k-1}^{(1+\sqrt{5})/2}\right\}, & \eta_{k-1}^{(1+\sqrt{5})/2} > 0.1, \end{cases} \tag{11}$$

where $\vartheta_k = \dfrac{F\left(x_k\right) - F\left(x_{k-1}\right) - F'\left(x_{k-1}\right)s_{k-1}}{F\left(x_{k-1}\right)}$, $k = 1, 2, \ldots$

3rd Given $\gamma \in (0, 1]$, $\omega \in (1, 2]$, $\eta_0 \in [0, 1)$, choose:

$$\eta_k = \begin{cases} \gamma\left(\dfrac{F\left(x_k\right)}{F\left(x_{k-1}\right)}\right)^{\omega}, & \gamma\left(\eta_{k-1}\right)^{\omega} \le 0.1, \\ \max\left\{\gamma\left(\dfrac{F\left(x_k\right)}{F\left(x_{k-1}\right)}\right)^{\omega}, \gamma\left(\eta_{k-1}\right)^{\omega}\right\}, & \gamma\left(\eta_{k-1}\right)^{\omega} > 0.1, \end{cases} \tag{12}$$

4th The last presented strategy was introduced in [1]. Let us denote:

$$\mathrm{Ared}_k(s_k) = ||F(x_k)|| - ||F(x_k + s_k)|| \tag{13}$$

$$\mathrm{Pred}_k(s_k) = ||F(x_k)|| - ||F(x_k) + F'(x_k)s_k) \tag{14}$$

then

$$r_k = \frac{\mathrm{Ared}_k\left(s_k\right)}{\mathrm{Pred}_k\left(s_k\right)} \tag{15}$$

In the 4th approach, the forcing term η_k is adjusted depending on the value of r_k. One can distinguish four situations which might take a place in practical applications. If $r_k \approx 1$, then the local linear model and nonlinear model will agree well on their scale and $\|F(x)\|$ will usually be reduced. 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. If $r_k < 0$, then the local linear model and nonlinear model disagree and $\|F(x)\|$ will be enlarged. In the last situation, if $r_k >> 0$, then the local linear model and nonlinear model also disagree, but $\|F(x)\|$ will be reduced greatly. If $\mathrm{Pred}_k(s_k) = 0$, then the solution is achieved and computations are stopped. One can choose forcing terms according to the value of r_k:

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

where $0 < p_1 < p_2 < p_3 < 1$ are prescribed at first and $p_1 \in \left(0, \frac{1}{2}\right)$.

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

The 2nd strategy reflects the agreement between $F(x)$ and its local linear model at the previous step. The 3rd choice reflects the reduction rate of $\|F(x)\|$ from x_{k-1} to x_k.

5. Case study: Catalyst mixing problem

The main goal is to determine the optimal mixing policy of two catalysts along the length of a tubular reactor [15]. The mixing ratio of the catalysts represents the control variable. The formulation of the dynamic optimization problem was described by the system of differential-algebraic equations:

$$\max_u z_3(1.0) \tag{17}$$

subject to:

$$\dot{y}_1 - u\left(10y_2 - y_1\right) = 0 \tag{18}$$

$$\dot{y}_2 - u\left(y_1 - 10y_2\right) + \left(1-u\right)y_2 = 0 \tag{19}$$

$$z_3 + y_1 + y_2 - 1 = 0 \tag{20}$$

$$x(0) = [y_1(0), y_2(0), z_3(0)]^T = [1.0; 0.0; 0.0]^T \tag{21}$$

$$u(t) \in [0.0; 1.0] \tag{22}$$

There are some reasons to describe the process using the DAE model. Depending on the application, it may be difficult to reformulate the problem as an ODE especially, when nonlinearities are present. The algebraic equations typically describe conservation laws or explicit equality constraints and they should be kept invariant. Furthermore, it is easier to vary design parameters in an implicit model. In the presented application, the main advantage of a such formulation is that the implicit model does not require the modeling simplifications often necessary to get an ODE and the variables keep their original physical interpretation.

In our research we concentrated upon how to solve the DAE system (18)-(20) with known feasible initial conditions (21) and assumed control function $u(t) = t$. The DAE system was considered in the time domain $t \in [0.0; 1.0]$. Then, the equations were discretized into equidistant points with a distance of 0.001. It resulted in 2000 differential and 1000 algebraic state variables. Afterwards, 3000 equality constraints from the backward Euler method were imposed. The Jacobian matrix was obtained analytically and stored as a 1000×1000 sparse matrix. This large-scale system of the algebraic equations was solved using GMRES algorithm [5, 18]. The inexact Newton method was used with the forcing terms adjusted by four presented approaches.

The simulations were executed in the Matlab environment using Wroclaw Centre for Networking and Supercomputing for a vector of initial conditions.

Let us denote the initial conditions vector as follows:

$$x(0) = [x_{1,1}, ..., x_{1,1000}, x_{2,1}, ..., x_{2,1000}, x_{3,1}, ..., x_{3,1000}]^T \tag{23}$$

In this case the vector $x(0) = [1.0, ..., 1.0, 0.0, ..., 0.0, 0.0, ..., 0.0]^T$, and $||F(x_0)|| = 0.0258$. The initial conditions like presented in eq. (23) are feasible for the discretized DAE model only at the mesh points.

The simulations were executed with the parameters $\gamma = 0.5$, $\omega = 1.5$, for proposition 3 and $p_1 = 0.25$, $p_2 = 0.6$ and $p_3 = 0.8$ for the 4th proposition.

The results after 6 iterations for proposition 1 and initial conditions as presented in (23) were plotted on Fig.1.

Results presented in Table 1 indicate, that the solution obtained by the inexact Newton algorithm with forcing terms adjusted as in proposition 1, have a high accuracy in 6 iterations. A similar situation can be observed in Table 2 for proposition 2, but the convergence is slower. The forcing terms adjusted as in 4th proposition give progress, which is slower than the previous two, but comparable with results for the 3rd proposition.

Using the Modellica DASSL solver, the derivatives $\dot{y}(t)$ are approximated by backward differentiation formulae (BDF), and the resulting nonlinear system at each time-step is solved by the Newton method. The implemented in Matlab procedure ode15s, utilizes the same algorithms as Modellica. These approaches are definitely different to the one presented in our researches. They belongs to the sequential methods. In the article, a simultaneous approach to solving differential-algebraic systems was investigated. In solving the catalyst mixing problem described as the DAE model, the differences are unnoticeable.

Table 1

Results for proposition 1

Iteration k	Proposition 1	
	η_k	$\|\|F(x)\|\|$
1	0.0258	0.0012
2	0.0012	6.7233e-4
3	6.7233e-4	2.0389e-4
4	2.0389e-4	2.2336e-5
5	2.2336e-5	1.1930e-8
6	1.1930e-8	2.4685e-13

Table 2

Results for propositions 2 and 3

Iteration k	Proposition 2		Proposition 3	
	η_k	$\|\|F(x)\|\|$	η_k	$\|\|F(x)\|\|$
1	0.9000	0.0039	0.9000	0.0256
2	0.8433	0.0033	0.4917	0.0253
3	0.7589	0.0025	0.4917	0.0250
4	0.6400	0.0016	0.4917	0.0247
5	0.4857	7.6601e-4	0.4917	0.0244
6	0.3108	2.3772e-4	0.4917	0.0242
7	0.1510	3.5672e-5	0.4917	0.0239
8	1.2772e-11	6.6590e-9	0.4917	0.0236

Table 3

Results for proposition 4

Iteration k	Proposition 4		
	η_k	r_k	$\|\|F(x)\|\|$
1	0.5000	0.0145	0.0256
2	0.4000	0.0145	0.0253
3	0.4000	0.0145	0.0250
4	0.4000	0.0145	0.0247
5	0.4000	0.0145	0.0244
6	0.4000	0.0145	0.0242
7	0.4000	0.0145	0.0239
8	0.4000	0.0145	0.0236

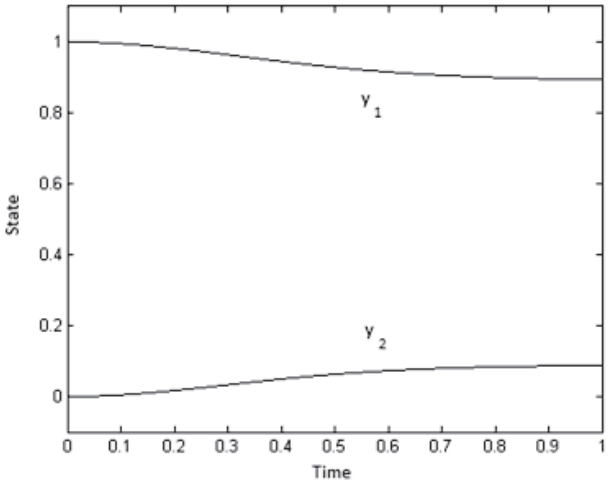

Fig. 1. Trajectory of the differential variables y_1 and y_2. Results for the forcing terms as in as proposition 1 after 6 iterations

6. Conclusion

In the article, new aspects of the inexact Newton methods for solving differential-algebraic models of the chemical processes in the fully implicit form were considered. The methods for the choice of the forcing terms for the inexact Newton method were presented and tested on the catalyst mixing problem.

The presented approach has some specific advantages which we want to emphasize. First of all, description a technological process using differential-algebraic equations enables us to exploit the system structure by problem-specific solvers. Hence, the implicit model does not require the modeling simplifications and the variables keep their original physical interpretation.

Rewriting the differential-algebraic model as a large-scale system of algebraic equations has a positive impact on the stability of the obtained solution. Recursive solutions do not have this property. This is the first step to take advantage of the effective numerical procedures for large and sparse matrices [3].

In the computations, matrix inversion was efficiently avoided.

Solving the large scale nonlinear differential-algebraic systems, especially with unbalanced nonlinearities, is a challenge. Attempts to solve these equations are still made [6]. One of the most famous examples of this type is the model of a kinetic batch reactor [7, 9].

For solving the Newton equation, the GMRES algorithm was used. The largest difficulty is associated with the calculation of the matrix-vector product. This calculation is time consuming and need storage of the matrix, which often has thousands of both rows and columns.

The future work will be concentrating on new reliable Jacobian-free Newton-Krylov methods, which could be successfully applied in optimal control of the chemical processes modeled by the nonlinear differential-algebraic equations.

This work was supported by the grant 'Młoda Kadra' B30036/I6 at Wroclaw University of Technology. The authors would like to thank all the reviewers for their helpful comments on an earlier version of the manuscript.

Referencess

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

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

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

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

[5] Brown P.N., Hindmarsh A.C., Petzold L.R., *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM Journal on Scientific Computing, Vol. 15, 1994, 1467-1488.

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

[7] Caracotsis M., Stewart W.E., *Sensitivity analysis of Initial Value Problems with mixed ODEs and algebraic equation*, Computers and Chemical Engineering, Vol. 9, 1985, 359-365.

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

[9] Drąg P., Styczeń K., *Inexact Newton method as a tool for solving differential-algebraic system*, Proceedings of the 2013 Federated Conference on Computer Science and Information Systems (FedCSIS), 2013, 639-642.

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

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

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

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

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

[15] Huang Y.J., Reklaitis G.V., Venkatasubramanian V., *Model decomposition based model for solving general dynamic optimization problems*, Computers and Chemical, Vol. 26, 2002, 863-873.

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

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

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

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

[20] Vassiliadis V.S., Sargent R.W.H., Pantelides C.C., *Solution of a Class of Multistage Dynamic Optimization Problems. 2. Problems with Path Constraints*, Ind. Eng. Chem. Res., Vol. 33, 1994, 2123-2133.