Petro STAKHIV[1], Liliana BYCZKOWSKA-LIPIŃSKA[2], Yuriy KOZAK[3]

Technical University of Łodz (1), The College of Computer Science in Łodz (2), Lviv Polytechnic National University (3)

The Impact of Calculation Precision on the Process of Mathematical Model Construction with the Use of Optimization

Abstract. *The usage of calculation power of GPU cards for macromodel construction based on optimization approach can lead to significant decrease of model construction time. Unfortunately, most GPU cards do not work well with double-precision calculations. In the paper the comparison of optimization process conducted using single-precision and double-precision has been done. It is shown that the reduction of computation precision to single-precision values does not worsen the precision of obtained model and the number of required iterations of optimization algorithm.*

Streszczenie. *Wykorzystanie mocy obliczeniowej procesorów kart graficznych do budowy makromodeli matematycznych może prowadzić do skrócenia czasu obliczeń. Niestety większość procesorów wykorzystywanych jako GPU nie pracuje w podwójnej precyzji. W artykule wykazano, że redukcja dokładności obliczeń do pojedynczej precyzji nie pogarsza jakości otrzymanego modelu, ani nie zwiększa liczby iteracji algorytmu optymalizacyjnego.* ***(Wpływ dokładności obliczeń na proces tworzenia zoptymalizowanych modeli matematycznych)***

Keywords: *mathematical model, macromodel, optimization, calculation precision, parallelization*
Słowa kluczowe: *model matematyczny, optymalizacja, dokładność obliczeń, zrównoleglenie obliczeń*

Introduction

The complexity of dynamical systems being developed and analyzed is constantly increasing. Currently such systems include elements of different nature which are described by models of different types. Another problem is the complexity of mathematical models of some components included in the system being designed. All these issues result in the complexity of simulation task in terms of required computation resources.

The use of macromodels in such conditions allows to replace some components or entire subsystems with the relatively simple mathematical models, and thus to significantly decrease required computation resources needed for the analysis. So for efficient usage of macromodels during the analysis of complex systems it is necessary to develop universal approaches to macromodel construction.

Macromodel construction with the use of optimization

Most macromodel construction techniques can not be considered as universal approaches because they apply many restrictions on the object and required input information which is not always available. An alternative approach, which does not impose such constraints, is the use of optimization.

Values of macromodel coefficients in this approach are found by minimization of some goal function $Q(\boldsymbol{\lambda})$. This function represents the deviation of the object behavior calculated using the model being constructed and experimental data as a function of model coefficients. The most commonly used expression for goal function calculation is a standard deviation:

$$Q(\boldsymbol{\lambda}) = \frac{1}{T}\int_0^T \left|\mathbf{y}(t) - \tilde{\mathbf{y}}(t)\right|^2 dt \,, \tag{1}$$

where $\mathbf{y}(t)$ is a vector of experimentally measured object output values; $\tilde{\mathbf{y}}(t)$ is a vector of object output values calculated using the model being constructed.

In case of discrete models, which are more convenient for computer simulations, this expression has the next form:

$$Q(\boldsymbol{\lambda}) = \frac{1}{N}\sum_{k=0}^{N-1} \left|\mathbf{y}_k - \tilde{\mathbf{y}}_k\right|^2 \tag{2}$$

where k is a discrete index.

By finding the point where mentioned function reaches its minimum we will find an optimal set of the model coefficients within the selected mathematical form of the model representation.

Described approach can be used for the construction of macromodels in any mathematical form, which is represented by a limited set of coefficients. Also it does not apply any restrictions on required input information except the obvious requirement to describe the object fully enough. Additionally, the utilization of optimization eliminates the calculation problems related to the fact that mathematical model identification is often an ill-conditioned problem.

The main problem which precludes the wide usage of an optimization approach is the complexity of the optimization task which requires significant computation resources.

One possible way to solve this problem is the usage of parallelization.

Optimization task parallelization using GPU cards

With the increase in computing power of GPU cards during last years there is a tendency to use their computational capabilities to solve tasks not related to graphics. This allows us to use the computational power of GPU cards to solve the optimization task for macromodel construction. There are already some papers where such efforts to use computational power of GPU cards at practice was described [1].

One important question here is the fact that GPU cards are mostly oriented on carrying out of calculations with single-precision values (7-8 significant decimal digits), while most implementations of optimization algorithms work with double-precision values (~16 significant decimal digits), or even values in extended format (~19 significant decimal digits).

The impact of the limited calculation precision on the process of macromodel construction on an example of Rastrigin's direct cone method has been analyzed in this paper.

The impact of the calculation precision on the process of model construction is important not only for GPU cards usage. Values of smaller precision use less space in the computer RAM. Therefore, we can get a difference in algorithm performance even in case when the time of mathematical operations execution would not depend on

the precision of the values being used. Generally such difference is a result of different amounts of data, which need to be transferred between CPU and RAM. This dependency exists at almost any computational system, including regular PCs and GPU cards.

In case of GPU cards there is one more not obvious advantage of single-precision values. It is caused by the fact that architecture of GPU cards is generally oriented to work with 4-bytes data elements. And the size of single-precision value is also equal to 4 bytes.

In general, there are many reasons owing to which the computation using single-precision values is performed quicker. But to make an optimal decision what precision should be used we have to consider the accuracy of the obtained result.

It is expected that smaller precision of calculations should result in the decrease of the accuracy of the obtained result. But in case of mathematical models construction we are interested not only in the accuracy of the result, but also in the number of iterations of the optimization algorithm needed to get the result. Small deviation of the solution from the optimal one is not generally a problem, and in the worst case can be fixed by execution of few more iterations of the algorithm using better precision of calculations. We should also mention here, that small deviation of model coefficients from the optimal solution does not necessarily imply a worse model.

Completely different situation is with the dynamics of the optimization process. We can state that the optimization task in case of mathematical models construction has generally such specific characteristics:

- high dimension of the parameters space which is equal to the number of coefficients in the constructed model;
- "ravine" type of the goal function which is caused by significant difference of the dependency level of the model precision on the different model coefficients [2];
- a large number of small local minima caused by the rounding errors. It is obvious that this characteristic significantly depends on the used calculation precision, so will be considered in more details below.

The mentioned characteristics are the reason of why stochastic algorithms are preferably used to solve the optimization task for model construction. And though such algorithms are not very sensitive to the local minima, the difference in optimization process dynamics may significantly depend on them.

Theoretical analysis of the possible impact of the calculation precision on the characteristics of the goal function allows to state that smaller calculation precision should result in deeper local minima but at the same time in the reduction of their total number.

It is also important that because of significant difference of the dependency of the goal function on the different model coefficients local minima generally are of the same "ravine" type as the global one. And together with high dimension of the parameters space we can expect a vast variety of possible situations and corresponding dynamics of the optimization task execution flow. So, in total, theoretical analysis does not provide any result except that the possible influence of calculation precision reduction can have both negative and positive impact on the dynamics of optimization process.

To verify the impact of the reduced calculation precision at practice authors performed construction of the model of single-phase asynchronous motor with starting capacitor in instantaneous values many times using both single and double precision of calculations.

Base information for mathematical model construction has been taken from experimental measurements of current consumption and rotor speed during motor start. These characteristics are shown in Fig. 1. The oscillograms were obtained using AD/DA converter 'ADA-1406'. The sampling frequency is of 2 kHz.

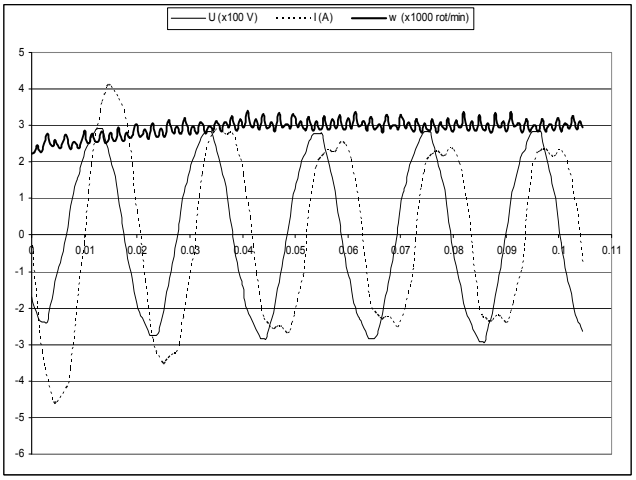

Fig.1 Voltage, current, and rotor speed during start of the single-phase asynchronous motor with starting capacitor

The instantaneous value of the voltage u applied to the motor has been selected as an input variable for the model of the single-phase asynchronous motor with starting capacitor in instantaneous values. The instantaneous value of current i and rotor speed ω have been selected as output variables. Thus vectors of input and output variables have the next form:

$$\mathbf{v}^{(k)} = \left(u^{(k)}\right),\ \mathbf{y}^{(k)} = \begin{pmatrix} i^{(k)} \\ \omega^{(k)} \end{pmatrix} \tag{3}$$

The model has been constructed in the discrete form of state variables:

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{F}\mathbf{x}^{(k)} + \mathbf{G}\mathbf{v}^{(k)} + \mathbf{\Phi}\left(\mathbf{x}^{(k)}, \mathbf{v}^{(k)}\right) \\ \mathbf{y}^{(k+1)} = \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{D}\mathbf{v}^{(k+1)} \end{cases} \tag{4}$$

where $\mathbf{v}$ is a vector of input values; $\mathbf{y}$ is a vector of output values; $\mathbf{x}$ is a vector describing the internal state of the object; $\mathbf{F}$, $\mathbf{G}$, $\mathbf{C}$, $\mathbf{D}$ are matrices of the model coefficients; $\mathbf{\Phi}$ is some nonlinear vector-function of many arguments; k is a discrete index.

The used optimization algorithm is based on the Rastrigin's direct cone method with adaptation procedures for step size and cone opening angle [5].

To obtain a statistically confidante results in both cases (for single and double precision of calculations) the process has been repeated 10 times. Fig. 1 shows the dependency of the value of the goal function from the number of executed iterations for every try.

We can see from Fig. 2 that there is no significant influence of computation precision on the dynamics of optimization process. The difference between tries caused by the randomness, which is typical for stochastic optimization algorithms, is much more significant than the difference caused by the computation precision. This can be more clearly seen if we do an averaging (geometrical mean is the most correct here). The result of such an averaging is shown in Fig. 3.

As we can see, except some fluctuations in the middle of optimization process, the number of iterations needed to find the appropriate solution in our case does not depend on the precision of calculations.

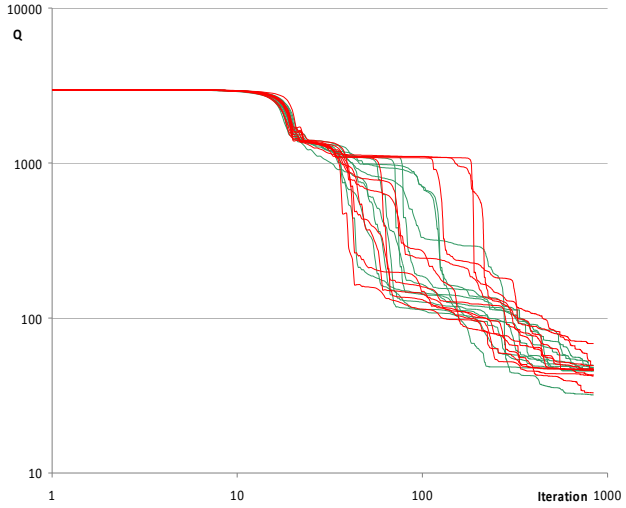

Fig. 2. Dependency of the value of the goal function on the number of executed iterations. Green lines – double precision, red lines – single precision.

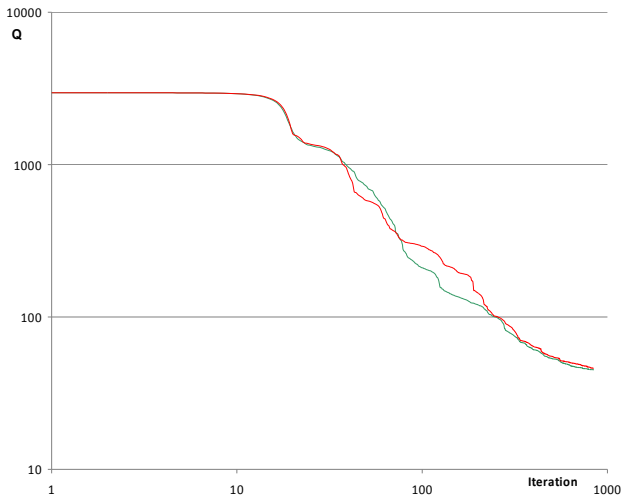

Fig. 3. Average dependency of the value of the goal function for the number of executed iterations. Green lines – double precision, red lines – single precision.

If we compare the difference between the lines in Fig. 3 with the dispersion between the lines in Fig. 2 we can clearly state that the difference in average values shown in Fig. 3 is most likely accidental and caused by the randomness of stochastic optimization algorithms.

One more point, which can be mentioned, is that the dispersion of the red lines in Fig. 2, which corresponds to the single precision, is wider than the dispersion of the red lines, which corresponds to the double precision. Thus single precision calculations have a bigger chance to come to the optimal solutions quicker as well as a bigger chance to get stuck near some not optimal solution. This effect can be explained by the fact that in case of less precise calculations local minima are deeper but their number is considerably less than in case of more precise calculations.

In general in experimental verification of the impact of calculation precision on the process of mathematical model construction with the use of optimization authors have not been able to detect any significant difference between single and double precision of calculations. Extreme cases where the step of optimization algorithm would be of the same magnitude as the calculation precision have not been tested because such situations can be easily eliminated by scaling the parameters space, which is a common practice for mathematical model construction using optimization.

Such result allows us to expect that switching to single-precision calculations is not going to cause any significant problems with the quality of obtained models. So it is recommended to use single-precision calculations for mathematical models construction using optimization in case if reduction of calculation precision would result in significant performance increase.

Conclusions

Experimental verification shows that the reduction of computation precision from double (~16 significant decimal digits) to single (7-8 significant decimal digits) values does not worsen the precision of the obtained model and the number of iterations of optimization algorithm, required for its construction. Thus it is desirable to use single-precision calculations for mathematical models construction using optimization if this allows one to increase the performance.

REFERENCES

[1] Byczkowska-Lipińska L., Stakhiv P., Strubytska I., Kozak Yu. Parallelization of calculations using GPU in optimization approach for macromodels construction, *Proc. of International Workshop CPEE'2011 "Computational Problems of Electrical Engineering",* Kostryna, Zakarpattia region, (Ukraine) September 5-7, 2011, P. 30.

[2] Kozak Yu., Melnyk B., Stakhiv P. Simplification of Optimization Process during Mathematical Models Creation, *Przegląd Elektrotechniczny*, Nr. 12/2008, 281-283.

[3] Kalman R., Falb P., Arbib M. Topics in Mathematical System Theory, McGraw Hill, New. York, 1969.

[4] Stachiw P., Kozak J., Melnyk B. Zastosowanie równoległych algorytmów w syntezie makromodelów wielobiegunników dynamicznych *Pracy XXVI Międzynarodowa Konferencja z Podstaw Elektrotechniki i Teorii Obwodów IC-SPETO 2003,* Gliwice-Niezica, Polska, 28-31.05.2003, Tom II, str. 263-266.

[5] Stakhiv P., Kozak Yu. Design of discrete models of complex dynamic systems based on optimization approach, *Proceedings of the 11th International Conference on "System modeling Control",* Zakopane, Poland, October 17-19, 2005, pp. 297-301.

***Authors**:*
Prof. Petro Stakhiv, Institute of Computer Sciences, Technical University of Łódz, ul. Wolczanska 215, Łódz, 90-924, Poland, spg@ics.p.lodz.pl,
Prof. Liliana Byczkowska-Lipińska, The College fo Computer Science in Lodz, lilip@ics.p.lodz.pl,
Assoc. Prof. Yuriy Kozak - Department of Theoretical Electrical Engineering, Institute of Electric Power and Control Systems, Lviv Polytechnic National University, 12 S. Bandera str, Lviv, 79013, Ukraine, ykozak@mail.ru