The Kaldor-Kalecki Model: Estimation By Using Modified Differential Evolution Heuristic

Zuzana Olajcova[1]

Keywords: Differential Evolution heuristic, Kaldor-Kalecki model estimation, optimization over continuous domain.

Introduction

The presented algorithm is based on the Differential Evolution (DE) algorithm, which is a branch of evolutionary programming developed by Rainer Storn and Kenneth Price [1] for optimization problems over continuous domains. Original heuristic was enhanced by adding a control variable representing the order of refinement of the differential weight parameter. The motivation of this enhancement lays in a complex form of the optimized objective function, which resulted from historical data of the U.S. economics and its sensitivity to the differential weight parameter. By optimization of this function an estimate of the new specification of the Kaldor-Kalecki model of the economic cycle is obtained.

Basic Algorithm

A basic frame of the DE algorithm works by having a population of candidate solutions (called agents). These agents are moved around in the search-space by using simple mathematical formulae to combine the positions of existing agents from the population. If the new position of an agent is an improvement it is accepted and forms part of the population, otherwise the new position is simply discarded. The process is repeated by doing so.

Algorithm 1 Differential Evolution

Initialize all agents $\mathbf{x} = (x_1, x_2, ..., x_N)$ with random positions in the search-space.
repeat
  for $i := 1$ do N **do**
- Pick agents $\mathbf{a}$,$\mathbf{b}$, and $\mathbf{c}$ from the population at random (distinct from each other as well as from agent $\mathbf{x}$)
- Pick a random index $R \in \{1, \ldots, n\}$ (n being the dimensionality of the problem to be optimized)
- Compute the agent's potentially new position $\mathbf{y} = [y_1, \ldots, y_n]$ as follows:
  - For each $i \in \{1, \ldots, n\}$, pick a uniformly distributed number $r_i \equiv U(0,1)$
  - **if** ($r_i < CR$ **or** $i = R$) **then** $y_i = a_i + F \times (b_i - c_i)$ **else** $y_i = x_i$
  - (In essence, the new position is the outcome of the binary crossover of agent $\mathbf{x}$ with the intermediate agent $\mathbf{z} = \mathbf{a} + F \times (\mathbf{b} - \mathbf{c})$.)
- If $f(\mathbf{y}) < f(\mathbf{x})$ then replace the agent in the population with the improved candidate solution, that is, replace $\mathbf{x}$ with $\mathbf{y}$ in the population.
- Pick the agent from the population that has the highest fitness or lowest cost and return it as the best found candidate solution.

  until termination criterion is met (e.g. number of iterations performed, or adequate fitness reached);

Modified Algorithm

A big value of differential weight parameter F causes that the result is uninterpretable and the value of the objective function flies to minus infinity and complex space. On the other hand, a small value of this parameter causes solution to stuck at a flat local minimum. Hence, a modification was implemented, which performs a rough approach by several steps with larger differential weight and refines it by half to perform more subtle approach. The number of repetitions of this procedure M represents the order of refinement of the differential weight parameter as a new control variable.

[1]CTU FNSPE, olajczuz@cvut.fjfi.cz

Algorithm 2 Modified Differential Evolution

Initialize all agents $\mathbf{x} = (x_1, x_2, ..., x_N)$ with random positions in the search-space.
for $j := 1$ do M **do**
  repeat
    for $i := 1$ do N **do**

- Pick agents **a**,**b**, and **c** from the population at random (distinct from each other as well as from agent $\mathbf{x}$)
- Pick a random index $R \in \{1, \ldots, n\}$ (n being the dimensionality of the problem to be optimized)
- Compute the agent's potentially new position $\mathbf{y} = [y_1, \ldots, y_n]$ as follows:
  - For each $i \in \{1, \ldots, n\}$, pick a uniformly distributed number $r_i \equiv U(0, 1)$
  - **if** $(r_i < CR$ **or** $i = R)$ **then** $y_i = a_i + F \times (b_i - c_i)$ **else** $y_i = x_i$
  - (In essence, the new position is the outcome of the binary crossover of agent $\mathbf{x}$ with the intermediate agent $\mathbf{z} = \mathbf{a} + F \times (\mathbf{b} - \mathbf{c})$)
- If $f(\mathbf{y}) < f(\mathbf{x})$ then replace the agent in the population with the improved candidate solution, that is, replace $\mathbf{x}$ with $\mathbf{y}$ in the population.
- Pick the agent from the population that has the highest fitness or lowest cost and return it as the best found candidate solution.

    until termination criterion is met (e.g. number of iterations performed, or adequate fitness reached);
Refine differential weight parameter $F = \frac{F}{2}$.

The Kaldor-Kalecki model

The Kaldor-Kalecki model is a bussines cycle model arosed out of combination of Kaldor's two dimensional model on national income and capital with Kaleckian idea of a gestation lag of investment between "investment order" and "investment installation". Since the Kaldor-Kalecki model becomes a system of delay differential equations of the form [6]

$$\begin{aligned} \dot{Y}(t) &= \alpha[I(Y(t), K(t)) - S(K(t))] \\ \dot{K}(t) &= I(Y(t-\theta), K(t-\theta)) - \delta K(t) \end{aligned} \tag{1}$$

where savings $S(Y, K) = \gamma Y$ are proportional to income ($\gamma \in (0, 1)$), investment $I(Y, K)$ is positively related to income Y and negatively related to capital K, δ represents the depreciation rate of capital stock ($\delta \in (0, 1)$) and $\alpha > 0$ is the adjustment coefficient in the goods market - small value means slow reaction of companies to unbalance (this behavior can be caused by aversion to risk or by market monopoly). In the picture 5 is provided a one of many numerical simulations of the Kaldor-Kalecki model specification adopted from W. Hu, H. Zha and T. Dong [7] - to clarify the motivation of this work.

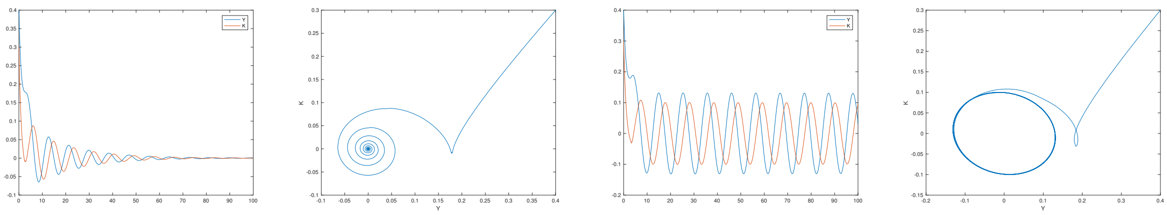


Figure 1: Output of the model for different setup of gestation lags.

On the left side, there is a figure of national income and capital stock with gestation lag set to $\theta = 2.5$ and corresponding phase diagram. This setup converges by damping oscillations to the asymptotically stable equilibrium. On the right side, there are figures of the same model with gestation lag set to $\theta = 3.1382$ showing divergent periodic oscillations, hence this setup has unstable equilibrium. From figures of both setups might be noticed periodical slumps of modeled variables into negative values - what is against the economic logic. The aim of this work is to estimate a specification of the Kaldor-Kalecki model with emphasis on realistic economic explanation. Model estimation is performed by using U. S. historical data from 1950 to 2014 and a new form of investment function:

$$I(Y(t), K(t)) = (\sum_{j=0}^{2} c_j (i_1 Y(t - \theta_j) + i_2 K(t - \theta_j))^{\epsilon})^{\frac{1}{\epsilon}}, \tag{2}$$

where ϵ can be interpreted as "elasticity" of substitution in between investments implemented in various years, i_1 is the contribution of income to investment growth, whereas i_2 plays an analogical role with regard to investment decline. Hence, expected values of these parameters are $i_1 \in (0, 1)$ and $i_2 \in (-1, 0)$. Parameters c_0 , c_1 and c_2

represent impact of investments implemented in individual periods in total sum and θ_0, θ_1, θ_2 respective gestation lags. Hence, expected values of these parameters are $c_0 \geq c_1 \geq c_2 > 0$. By installing the investment function 2 into the Kaldor-Kalecki model we get

$$\begin{aligned} \dot{Y}(t) &= \alpha(i_1 Y(t) + i_2 K(t) - \gamma Y(t)), \\ \dot{K}(t) &= (\sum_{j=0}^{2} c_j (i_1 Y(t-\theta_j) + i_2 K(t-\theta_j))^{\epsilon})^{\frac{1}{\epsilon}} - \delta K(t), \end{aligned} \tag{3}$$

what represents a new specification with ability to sum the impact of investments over several recent periods.

Estimation of the new specification

Objective of this work is to estimate parameters of a new specification of the Kaldor-Kalecki model 6, namely α, i_1, i_2, γ, c_0, c_1, c_2, ϵ and δ by using U. S. historical data.

Since in the system act apart of $Y(t)$ and $K(t)$ members also their derivatives $\dot{Y}(t)$ and $\dot{K}(t)$, estimation of their values is performed by applying Savitzky-Golay filter [5]. The filters principle is an approximation of the group of successive observations by the polynomial. Output is a polynomial value at the central point of the current group of observations. In order to minimize the loss of observation, the first degree polynomial was chosen, with five observations in a group. This setup corresponds to convolutional coefficients -2, -1, 0, 1, 2 and filter of the form

$$\dot{Y}_t = \frac{-2Y_{t-2} - Y_{t-1} + Y_{t+1} + 2Y_{t+2}}{10}, \quad \dot{K}_t = \frac{-2Y_{t-2} - Y_{t-1} + Y_{t+1} + 2Y_{t+2}}{10}. \tag{4}$$

Estimates of derivatives of income $\dot{Y}(t)$ and capital $\dot{K}(t)$ are obtained by applying the filter to underlying data.

The estimates of parameters are obtained as a solution of a following nonlinear least squares problem:

$$\sum_{j=3}^{n-2} (\dot{Y}_j - \alpha(i_1 Y_j + i_2 K_j - \gamma Y_j))^2 + \sum_{j=3}^{n-2} (\dot{K}_j - (\sum_{j=0}^{2} c_j (i_1 Y(t-\theta_j) + i_2 K(t-\theta_j))^{\epsilon})^{\frac{1}{\epsilon}} - \delta K(t))^2 \rightarrow min, \tag{5}$$

Investment lags are set to $\theta_0 = 0$, $\theta_1 = 1$, $\theta_2 = 2$ in order to avoid loss of data accuracy by potential interpolation. Since 5 represents a multimodal problem, there is a risk of finding a local minimum as a solution.

Table 1: Solution - Simple Algorithm, pt.1

F	2	1	0.5
α	0	0	0.5591
i_1	0.2317	0.2678	0.3000
i_2	-0.1007	-0.1005	-0.0900
γ	0.1341	0.0088	0.0000
c_0	0.6425	0.6612	0.9979
c_1	0.4978	0.1881	0.0000
c_2	0.4528	0.4452	1.0000
ϵ	0.0001	0.0004	0.2607
δ	0.5345	0.0500	0.0500
$f(\alpha, i_1, i_2, \gamma, c_0, c_1, c_2, \epsilon, \delta)$	3.6987e+6 - $i\infty$	$-\infty + NaNi$	1.3073e+8 - 1.6565e+7i

Table 2: Solution - Simple Algorithm, pt.2

F	0.25	0.1	0.05	0.001
α	0.0004	0.0004	0.5695	0.1659
i_1	0.3000	0.2405	0.2994	0.2894
i_2	-0.0901	-0.0900	-0.0900	-0.0945
γ	0.0000	0.0206	0.0313	0.1610
c_0	0.1356	0.2678	0.0501	0.0076
c_1	0.0337	0.1232	0.9173	0.1139
c_2	0.1381	0.0911	0.5046	0.1889
ϵ	0.3608	0.0890	0.1309	0.2339
δ	0.0500	0.0541	0.1253	0.2405
$f(\alpha, i_1, i_2, \gamma, c_0, c_1, c_2, \epsilon, \delta)$	2.85e+8 - 4.36e+4i	3.17e+8 - 1.17e+1i	6.35e+8 - 3.70e+7i	3.49e+9 - 2.28e+3i

Table 3: Solution - Modified Algorithm, pt.1

M	10	8	6	4	2
F	0.1	0.1	0.1	0.1	0.1
α	0.0224	1.0865	0.2759	0.1461	0.0477
i_1	0.2999	0.2996	0.2914	0.3000	0.2995
i_2	-0.0902	-0.0900	-0.0901	-0.0911	-0.0901
γ	0.0148	0.0001	0.0000	0.0003	0.2608
c_0	0.2300	0.0455	0.1667	0.1161	0.4690
c_1	0.0130	0.2164	0.2203	0.9997	0.1338
c_2	0.3209	0.3212	0.1534	0.2934	0.1674
ϵ	0.5711	0.8512	0.3246	0.6667	0.3239
δ	0.0573	0.1026	0.3897	0.0530	0.2290
$f(\alpha, i_1, \ldots, \delta)$	3.3e+8 - 2.9e+5i	7.9e+8 - 1.5e+5i	8.47e+9 - 5.79e+5i	2.8e+8 - 5.9e+5i	3.2e+9 - 1.7e+6i

Table 4: Solution - Modified Algorithm, pt.2

M	10	8	6	4
F	0.05	0.05	0.05	0.05
α	1.0068	0.0398	0.0284	0.0673
i_1	0.2949	0.2999	0.3000	0.3000
i_2	-0.0901	-0.0900	-0.0900	-0.0901
γ	0.0012	0.0001	0.0447	0.0001
c_0	0.9991	0.4413	0.4101	0.3160
c_1	0.1876	0.1487	0.5009	0.1390
c_2	0.0588	0.1473	0.6119	0.1325
ϵ	0.0891	0.7627	0.1593	0.5037
δ	0.0502	0.0500	0.0500	0.0500
$f(\alpha, i_1, i_2, \gamma, c_0, c_1, c_2, \epsilon, \delta)$	2.31e+8 - 9.01e+6i	2.71e+8 - 1.99e+5i	1.36e+8 - 1.69e+7i	2.78e+8 - 2.67e+5i

By the basic algorithm we get the results listed in the Table 1 and Table 2. Setups with various differential weight parameters were tested. Every solution apart of the one for $F = 0.001$ is not economically interpretable. In addition, the value $F = 0.001$ is so small, that no newfound position produced an improvement during the computation.

By the modified algorithm we get the results listed in the Table 3 and Table 4. Setups with various orders of refinement of the differential weight parameter M were tested, with differential weight parameter set to $F = 0.1$ and $F = 0.05$. Acceptable solutions were obtained by using $F = 0.1, M = 8$, $F = 0.1, M = 6$, $F = 0.05, M =$

10, $F = 0.05, M = 8$ and $F = 0.05, M = 4$. The best one arosed out of combining $F = 0.05$ and $M = 4$, having a shape close to the modeled data. The estimate meets the expected parameter values as the rate of income response to the increment in investments α is positive, the propensity for savings meets $\gamma \in (0, 1)$, as well as for the capital depreciation parameter applies $\delta \in (0, 1)$. For the "elasticity" parameter of the substitution in between investments completed in different years applies $\epsilon \in (0, 1)$ and rates of contribution of income/capital to investment meets $i_1 \in (0, 1)$ / $i_2 \in (-1, 0)$. However, parameters representing impact of investments implemented in individual periods in total sum do not meet expected inequalities $c_0 \geq c_1 \geq c_2 > 0$. According to our solution, the investments implemented in the last period considered has a bigger impact on capital stock than investment implemented in the prelast period considered. Despite of this flaw, the biggest influence of investments implemented in current period is met. By using $F = 0.05, M = 6$ we got uninterpretable solution.

Numerical solver

The integration of delay differential equation with constant delay is expressed as follows [3]:

$$\begin{aligned}
&k_i = f(t_0 + a_1 h, x_0 + h \sum_{j=1}^{s} b_{ij} k_j, \phi(t_0 + a_i h)) \qquad \text{for} \quad i = 1, \ldots, s \\
&x_1 = x_0 + h \sum_{j=1}^{s} c_j k_j \\
&t_1 = t_0 + h
\end{aligned}$$

which is a notation similar to the ordinary case, with the difference in the member $\phi(t_0 + a_i h)$. Because the initial condition of u is a discretized function, some $\phi(t_0 + a_i h)$ values may be unknown. Therefore, they must be interpolated using known data, which is also the reason for a more complicated implementation and a new source of error. The adaptation of the RKCeM4 iterative scheme for a delayed case is as follows [4]:

$$x_{n+1} = x_n + \frac{h}{3}\left[\frac{2}{3}\sum_{i=1}^{3}\left(\frac{k_i^2 + k_i k_{i+1} + k_{i+1}^2}{k_i + k_{i+1}}\right)\right],$$
$$\text{where } k_i = f(x_n + a_i h, y_n + h\sum b_{ij}k_j, \phi(x_n + a_i h))$$

$$\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
\frac{1}{2} & \frac{1}{24} & \frac{11}{24} & 0 \\
1 & \frac{11}{132} & \frac{-25}{132} & \frac{73}{66} \\
\hline
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{array}$$

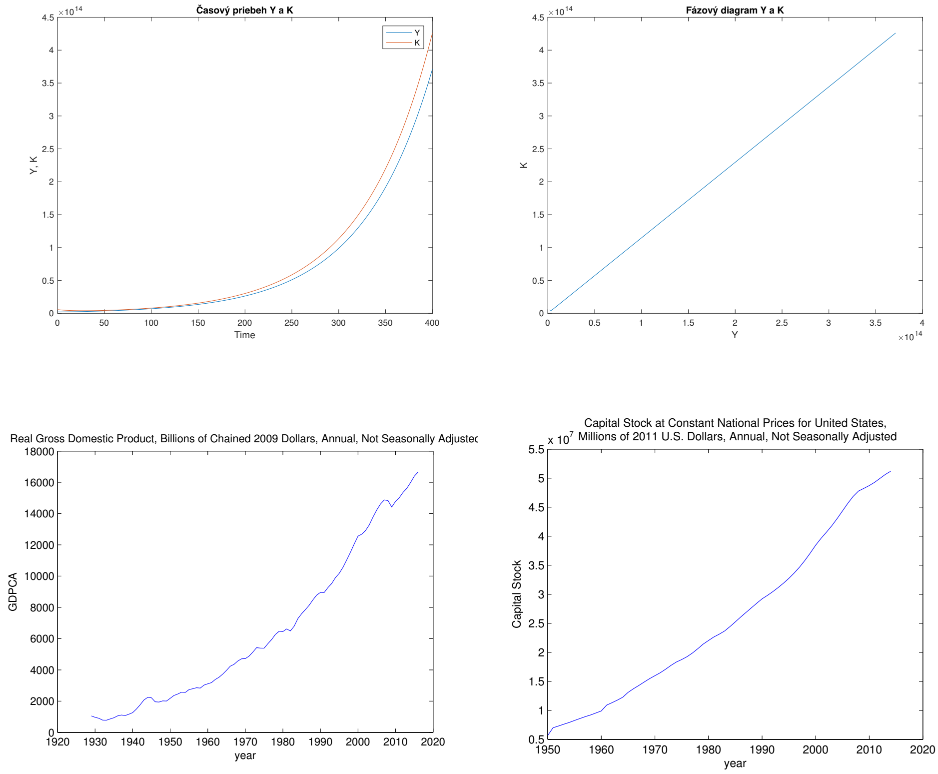
Hermite interpolation was used to approximate a delayed member. The order of interpolations must be adapted to the order of the method, because for arbitrary adaptation of the Runge-Kutta method for ordinary differential equations to delay differential, it holds a convergence order equal to the minimum of the order of the method and of the number of interpolation support points.

0.1 Application

By applying the numerical solver to the estimated model

$$\begin{aligned}
&\dot{Y}(t) = 0.0673(0.3000Y(t) - 0.0901K(t) - 0.0001Y(t)), \\
&\dot{K}(t) = (\sum_{j=0}^{2} c_j(0.3000Y(t-\theta_j) - 0.0901K(t-\theta_j))^{0.5037})^{\frac{1}{0.5037}} - 0.0500K(t),
\end{aligned} \tag{6}$$

where $c_0 = 0.3160$, $c_1 = 0.1390$, $c_2 = 0.1325$, $\tau_0 = 0$, $\tau_1 = 1$ and $\tau_2 = 2$, the result is by exponential shape visually related to historical data:



Graphical review of the solutions

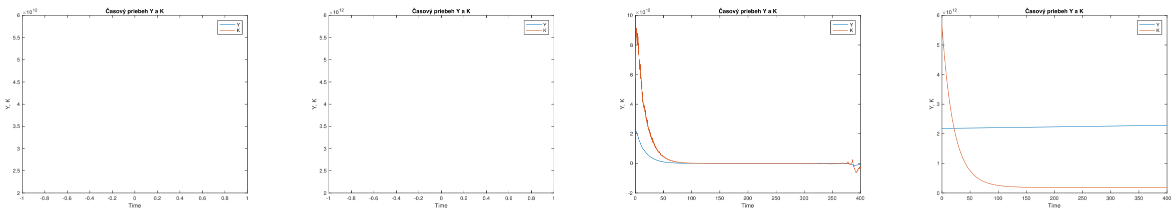


Figure 2: Simple Algorithm: setups $F = 2$, $F = 1$, $F = 0.5$ and $F = 0.25$.

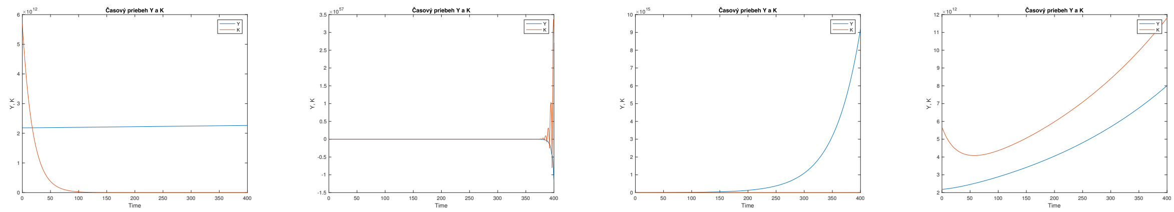


Figure 3: Simple Algorithm: setups $F = 0.1$, $F = 0.05$, $F = 0.001$, Modified Algorithm: setup $F = 0.1$, $M = 10$.

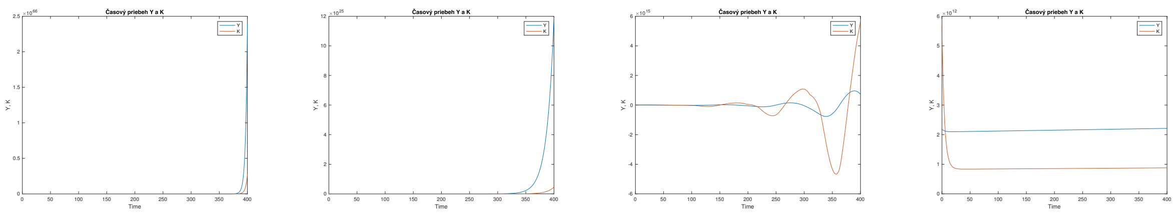


Figure 4: Modified Algorithm: setups $F = 0.1$, $M = 8$; $F = 0.1$, $M = 6$; $F = 0.1$, $M = 4$; $F = 0.1$, $M = 2$.

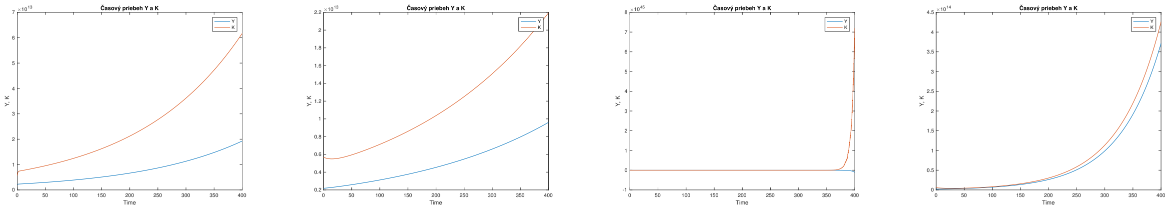


Figure 5: Modified Algorithm: setups $F = 0.05$, $M = 10$; $F = 0.05$, $M = 8$; $F = 0.05$, $M = 6$; $F = 0.05$, $M = 4$.

Performance on generally known test functions

Beside objective function for nonlinear regression of the Kaldor-Kalecki model with unknown minimum, generally known problems with known solution will be tested in this section. Specifically, Dejongl, Rastrigin and Grienwangk functions with global minimums in $f(x) = 0; x(i) = 0, i = 1 : n$ from [2]. Number of enumerations needed for adequate fitness (as close as $1e - 20$ to the solution) reached will be tested. Results for basic algorithm are listed in Table 5. Averages were obtained from 100 repetitions of calculation. The basic DE algorithm shows the best results with $F = 0.25$ for Dejong1 function and with $F = 0.5$ for Grienwangk function. For Rastrigin function no setup was sufficient.

Table 5: Basic DE: Average number of steps for the $1e - 20$ acceptance threshold value

	F	2	1	0.5	0.25	0.1	0.05	0.001
DEJONG1	avg steps	325.6200	173.8200	110.1000	87.4500	1473.1	19004	20000
RASTRIGIN		20000	20000	20000	20000	20000	20000	20000
GRIENWANGK		539.2400	236.1600	188.8900	380.1200	9071.6	18806	20000

Results for modified algorithm are listed in Table 6 and Table 7 . Unlike simple algorithm, modified one works for Rastrigin function, however, performance has worsened for both Dejong1 and Grienwangk functions. For Rastrigin function the modified DE algorithm shows the best results with setup $F = 0.1, M = 2$.

Table 6: Modified DE: Average number of steps for the $1e - 20$ acceptance threshold value

	M	10	8	6	4	2
	F	0.1				
DEJONG1	avg steps	16235	16016	16016	16193	16193
RASTRIGIN		20400	19600	18800	18000	17200
GRIENWANGK		17340	17278	17134	16914	17340

Table 7: Modified DE: Average number of steps for the $1e - 20$ acceptance threshold value

	M	10	8	6	4	2
	F	0.05				
DEJONG1	avg steps	19523	19421	18382	17702	17082
RASTRIGIN		20400	19600	18800	18000	17200
GRIENWANGK		19962	19600	18522	17604	17200

References

[1] STORN, R., PRICE, K. Differential Evolution A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization.* 1997, 11(4), 341359.

[2] Example Functions for Parametric Optimization [online]. [cit. 2018-05-10] http://www.geatbx.com/docu/fcn index-01.html#P89_3085

[3] ISMAIL, F., AL-KHASAWNEH, R. A., LWIN, A. S., SULEIMAN, M. *Numerical Treatment of Delay Differential Equations by Runge-Kutta Method Using Hermite Interpolation* [online]. [cit. 2018-04-06]dor http://www.matematika.utm.my/index.php/matematika/article/view/121

[4] PUSHPAM, E. K. A., SHAALINI, V. J. *Solving Delay Differential Equations with Constant Lags using RKCeM Method* [online]. [cit. 2018-04-06] https://www.ijirset.com/upload/2016/april/11_Solving.pdf

[5] SAVITZKY, A., GOLAY, M.J.E. (1964).Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36 (8): 162739. doi:10.1021/ac60214a047.

[6] SZYDLOWSKI , M. & KRAWIEC, A. (2001) The Kaldor-Kalecki Model of Business Cycle as a Two-Dimensional Dynamical System, *Journal of Nonlinear Mathematical Physics*, 8:sup1, 266-271, DOI: 10.2991/jnmp.2001.8.s.46

[7] WANG, L., WU, X., *Bifurcation Analysis of a Kaldor-Kalecki Model of Business Cycle with Time Delay* [online]. [cit. 2018-04-06]. https://digitalcommons.kennesaw.edu/cgi/viewcontent.cgi?article=2103&context=facpubs