A MATLAB PACKAGE FOR ORTHOGONAL COLLOCATIONS ON FINITE ELEMENTS IN DYNAMIC OPTIMISATION

M. Čižniar * **D. Salhi** ** **M. Fikar** * **M.A. Latifi** **

* *Department of Information Engineering and Process Control,*
Faculty of Chemical and Food Technology, STU
Radlinského 9, 83102 Bratislava
fax : +421 2 52496469 and e-mail : miroslav.fikar@stuba.sk
** *Laboratoire des Sciences du Génie Chimique*
CNRS-ENSIC, B.P. 451, 1 rue Grandville, 54001 Nancy Cedex
fax : +33 (0)3 83 17 53 26 and e-mail : latifi@ensic.inpl-nancy.fr

Abstract: This paper deals with dynamic optimisation of processes. It consists in searching for optimal profiles of decision variables which optimise a given performance index under specified constraints. The method of orthogonal collocations on finite elements has been developed and implemented within MATLAB environment. The original optimisation problems are then converted into NLP problems which are solved using appropriate NLP solvers, i.e. SQP methods. The gradients of the performance index as well as constraints needed in the NLP solver are analytically computed using formal calculus. Several applications are successfully tested.

Keywords: dynamic optimisation, orthogonal collocations, finite elements, process applications

1. INTRODUCTION

The objective of dynamic optimisation is to determine, in open loop control, a set of decision variable time profiles (pressure, temperature, flow rate, current, heat duty, . . .) for a dynamic system that optimise a given performance index (or cost functional or optimisation criterion)(cost, time, energy, selectivity, . . .) subject to specified constraints (safety, environmental and operating constraints). Optimal control refers to the determination of the best time-varying profiles in closed loop control.

The numerical methods used for the solution of dynamic optimisation problems may be grouped into two categories: indirect and direct methods. In this work only direct methods are considered. In this category, there are two strategies: sequential method and simultaneous method. The sequential strategy, often called control vector parameterisation (CVP), consists in an approximation of the control trajectory by a function of only a few parameters and leaving the state equations in the form of the original differential algebraic equation (DAE) system (Goh and Teo, 1988). In the simultaneous strategy, both the control and state variables are discretised using polynomials (e.g., Lagrange polynomials) of which the coefficient become the decision variables in a much larger NLP problem (Cuthrell and Biegler, 1987).

In this paper, the method of orthogonal collocation is developed. Moreover, the finite elements are used in order to handle sharp variations or control discontinuities.

In the next section we review the general NLP formulation for optimal control problems using orthogonal collocation on finite elements method, which is implemented in the dynamic optimisation package (*dynopt*). In section 3, we present some examples which are then solved and discussed in section 4.

The main aim of this work that is to implement a user friendly interface to dynamic optimisation based on orthogonal collocation within the MATLAB environment.

2. NLP FORMULATION PROBLEM

In this paper, it is assumed that the dynamic model can be described by a set of ordinary differential equations (ODE).

Consider the following general control problem for $t \in [a, b]$

$$\min_{\boldsymbol{u}(t)} \{\Psi[\boldsymbol{x}(b)] + \int_a^b G[\boldsymbol{x}(t), \boldsymbol{u}(t)]\mathrm{d}t\} \tag{1}$$

such that

$$\begin{gathered}\dot{\boldsymbol{x}}(t) = F[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \\ \boldsymbol{x}(a) = \boldsymbol{x}_0 \\ \boldsymbol{h}[\boldsymbol{u}(t), \boldsymbol{x}(t)] = \boldsymbol{0} \\ \boldsymbol{g}[\boldsymbol{u}(t), \boldsymbol{x}(t)] \le \boldsymbol{0} \\ \boldsymbol{x}(t)^L \le \boldsymbol{x}(t) \le \boldsymbol{x}(t)^U \\ \boldsymbol{u}(t)^L \le \boldsymbol{u}(t) \le \boldsymbol{u}(t)^U\end{gathered}$$

where

$\Psi[\boldsymbol{x}(b)]$ – component of objective function evaluated at final conditions,
$\int_a^b G[\boldsymbol{x}(t), \boldsymbol{u}(t)]\mathrm{d}t$ – component of objective function over a period of time,
$\boldsymbol{h}$ – equality design constraint vector,
$\boldsymbol{g}$ – inequality design constraint vector,
$\boldsymbol{x}(t)$ – state profile vector,
$\boldsymbol{u}(t)$ – control profile vector,
$\boldsymbol{x}_0$ – initial conditions for state vector,
$\boldsymbol{x}(t)^L, \boldsymbol{x}(t)^U$ – state profile bounds,
$\boldsymbol{u}(t)^L, \boldsymbol{u}(t)^U$ – control profile bounds.

In order to derive the NLP problem the differential equations are converted into algebraic equations using collocations on finite elements. Residual equations are then formed and solved as a set of algebraic equations. These residuals are evaluated at the shifted roots of the Legendre polynomial. The procedure is then following: Consider the initial-value problem over a finite element i with time $t \in [\zeta_i, \zeta_{i+1}]$:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}[t, \boldsymbol{x}(t), \boldsymbol{u}(t)] \qquad t \in [a, b] \tag{2}$$

The solution is approximated by Lagrange polynomials over the element i, $\zeta_i \le t \le \zeta_{i+1}$ as follows:

$$\boldsymbol{x}_{K+1}(t) = \sum_{j=0}^{K} \boldsymbol{x}_{ij}\phi_j(t); \qquad \phi_j(t) = \prod_{k=0,j}^{K} \frac{(t - t_{ik})}{(t_{ij} - t_{ik})} \tag{3}$$

$$\text{in element } i \qquad i = 1, \ldots, \mathrm{NE}$$

$$\boldsymbol{u}_K(t) = \sum_{j=1}^{K} \boldsymbol{u}_{ij}\theta_j(t); \qquad \theta_j(t) = \prod_{k=1,j}^{K} \frac{(t - t_{ik})}{(t_{ij} - t_{ik})} \tag{4}$$

$$\text{in element } i \qquad i = 1, \ldots, \mathrm{NE}$$

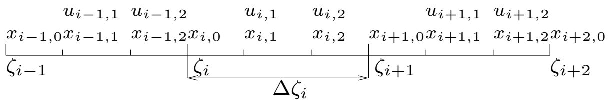

Fig. 1. Finite-element collocation discretisation for state profiles, control profiles and element lengths

Here $k = 0, j$ means that k starts from 0 and $k \neq j$, NE is the number of elements. Also $\boldsymbol{x}_{K+1}(t)$ is a $(K+1)$th order (degree $< K+1$) piecewise polynomial and $\boldsymbol{u}_K(t)$ is Kth order (degree $< K$) piecewise polynomial. The difference in orders is due to the existence of the initial conditions for $\boldsymbol{x}(t)$, for each element i. Also, the Lagrange polynomial has the desirable property that (for $\boldsymbol{x}_{K+1}(t)$, for example)

$$\boldsymbol{x}_{K+1}(t_{ij}) = \boldsymbol{x}_{ij} \tag{5}$$

which is due to the Lagrange condition $\phi_k(t_j) = \delta_{kj}$, where δ_{kj} is the Kronecker delta. This polynomial form allows direct bounding of the states and controls, i.e., path constraints can be imposed on the problem formulation.

By using K points of orthogonal collocations on finite elements as shown in Figure 1, and by defining the basis functions so that they are normalised over each element $\Delta\zeta_i (\tau \in [0, 1])$, one can write the residual equations as follows:

$$\Delta\zeta_i \boldsymbol{r}(t_{ik}) = \sum_{j=0}^{K} \boldsymbol{x}_{ij}\dot{\phi}_j(\tau_k) - \Delta\zeta_i \boldsymbol{F}(t_{ik}, \boldsymbol{x}_{ik}, \boldsymbol{u}_{ik}) \tag{6}$$

$$i = 1, \ldots, \mathrm{NE} \qquad k = 1, \ldots, K$$

where $\dot{\phi}_j(\tau_k) = \mathrm{d}\phi_j/\mathrm{d}\tau$, and can be calculated off-line. Note that $t_{ik} = \zeta_i + \Delta\zeta_i\tau_k$. This form is convenient to work with when the element lengths are included as decision variables. The element lengths are also used to find possible points of discontinuity for the control profiles and to insure that the integration accuracy is within the desired numerical tolerance. Additionally, we enforce the continuity of the states at element endpoints (interior knots $\zeta_i, i = 2, ..., \mathrm{NE}$), but we allow the control profiles to have discontinuities at these endpoints. Here

$$\boldsymbol{x}^i_{K+1}(\zeta_i) = \boldsymbol{x}^{i-1}_{K+1}(\zeta_i) \qquad i = 2, \ldots, \mathrm{NE} \tag{7}$$

or

$$\boldsymbol{x}_{i0} = \sum_{j=0}^{K} \boldsymbol{x}_{i-1,j}\phi_j(\tau = 1) \qquad i = 2, \ldots, \mathrm{NE} \quad j = 0, \ldots, K \tag{8}$$

These equations extrapolate the polynomial $\boldsymbol{x}^{i-1}_{K+1}(t)$ to the endpoints of its element and provide an accurate initial condition for the next element and polynomial $x^i_{K+1}(t)$.

At this point a few additional comments concerning the construction of control polynomials must be made. Note that these polynomials use only K coefficients per element and are of lower order than the state polynomials. As a result these profiles are constrained or bounded only at collocation points. The constraints of the control profile are carried out by bounding the values of each control polynomial at both ends of element. This can be done by writing the equations:

$$\boldsymbol{u}_i^L \leq \boldsymbol{u}_K^i(\zeta_i) \leq \boldsymbol{u}_i^U \quad i = 1, \dots, \mathrm{NE} \tag{9}$$

$$\boldsymbol{u}_i^L \leq \boldsymbol{u}_K^i(\zeta_{i+1}) \leq \boldsymbol{u}_i^U \quad i = 1, \dots, \mathrm{NE} \tag{10}$$

Note that since the polynomial coefficients of the control exist only at the collocation points, enforcement of these bounds can be done by extrapolating the polynomial to the endpoints of the element. This is easily done by using:

$$\boldsymbol{u}_K^i(\zeta_i) = \sum_{j=1}^{K} \boldsymbol{u}_{ij}\theta_j(\tau = 0) \qquad i = 1, \dots, \mathrm{NE} \tag{11}$$

and

$$\boldsymbol{u}_K^i(\zeta_{i+1}) = \sum_{j=1}^{K} \boldsymbol{u}_{ij}\theta_j(\tau = 1) \qquad i = 1, \dots, \mathrm{NE} \tag{12}$$

Adding these constraints affects the shape of the final control profile and the net effect of these constraints is to keep the endpoint values of the control profile from varying widely outside their ranges $[\boldsymbol{u}_i^L, \boldsymbol{u}_i^U]$. Note that the $(\phi_j(\tau),\ \theta_j(\tau))$ and the $(\dot{\phi}_j(\tau))$ terms (basis functions and their derivatives) are calculated beforehand, since they depend only on the Legendre root locations.

The NLP formulation consists of the ODE model (1) discretised on finite elements, continuity equation for state variables, and any other equality and inequality constraints that may be required. It is given by

$$\min_{\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}, \Delta\zeta_i} \left[\Psi(\boldsymbol{x}_f) + \sum_{i=1}^{\mathrm{NE}} \sum_{j=1}^{K} \boldsymbol{w}_{ij} G(\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}, \Delta\zeta_i)\right] \tag{13}$$

such that

$$\boldsymbol{x}_{10} - \boldsymbol{x}_0 = \boldsymbol{0} \tag{14}$$

$$\Delta\zeta \boldsymbol{r}_{ij} = \dot{\boldsymbol{x}}_{k+1}(\tau_j) - \Delta\zeta_i \boldsymbol{F}(\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}) = \boldsymbol{0}, \quad i = 1, \dots, \mathrm{NE} \quad j = 1, \dots, K \tag{15}$$

$$\boldsymbol{x}_{i0} - \boldsymbol{x}_{K+1}^{i-1}(\zeta_i) = \boldsymbol{0}, \quad i = 2, \dots, \mathrm{NE} \tag{16}$$

$$\boldsymbol{x}_f - \boldsymbol{x}_{K+1}^{NE}(\zeta_{NE+1}) = \boldsymbol{0} \tag{17}$$

$$\boldsymbol{u}_i^L \leq \boldsymbol{u}_K^i(\zeta_i) \leq \boldsymbol{u}_i^U, \quad i = 1, \dots, \mathrm{NE} \tag{18}$$

$$\boldsymbol{u}_i^L \leq \boldsymbol{u}_K^i(\zeta_{i+1}) \leq \boldsymbol{u}_i^U, \quad i = 1, \dots, \mathrm{NE} \tag{19}$$

$$\Delta\zeta_i^L \leq \Delta\zeta_i \leq \Delta\zeta_i^U \quad i = 1, \dots, \mathrm{NE} \tag{20}$$

$$\boldsymbol{c}(\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}, \Delta\zeta_i) = \boldsymbol{0} \tag{21}$$

$$\boldsymbol{g}_f(\boldsymbol{x}_f) \leq \boldsymbol{0} \tag{22}$$

$$\boldsymbol{x}_{ij}^L \leq \boldsymbol{x}_{K+1}(\tau_j) \leq \boldsymbol{x}_{ij}^U, \quad i = 1, \dots, \mathrm{NE} \quad j = 0, \dots, K \tag{23}$$

$$\boldsymbol{u}_{ij}^L \leq \boldsymbol{u}_K(\tau_j) \leq \boldsymbol{u}_{ij}^U, \quad i = 1, \dots, \mathrm{NE} \quad j = 1, \dots, K \tag{24}$$

$$\sum_{i=1}^{NE} \Delta\zeta_i = \zeta_{\text{total}} \tag{25}$$

where

i – refers to the element,
j – refers to the collocation point,
$\boldsymbol{w}_{ij}$ – positive quadrature weights,
$\Delta\zeta_i$ – finite-element lengths
$\boldsymbol{x}_0 = \boldsymbol{x}(a)$ – the value of the state at time $t = a$,
$\boldsymbol{x}_f = \boldsymbol{x}(b)$ – the value of the state at the final time $t = b$,
$\boldsymbol{g}_f$ – the constraint evaluated at the final time $t = b$,
$\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}$ – the collocation coefficients for the state and control profiles,

In this formulation the knot positions, ζ_i, are formulated as decision variables and found by optimisation procedure as points of control profile discontinuities. With the knot positions as decision variables, we now have an accurate and efficient strategy to solve very general and difficult optimal control problems, as long as orthogonal collocation approximates the state profiles accurately within each element.

Problem (9) can be now solved by any large scale nonlinear programming solver.

To solve this problem within MATLAB, we used the optim toolbox which includes several programs for treating optimisation problems. In this case function *fmincon* was choosen. This can minimise/maximise a given objective function subject to nonlinear equality and inequality constraints. In order to use this function it was neccessary to create and program additional functions (Čižniar et al., 2005). The resulting code is called *dynopt*.

3. CASE STUDIES

In this section we present the examples from literature sloved by *dynopt*.

3.1 *Example 1a*

Consider the following unconstrained problem (Luus, 1991; Rajesh et al., 2001)

$$\min_{\boldsymbol{u}(t)} J = x_2(t_f) \tag{26}$$

such that

$$\dot{x}_1 = u$$
$$\dot{x}_2 = x_1^2 + u^2$$
$$\boldsymbol{x}(0) = [1\ 0]^{\mathrm{T}}$$
$$t_f = 1$$

where

$x_1(t),\ x_2(t)$ – state vectors,
$u(t)$ – control vector.

3.2 *Example 1b*

Consider the following constrained problem (Luus, 1991; Rajesh et al., 2001)

$$\min_{\boldsymbol{u}(t)} J = x_2(t_f) \tag{27}$$

such that

$$\begin{aligned}\dot{x}_1 &= u\\ \dot{x}_2 &= x_1^2 + u^2\\ \boldsymbol{x}(0) &= [1\ 0]^{\mathrm{T}}\\ x_1(1) &= 0\\ t_f &= 1\end{aligned}$$

where

$x_1(t),\ x_2(t)$ – state vectors,
$u(t)$ – control vector.

3.3 *Example 2*

Consider the following nonlinear unconstrained problem (Luus, 1990; Rajesh et al., 2001)

$$\min_{\boldsymbol{u}(t)} J = x_4(t_f) \tag{28}$$

such that

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -x_3 u + 16t - 8\\ \dot{x}_3 &= u\\ \dot{x}_4 &= x_1^2 + x_2^2 + 0.0005(x_2 + 16t - 8 - 0.1x_3u^2)^2\\ \boldsymbol{x}(0) &= [0\ -1\ -\sqrt{5}\ 0]^{\mathrm{T}}\\ -4 &\le u \le 10\\ t_f &= 1\end{aligned}$$

where

$x_1(t) - x_4(t)$ – state vectors,
$u(t)$ – control vector.

3.4 *Example 3*

Consider a tubular reactor with following parallel reaction (Dadeo and McAuley, 1995; Logsdon and Biegler, 1989; Rajesh et al., 2001):
$A \rightarrow B$
$A \rightarrow C$

$$\min_{\boldsymbol{u}(t)} J = -x_2(t_f) \tag{29}$$

such that

$$\begin{aligned}\dot{x}_1 &= -[u + 0.5u^2]x_1\\ \dot{x}_2 &= ux_1\\ \boldsymbol{x}(0) &= [1\ 0]^{\mathrm{T}}\\ 0 &\le u \le 5\\ t_f &= 1\end{aligned}$$

where

$x_1(t)$ – dimensionless concetration of A,
$x_2(t)$ – dimensionless concentration of B,
$u(t)$ – control vector

3.5 *Example 4*

Consider a batch reactor (Dadeo and McAuley, 1995; Rajesh et al., 2001) with the following consecutive reactions:
$A \rightarrow B \rightarrow C$

$$\min_{\boldsymbol{u}(t)} J = -x_2(t_f) \tag{30}$$

such that

$$\begin{aligned}\dot{x}_1 &= -k_1x_1^2\\ \dot{x}_2 &= k_1x_1^2 - k_2x_2\\ \boldsymbol{x}(0) &= [1\ 0]^{\mathrm{T}}\\ k_1 &= 4000e^{(-\frac{2500}{T})}\\ k_2 &= 620000e^{(-\frac{5000}{T})}\\ 298 &\le T \le 398\\ t_f &= 1\end{aligned}$$

$x_1(t)$ – concentration of A,
$x_2(t)$ – concentration of B,
T – temperature.

3.6 *Example 5*

Consider a catalytic plug flow reactor (Dadeo and McAuley, 1995; Rajesh et al., 2001) with the following reactions:
$A \leftrightarrow B \rightarrow C$

$$\max_{\boldsymbol{u}(t)} J = 1 - x_1(t_f) - x_2(t_f) \tag{31}$$

such that

$$\begin{aligned}\dot{x}_1 &= u[10x_2 - x_1]\\ \dot{x}_2 &= -u[10x_2 - x_1] - [1-u]x_2\\ \boldsymbol{x}(0) &= [1\ 0]^{\mathrm{T}}\\ 0 &\le u \le 1\\ t_f &= 12\end{aligned}$$

$x_1(t)$ – mole fraction of A,
$x_2(t)$ – mole fraction of B,
$u(t)$ – fraction of type 1 catalyst.

4. RESULTS AND DISCUSSION

For all the aformentioned examples, 4 collocation points and 5 elements have been used to obtain accurate solution to the fourth decimal place. Problems (26), (27), and (28) are examples of purely mathematical systems. The first problem (26) does not have any constraint and for this problem a minimum (0.76519) was determined by Luus (1991). Another optimal value for the performance index was found by Rajesh et al. (2001) (0.76238). The solution obtained by *dynopt* is shown in Table 1. The control profiles and state variables are shown in Figure 2.

The problem (27) has a terminal constraint. For this case the value of the minimum (0.92518) was obtained by Luus (1991) and (0.92547) obtained by Rajesh et al. (2001). The solution obtained by *dynopt* in this case is shown in Table 2 and Figure 3.

Problem (28) is a four-state variable system treated by Luus (1990); Rajesh et al. (2001). For

	Numerical	Analytical
optimal value	0.7616	0.7616
number of iterations	43	45
funccount	3570	353
CPU-time (s)	50.5730	18.6770

Table 1. Comparison between gradients calculated numerically and analytically for example 1a

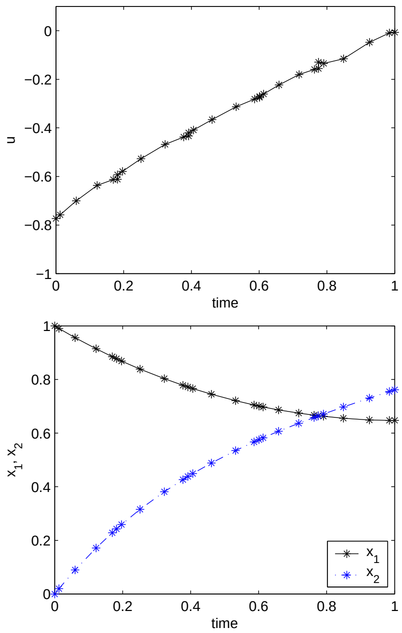

Fig. 2. Control and state profiles obtained by *dynopt* for problem (26)

	Numerical	Analytical
optimal value	0.9243	0.9243
number of iterations	26	23
funccount	2154	193
CPU-time (s)	32.0660	13.6790

Table 2. Comparison between gradients calculed numerically and analytically for example 1b

the state variable x_4, a value of the minimum (0.12011) was obtained by Luus (1990). Rajesh et al. (2001) computed the optimum of x_4 at final time (0.1290). With *dynopt* in this case we were able to reach the values shown in Table 3 and Figure 4.

	Numerical	Analytical
optimal value	0.1217	0.1212
number of iterations	96	126
funccount	12555	586
CPU-time (s)	239.5850	44.8450

Table 3. Comparison between gradients calculated numerically and analytically for example 2

Problem (29) is a tubular reactor control problem where the state variable x_2 at final time has to be maximised. This problem was treated by Dadeo and McAuley (1995); Logsdon and

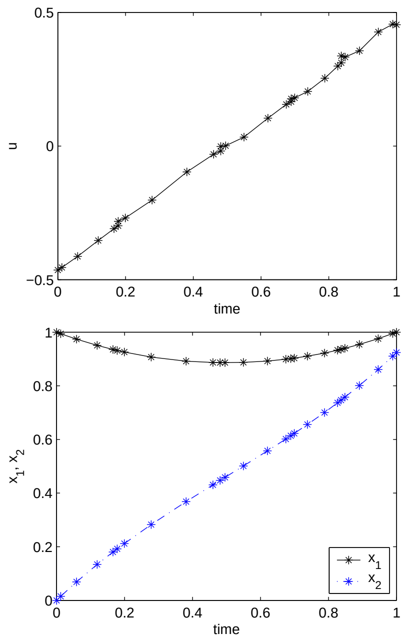

Fig. 3. Control and state profiles obtained by *dynopt* for problem (27)

Biegler (1989); Rajesh et al. (2001) and the optimal value (0.57353) was reported by Dadeo and McAuley (1995); Logsdon and Biegler (1989) and optimal value (0.57284) was given by Rajesh et al. (2001). Table 4 and Figure 5 show the optimal values found by the orthogonal collocation of finite elements method used in *dynopt*.

	Numerical	Analytical
optimal value	0.5727	0.5725
number of iterations	74	68
funccount	5951	370
CPU-time (s)	93.3040	26.3770

Table 4. Comparison between gradients calculated numerically and analytically for example 3

The objective in problem (30) is to obtain the optimal temperature profile that maximizes x_2 at the end of a specified time. The problem was solved by Logsdon and Biegler (1989); Rajesh et al. (2001) and the reported optimum (0.610775) was found by Logsdon and Biegler (1989) and (0.61045) obtained by Rajesh et al. (2001). We were able to obtain the values described by Table 5 and Figure 6.

	Numerical	Analytical
optimal value	0.6102	0.6102
number of iterations	9	10
funccount	780	107
CPU-time (s)	13.0380	9.9540

Table 5. Comparison between gradients calculated numerically and analytically for example 4

Optimisation of problem (31) has also been analyzed. This problem was solved by Logsdon and

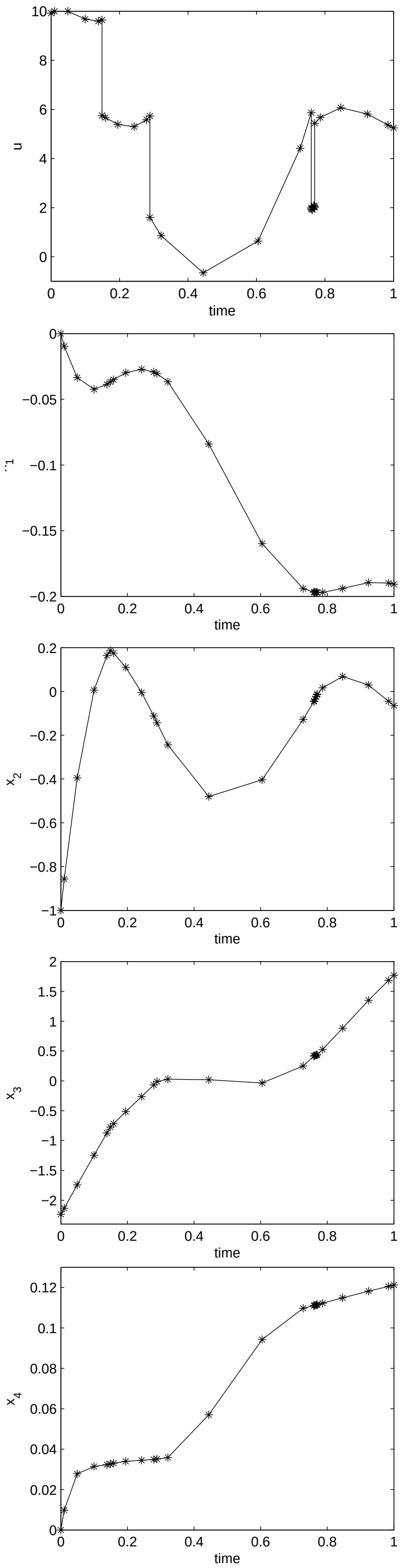

Fig. 4. Control and state profiles obtained by *dynopt* for problem (28)

Biegler (1989); Rajesh et al. (2001) and the optima (0.476946, 0.47615) were found. Values obtained by using *dynopt* are shown in Table 6. The coresponding control and state profiles are shown in Figure 7

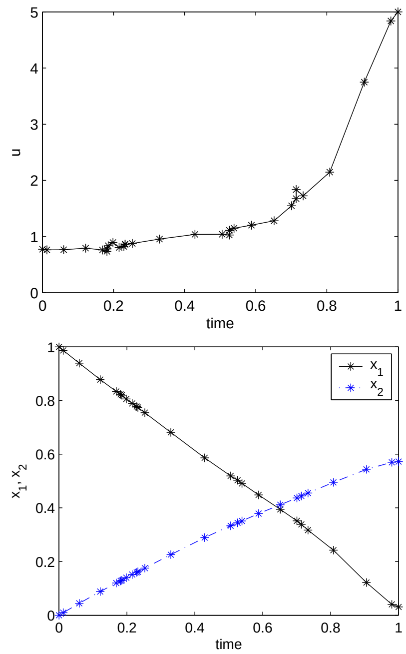

Fig. 5. Control and state profiles obtained by *dynopt* for problem (29)

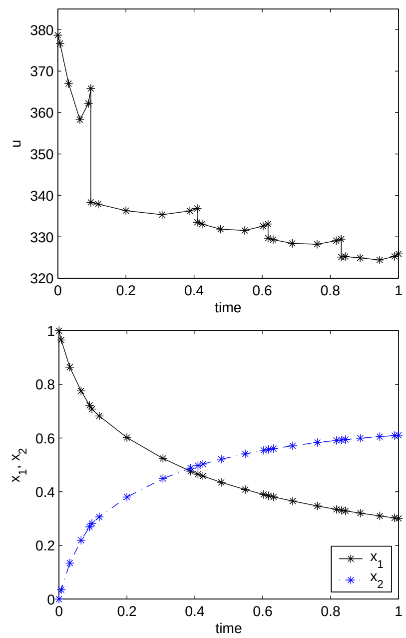

Fig. 6. Control and state profiles obtained by *dynopt* for problem (30)

	Numerical	Analytical
optimal value	0.4790	0.4790
number of iterations	36	31
funccount	3474	218
CPU-time (s)	63.7520	23.3360

Table 6. Comparison between gradients calculated numerically and analytically for example 5

Note, that all the results obtained by orthogonal collocation on finite elements method implemented within MATLAB-*dynopt* are only local

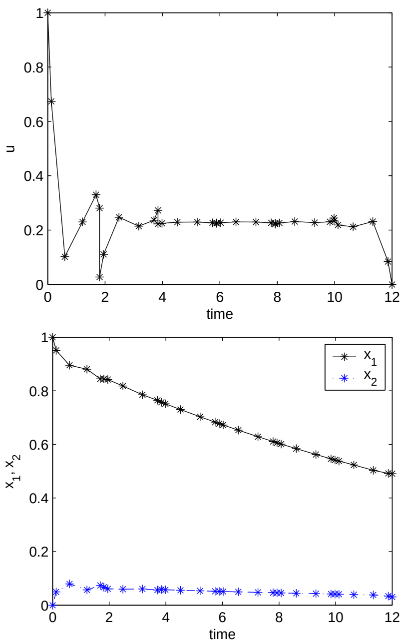

Fig. 7. Control and state profiles obtained by *dynopt* for problem (31)

in nature, since NLP solvers are only based on necessary conditions for optimality.

5. CONCLUSION

The orthogonal collocation on finite elements has been developed and implemented within MATLAB environment and it has been tested on six examples from the literature. The details of the problems are provided in section 3. The examples were chosen to illustrate the ability of the dynopt package to treat the problems of varying levels of difficulty. In all the considered examples, two different methods of gradients computation were used: numerical approximation and analytical computation. The resulting performances of each method for the case studies are presented in tables 1, 2, 3, 4, 5, 6. As expected, they show that the performances of analytical computations are superior. On the other hand, as mentioned before, the optima obtained are local in nature, the coming work will be devoted to the solution of the resulting NLP problems to global optimality.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under grant No. 1/0135/03. Part of the work has been completed under the French – Slovak program Stefanik (Dynamic and Global Optimisation of Processes) and within the framework of the European Social Fund (Develompment of Human Resources for Research and Progress in Automation, JPD 3 2004/1-056), and under the Financial support from ERASMUS SOCRATES Foundation.

References

J. E. Cuthrell and L. T. Biegler. On the optimization of differential-algebraic process systems. *AIChE Journal*, 33:1257–1270, 1987.

S.A. Dadeo and K.B. McAuley. Dynamic optimization of constrained chemical engineering problems using dinamic programming. *Computers chem. Engng.*, 19:513–525, 1995.

C. J. Goh and K. L. Teo. Control parametrization: A unified approach to optimal control problems with general constraints. *Automatica*, 24:3–18, 01 1988.

J. S. Logsdon and L. T. Biegler. Accurate solution of differential-algebraic optimization problems. *Chem. Eng. Sci.*, (28):1628–1639, 1989.

R. Luus. Optimal control by dynamic programming using systematic reduction in grid size. *Int. J. Control*, 51:995–1013, 1990.

R. Luus. Application of iterative dynamic to state constrained optimal control problems. *Hung. J. Ind. Chem.*, 19:245–254, 1991.

J. Rajesh, K. Gupta, H.S. Kusumakar, V.K. Jayaraman, and B.D. Kulkarni. Dynamic optimization of chemical processes using ant colony framework. *Computers and Chemistry*, 25:583–595, 2001.

M. Čižniar, M. Fikar, and M.A. Latifi. *User's Guide for MATLAB Dynamic Optimisation Code DYNOPT*, 05 2005.