Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 597–602
DOI: 10.15439/2014F216
ACSIS, Vol. 2

Inexact Newton matrix-free methods for solving complex biotechnological 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.edu.pl

Marlena Kwiatkowska
Institute of Environmental Protection Engineering,
Faculty of Environmental Engineering,
Wrocław University of Technology
Pl. Grunwaldzki 9, 50-377 Wrocław, Poland
Email: marlena.kwiatkowska@pwr.edu.pl

***Abstract*—In the article a new approach for solving complex and highly nonlinear differential-algebraic equations (DAEs) was presented. An important kind of applications of DAE systems is modeling of biotechnological processes, which can have a very different course. An efficient solving of equations describing biotechnological industrial inlets results in better optimization of the processes and has a positive impact on the environment. Some of the mentioned processes were characterized by a highly nonlinear dynamics. To obtain the trajectories of the state numerically, the backward differentiation formula was used in the presented method. As a result, a large-scale system of nonlinear algebraic equations was obtained. To solve a such system, the inexact Newton matrix-free approach was proposed. The new algorithm was tested on a mathematical model of a fed-batch fermentor for penicillin production. The numerical simulations were executed in MATLAB using Wroclaw Center for Networking and Supercomputing.**

Keywords*—inexact Newton methods, matrix-free methods, DAE systems, system of nonlinear equations.

I. Introduction

NOWADAYS, biotechnological processes are widely used in many real-life industrial plants. They can be often met in food industry, production of medicines and in many other sectors of industry, especially, when biodegradable components have to be used for the protection of the environment [16], [21].

Very often, the mathematical models of the bioprocesses have highly nonlinear dynamics. Technological and resources constraints on both the state and the control variables are also frequently present. Hence, a commonly used way to describe complex processes are both nonlinear ordinary differential equations and differential-algebraic equations [3], [4].

In recent years, many efforts have been devoted to the model-based optimization of processes in biotechnology and bioengineering. An example of a problem which has received major attention is the dynamic optimization of fed-batch bioreactors [17]. Dynamic optimization allows the computation of the optimal operating policies for these units to ensure the maximization of a predefined performance index. The performance index reflects a productivity or an economical index derived from both the operation profile and the final concentrations [22].

The development of information technology, robust numerical methods and computing capacity, enables to obtain optimal operating policies of the complex biotechnological processes. An efficient solving of the differential-algebraic systems enables the use of optimization strategies, what can improve a process flow significantly [5], [10].

In this work, the general problem of solving dynamical models of bioprocesses described by nonlinear differential-algebraic equations was considered. A solution strategy based on the matrix-free inexact Newton method was presented.

The article consists of 5 sections. The problem of solving complex and highly nonlinear differential-algebraic equations (DAEs) will be introduced in the next section. In the 3*rd* section the matrix-free Newton-Krylov method will be presented. The inexact Newton method will be discussed in the 4*th* section. The inexact Newton matrix-free approach will be tested on the fed-batch fermentor for penicillin production. The numerical results will be presented in the 5*th* section.

II. Statement of the problem

In general, real-life biotechnological systems with dynamics and conservation laws can be described in a *fully-implicit* form

$$\mathcal{B}\Big(\dot{y}(t), y(t), z(t), u(t), p, t\Big) = 0. \tag{1}$$

Here $y(t) \in \mathcal{R}^{n_y}$ represents the differential state trajectory, whereas $z(t) \in \mathcal{R}^{n_z}$ denotes the algebraic state trajectory, $u(t) \in \mathcal{R}^{n_u}$ a vector representing control function and $p \in \mathcal{R}^{n_p}$ indicates a vector of parameters constant in the time. Then, the nonlinear vector-valued function is considered

$$\mathcal{B} : \mathcal{R}^{n_y \times n_z \times n_u \times 1} \rightarrow \mathcal{R}^{n_\mathcal{B}}. \tag{2}$$

On the other hand, when only dynamical features of the systems are pondered, the ordinary differential equations are enough

$$\dot{y}(t) = \mathcal{G}\big(y(t), u(t), p, t\big). \tag{3}$$

Hence, some interesting relations between variables and their physical interpretations can be lost [6].

The first general technique for the numerical solution of the *fully-implicit* DAEs was the backward differential formula. The idea of this technique was that the derivative $\dot{y}(t)$ could be approximated by a linear combination of the solution $y(t)$ at the current mesh point and at several previous mesh points [19].

Previously, the backward differential formula was defined for the differential equations systems coupled to the algebraic

978-83-60810-58-3/$25.00

equations. The application of this method was soon extended to any fully-implicit system of the differential-algebraic equations.

The first order backward differential formula has been considered as the simplest method for solving differential-algebraic systems [14]. It consists of replacing the derivative in eq. (1) by the backward difference quotient

$$F\Big(\frac{y_{n+1}-y_n}{h}, y_{n+1}, z_{n+1}, t_{n+1}\Big) = 0. \tag{4}$$

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

This procedure results in system of nonlinear equations for y_{n+1} at each step. To obtain the solution from time t_n to time t_{n+1}, the system of equations (4) should be solved.

There are two main assumptions to solve the system (4). The initial value $y(t_0)$ is known and t (time) is the independent variable.

In practical applications, if the time interval, in which the system has to be considered, is known, it can be scaled to the interval $[0, 1]$.

The presented methodology leads to the following equation

$$F(\chi) = 0. \tag{5}$$

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

(1) There exists a point $\chi^* \in \mathcal{R}^n$ with $F(\chi^*) = 0$.

(2) F is continuously differentiable in a neighborhood of χ^*.

(3) The Jacobian matrix $F'(\chi^*) \equiv \mathcal{J}(\chi^*)$ is nonsingular and for

$$F(\chi) = [F_1, F_2, \cdots, F_n] \tag{6}$$

and

$$\chi \in \mathcal{R}^n, \tag{7}$$

the (i, j)th element (ith row, jth column) of the Jacobian matrix is calculated as

$$\mathcal{J}_{i,j} = \frac{\partial F_i(\chi)}{\partial \chi_j}. \tag{8}$$

There have been a lot of methods for solving the nonlinear equations (5). The most popular and important are both the Newton and different variations of the inexact Newton methods [18].

III. Matrix-free Newton-Krylov method

The matrix-free Newton-Krylov method stands the iterative approach consisting of some nested levels, generally, from two to four. The name of the method come from the primary levels, which are *the Newton correction step* and the loop building up *the Krylov subspace*, out of which each Newton correction is computed [15].

In some applications two additional levels are present. There is *a preconditioner* in the interior to the Krylov loop, and, outside of the Newton loop, *a globalization method* is often required.

A. Newton method

The Newton iteration for $F(\chi) = 0$ derives from a multivariate Taylor expansion about a current point χ_k

$$F(\chi_{k+1}) = F(\chi_k) + F'(\chi_k)(\chi_{k+1} - \chi_k) + \cdots. \tag{9}$$

Neglecting the terms of the higher-order curvature and setting the left-hand side to zero yields a strict Newton method. It is as an iterative process of solving the sequence of the linear systems

$$\mathcal{J}(\chi_k)\delta\chi_k = -F(\chi_k), \tag{10}$$

to obtain $\delta\chi_k$ and to determine

$$\chi_{k+1} = \chi_k + \delta\chi_k, \qquad k = 0, 1, \cdots, \tag{11}$$

where the starting point χ_0 is given, $F'(\chi)$ is a vector-valued function of nonlinear residuals, $\mathcal{J}(\chi)$ is the Jacobian matrix associated with $F'(\chi)$, χ stands the state vector to be found, and k is a nonlinear iteration index.

The Newton iteration is terminated based on a required decrease in the norm of the nonlinear residual

$$\frac{\|F(\chi_k)\|}{\|F(\chi_0)\|} < \Delta_{res}, \tag{12}$$

and a sufficiently small Newton update

$$\frac{\|\delta\chi_k\|}{\|\chi_k\|} < \Delta_{update}. \tag{13}$$

In a scalar example, there is a one-to-one mapping between grid points and rows in the Jacobian. But forming each element of $\mathcal{J}$ requires taking analytic or discrete derivatives of the system of equations with respect to χ. This can be both time consuming and possible source of error for many problems in control and optimization of the biotechnological processes.

B. Krylov method

Krylov subspace methods are approaches for solving large-scale linear systems. They are projection or generalized projection methods for solving

$$A\chi = b, \tag{14}$$

using the Krylov subspace $\mathcal{K}_j$ defined as

$$\mathcal{K}_j = span(r_0, Ar_0, A^2r_0, \cdots, A^{j-1}r_0), \tag{15}$$

where $r_0 = b - A\chi_0$.

These methods require only matrix-vector products, not the individual elements of the matrix A, to perform the iteration. This is the key to their use with the Newton method.

C. Matrix-free Newton-Krylov methods

In the matrix-free Newton–Krylov approach, a Krylov method is used to solve the linear system of equation given by eq. (10). For the Newton step, an initial linear residual r_0 is defined, and an initial guess $\delta\chi_0$ is given

$$r_0 = -F(\chi) - \mathcal{J}(\chi)\delta\chi_0. \tag{16}$$

The nonlinear iteration index k has been omitted, because the Krylov iteration is performed at a fixed k. Let j be the Krylov iteration index. Since the Krylov solution is a Newton correction, and a locally optimal move was just made in the direction of the previous Newton correction, the initial iterate for the Krylov iteration for $\delta\chi_0$ is typically zero. This is asymptotically a reasonable guess in the context of the Newton step, as the converged value for $\delta\chi_0$ should approach zero in late Newton iterations.

When the Generalized Minimal RESidual method (GMRES) is used, in the jth iteration $\|\mathcal{J}\delta\chi_j + F(\chi)\|_2$ is minimized within a subspace of small dimension, relative to the number of unknowns, in a least-square sense [20]. $\delta\chi_j$ is drawn from the subspace spanned by the Krylov vectors, $\{r_0, \mathcal{J}r_0, \mathcal{J}^2 r_0, \cdots, \mathcal{J}^{j-1}r_0\}$, and can be written as

$$\delta\chi_j = \sum_{i=0}^{j-1} \beta_i \mathcal{J}^i r_0, \tag{17}$$

where the scalars β_i minimize the residual.

Upon examining eq. (17) one can see, that GMRES requires the Jacobian only in the form of the matrix-vector products, which may be approximated by

$$\mathcal{J}v \approx [F(\chi + \varepsilon v) - F(\chi)]/\varepsilon, \tag{18}$$

where ε is a small perturbation.

Equation (18) is a first order Taylor series expansion approximation to the product of the Jacobian $\mathcal{J}$ and a vector v.

In a simple case, when the two coupled nonlinear equations are considered $F_1 = (\chi_1, \chi_2) = 0, F_2 = (\chi_1, \chi_2) = 0$, the Jacobian matrix takes a form

$$\mathcal{J} = \begin{bmatrix} \frac{\partial F_1}{\partial \chi_1} & \frac{\partial F_1}{\partial \chi_2} \\ \\ \frac{\partial F_2}{\partial \chi_1} & \frac{\partial F_2}{\partial \chi_2} \end{bmatrix}. \tag{19}$$

The matrix-free Newton-Krylov method does not require the formation of this matrix. Instead, a result vector, that approximates this matrix multiplied by a vector, was formed.

$$\frac{F(\chi + \varepsilon v) - F(\chi)}{\varepsilon} = \begin{bmatrix} \frac{F_1(\chi_1+\varepsilon v_1, \chi_2+\varepsilon v_2) - F_1(\chi_1,\chi_2)}{\varepsilon} \\ \\ \frac{F_2(\chi_1+\varepsilon v_1, \chi_2+\varepsilon v_2) - F_2(\chi_1,\chi_2)}{\varepsilon} \end{bmatrix}. \tag{20}$$

Approximation of $F(\chi+\varepsilon v)$ with a first order Taylor series expansion about χ takes a form

$$F'(\chi_1, \chi_2) \approx \begin{bmatrix} \frac{F_1(\chi_1,\chi_2) + \varepsilon v_1 \frac{\partial F_1}{\partial \chi_1} + \varepsilon v_2 \frac{\partial F_1}{\partial \chi_2} - F_1(\chi_1,\chi_2)}{\varepsilon} \\ \\ \frac{F_2(\chi_1,\chi_2) + \varepsilon v_1 \frac{\partial F_2}{\partial \chi_1} + \varepsilon v_2 \frac{\partial F_2}{\partial \chi_2} - F_2(\chi_1,\chi_2)}{\varepsilon} \end{bmatrix}, \tag{21}$$

which simplifies

$$\mathcal{J}v = \begin{bmatrix} v_1 \frac{\partial F_1}{\partial \chi_1} + v_2 \frac{\partial F_1}{\partial \chi_2} \\ \\ v_1 \frac{\partial F_2}{\partial \chi_1} + v_2 \frac{\partial F_2}{\partial \chi_2} \end{bmatrix}. \tag{22}$$

The error in this approximation is proportional to ε.

The most attractive advantages of the matrix-free approach is a Newton-like nonlinear convergence without costs of forming and storing the true Jacobian. In practice, one forms a matrix for preconditioning purposes. However, the matrices employed in preconditioning can be simpler than true Jacobian of the problem, so the algorithm is properly said to be *Jacobian-free* [15].

Since the use of an iterative technique to solve eq. (10) does not require the exact solution of the linear system, the resulting algorithm is categorized as *the inexact Newton method*.

IV. Inexact Newton method

The Newton method is attractive because its quadratically rate of convergence from any sufficiently good initial point. But the computational cost can be expensive, especially, when the size of the problem is very large. In each iteration step the Newton equation

$$F(\chi_k) + \mathcal{J}(\chi_k)\delta\chi_k = 0 \tag{23}$$

should to be solved. Here χ_k denotes the current iterate, and $\mathcal{J}(\chi_k)$ is the Jacobian matrix of $F(x)$ at point χ_k. The solution $\delta\chi_k^N$ of the Newton equation is known as the Newton correction or the Newton step. Once the Newton step is obtained, the next iterate is given by

$$\chi_{k+1} = \chi_k + \delta\chi_k^N. \tag{24}$$

The inexact Newton method is a generalization of the Newton method [8], [12]. It is any method, which for given an initial guess χ_0, generates a sequence χ_k of approximations to χ^* as in Algorithm 1.

ALGORITHM 1. The inexact Newton method
1. Given $\chi_0 \in \mathcal{R}^n$
2. For $k = 0, 1, 2, \cdots$ until χ_k converges
   2.1 Choose some $\eta_k \in [0, 1)$
   2.2 Inexactly solve the Newton equation (10) and obtain a step $\delta\chi_k$, such that
   $\|F(\chi_k) + \mathcal{J}(\chi_k)\delta\chi_k\| \le \eta_k \|F(\chi_k)\|$. $(\star)$
   2.3 Let $\chi_{k+1} = \chi_k + \delta\chi_k$.

In the Algorithm 1, η_k is the forcing term in the kth iteration, $\delta\chi_k$ is the inexact Newton step and $(\star)$ is the inexact Newton condition.

In each iteration step of the inexact Newton method, a real number $\eta_k \in [0, 1)$ should be chosen. Then the inexact

Newton step $\delta\chi_k$ was obtained by solving the Newton equation approximately.

Since $F(\chi_k) + \mathcal{J}(\chi_k)\delta\chi_k$ is both residual of the Newton equations and the local linear model of $F(\chi)$ at χ_k, the inexact Newton condition ($\star$) reflects both the reduction in the norm of the local linear model and certain accuracy in solving the Newton equations. In this way, the role of forcing terms is to control the accuracy degree 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 continuously differentiable, $\chi^* \in \mathcal{R}^n$ such that $\mathcal{J}(\chi^*)$ is nonsingular. Let $0 < \eta_{max} < \beta < 1$ be the given constants. If the forcing terms η_k in the inexact Newton method satisfy $\eta_k \leq \eta_{max} < \beta < 1$ for all k, then there exists $\varepsilon > 0$, such that for any $\chi_0 \in N_\varepsilon(\chi^*) \equiv \{\chi : \|\chi - \chi^*\| < \varepsilon\}$, the sequence $\{\chi_k\}$ generated by the inexact Newton method converges to χ^*, and

$$\|\chi_{k+1} - \chi^*\|_* \leq \beta\|\chi_k - \chi^*\|_*, \tag{25}$$

where $\|v\|_* = \|\mathcal{J}(\chi^*)v\|$.

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 continuously differentiable, $\chi^* \in \mathcal{R}^n$ such that $\mathcal{J}(\chi^*)$ is nonsingular. If the sequence $\{\chi_k\}$ generated by the inexact Newton method converges to χ^*, then

(1) χ_k converges to χ^* superlinearly when $\eta_k \to 0$;

(2) χ_k converges to χ^* quadratically if $\eta_k = \mathcal{O}(\|F(\chi_k)\|)$ and $\mathcal{J}(\chi)$ is Lipschitz continuous at χ^*.

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

Various ways for selection the forcing terms have been widely discussed and tested in [1] and [13].

V. Case study

As the case study a fed-batch reactor for the production of penicillin [2] was considered. The objective was to maximize the amount of penicillin using the feed rate as the control variable. The duration of the process was specified at 120 hours.

The mathematical statement of the dynamical optimization problem is as follows.

Find $u(t)$ and t_f over $t \in [t_0, t_f]$ to maximize

$$J = x_2(t_f) \cdot x_4(t_f) \tag{26}$$

subject to differential-algebraic system

$$\frac{dx_1}{dt} = h_1 x_1 - u\left(\frac{x_1}{500x_4}\right), \tag{27}$$

$$\frac{dx_2}{dt} = h_2 x_1 - 0.01x_2 - u\left(\frac{x_2}{500x_4}\right), \tag{28}$$

$$\frac{dx_3}{dt} = -h_1\frac{x_1}{0.47} - h_2\frac{x_1}{1.2} - x_1\frac{0.029x_3}{0.0001 + x_3} + \frac{u}{x_4}\left(1 - \frac{x_3}{500}\right), \tag{29}$$

$$\frac{dx_4}{dt} = \frac{u}{500}, \tag{30}$$

$$h_1 = 0.11\left(\frac{x_3}{0.006x_1 + x_3}\right), \tag{31}$$

$$h_2 = 0.0055\left(\frac{x_3}{0.0001 + x_3(1 + 10x_3)}\right), \tag{32}$$

where x_1, x_2 and x_3 are the biomass, penicillin and substrate concentration (g/L), and x_4 is the volume (L). The initial conditions are

$$x(t_0) = [1.5 \quad 0 \quad 0 \quad 7]^T. \tag{33}$$

There are several path constraints for state variables

$$0 \leq x_1 \leq 40, \tag{34}$$

$$0 \leq x_2 \leq 25, \tag{35}$$

$$0 \leq x_3 \leq 10. \tag{36}$$

The upper and lower bounds on the control variable (feed rate of substrate) are

$$0 \leq u \leq 50. \tag{37}$$

The control problem of the fed-batch fermentor for penicillin production was solved with the matrix-free inexact Newton method, presented in the article.

At first, the overall time domain was divided into 1200 equidistant intervals. The resulting model consisted of 7200 nonlinear algebraic equations and the same number of variables and it was of the form

$$x_{1,n+1} - x_{1,n} - \Delta t\Big(h_{1,n+1}x_{1,n+1} - u\frac{x_{1,n+1}}{500x_{4,n+1}}\Big) = 0, \tag{38}$$

$$\vdots$$

$$x_{4,n+1} - x_{4,n} - \Delta t\frac{u}{500} = 0, \tag{39}$$

$$h_{1,n+1} - \Delta t\Big(0.11 \times \frac{x_{3,n+1}}{0.006x_{1,n+1} + x_{3,n+1}}\Big) = 0, \tag{40}$$

$$h_{2,n+1} - \Delta t\Big(0.0055 \times \frac{x_{3,n+1}}{0.0001 + x_{3,n+1}(1 + 10x_{3,n+1})}\Big) = 0, \tag{41}$$

for $\quad n = 0, 1, \cdots, 1200.$

The initial conditions were known only for the first stage $n = 0$. In this way, there are 7200 decision variables connected with initial values for both differential and algebraic state variables. There is one variable, which is the assumed value of the feed rate and represents the control variable.

The initial values for the decision variables were as follows

$$\chi_{1,x_{1,1}}, \cdots, \chi_{1200,x_{1,1200}} = 1.5, \tag{42}$$

$$\chi_{1201,x_{2,1}}, \cdots, \chi_{2400,x_{2,1200}} = 0.0, \tag{43}$$

$$\chi_{2401,x_{3,1}}, \cdots, \chi_{3600,x_{3,1200}} = 0.0, \tag{44}$$

$$\chi_{3601,x_{4,1}}, \cdots, \chi_{4800,x_{4,1200}} = 7.0, \tag{45}$$

$$\chi_{4801,h_{1,1}}, \cdots, \chi_{6000,h_{1,1200}} = 10.0, \tag{46}$$

$$\chi_{6001,h_{2,1}}, \cdots, \chi_{7200,h_{2,1200}} = 10.0, \tag{47}$$

In the simulations the following rule choice of the forcing terms was used

$$\eta = \min\left\{\frac{1}{k_{iter}+2}, \|F(\chi_{iter})\|\right\}, \tag{48}$$

where $iter$ denotes the number of the previously iterate [9].

For the constant control function u, the final value of the objective function was $81.1943g$. The obtained value of the control function was $u_{const} = 12.5000$. The assumed duration of the whole process was adjusted to 120 hours. Simulations were performed with the accuracy $\Delta_{res} = \Delta_{update} = 10^{-6}$. There are the optimal trajectories of both the biomass and penicillin concentrations in the Fig. 1.

In the simulation for solving the Newton equation (10), the Generalized Minimal RESidual method (GMRES) was used. In GMRES, the Arnoldi basis vector form the trial subspace out of which the solution was constructed. One matrix-vector product was required per iteration to create each new trial vector, and the iterations are terminated based on a by-product estimate of the residual that does not require explicit construction of intermediate residual vector of solutions. It was a major beneficial feature of the algorithm.

In the case study, the Jacobian matrix in the Newton equation consisted on more than $50 \cdot 10^6$ cells. It means, that the matrix-vector product would be impossible to obtain by ordinary methods.

The first proposition was to use the sparsity of the matrix, especially for the storage and speed-up of the computations. In the Jacobian matrix only 0.048% elements has another value than zero. The second proposition is the Jacobian-free approach.

These two remarks, enables us to solve the fed batch fermentor for penicillin production described by the nonlinear differential-algebraic equations.

The numerical simulations were executed in MATLAB using Wroclaw Center for Networking and Supercomputing

VI. Conclusion

In this paper the new approach for solving the nonlinear differential-algebraic equations in the *fully-implicit* form was presented. The method consists of two main remarks. The first, that the Newton equation can be solved inexactly. The appropriate choice of the forcing terms to obtain the well behaved inexact Newton method preserve locally the superlinearly convergence rate. The second remark is that, the matrix-free approach enables us to consider a large-scale systems

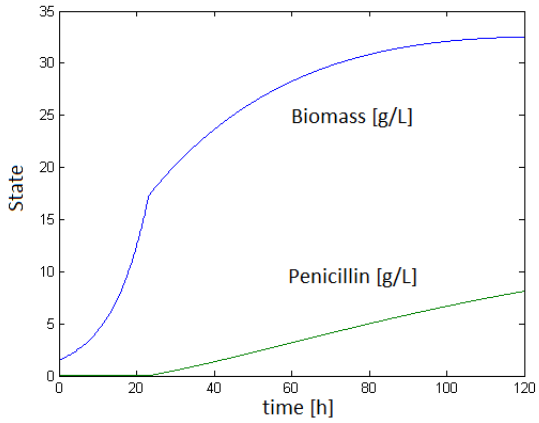

Fig. 1. The optimal trajectories of both the biomass and penicillin concentrations.

with thousands of variables. The sparse representation of the Jacobian matrix and a function, which calculate the matrix-vector product effectively, makes large-scale computations possible.

The algorithm was tested on the nonlinear DAE system, which described the fed batch fermentor for penicillin production. The discretized large-scale model consisted on 7200 nonlinear algebraic equations and the same number of variables.

The presented approach can be applied in real-life industrial plants, to optimize and control the biotechnological processes [7], [23]. The high degree of utilization of resources ensures a high profit and negligible waste.

At the next step, the new preconditioned Jacobian-free algorithms, which could solve large-scale optimization problems efficiently, will be studied and adjusted for new challenges in solving the dynamical optimization problems [11].

Acknowledgment

The project was supported by the grant Młoda Kadra B30036 at Wrocław University of Technology.

References

[1] H.-B. An, Z.-Y. Mo, X.-P. Liu. 2007. A choice of forcing terms in inexact Newton method. Journal of Computational and Applied Mathematics. 200:47-60, http://dx.doi.org/10.1016/j.cam.2005.12.030.

[2] J.R. Banga, E. Balsa-Canto, C.G. Moles, A.A. Alonso. 2005. Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. Journal of Biotechnology. 117:407-419, http://dx.doi.org/10.1016/j.jbiotec.2005.02.013.

[3] J.T. Betts. 2010. Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition. SIAM, Philadelphia, http://dx.doi.org/10.1137/1.9780898718577.

[4] L.T. Biegler. 2010. Nonlinear Programming. Concepts, Algorithms and Applications to Chemical Processes. SIAM, Philadelphia, http://dx.doi.org/10.1137/1.9780898719383.

[5] L.T. Biegler, S. Campbell, V. Mehrmann. 2012. DAEs, Control, and Optimization. Control and Optimization with Differential-Algebraic Constraints. SIAM, Philadelphia, http://dx.doi.org/10.1137/9781611972252

[6] K.E. Brenan, S.L. Campbell, L.R. Petzold. 1996. Numerical Solution of Initial- Value Problems in Differential-Algebraic Equations. SIAM, Philadelphia, http://dx.doi.org/10.1137/1.9781611971224

[7] R. Brunet, G. Guillen-Gosalbez, L. Jimenez. 2010. Cleaner design of single-product biotechnological facilities through the integration of process simulation, multiobjective optimization, life cycle assessment, and principal compopnent analysis. Ind. Eng. Chem. Res. 51:410-424, http://dx.doi.org/10.1021/ie2011577.

[8] R.S. Dembo, S.C. Eisenstat, T. Steihaug. 1982. Inexact Newton Methods. SIAM Journal on Numerical Analysis. 19:400-408, http://dx.doi.org/10.1137/0719025.

[9] R.S. Dembo, T. Steihaug. 1983. Truncated-Newton algorithm for large-scale unconstrained optimization. Mathematical Programming. 26:190-212, http://dx.doi.org/10.1007/BF02592055.

[10] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, F. Allgower. 2002. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Journal of Process Control. 12:577-585, http://dx.doi.org/10.1016/S0959-1524(01)00023-3.

[11] P. Drąg, K. Styczeń. 2012. A Two-Step Approach for Optimal Control of Kinetic Batch Reactor with electroneutrality condition. Przegląd Elektrotechniczny. 6:176-180.

[12] S.C. Eisenstat, H.F. Walker. 1994. Globally convergent inexact Newton methods. SIAM Journal on Optimization. 4:393-422, http://dx.doi.org/10.1137/0804022.

[13] S.C. Eisenstat, H.F. Walker. 1996. Choosing the forcing terms in an inexact Newton method. SIAM Journal on Scientific Computing. 17:16-32, http://dx.doi.org/10.1137/0917003.

[14] C.W. Gear. 1971. The simultaneous numerical solution of differential-algebraic equations. IEEE Transactions on Circuit Theory. 18:89-95, http://dx.doi.org/10.1109/TCT.1971.1083221.

[15] D.A. Knoll, D.E. Keyes. 2004. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. Journal of Computational Physics. 193:357-397, http://dx.doi.org/10.1016/j.jcp.2003.08.010.

[16] M. Kwiatkowska. 2012. Antimicrobial PVC composites. Processing technologies and functional properties of polymer nanomaterials for food packaging : International COST Workshop, Wroclaw, Poland, September 11-12, pp. 40-41.

[17] D. Niu, M. Jia, F. Wang, D. He. 2013. Optimization of nosiheptide fed-batch fermentation process based on hybrid model. Ind. Eng. Chem. Res. 52:3373-3380, http://dx.doi.org/10.1021/ie3022169.

[18] J. Nocedal, S.J. Wright. 2006. Numerical Optimization. Second Edition. Springer, New York, http://dx.doi.org/10.1007/978-0-387-40065-5

[19] L. Petzold. 1982. Differential/Algebraic Equations are not ODEs. SIAM Journal on Scientific Computing. 3:367-384, http://dx.doi.org/10.1137/0903023.

[20] Y. Saad, M. H. Schultz. 1986. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. 7:856-869, http://dx.doi.org/10.1137/0907058.

[21] D.L. Stoner, A.P. Poloski, J.A. Johnson, C.R. Tolle. 2001. Optimization and Control of Dynamic Bioprocesses. Organic Process Research and Development. 5:299-307, http://dx.doi.org/10.1021/op0100091.

[22] V.S Vassiliadis, R.W.H. Sargent, C.C. Pantelides. 1994. Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems without Path Constraints. Ind. Eng. Chem. Res. 33:2111-2122, http://dx.doi.org/10.1021/ie00033a014.

[23] S.R.R. Vetukuri, L.T. Biegler, A. Walther. 2010. An inexact trust-region algorithm for the optimization of periodic adsorption processes. Ind. Eng. Chem. Res. 49:12004-12013, http://dx.doi.org/10.1021/ie100706c.