XXI ENMC
ENCONTRO NACIONAL DE MODELAGEM COMPUTACIONAL
IX ECTM
ENCONTRO DE CIÊNCIA E TECNOLOGIA DE MATERIAIS

08 a 11 de Outubro de 2018
Instituto Federal Fluminense
Búzios - RJ

DYNAMIC OPTIMIZATION USING ADAPTIVE GRID REFINEMENT WITH WAVELETS TRANSFORM in PYTHON

Isabella da Rocha Faria[1] - isabellaf@id.uff.br
Diego Martinez Prata[1] - pratadiego@gmail.com
Lizandro de Sousa Santos[1] - lizandrosousa@id.uff.br
[1]Departamento de Engenharia Química e de Petróleo, Escola de Engenharia, Rua Passo da Pátria, 156
CEP 24210-240, Universidade Federal Fluminense, Niterói, RJ, Brazil.

Abstract. *The dynamic optimization is a useful mathematical tool to find the optimal solution of problems where a significant variability with the time of the present variables is observed. The solution to these problems may become difficult in some cases, such as problems with temporal constraints or changes in the solution profile over time. In such cases, both the quality of the solution and the computational cost can be improved. This work aims to develop an adaptive numerical method, based on wavelets analysis, to solve dynamic optimization problems. The method was developed in the Python programming language, using the Gekko package. Two cases of dynamic optimization were solved and the results showed that the computational cost could be significantly reduced.*

Keywords: *dynamic optimization, wavelets, grid adaptation*

1. INTRODUCTION

Dynamic optimization issues are of importance both industrially and scientifically. Algorithms to solve such problems, in specific, direct methods, usually require significant computational burden, mainly because the discretization level needed to capture the discontinuities of the model can be very high. In this sense, large-scale optimization problems may be difficult to be solved (Logist et al., 2012; Santos et al., 2014).

This paper seeks new implementations of numerical techniques for solving problems using the direct sequential method with the aid of the wavelets transform. The method is based on the works of Binder et al. (2000), Hartwich et al. (2010) and Assasa and Marquardt (2014) that used the adaptive wavelets for solving dynamic optimization problems.

This work aims to contribute with a different strategy in adaptive method application using the wavelets transform. The main contribution is the application of the wavelets to calculate the optimal discretization of the dynamic optimization problem in a single step (insertion of points), unlike the one proposed by Assassa and Marquardt (2014) that use an algorithm of insertion and elimination of points two steps). The main advantage of this methodology is the increasing discretization of the discretization mesh, without the need to eliminate unnecessary points, according to the criterion adopted. Another contribution was the implementation of the method in Python software, through the computational package Gekko. To date there are no adaptive wavelet methods applied in the Python language.

This paper is organized as follows. In Section 2, we present a general formulation of dynamic optimization problems. Section 3 a brief introduction to wavelets analysis. Section 4 presents the adaptive algorithm and Section 5 compares adaptive and equidistant grids. Finally, in Section 6, concludes the paper by summarizing the results and achieve objectives.

2. GENERAL DYNAMIC OPTIMIZATION PROBLEM

A class of typical dynamic optimization problems of chemical engineer can be formulated as follows:

$$\min_{u(t),p,t_f} \left[\int_{t_0}^{tf} J_0(x(t), y(t), u(t), p, t)dt + J_1(x(t_f), y(t_f), u(_f), p, t_f) \right] \tag{1}$$

$$\begin{aligned} s.t.\ \ & f[x(t); x(t), y(t), u(t), p, t] = 0, \ \ x(t_0) = x_0 \\ & g[x(t); x(t), y(t), u(t), p, t] \leq 0 \\ & h[x(t), y(t), u(t), p, t] = 0 \\ & e[x(t_f), y(t_f), u(t_f), p, t_f] = 0 \\ & x(t) \in [x_L, x_U], \ \ y(t) \in [y_L, y_U], \ \ u(t) \in [u_L, u_U] \end{aligned}$$

Where J is the objective functional, f represents the differential-algebraic equations (DAEs) system, $g(t)$, $h(t)$ and $e(t_f)$ are, respectively, the inequality, the equality and endpoint constraints. p refers to time-independent parameters, $u(t)$ are the control variables, $x(t)$ and $y(t)$ are the differential, and the algebraic variables, respectively. The index L denotes lower bound and the index U, upper bound.

Each component of $u(t)$ is parameterized whithin the domain $t \in [t_0, t_f]$ inn direcy sequential methods. Dividing this domain of optimization is a convenient way to parameterize control variable. In $n_s \geq 1$, n_s is a control stages, $t = [t_1, ..., t_{n_s}]$. A polynomial is used on each time interval as:

$$u(t) = \varrho(t, \tilde{\upsilon}) \tag{2}$$

$\varrho(t, \tilde{\upsilon})$ is a polynomial with the coefficientes $\tilde{\upsilon} = \{\upsilon_1, ..., \upsilon_{n_s}\}$. This work considers the control variable to be constant in the stages, so the algorithm calculates a constant $u(t)$ in each stage $\upsilon_i = u(t_i), i = 1, ..., n_s$.

3. WAVELETS

Wavelets are advantageous in the analysis of non-stationary signals. Its application can be encountered in several areas such as engineering, physics, mathematics, among others (Santos et al., 2014).

Basically, wavelets are a family if functions derived from one single function defined as the mother wavelet, ψ:

$$\psi_{n,m}(t) = 2^{n/2}\psi(2^n t - m), \quad m = 0, \ldots, 2^n - 1, \quad n = 0, \ldots, K - 1 \tag{3}$$

The term denoting the decomposition level of the wavelet is n, the integral index, K is the maximum n value and m refers to the translation of the index at a specific level.

The control variable function transformation, $u(t)$, into the wavelet domain as:

$$d_{n,m} = \langle u(t), \psi_{m,n}(t) \rangle \tag{4}$$

where $d_{n,m}$ is a scalar wavelet detail which stores the function characteristics in the indices, this equation shows the inner product between $u(t)$ and $\psi_{m,n}(t)$ in Euclidian space, and the function $u(t)$ is defined:

$$u(t) = \wp(t, \tilde{v}), \quad \tilde{v} = v_1, v_2, \ldots, v_{n_{s-1}}, v_{n_s} \tag{5}$$

The Equation 5 can be approximated by the wavelet expansion:

$$u(t) \approx c_{0,0} \cdot \phi_{0,0}(t) + \sum_{n=0}^{k} -1 \sum_{m=0}^{2^n-1} d_{n,m} \cdot \psi_{n,m}(t) \tag{6}$$

$c_{0,0}$ is the approximated coefficient and the other wavelet form, $\phi_{0,0}(t)$, is the scalar function.

The wavelets presented in this definition are quite frequently used to solve different problems. In this work, the objective is to detect system discontinuities, in these cases, the short wavelets are more effective, so the one used as the Haar wavelet and is written as:

$$\psi(t) = \begin{cases} 1, & t \in [0, 1/2) \\ -1, & t \in [1/2, 1) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

and

$$\phi(t) = \begin{cases} 1, & t \in [0, 1) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The algorithm used in Python allows to calculate $\tilde{d}$ from $\tilde{v}$ by using:

$$\tilde{d} = \tilde{v} \cdot \psi \tag{9}$$

4. ADAPTIVE WAVELET ALGORITHM

The proposed algorithm uses the sequential dynamic optimization solution model, using the Gekko free software, which solves constrained nonlinear optimization problems by providing five solvers: APOPT (Advanced Process OPTimizer), BPOPT (BIM-based Performance Optimization), IPOPT (Interior Point OPTimizer), MINOS and SNOPT (Sparse Nonlinear OPTimizer) (Beal et al., 2018). The IPOPT was the solver chosen for the resolution of the cases proposed in this work. The choice of this solver was based on its effectiveness in solving high-dimension dynamic optimization problems resultant of numerical approximation.

Other free software, PyWavelets (Lee et al., 2006), was used in the application of the wavelet transform, this paper uses the discrete wavelet transform (DWT), which returns the values of the high and low-frequency coefficients.

In the algorithm, each iteration performs the signal analysis of the control variable resulted from the dynamic optimization process. The resulted coefficients after teh wavelet transform represnets the high frequency part of the signal. Thus, the coefficients that have values greater than the correspoding mean value are selected to be refined. In this way, a new point is added in the middle of the selected stage to the optimization be computed in he next iteration.

All tests are performed on PC with intel® Core™ i7i7-8650U Processor (8M Cache, up to 4.20 GHz).

5. CASE STUDIES

In this paper, two case studies are presented and the results are demonstrated in the following sections. In each case, the adaptive wavelets algorithm is compared with the conventional method, using equidistant discretization.

5.1 Case Study I: scalar optimal control problem (Chachuat, 2007):

In this example the goal is miniming the objective function represented by Equation 10, according to the time horizon $t \in [0, 2]$ with the initial conditions $x_1(0) = x_2(0) = 1$. The mathematical model is written as:

$$J = \int_0^2 \frac{1}{2}[x_1(t)]^2 \tag{10}$$

$$\begin{aligned} s.t.\quad & \dot{x}_1(t) = x_2(t) + u(t) \\ & \dot{x}_2(t) = -u(t) \\ & x_1(2) = x_2(2) = 0 \\ & -10 \leq u(t) \leq 10 \end{aligned}$$

Table 1 shows the results obtained from the solution of case I in which the wavelet method is indicated as ”Method I” and the method without the application of wavelets, only using equidistant time grid sizes is represented as ”Method II”. The n_s is the number of stages, time is the solution time and J is objective function value.

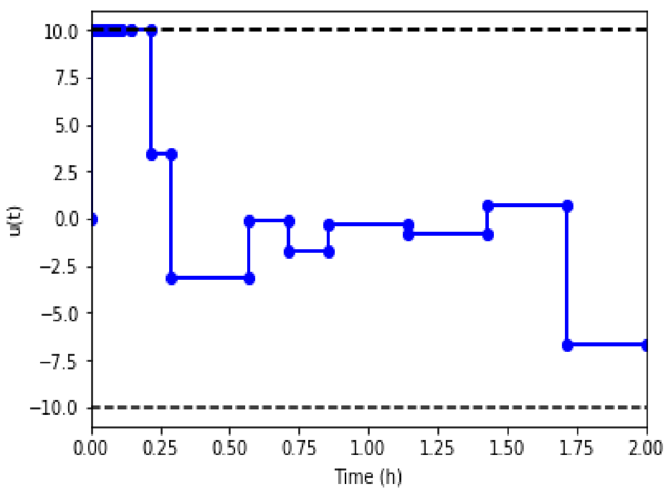

Figure 1- Optimal control profiles for case I.

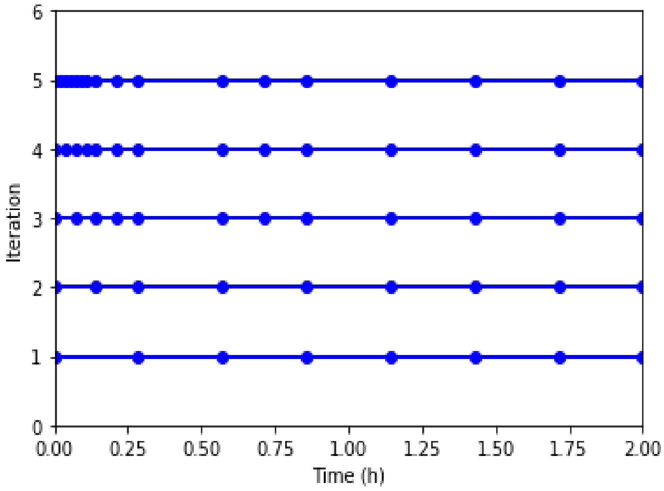

Figure 2- Evolution of the time grids for case I.

Table 1- Test results for fedbatch fermentor problem

Method I	n_s	Time (s)	J	Method II	n_s	Time (s)	J
	8	0.05	5.32		8	0.05	5.32
	9	0.06	4.96		16	0.12	4.93
	12	0.09	4.80		32	0.38	4.75
	14	0.11	4.70		64	1.89	4.67
	17	0.14	4.65		128	12.84	4.63

According to Table 1, the best performance was obtained by the use of the adaptive method resulting in a computation time of 0.14 seconds in 17 stages compared to the Method II which resulted in 12.84 seconds and 128 stages. This corresponds to a reduction of 98.91% regarding the solution time. Notice that, maintaining the quality of the solution, the difference between the objective function values is only 0.43%.

The evaluation of time grids in an adaptative method is present in the Figure 2 and the Figure 1 represents the optimal control for the proposed case. It can be observed in Figure 1 that the profile of the control variable has some discontinuities. An explanation for this result is the low sensitivity of the objective function to the possible discontinuities located in the control variables, causing the optimization algorithm to reach local convergence. However, as shown in Table 1, it is noted that the resulting objective function was satisfactory, within the established error criterion.

5.2 Case Study II: fed-batch fermentor problem (Balsa-Canto et al., 20001)

This problem aims to obtain the maximum concentration of penicillin produced in an feed-batch reactor, with feed rate being the manipulated variable of the optimization problem. The process model is written as:

$$J = x_2(t_f)x_4(t_f) \tag{11}$$

$$
\begin{aligned}
s.t.\quad & \dot{x}_1 = h_1 x_1 - \frac{x_1}{500 x_4} u \\
& \dot{x}_2 = h_2 x_1 - 0.01 x_2 - \frac{x_2}{500 x_4} u \\
& \dot{x}_3 = -\frac{h_1 x_1}{0.47} - \frac{h_2 x_1}{1.2} - \frac{0.029 x_3 x_1}{0.0001 + x_3} + \left(1 - \frac{x_3}{500}\right) \frac{u}{x_4} \\
& \dot{x}_4 = \frac{u}{500} \\
& h_1 = \frac{0.11 x_3}{0.006 x_1 + x_3} \\
& h_2 = \frac{0.0055 x_3}{0.0001 + x_3 (1 + 10 x_3)} \\
& 0 \leq x_1 \leq 40 \\
& 0 \leq x_3 \leq 25 \\
& 0 \leq x_4 \leq 10 \\
& 0 \leq u \leq 50
\end{aligned}
$$

where the optimization domain is $t \in [0, t_f]$, with terminal time t_f equals to 132 h. In the model, J is the objective function to be maximized, u is the feed rate, the biomass concentration, penicillin and substrate are defined as x_1, x_2 and x_3 respectively in grams per liter and the reactor volume in liters is the variable x_4.

The results of the optimization problem is summarized in Table 2. As observed, the fed-batch fermentor problem with proposed method ("Method I") resulted in $J = 87.95$, and 2.13 seconds. On the other hand, these correponding values for the Method-II presented an objective function value of 87.80 and a computational time of 2254.77 seconds. The table also shows that for Method-I only 14 stages of discretization were required, whereas for Method-II, 128 stages were used.

The Figure 3 represents the graphical behavior of the control variable in time and the figure 4 the time grid in each iteration.

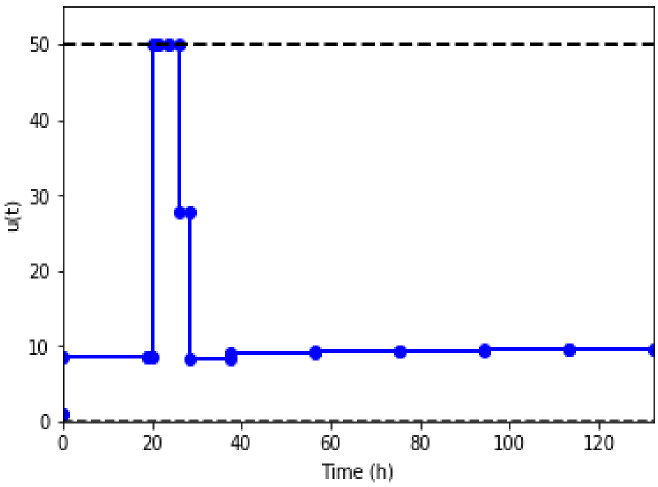

Figure 3- Optimal control profiles for case II.

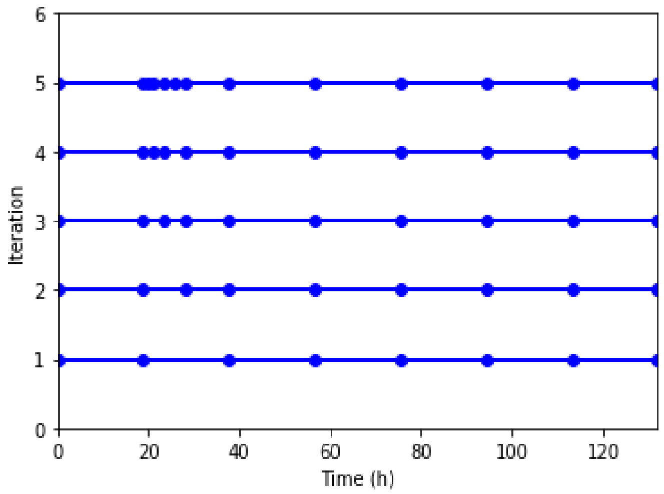

Figure 4- Evolution of the time grids for case II.

Table 2- Test results for fedbatch fermentor problem

Method I	n_s	Time (s)	J	Method II	n_s	Time (s)	J
	8	0.36	86.85		8	0.34	86.85
	10	0.58	87.83		16	1.98	87.60
	11	1.11	87.89		32	19.68	87.53
	12	1.24	87.90		64	176.01	87.72
	14	2.13	87.95		128	2254.77	87.80

6. CONCLUSIONS

The adaptive wavelets method proposed for a refinement in the grid number showed a satisfactory result, meeting expectations, saving computing time and maintaining similar or improving the response quality of the two cases solved. It was possible to verify that the application of the wavelets is useful to optimize the discretization trajectory in the sequential method. In both analyzed cases, there was a significant reduction of the computational cost for a given solution.

It should be noted that improvements can still be realized in the method, such as the incorporation of new wavelet functions, as well as deterministic methods to improve the selection procedure of wavelet coefficients.

Acknowledgements

The authors would like to thank the research funding agency CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for the scholarship granted to the post-graduate student participating in the study.

REFERENCES

Assassa, F.; Marquardt, W. (2014), "Dynamic optimization using adaptive direct multiple sho-oting", *Computers Chemical Engineering*, Elsevier, vol 60, 242-259.

Beal, L.D.R.; Hill, D.C.; Martin, R. A.; Hedengren, J.D. (2018), "GEKKO Optimization Suite", *Processes*, vol 6, 106.

Binder, T.; Cruse, A.; Villar, C. C.; Marquardt, W. (2000), "Dynamic optimization using a wavelet-based adaptive control vector parameterization strategy", *Computers & Chemical Engineering*, Elsevier, vol 24, 1201–1207.

Chachuat, B. (2007), "Nonlinear and dynamic optimization: From theory to practice".

Haar, A. (1910), "DZur theorie der orthogonalen funktionensysteme", *Mathematische Annalen*, Springer, vol 69, 331–371.

Hartwich, A.; Marquardt, W. (2010), "Dynamic optimization of the load change of a large-scale chemical plant by adaptive single shooting", *Computers & Chemical Engineering*, Elsevier, vol 34, 1873–1889.

Logist, F.; Vallerio, M.; Houska, B.; Diehl, M.; Impe, J. V. (2012), "Multi-objective optimal control of chemical processes using acado toolkit", *Computers & Chemical Engineering*, Elsevier, vol 37, 191–199.

Lee, G.; Wasilewski, F.; Gommers, R.; Wohlfahrt, K.; O'Leary, A.; Nahrstaedt, H. (2006), "PyWavelets - Wavelet Transforms in Python", https://github.com/PyWavelets/pywt, accessed 2018-08-16.