Dynamic Optimization of a Batch Reactor using the capabilities of an MINLP process synthesizer MIPSYN

M. Ropotar,[a] Z. Kravanja,[b]

[a] *Tanin Sevnica kemična industrija d.d., Hermanova cesta 1, 8290 Sevnica, Slovenia*
[b] *Faculty of Chemistry and Chemical Engineering, University of Maribor, P.O. Box 219, Maribor, Slovenia*

Abstract

This contribution describes the development of various strategies for the dynamic optimization of a batch reactor in order to obtain a robust model, suitable for nonlinear (NLP) or mixed-integer nonlinear programming (MINLP) problems. Different Orthogonal Collocation on Finite Element (OCFE) schemes and various formulations of the MINLP model have been studied to increase its robustness. It has been found that none of the MINLP model formulation is as efficient as NLP. Various strategies have been applied to NLP and MINLP models, and in addition, their efficiencies and robustness have been compared.

Keywords: Batch reactor, orthogonal collocation, NLP, MINLP, process synthesis

1. Introduction

Over the last decade modeling, dynamic optimization and on-line optimization have been the main research areas in optimization of batch reactors. The modeling category is usually oriented towards a more realistic description of a batch reactor (Zaldivar et al., 1996) and towards the use of special modeling techniques and strategies in cases of imperfect knowledge of kinetic studies involved, e.g. the use of tendency models (Fotopoulos et al., 1998) or a sequential experiment design strategy based on reinforcement learning (Martinez, 2000). The second category is related to more advanced aspects of dynamic optimization of batch reactors, e.g. robust optimization of models, characterized by parametric uncertainty (Ruppen et al., 1995), or stochastic optimization of multimodal batch reactors (Carrasco and Banga, 1997). And finally, in work related to on-line optimization, which is currently the prevailing activity, different control schemes were proposed, e.g. feedforward/state feedback laws in the presence of disturbances, and nonlinear state feedback laws for batch processes with multiple manipulated inputs were developed (Raman and Palanki, 1996, 1998).

Kinetics in batch reactors is described using differential equations. These equations represent complex optimization problems, even in small and simple examples. The use of Orthogonal Collocation on Finite Elements (OCFE) in optimization models of batch reactors has become a well-established numerical method. The OCFE method with a fixed finite element is the most straightforward and easiest. However, when using fixed finite elements directly it is impossible to explicitly model the optimal retention times of the batch reactors and the optimal outlet concentrations and conditions. Consequently, the use of flexible finite elements is regarded as a conventional approach for overcoming these difficulties (Cuthrell and Biegler, 1989). This model, however, seems to have become more nonlinear because the length of the final element is converted into a variable.

In recent research (Ropotar and Kravanja, 2006a), NLP and MINLP models were developed for the dynamic optimization of batch reactors. A differential-algebraic optimization problem (DAOP) model was initially converted into a robust nonlinear programming (NLP) model by the use of Orthogonal Collocation on a fixed, rather than flexible, Finite Element.

This paper describes the use of various strategies for the dynamic optimization of a batch reactor in order to obtain a robust model, suitable for NLP or MINLP synthesis problems. Different schemes for OCFE were studied to increase the robustness of the model. Finally, in the case of the MINLP model, the robustness of the model is studied with respect to the use of different model formulations. Different model formulations are then compared, in order to find out which of them are more efficient and robust. Models were developed on the example of a batch reactor described by Ropotar and Kravanja, 2005.

2. OCFE schemes and strategies

Different OCFE schemes with a fixed (NLP) or changing number of finite elements (MINLP), with moving or fixed finite elements, and with an end and/or inner optimal point in the Legendre polynomial representation were investigated in order to increase the efficiency of the NLP and MINLP models. In the case of NLP optimization, the number of finite elements has to be set in advance and is, thus, usually oversized in order to satisfy a given error tolerance, whereas, in the MINLP cases it is explicitly modeled in order to be simultaneously adjusted during the optimization process to the minimal number of elements.

2.1. NLP model with moving finite elements

In the case of NLP and MINLP formulations with moving finite elements, additional nonlinearities of algebraic constraints are introduced in the model due to the presence of the variables, which represent finite element’s lengths. On the other hand, some nonlinearities vanish because optimal time is moved to the end of the final element and several equations become linear. Additionally, inequality constraints for the approximation error were included in the model. Complete NLP model (F-NLP) with flexible finite elements and inequations for approximation error is shown below. The model was developed on the example of a batch reactor problem (Fig. 1), where consecutive reaction A → B → C is carried out and B is the desired product. Since the reaction is endothermic, the system can be heated and/or preheated. As far as the optimal inlet temperature is higher than the one defined by the user, the inlet must be preheated. The kinetics of this reaction is following:

$$\frac{\mathrm{d}c_{\mathrm{A}}}{\mathrm{d}t} = -k_0 \cdot \mathrm{e}^{\frac{-E_{\mathrm{a,A}}}{\mathrm{R \cdot T}}} \cdot c_{\mathrm{A}}$$

$$\frac{\mathrm{d}c_{\mathrm{B}}}{\mathrm{d}t} = k_0 \cdot \mathrm{e}^{\frac{-E_{\mathrm{a,A}}}{R \cdot T}} \cdot c_{\mathrm{A}} - k_0 \cdot \mathrm{e}^{\frac{-E_{\mathrm{a,B}}}{R \cdot T}} \cdot c_{\mathrm{B}}$$

$$\frac{\mathrm{d}c_{\mathrm{C}}}{\mathrm{d}t} = k_0 \cdot \mathrm{e}^{\frac{-E_{\mathrm{a,B}}}{\mathrm{R \cdot T}}} \cdot c_{\mathrm{B}}$$

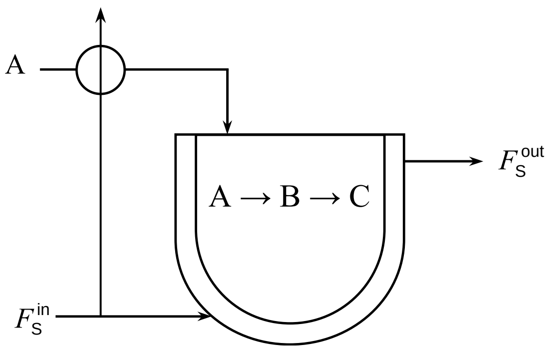

Figure 1: Batch reactor.

$$\max_{\alpha, c_{\mathrm{A}}, c_{\mathrm{B}}, c_{\mathrm{B}}, \Phi_{\mathrm{preheat}}, \Phi_{\mathrm{S}}} Z = \frac{28800}{t_{\mathrm{tot}}^{\mathrm{opt}} + 600} \cdot \left(\begin{array}{l} C_1 c_{\mathrm{B},l=NE}^{\mathrm{opt}} V - C_2 \cdot c_{\mathrm{A}}^0 V - C_3 \cdot c_{\mathrm{C},l=NE}^{\mathrm{opt}} V - C_4 \cdot \Phi_{\mathrm{preheat}} - \\ C_5 \cdot \sum_{l=1}^{NE} \frac{\Delta \alpha_l}{2} \sum_{n=1}^{N} A_n \sum_{j=1}^{K} \Phi_{\mathrm{S},jl} \cdot \prod_{k=0,k \neq j}^{K} \frac{\frac{1}{2}(x_n + 1) - t_k}{t_j - t_k} \end{array} \right)$$

Residual equations and component balances:

$$\left. \begin{array}{l} R_{\mathrm{B},il}(t_{il}) = \sum_{j=0}^{K} c_{\mathrm{B},jl} \cdot \prod_{k=0,k \neq j}^{K} \frac{t_{il} - t_{kl}}{t_{jl} - t_{kl}} - k_0 \cdot \mathrm{e}^{\frac{-E_{\mathrm{a,A}}}{\mathrm{R} \cdot T_{il}}} \cdot c_{\mathrm{A},il} + k_0 \cdot \mathrm{e}^{\frac{-E_{\mathrm{a,B}}}{\mathrm{R} \cdot T_{il}}} \cdot c_{\mathrm{B},il} = 0 \\ R_{\mathrm{C},il}(t_{il}) = \sum_{j=0}^{K} c_{\mathrm{C},jl} \cdot \prod_{k=0,k \neq j}^{K} \frac{t_{il} - t_{kl}}{t_{jl} - t_{kl}} - k_0 \cdot \mathrm{e}^{\frac{-E_{\mathrm{a,B}}}{\mathrm{R} \cdot T_{il}}} \cdot c_{\mathrm{B},il} = 0 \\ \left(c_{\mathrm{A}}^0 - c_{\mathrm{A},il}\right) = \left(c_{\mathrm{B},il} - c_{\mathrm{B}}^0\right) + \left(c_{\mathrm{C},il} - c_{\mathrm{C}}^0\right) \end{array} \right\} \begin{array}{l} \forall i = 1,2,...K \\ \forall l = 1,2,...NE \end{array}$$

Energy balance:

$$R_{T,il}(t_{il}) = \sum_{j=0}^{K} T_{il} \cdot \prod_{k=0,k\neq j}^{K} \frac{t_{il}-t_{kl}}{t_{jl}-t_{kl}} + \frac{\Delta_r H_B}{\rho \cdot c_p} \cdot \left(k_0 \cdot e^{\frac{-E_{a,A}}{R \cdot T_{il}}} \cdot c_{A,il} - k_0 \cdot e^{\frac{-E_{a,B}}{R \cdot T_{il}}} \cdot c_{B,il} \right) +$$

$$\frac{\Delta_r H_C}{\rho \cdot c_p} \cdot k_0 \cdot e^{\frac{-E_{a,B}}{R \cdot T_{il}}} \cdot c_{B,il} - \frac{\Phi_{S,il}}{V \cdot \rho \cdot c_p} = 0, \quad \forall i = 1,2,...K,\ \forall l = 1,2,...NE$$

where K and NE are collocation points and final elements, respectively. Optimal outlet point by Legendre polynomials:

$$\left.\begin{aligned} & c_{B,l}^{opt}\left(t_l^{opt}\right) = \sum_{j=0}^{K} c_{B,jl} \cdot \prod_{k=0,k\neq j}^{K} \frac{1-t_k}{t_j-t_k},\ T_l^{opt}\left(t_l^{opt}\right) = \sum_{j=0}^{K} T_{jl} \cdot \prod_{k=0,k\neq j}^{K} \frac{1-t_k}{t_j-t_k} \\ & c_{C,l}^{opt}\left(t_l^{opt}\right) = \sum_{j=0}^{K} c_{C,jl} \cdot \prod_{k=0,k\neq j}^{K} \frac{1-t_k}{t_j-t_k},\ \Phi_{S,l}^{opt}\left(t_l^{opt}\right) = \sum_{j=0}^{K} \Phi_{S,jl} \cdot \prod_{k=0,k\neq j}^{K} \frac{1-t_k}{t_j-t_k} \\ & c_{A,l}^{opt} = c_{A,i=0,l} - \left(c_{B,l}^{opt} - c_B^0\right) - \left(c_{C,l}^{opt} - c_C^0\right) \end{aligned}\right\} \forall l = 1,2,...NE \qquad (1)$$

The point at the interior knot is defined as the optimal interior point from the previous finite element defined by Legendre polynomials (eq. 1):

$$\left.\begin{aligned} & c_{A,i=0,l} = c_{A,l-1}^{opt} \quad T_{i=0,l} = T_{l-1}^{opt} \\ & c_{B,i=0,l} = c_{B,l-1}^{opt} \quad \Phi_{S,i=0,l} = \Phi_{S,l-1}^{opt} \\ & c_{C,i=0,l} = c_{C,l-1}^{opt} \\ & \Delta\alpha_l = \Delta\alpha_{l-1} \end{aligned}\right\} \forall l = 2,3,...NE$$

Inequality constraints for approximation error $\forall i = 1,2,...K,\ \forall l = 1,2,...NE$:

$$-\varepsilon \leq \sum_{j=0}^{K} c_{B,jl} \cdot \prod_{k=0,k\neq j}^{K} \frac{t_{il}-t_{kl}}{t_{jl}-t_{kl}} - k_0 \cdot e^{\frac{-E_{a,A}}{R \cdot T_{0,l+1}}} \cdot c_{A,i=0,l+1} + k_0 \cdot e^{\frac{-E_{a,B}}{R \cdot T_{i=0,l+1}}} \cdot c_{B,i=0,l+1} \leq \varepsilon$$

$$-\varepsilon \leq \sum_{j=0}^{K} c_{C,jl} \cdot \prod_{k=0,k\neq j}^{K} \frac{t_{il}-t_{kl}}{t_{jl}-t_{kl}} - k_0 \cdot e^{\frac{-E_{a,B}}{R \cdot T_{i=0,l+1}}} \cdot c_{B,i=0,l+1} \leq \varepsilon$$

$$-\varepsilon \leq \sum_{j=0}^{K} T_{il} \cdot \prod_{k=0,k\neq j}^{K} \frac{t_{il}-t_{kl}}{t_{jl}-t_{kl}} + \frac{\Delta_r H_B}{\rho \cdot c_p} \cdot \left(k_0 \cdot e^{\frac{-E_{a,A}}{R \cdot T_{i=0,l+1}}} \cdot c_{A,i=0,l+1} - k_0 \cdot e^{\frac{-E_{a,B}}{R \cdot T_{i=0,l+1}}} \cdot c_{B,i=0,l+1} \right) +$$

$$\frac{\Delta_r H_C}{\rho \cdot c_p} \cdot k_0 \cdot e^{\frac{-E_{a,B}}{R \cdot T_{i=0,l+1}}} \cdot c_{B,il+1} - \frac{\Phi_{S,i=0,l+1}}{V \cdot \rho \cdot c_p} \leq \varepsilon \qquad \text{(F-NLP)}$$

where ε is an error tolerance, e.g. 10^{-3}. Note that optimal time is, in contrast to the model with fixed final elements, at the end of the final element and, thus, Legendre polynomials are simplified. Furthermore, the heat flow in the objective function is integrated over the whole length of the final element, which is optimized. It also should be noted that the profit and number of batches are defined for production covering 8 hours and a 600 sec non-operating period between batches. Thus, the number of batches is 28880/($t_{\text{tot}}^{\text{opt}}$ +600). Total time is defined as a sum of all optimal times, which are actually the length of the final elements, in all finite elements: $t_{\text{tot}}^{\text{opt}} = \sum_{l=1}^{NE} \Delta\alpha_l$ and $0 \le \Delta\alpha_l \le \Delta\alpha_l^{\text{UP}}$. Thus, each fixed final element is defined as between zero and $\Delta\alpha_l^{\text{UP}}$.

2.2. MINLP model with moving final elements

The MINLP model is similar to the NLP with the exception of some additional constraints. These are applied in order select the optimal number of finite elements:

$$y_l \le y_{l-1} \quad \forall l = 2,3,...NE \tag{2}$$

$$\left.\begin{array}{l} \Delta\alpha_l \le \Delta\alpha_l^{\text{UP}} \cdot y_l \quad (3) \\ \Delta\alpha_l \ge \Delta\alpha_l^{\text{LO}} \cdot y_l \quad (4) \end{array}\right\} \forall l = 1,2,...NE$$

Ineq. (2) is applied to ensure that all finite elements up to the last selected one are, in fact, selected. If the corresponding finite element is rejected, ineq. (3) forces $\Delta\alpha_l$ to zero. On the other hand, when the element is selected, ineqs. (3)-(4) are applied to vary the $\Delta\alpha_l$ of each finite element between bounds. Note that, in contrast to the NLP model where the integration is distributed equally and continuously within all the finite elements, here the integration is applied only to the selected finite elements.

Ropotar and Kravanja (2006b) developed an alternative convex-hull model formulation and implemented it in the process synthesizer MIPSYN, the successor of PROSYN-MINLP (Kravanja and Grossmann, 1994). MIPSYN enables automated execution of simultaneous topology, and parameter optimization of the processes. Optimization of each NLP subproblem is performed only on the existing units rather than on the entire superstructure, which substantially reduces the size of the NLP subproblems. An NLP initializer, model generator and a comprehensive library of models for basic process units and interconnection nodes, together with a comprehensive library of basic physical properties for the most common chemical components were developed in order to facilitate different types of computation (for example initialization, optimizing fixed structures).

In the case of the MINLP model the robustness of the model was studied with respect to the use of different model formulations. Namely, Big-M formulation, conventional convex hull (CCH) and alternative convex hull formulation (ACH) and the following representations of outer approximations (OA) for the Outer Approximation/Equality Relaxation algorithm are compared:

Big-M formulation: $$h\left(\boldsymbol{x}^{1}\right)+\nabla_{x} h\left(\boldsymbol{x}^{1}\right)^{\mathrm{T}} \boldsymbol{x}-\nabla_{x} h\left(\boldsymbol{x}^{1}\right)^{\mathrm{T}} \boldsymbol{x}^{1} \leq M(1-y) \quad (5)$$

Convex hull representation: $$\nabla_{x} h\left(\boldsymbol{x}^{1}\right)^{\mathrm{T}} \boldsymbol{x} \leq\left(\nabla_{x} h\left(\boldsymbol{x}^{1}\right)^{\mathrm{T}} \boldsymbol{x}^{1}-h\left(\boldsymbol{x}^{1}\right)\right) y \quad (6)$$

Alternative formulation: $$\nabla_{x} h\left(\boldsymbol{x}^{1}\right)^{\mathrm{T}} \boldsymbol{x} \leq \nabla_{x} h\left(\boldsymbol{x}^{1}\right)^{\mathrm{T}} \boldsymbol{x}^{\mathrm{f}}+\left(\nabla_{x} h\left(\boldsymbol{x}^{1}\right)^{\mathrm{T}}\left(\boldsymbol{x}^{1}-\boldsymbol{x}^{\mathrm{f}}\right)-h\left(\boldsymbol{x}^{1}\right)\right) y \quad (7)$$

Unlike CCH representation, where the continuous variables $\boldsymbol{x}$ are usually forced into zero values when the corresponding disjunctives are false, in ACH the variables are forced into arbitrarily-forced values, $\boldsymbol{x}^{\mathrm{f}}$.

3. Results and comparison

As stated above, all schemes and strategies were performed on the batch reactor example with 50 finite elements. Results for different OCFE schemes are given in Table 1 in order to compare NLP and MINLP solutions with fixed final elements to those with flexible final elements. The last two columns outline NLP and MINLP solutions obtained considering approximation error tolerance $\varepsilon = 10^{-3}$. The comparison between three different MINLP model formulations is given in Table 2.

Table 1: Comparison among different schemes and strategies.

scheme/ strategy	NLP (fixed FE)	MINLP (fixed FE)	NLP (flexible FE)	MINLP (flexible FE)	NLP ($\varepsilon = 10^{-3}$) (flexible FE)	MINLP ($\varepsilon = 10^{-3}$) (flexible FE)
c_A^{opt} (mol/l)	0.101	0.101	0.101	0.101	0.101	0.101
c_B^{opt} (mol/l)	0.605	0.605	0.605	0.604	0.607	0.605
c_C^{opt} (mol/l)	0.094	0.094	0.094	0.095	0.092	0.094
T^{opt} (K)	369.1	369.3	369.3	382.3	369.2	383.2
t^{opt} (s)	142.55	138.69	139.95	132.61	173.65	139.97
Z (k$)	36.996	37.024	36.998	37.139	36.574	36.881
CPU time (s)	11.46	244.48	7.07	328.61	33.96	667.05

It can be seen that the solutions are very similar: small differences occur in temperatures, total optimal time, and profit. However, the CPU time for solving the NLP model is significantly smaller than the MINLP one because 6 major MINLP iterations have to be performed to obtain the optimal solution. Also, the values of the objective function obtained with MINLP model are somewhat larger than the NLP ones. With 50 finite elements the NLP and MINLP models were able to tolerate an approximation error tolerance less than 10^{-3}. When the approximation error tolerance is considered explicitly in the model, the value of the objective function is, as expected, somewhat smaller. It should be noted that Big-M model formulation was used in the case of MINLP.

Table 2: Comparison among three different MINLP model formulations.

formulation	BIG-M (fixed FE)	CCH (fixed FE)	ACH (fixed FE)	BIG-M (flexible FE)	CCH (flexible FE)	ACH (flexible FE)
c_A^{opt} (mol/l)	0.101	0.101	0.101	0.101	0.101	0.101
c_B^{opt} (mol/l)	0.605	0.605	0.605	0.604	0.604	0.604
c_C^{opt} (mol/l)	0.094	0.094	0.094	0.095	0.095	0.095
T^{opt} (K)	369.3	369.3	369.3	382.3	379.9	383.0
t^{opt} (s)	138.69	139.85	139.85	132.61	132.53	132.47
Z (k$)	37.024	36.999	36.999	37.139	37.141	37.142
CPU time (s)	627.26	639.21	844.09	722.58	527.74	1266.96

Finally, three different MINLP model formulations were compared, the Big-M, CCH and ACH, Table 2. It can be seen that when fixed final elements were used, CPU time needed for solving 11 major iterations is comparable for Big-M and CCH formulation, while for the ACH formulation it is somewhat bigger. When flexible final elements were used, ACH formulation needed twice as much CPU time for solving 11 major iterations as CCH formulation, while Big-M formulation needed slightly more CPU time than the CCH one. Also, it can be seen from both tables that when using flexible final elements better solutions were obtained than with fixed elements.

4. Conclusions

The main goal of the research described in this contribution is to obtain a robust model, suitable for nonlinear (NLP) or mixed-integer nonlinear programming (MINLP) problems. In order to achieve that, different OCFE schemes and strategies were developed and, furthermore, different model formulations were studied in the case of the MINLP model.

In the NLP model with flexible final elements, some nonlinearities were reduced and CPU time was also decreased. On the contrary, in the case of MINLP model, CPU time was increased when flexible final elements were used, except in CCH formulation. Although convex hull formulations are usually more efficient than Big-M ones, which was shown also by Ropotar and Kravanja (2006b), in our example of batch reactor it was just the opposite. It was noticed that Big-M formulation is in our example most straightforward and contains considerably less equations and variables.

The NLP model is most suitable for the optimization of stand-alone reactors when the approximation error is large. Since the disjunctive MINLP model can adjust automatically an appropriate number of finite elements in order to tolerate a given approximation error, it can be more efficient than the NLP model when the approximation error is small. This could be especially significant when MINLP synthesis of reactors is performed within the overall process schemes where it is important not to burden the NLP computation by carrying unnecessary final elements through MINLP iterations. On the other hand, it is promising that the NLP model would behave better during the process synthesis than the MINLP model, especially because the combinatorics of the model can be significantly reduced since the selection of the final element as performed in the MINLP model is avoided. Moreover, other MINLP formulations were used and all of them were more expensive in CPU time than NLP. Hence, the future research is oriented towards the use of NLP rather than MINLP model for reactors in process synthesis.

References

Carrasco, E. F. and Banga, J. R., (1997) *Industrial & Engineering Chemistry Research*, 36 (6) 2252-2261.

Fotopoulos, J., Georgakis, C. and Stenger, H. G., (1998) *Chemical Engineering and Processing*, 37 (6) 545-558.

Cuthrell, J.E., Biegler, L.T., (1989) *Computers & Chemical Engineering*, 13 (1-2) 49-62.

Kravanja, Z., Grossmann, I.E., (1994) *Computers & Chemical Engineering*, 18 1097-1114.

Martinez, E. C., (2000) *Computers & Chemical Engineering*, 24 (2-7) 1187-1193.

Rahman, S., Palanki, S., (1996) AICHE Journal, 42 (10) 2869-2882.

Rahman, S., Palanki, S., (1998) *Computers & Chemical Engineering*, 22 (10) 1429-1439.

Ropotar, M., Kravanja, Z., (2005) *International Conference of Computational Methods in Science and Engineering ICCMSE 2005*, Loutraki, 21-26 October 2005, Greece.

Ropotar, M., Kravanja, Z., (2006a) *International Conference of Computational Methods in Science and Engineering ICCMSE 2006*, Chania, Crete, 27 October - 1 November 2006, Greece.

Ropotar, M., Kravanja, Z., (2006b) *16th European symposium on computer aided process engineering and 9th International symposium on process systems engineering,* Garmish-Partenkirchen, 9 – 13 July 2006.

Ruppen, D., Benthack, C. and. Bonvin, D., (1995) *Journal of Process Control*, 5 (4) 235-240.

Zaldivar, J. M., Hernandez, H. and Barcons, C., (1996) *Thermochimica Acta*, 289 (2) 267-302.