 doi:10.20944/preprints201707.0087.v1

Article

Domain's Sweep, a Computational Method to Optimise Chemical and Physical Processes

Alexandre César Balbino Barbosa Filho

Department of Chemical Engineering, Federal University of Campina Grande, Campina Grande, PB 882, BRAZIL; Acbbarbosaf2@gmail.com or alexandrecb.barbosa@eq.ufcg.edu.br; Tel.: +55(82)99610-1789

Abstract: Many of engineering's problems are about to optimise equations, functions and process models equations which appear on common or even complex science's cases. Most applied optimisation methods nowadays can only be used on particular cases, so the main objective of this article is to define a computational method that can optimise or even find target values for specified objective functions and variables, by just using computer effort. Chemical plants involve many differential equations and they can be optimised with more facility now. On this paper, there are shown as example, an optimisation for a chemical reactor, which is found the optimal temperature and volumetric flow rate of feed, for the given objective function being the composition of the desired product. Named Domain's Sweep, it is an algorithm that evaluate given(s) mathematical function(s) and/or equation(s) by varying your(s) independent variable(s) through loops with a given step size and solving it after closing the degrees of freedom, and finally, with some condition statements, store all the optimum values of given or created objective functions with its respective independent variables. In another words, the user create an objective function and this method find the function's maximum, minimum or a certain chosen target value, even if it does not have an inflection point in the given search interval of the independent variables.

Keywords: computational methods; optimisation; plant optimisation; chemical processes; physical processes; computational intelligence; maximum value; minimum value; functions;

1. Introduction

Numerical methods for optimisation of equations usually uses mathematical fundamentals and theories over each specific situation and on most of them, it is necessary a given stop criteria and these methods are just used to find the maximum and the minimum value of a function. Sometimes at some cases, specially the complex ones, these methods do not converge because of the given initial estimate or even the given search interval, or because it does not have a maximum or minimum in the function that is being optimised at a certain range of independent variables values. It is commonly used numerical methods to optimise because most of the problems do not have an analytical solution, like partial differential equations of 3 independent variables and some other non-linear equations that a linearization does not satisfy the specific case because the approximation would not give a true value [1]. So the computational intelligence on some cases is a good tool to be used to solve complex problems and to facilitate its solution.

The computational intelligence is evolving over the years, and it is functionally worth to deal with it on complex tasks that human being could not do by their own hands because of the hard work and the resolution time. With the developing of the CI and the search engineering, many discovered solutions are beyond the human engineer imagination [2]. CI is used to do a lot of tasks, but in some cases it is not preferred because of the difficulty of its architecture, but with the pass of the years it is

changing. One of its uses is a linear search algorithm that seeks a collection and a target item and finds whether the target is in the collection, the run time is linear and the target is found by an organized order of searching through the items of the collection, stopping when it finds the target [3]. A several number of CI algorithms have been developed and published during the last years but as they were not unified, complicated the excellent use of this computational science [4]**.** With the CI, human being tasks (complex or even simple tasks) had been solved by a machine with the implementation of a computing algorithm on the machine`s software, so, the CI theories show interests in knowing how the problems (specified by human being) can be represented and solved by a machine [5].

On many years, chemical and physical processes were optimised based on objective functions, which were linked to the financial business. And instead of finding the operational conditions of a chemical plant's equipment based on its best performance (for example, the maximum conversion of a chemical reactor or the complete separation of a mixture on a column distillation), plants were constructed with clearance.

A degree of freedom (DOF) is the main analysis that must be done before solving any equation, as it is already known, an equation can only be solved if the DOF is equal to zero, which is the number of equations subtracted by the number of independent variables of each equation. The main idea for the algorithm presented in this paper is to close the degree of freedom of the user equation(s) by specifying values for its independent variables at order, through closed loops, and the objective of this paper is to show a computational method that find the optimum or desired value for chosen independents variables in equations, as users will, that flees the divergence problem and lead to the optimum stage of chemical and physical processes.

2. Methodology

Suppose it is required to find the amount of **N** optimum independent variables values that leads to the maximum, minimum or even a certain specified target value of a function (must be defined by the user) on a specified interval of the independent function's variables. It's only needed a computational algorithm that does the following:

I) Specify values to store functions (its value is updated according the hike of the sweep) that will be compared into those ones found by **III** or **IV,** on posterior items **V**, **VI** and **VII**. A minimum store function value must have an initial estimated value of a high positive number, because thus the interval of possible values will be large and no problem will occur at storing the first minimum value. On about storing a maximum store function value it is needed a high negative number as initial estimated value. The target store function value is determined by the user.

II) Sweep the determined function(s) or equation(s) domain with **N** loops, where **N** is the number of function's independent variable. A "sweep" is an expression to refer that a loop is being done to vary a function's independent variable to evaluate its value. The sweep is always in the direction of minus infinite to positive infinite. Each loop varies a different independent variable starting at an initial chosen value to a final chosen value with a given step size. The loops must be done one inside another.

III) Solve equation(s) or function(s) for independent variables in the last loop;

IV) Still in the last loop, define new functions if wanted, that have the same independent variables which values were found in item **III**;

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

V) Then in the same loop, it is put a condition statement that can express that if the current calculated function's value is higher than the current maximum store function value, then, store all the independent variables value and update the maximum store function value to the current value of the calculated function. At the end of all loops, it will have stored all the optimum independent variables values that lead to the maximum value of the defined function by the user.

VI) After closing the first condition statement in last loop, it is put another one that can express that if the current calculated function's value is lower than the current minimum store function value, then, store all the independent variables value and update the minimum store function value to the current value of the calculated function. At the end of all loops, it will have stored all the optimum independent variables values that lead to the minimum value of the defined function by the user.

VII) Before closing the last loop, the last condition statement may do that if the currently calculated function's value is between the limits of 0.999 and 1.001 of the specified target store function value, then, store all the function's and independent variable's value. At the end of all loops, it will have stored the independent variables values that lead to the specified target function's value with the given tolerance limits, and it is common on a function of more than one variable to have several points that gives the desired target function value, so, it must store all these values in a vector. About the tolerance limits, it can be changed with the will of the user.

On some cases it is optionally to the user to put a condition statement in any of other loops (not necessary in the last loop), but remembering that it is not occasionally done in optimisation, because the need of sweep all the function's domain. User may choose which conditions statements will be used (**V**, **VI** and **VII**).

2.1. Algorithm's schema

Since ''V_i'' is any independent variable, a step scheme algorithm of the method for a function of **N** independent variables should be:

Step 1: Function target value = input value;
Step 2: Maximum store function = -10^{32};
Step 3: Minimum store function = 10^{32};
Step 4: Loop 1: Vary V_1 with a given step to a given maximum range
Step 5: Loop 2: Vary V_2 with a given step to a given maximum range
… And so on till loop N
Step 6: Loop N: Vary V_N with a given step to a given maximum range
Step 7: Solve equation(s) or function(s) declared by the user and define new functions (if wanted), based on the independent variables;
Step 8: If 1: (Function's value) > Maximum store function
Then: Maximum store function = (Function's value);
$V_1^{max} = V_1$;
$V_2^{max} = V_2$;
…
$V_N^{max} = V_N$.
End If 1
Step 9: If 2: (Function's value) < Minimum store function

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

Then: Minimum store function = (Function's value);

$V_1^{min} = V_1$;

$V_2^{min} = V_2$;

…

$V_N^{min} = V_N$.

End If 2

Step10: If 3: (Function's value) >= 0.999x (Function target value) and <=1.001x (Function target value)

Then: Store the function and independent variables values in a vector;

$V_1^{target} = V_1$;

$V_2^{target} = V_2$;

…

$V_N^{target} = V_N$.

End If 3;

End loop 1;

End loop 2;

…

End loop N.

When the algorithm's run finishes, all the function(s) maximum, minimum and even a target value are stored with its respective independent variables. Remember that solutions found on **step 8** must converge, so then computer will not enter on an infinite loop. Because of that, it can be optionally put another condition statement to warn if the problem is not converging or the function(s) value diverge, or even to ignore that and keep running.

3. Real-life Impact and applications

3.1. Chemical reactor optimisation

This method is very useful in day-to-day on processes that can be described by algebraic equations. One application for example, is to find the optimum operational conditions in a chemical reactor by solving the process model equations for independent variables and creating an objective function, like the conversion of a specific reactant or even the exit's composition of a desired product and so on.

Now, it is shown the optimisation of a continuous stirred tank reactor (CSTR), which the objective was to maximize the composition of the desired product (B) at the reactor's exit [6-9]. The example was performed by MATLAB® (R2013a, Mathworks, Natick, MA, USA):

Reactions occurring:

$$A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C \quad (1)$$

$$A \overset{k_3}{\rightarrow} \frac{1}{2} D \quad (2)$$

Energy balance for reactor:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

$$\frac{dT}{dt} = \frac{T_f - T}{\tau} + \frac{(-Q_c + Q_r)}{\rho . V . c_p} \quad (3)$$

Energy balance for coolant jacket:

$$\frac{dT_w}{dt} = \frac{T_{wo} - T_w}{\tau_j} + \frac{Q_c}{\rho_c . V_j . c_{pj}} \quad (4)$$

Component molar balances:

$$\frac{dC_a}{dt} = \frac{C_{af} - C_a}{\tau} - r_1 - r_3 \quad (5)$$

$$\frac{dC_b}{dt} = \frac{C_{bf} - C_b}{\tau} + r_1 - r_2 \quad (6)$$

$$\frac{dC_c}{dt} = \frac{C_{cf} - C_c}{\tau} + r_2 \quad (7)$$

$$\frac{dC_d}{dt} = \frac{C_{df} - C_d}{\tau} + \frac{r_3}{2} \quad (8)$$

Where,

$$Q_r = -\Delta H_{r1} . r_1 - \Delta H_{r2} . r_2 - \Delta H_{r3} . r_3 \quad (9)$$

$$Q_c = U . A_L . (T - T_w) \quad (10)$$

$$r_1 = k_1 . e^{(\frac{-Ea_1}{T})} . C_a \quad (11)$$

$$r_2 = k_2 . e^{(\frac{-Ea_2}{T})} . C_b \quad (12)$$

$$r_3 = k_3 . e^{(\frac{-Ea_3}{T})} . {C_a}^2 \quad (13)$$

The table below shows the meaning of each variable used on this application:

Table 1. Variables meaning.

Variable	Meaning
ρ	Fluid density inside the reactor
MW	Molecular weight
C_{af}	Feed concentration of A
C_{bf}	Feed concentration of B

Table 1. *Cont.*

Variable	Meaning
C_{cf}	Feed concentration of C

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

C_{df}	Feed concentration of D
ΔH_{r1}	Enthalpy of the reaction 1
ΔH_{r2}	Enthalpy of the reaction 2
ΔH_{r3}	Enthalpy of the reaction 3
c_p	Inside reactor's fluid heat capacity
T_{wo}	Feed Coolant's jacket fluid temperature
ρ_c	Coolant's jacket fluid density
c_{pj}	Coolant's jacket fluid heat capacity
k_1	Pre-exponential factor of reaction 1
k_2	Pre-exponential factor of reaction 2
k_3	Pre-exponential factor of reaction 3
Ea_1	Activation energy of reaction 1
Ea_2	Activation energy of reaction 2
Ea_3	Activation energy of reaction 3
F_j	Coolant's jacket fluid volumetric flow rate
D	Reactor's diameter
R	Reactor's radius
h_o	Reactor's fluid level
U	Global heat transfer coefficient
esp	Coolant's jacket thickness
A_L	Reactor's lateral area
V	Reactor's volume
V_j	Coolant's jacket volume
τ_j	Coolant's jacket residence time
C_{af}	Feed concentration of A
τ	Reactor's residence time
T	Reactor's temperature
T_w	Coolant's jacket temperature
C_a	Molar concentration of A
C_b	Molar concentration of B
C_c	Molar concentration of C
C_d	Molar concentration of D
X_B	Composition of B at Reactor's exit
T_f	Feed temperature
F_e	Reactor's fluid volumetric flow rate

Then it was specified fixed parameters of the process model equations:

Table 2. Parameters for the simulation.

Parameter	Value	Unit
ρ	1000	$Kg.m^{-3}$
MW	40	$Kg.Kmol^{-1}$
C_{bf}	0	$Kmol.m^{-3}$
C_{cf}	0	$Kmol.m^{-3}$
C_{df}	0	$Kmol.m^{-3}$
ΔH_{r1}	4200	$KJ.Kmol^{-1}$

Table 2. *Cont.*

Parameter	Value	Unit
ΔH_{r2}	-11000	$KJ.Kmol^{-1}$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

ΔH_{r3}	-41850	$KJ.Kmol^{-1}$
c_p	2.8121	$KJ.Kg^{-1}.K^{-1}$
T_{wo}	306	K
ρ_c	960	$Kg.m^{-3}$
c_{pj}	1.6	$KJ.Kg^{-1}.K^{-1}$
k_1	$1.287x10^{12}$	min^{-1}
k_2	$1.287x10^{12}$	min^{-1}
k_3	$9.043x10^{9}$	$m^3.min^{-1}.Kmol^{-1}$
Ea_1	9758.3	K
Ea_2	9758.3	K
Ea_3	8560	K
F_j	30	$m^3.min^{-1}$
D	2	m
R	1	m
h_o	4	m
U	17.7	$KJ.m^{-2}.K^{-1}.min^{-1}$
esp	0.25	m

Some other terms appears in the balance equations, so it must be calculated before the simulation. Assuming an approximately cylinder volume for reactor and that the coolant jacket includes the top, base and the side of the reactor and has a specified thickness of refrigerant fluid, it follows that the geometric calculus arrive to:

$$A_L = 2.\pi.h_o.R = 25.1327\ m^2 \tag{14}$$

$$V = \pi.R^2.h_o = 12.5664\ m^3 \tag{15}$$

Then, the volume of the coolant jacket is represented by a cylindrical shell with a specified thickness:

$$V_j = \pi.h_o.esp.(2.R + esp) = 7.0686\ m^3 \tag{16}$$

The refrigerant fluid used was the DOWTHERM-Q and your residence time is:

$$\tau_j = \frac{V_j}{F_j} = 0.17017\ m^3 \tag{17}$$

As shown in Table 1, it was adopted a value of 40 $Kg.Kmol^{-1}$ to the molecular weight of the reactant A, and a medium value of $1000\ Kg.m^{-3}$ for the reactor's fluid density. Then the feed concentration value of A is given by:

$$C_{af} = \frac{\rho}{MW} = 25\ \frac{Kmol}{m^3} \tag{18}$$

After declared all fixed parameters and store functions, it was used the domain's sweep algorithm to architecture the optimisation of the process. It was varied through the loops, the feed temperature and the feed volumetric flow of the reactant "A" on an interval of 307 to 407 and 0.1 to 100, respectively, with a step size of 0.1 for both independent variables. But to solve the balance

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

equations it needs the value of the reactant "A" residence time inside the reactor, so it was written in the last loop of the code its formula:

$$\tau = \frac{V}{F_e} \tag{19}$$

Note that the value of τ changes with the change of the feed volumetric flow of reactant A, so its value is updated at each iteration. Also in the last "loop" of the algorithm's code it was solved the balance equations using Runge-Kutta's method for ordinary differential equations (ODE), declared conditions statements to store the optimum values over the store functions and created the objective function of interest (The composition of product B). As the example was performed by MATLAB®, the command to do the loop, the conditional statements and to solve the system of ODES were "for" and "if" respectively, according to its programming language.

The simulation (Solving the balance equations) was done in the last "for" and it was done by setting initial conditions, based on feed properties:

Table 3. Initial conditions for the simulation (Same values of Table 2).

Dependent variable	Initial condition
T	T_f (it is being varied)
T_w	T_{wo}
C_a	C_{af}
C_b	C_{bf}
C_c	C_{cf}
C_d	C_{df}

As follows the Domain's Sweep algorithm and with the results in hands, the objective function created was the composition of product B that is function of the steady-state value (SSV) of the concentration of reactant A, product B, C and D found in the simulation of the balance equations. So the simulation was run with a big considerably time so that the values could arrive the steady-state. After the MATLAB® code was run the maximum value of the composition of product B (objective function) and its feed temperature and feed volumetric flow of the reactant "A" (operational conditions) attached were:

$$X_B = \frac{C_{b-ssv}}{C_{a-ssv} + C_{b-ssv} + C_{c-ssv} + C_{d-ssv}} = 0.1924 \tag{20}$$

$$T_f = 366.6\ K \tag{21}$$

$$F_e = 100\ \frac{m^3}{min} \tag{22}$$

Remembering that since the operational conditions are the values, which have the maximum value for the objective function, on this case, the store function created before the loops was a maximum store function that has its values updated by doing *step 8* of algorithm's schema. Also in the same step, the independent variables (T_f, F_e) were being stored in a store function when the conditional statement was true.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

The results of the simulation of the process model are shown. First it is shown an analysis of sensibility of the composition of product "B" with a fixed optimum value of 366.6 K for the feed temperature:

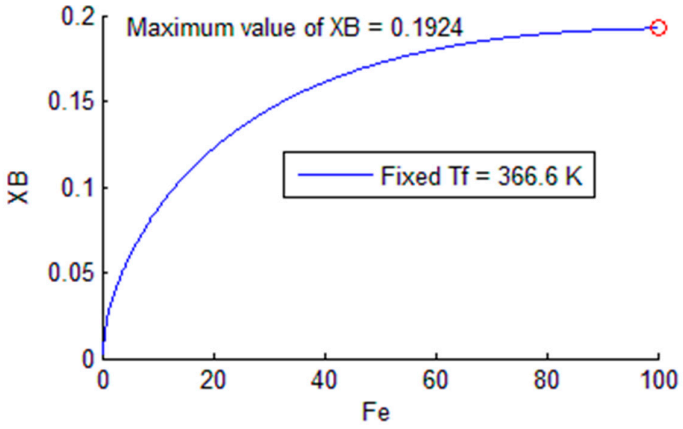

Figure 1. Analysis of sensitivity: X_B versus F_e to show the consistence of the results.

As the found optimum values that give the maximum value for the composition of B were a feed temperature of 366.6 K and a feed volumetric flow of reactant A of 100 m^3, then, if it is plotted a graph of X_B versus F_e, the maximum value must be at the point (100, 0.1924) and as it can be seen in Figure 1, the composition of B has a maximum value at the found optimum operational conditions ($T_f = 366.6\ K$ and $F_e = 100$). But it is not enough at all, on a chemical process just to do the optimisation, so it is needed a proof that the process is real and stable, so since the simulation was done when it was also doing the optimisation, it is also plotted below the dynamic plot of the process variables of the system. Figure 2 and 3 show that the temperature and the concentration of the reactants and products remains with real values in the whole simulation and get a constant and real value as they reach the steady-state, proofing, that the system is real, stable and reachable.

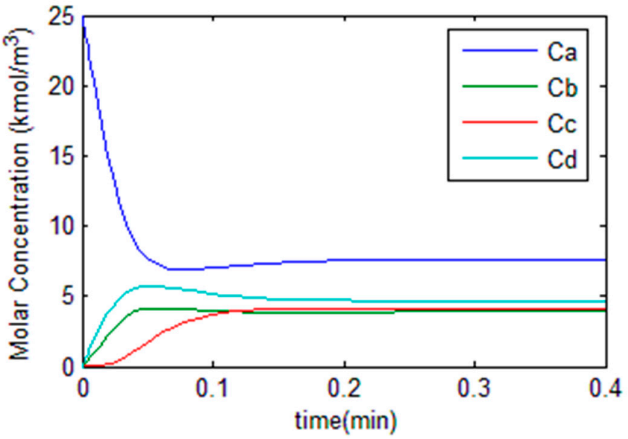

Figure 2. Molar concentrations of reactants and products.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

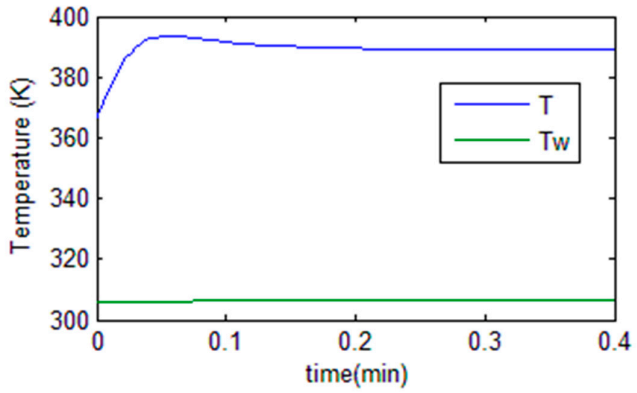

Figure 3. Reactor's and coolant jacket's temperature.

The system is stable as shown on the graphs, and with the found optimum operational conditions, the steady-state values for the reactor's temperature, coolant jacket's temperature and the exit molar concentration of A, B, C and D with their given units are 389.286, 306.55, 7.636, 3.912, 4.107 and 4.672, respectively.

Also, a 3D-plot and a contour graph were done to enhance the reader's understanding, as follow in Figure 4 and 5:

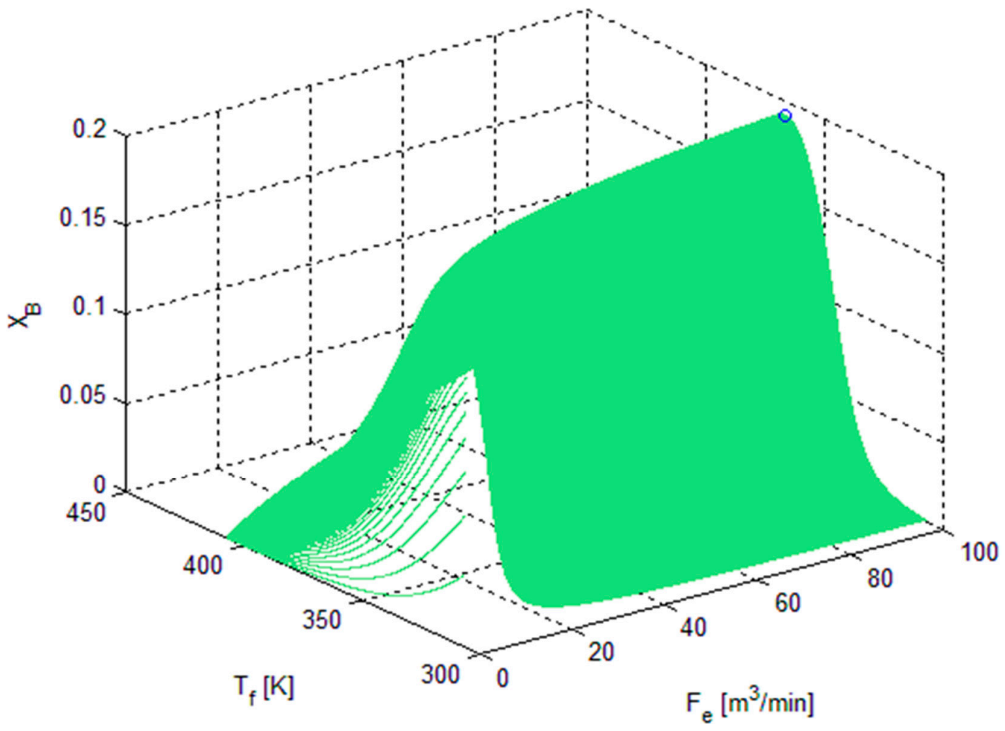

Figure 4. 3D-plot of the composition of product B, as a function of feed temperature and feed volumetric flow rate with a blue circle on the maximum value.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 July 2017 doi:10.20944/preprints201707.0087.v1

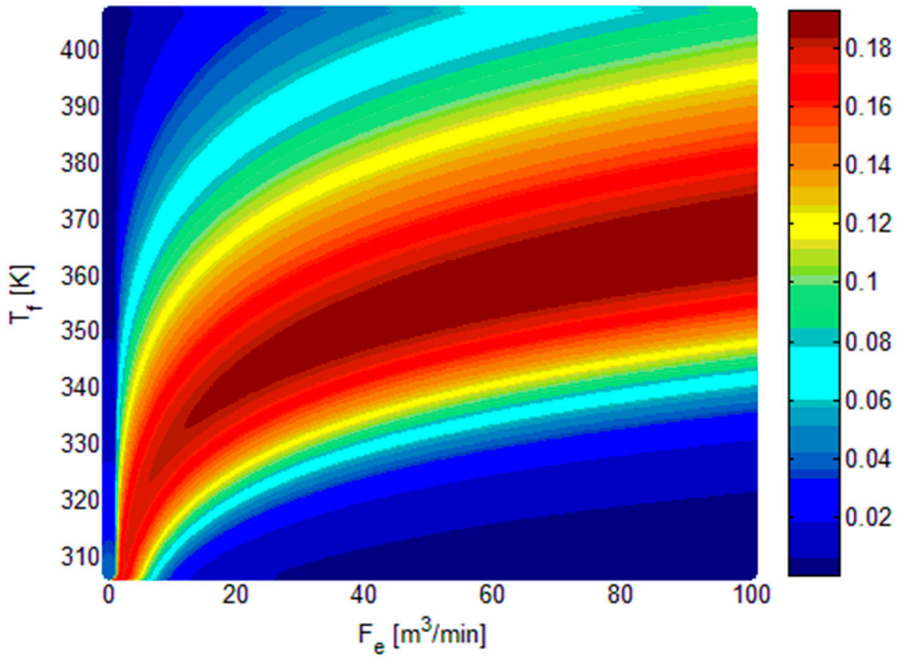

Figure 5. Contour graph showing the composition values of product B in the colorbar.

4. Conclusions

Of easy understanding and application, it is a method to find the optimum value as the user choices, remembering that according to the user's code, it can find the maximum, minimum or a determined function value and the accuracy of the found value compared to the real value is linked to the step's size criteria on about varying the independent variables of the function(s). Smaller the step size, higher the accuracy of the optimum value and higher the machine processing time. In respect of time, the Domain's Sweep algorithm's delay time to finish is proportional to step's size criteria, specified sweep's range and the number of functions independent variables.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Chapra, Steven C.; Canale, Raymond P. *Numerical Methods for engineers*, 6th ed.; McGraw-Hill: 1221 Avenue of the Americas, New York, NY 10020, USA, 2010; pp. 3–21.
2. Downey, Allen B. *Think Complexity*, version 1.2.3; Green Tea Press: 9 Washburn Ave, Needham 02492, USA, 2012; pp. 6–8.
3. Downey, Allen B. *Think Complexity*, version 1.2.3; Green Tea Press: 9 Washburn Ave, Needham 02492, USA, 2012; pp. 23-24.
4. Xing, B.; Gao, W.J. *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, volume 62; Springer International Publishing: Switzerland, 2014; pp. 3-9.
5. Konar, A. *Computational Intelligence: Principles, Techniques and Applications*, 1st ed.; Springer International Publishing: Switzerland, 2005; p. 5.
6. Incropera, Frank P.; DeWitt, David P.; Bergman, Theodore L.; Lavine, Adrienne S. *Fundamentals of Heat and Mass Transfer*, 6th ed.; John Wiley & Sons, Inc.: New Jersey, USA, 2007; pp. 62-84.
7. Seborg, Dale E.; Edgar, Thomas F.; Mellichamp, Duncan A.; Doyle III, Francis J. *Process Dynamics and Control*, 3rd ed.; John Wiley & Sons, Inc.: New Jersey, USA, 2011; pp. 14-33.

 doi:10.20944/preprints201707.0087.v1

8. Magalhães, Otto I. B. *Desenvolvimento de um sistema de otimização dinâmica em tempo real*. Master's Program, UFRJ, Rio De Janeiro, Brazil, 2010; pp. 69-73.

9. Smith, J. M.; Van Ness, H. C.; Abbott, M. M. *Introduction to Chemical Engineering Thermodynamics*, 6th ed.; McGraw-Hill Education: New York, NY, 2001; pp. 34-39.