V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal, 14–17 June 2010

USING GRID COMPUTING TO MODEL BIOSENSORS ACTING IN STIRRED AND NON-STIRRED SOLUTIONS

V. Ašeris[*], R. Baronas[*,†]

[*]Department of Software Engineering, Vilnius University
Naugarduko 24, LT-03225 Vilnius, Lithuania
e-mail: Vytautas.Aseris@mif.vu.lt

[†]Institute of Mathematics and Informatics
Akademijos 4, LT-08663 Vilnius, Lithuania
e-mail: Romas.Baronas@mif.vu.lt

Key words: Grid computing, Computer modeling, Simulation, Biosensor, Reaction-diffusion

Abstract. *In this paper the response of the biosensor based on a chemically modified electrode is modeled numerically by using grid computing. Computer models intended to investigate characteristics of biosensors are commonly parameterized. In most cases the same problem is to be manifold solved at different sets of parameter values. Running multiple simulations is a time-consuming task, which can be accelerated by using the grid computing.*

Biochemical behavior of the biosensor based on a chemically modified electrode[1] was investigated. The mathematical model of the biosensor is based on a system of non-linear reaction-diffusion equations. The Crank–Nicolson finite difference method has been used when approximating the model[2]. The biosensor action was modeled in stirred and non-stirred solutions.

In this paper the input, output and operational requirements are defined for computer models to be executed in computational grids. A technique assuring an efficient usage of the grids to investigate the peculiarities of the biosensor response is defined. The introduced technique of computational grid usage was applied to perform computer simulation to investigate the peculiarities of the above-mentioned biosensor.

An efficiency of the developed technique was investigated by comparing the time of calculation in the BalticGrid environment with the corresponding time in a local computer. The technique dependency on the size of the whole computational task and the quantity of subtasks distributed among different computing nodes in the grid was investigated. In the investigation there was noticeable delay when simulating biosensors response in the computational grid. This delay is caused by an approximately constant time for the grid middleware to process all tasks in the grid environment and the time of random tasks delayed in some computational nodes.

1 INTRODUCTION

Grid computing is rapidly developing technology. A cluster is a group of linked computers which in many aspects can be viewed as a single computer. Computational grids are formed by connecting independent clusters. The computational grids are used in a wide variety of scientific researches – molecular physics, thermodynamics[3], cosmology[4] and other areas. In this paper the grid computing is used to model the behavior of a chemically modified biosensor.

Biosensors are sensing devices that transform a biological recognition into an electrical signal. During the biosensor operation the substrate to be analyzed is biochemically converted to a product. The signal is usually proportional to the concentration of the reaction product[5]. Biosensors are widely used to detect very accurate chemical compounds and to define them[6]. The biochemical behavior of the biosensor can be described by a mathematical model. A computer simulation is the way to solve the mathematical model in a general case. It is crucial to know the influence of parameters for biosensors response when creating new biosensors or modifying existing ones. Large sets of parameter values are used when modeling the biocatalysis process in a specific biosensor. The modeling tasks can be solved independently and parallelly when the same computer model is used with many different sets of parameters values. A usage of computational grids is a perspective way to accelerate the investigation of biosensors peculiarities[7].

The computational grid is a system that coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service[8]. The computer software designed to solve problems of parameterized computer modeling already exists[9,10,11]. Specialized software highly depends on the middleware installed in a specific computational grid. The purpose of this work was to develop a technique assuring an efficient usage of the computational grids. This technique was primarily designed to investigate the peculiarities of the biosensor response without attaching to specific middleware. The efficiency of the developed technique was investigated in *BalticGrid* computational grid. Developed technique was applied to compare the detail model of the chemically modified biosensor and its quasi-steady state approximation.

2 TECHNIQUE

Computer modeling software has to meet specific requirements in order to ensure efficient and convenient usage of grid computing.

2.1 Requirements for computer models

The most important requirements for passing parameters and the environment of software are the following ones:

- Data input and output. The ability to change model's parameters without changing model itself is crucial. The most efficient way to assure this requirement is to develop the software which would read modeling data from a file defined in models parameters. Data input and output must be independent of the operating system.
- Operation of the software. Computer models have to operate in command line mode as in the computational grids they are executed automatically. Occurring

mistakes should be logged to the error log file because it the only way to assure that no information is lost during the process.

2.2 Grid computing usage scheme

Computer models have many parameters in most cases. The most common and important task in parameterized simulations is to define the dependency between the initial data and the results of computer simulation[12]. Specific characteristic's dependency on the initial parameters during simulation process can be described as follows:

$$y = f(p_1, p_2, ..., p_N), \tag{1}$$

where y stands for the above mentioned specific characteristic, N – the number of parameters, $Q = \{p_1, p_2, ..., p_n\}$ – the parameter queue where values $p_1, p_2, ..., p_n$ affects the investigated characteristic y, f denotes the simulation process.

In many cases of the comprehensive analysis the simulations are carried out with different sets of parameters called the parameter sweeps[13]. The parameter sweep is a collection of parameter queues. It can be created by enumerating particular values of parameters or can be generated. When generating parameter queues the following values must be specified[14]:

- The interval of parameter values:

$$p_A \le p_i \le p_B, \tag{2}$$

  where p_A stands for the starting value of the concrete parameter, p_B – last value, and p_i – any parameter from the parameter queue, $i = 1, 2, ..., N$.

- The progression type to define whether arithmetic or geometric progression is used to generate the parameter.

Computer models designed to investigate the peculiarities of biosensors are parameterized as well. To perform the parameterized simulation in the computational grid a group G of total N_G tasks is submitted to the grid:

$$G = \{M, PS_i\}, \tag{3}$$

where G stands for a group of tasks, M is the computational model, PS_i – parameter sweeps where every parameter queue consists of parameters $p_1, p_2, ..., p_N$ and $i = 1, 2, \ldots, N_G$.

Utilization of grid computing to solve parameterized tasks can be defined by the following steps:

- The rules to form parameter sweep are defined. The parameter sweep can be enumerated by hands, generated using progression or both ways can be used at the same time. In such case, a part of the parameter queues are generated and part of them are enumerated manually.
- The main parameter sweep is prepared according to the defined rules.
- Produced parameter sweep is devided into N_G amount of smaller sweeps. It is recommended to seek that smaller sweeps would be simmilar in size.
- The group of tasks (3) is submitted to the computational grid.
- The results are retrieved after the simulation is finished.

2.3 Implementation of the technique

It is commonly required that some parameter values would be enumerated, some to be generated and some to remain constant in all parameter queues. In order to meet this requirement we have developed a method for the parameter sweep generation. Directions for the generation of parameters are provided in the parameter description file in the following syntax:

$$\begin{array}{l} M_{PQ}; \\ \left.\begin{array}{l} Type; Guidelines; \\ \ldots \\ Type; Guidelines; \end{array}\right\} N \end{array},$$

where M_{PQ} stands for the number of the desired parameter queues in the main parameter sweep (which also equals to the total number of simulations), N is the total number of parameters, *Type* defines the type of parameter generation and *Guidelines* defines an additional information needed for the current parameter. Values allowed for *Type* and *Guidelines* are defined in Table 1.

Type	*Type* description	*Guidelines*	*Guidelines* description
0	Parameter is not changed during the simulations.	*ConcreteValue*	A single value with semicolon at the end.
1	Parameter is changed by enumerating its values.	*ListOfValues*	The enumerated list of parameter values separated by semicolons. If *ListOfValues* count is less than M_{PQ}, then the last value of the given list is used for the remaining parameter queues, if *ListOfValues* count is more than M_{PQ}, then the remaining values are ignored.
2	Parameter is generated using arithmetic progression.	*StartingValue; EndValue*	*StartingValue* stands for the first value in desired interval of parameter's values; *EndValue* is the final value (number M_{PQ}) in mentioned interval.
3	Parameter is generated using geometric progression.	*StartingValue*; *EndValue*;	*StartingValue* stands for the first value in desired interval of parameter's values; *EndValue* is the final value (number M_{PQ}) in mentioned interval.

Table 1. Parameter values defined in the description file.

A computer program was developed to generate the main parameter sweep using parameter description file.

Entire technique of computational grid usage was implemented as follows:

- The main parameter sweep is generated using the above mentioned syntax.
- The main parameter sweep of M_{PQ} parameter queues is divided into N_G smaller parameter sweeps. An average amount of parameter queues in smaller parameter sweep is found:

$$N_{avg} = M_{PQ} / N_G. \quad (4)$$

Let $N_{avg,int}$ be the integer part of the average N_{avg}. Then a total number of A parameter sweeps will have $N_{avg,int}$ parameter queues. A total number of B parameter sweeps will have $N_{avg,int}$ + 1 parameter queues. A and B stands for integer numbers satisfying the following equation:

$$M_{PQ} = AN_{avg,int} + B(N_{avg,int} + 1). \tag{5}$$

In accordance with the equation (5) we assure that every smaller parameter sweep would be similar in the size.

- The group of N_G tasks (3) is submitted to the computational grid.
- The modeling software writes results to the output file independently in its own computing node.
- The output files from different computing nodes are collected and joined after the separate modeling tasks are completed.

3 MODELING OF BIOSENSORS USING DEVELOPED TECHNIQUE

The developed technique was applied to model biosensors behavior. Three materials are present – enzyme (E), substrate (S) and mediator (M) in the biocatalysis process of the studied biosensor.

$$\mathrm{E_{ox}} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{ES} \overset{k_2}{\longrightarrow} \mathrm{E_{red}} + \mathrm{P_1}, \tag{6}$$

$$\mathrm{E_{red}} + \mathrm{M} \overset{k_3}{\longrightarrow} \mathrm{E_{ox}} + \mathrm{P}. \tag{7}$$

The model involves three regions: the enzyme layer where the biochemical reactions (6) and (7) as well as the mass transport by diffusion takes place, a diffusion limiting region where only the mass transport by diffusion takes place and a convective region where the substrate concentration is maintained constant.

A mathematical model of a generic amperometric biosensor based on a chemically modified electrode (CME) has been proposed[1]. The biosensor response was numerically modeled under the quasi-steady state assumption[15]. The goal of this investigation was to develop a detail mathematical and the corresponding numerical models of the biosensor based on the CME, where no quasi-steady state assumption is applied, and to evaluate the conditions at which the biosensor action can be simulated under quasi-steady state approximation (QSSA) for an accurate prediction of the biosensor response.

3.1 Mathematical model of the biosensor

Coupling the enzyme catalysed reactions (1) and (2) in the enzyme layer with the one-dimensional-in-space diffusion, described by Fick's law, leads to the following equations[1] of the reaction–diffusion type ($t > 0$, $0 < x < d_e$):

$$\frac{\partial s_e}{\partial t} = D_{Se}\frac{\partial^2 s_e}{\partial x^2} + k_{-1}e_S - k_1 e_{ox} s_e, \tag{8a}$$

$$\frac{\partial m_e}{\partial t} = D_{Me}\frac{\partial^2 m_e}{\partial x^2} - k_3 e_{red} m_e, \tag{8b}$$

$$\frac{\partial p_e}{\partial t} = D_{Pe}\frac{\partial^2 p_e}{\partial x^2} + k_3 e_{red} m_e, \tag{8c}$$

$$\frac{\partial e_{ox}}{\partial t} = k_3 e_{red} m_e + k_{-1} e_S - k_1 e_{ox} s_e , \tag{8d}$$

$$\frac{\partial e_{red}}{\partial t} = k_2 e_S - k_3 e_{red} m_e , \tag{8e}$$

$$\frac{\partial e_S}{\partial t} = k_1 e_{ox} s_e - k_{-1} e_S - k_2 e_S , \tag{8f}$$

where x and t stand for space and time, respectively, $s_e(x, t)$, $m_e(x, t)$, $p_e(x, t)$, $e_S(x, t)$, $e_{red}(x, t)$ and $e_{ox}(x, t)$ are the molar concentrations of the substrate S, the mediator M, the product P, the oxidized enzyme E_{ox}, the reduced enzyme E_{red} and the enzyme substrate ES, respectively, d_e is the thickness of the enzyme layer, and D_{Se}, D_{Me}, D_{Pe} are the diffusion coefficients.

Outside the enzyme layer only the mass transport by diffusion of the substrate, the mediator and the product takes place. We assume that the external mass transport obeys a finite diffusion regime[1] ($t > 0$, $d_e < x < d_e + d_d$):

$$\frac{\partial s_d}{\partial t} = D_{Sd} \frac{\partial^2 s_d}{\partial x^2}, \ \frac{\partial m_d}{\partial t} = D_{Md} \frac{\partial^2 m_d}{\partial x^2}, \ \frac{\partial p_d}{\partial t} = D_{Pd} \frac{\partial^2 p_d}{\partial x^2}, \tag{9}$$

where $s_d(x, t)$, $m_d(x, t)$ and $p_d(x, t)$ stand for the molar concentrations of the substrate, the mediator and the product in the diffusion layer, d_d is the thickness of the external diffusion layer, D_{Sd}, D_{Md} and D_{Pd} are the diffusion coefficients.

The biosensor operation starts when some substrate appears in the bulk solution ($t = 0$),

$$s_e(x,0) = p_e(x,0) = 0, \quad 0 \le x \le d_e, \tag{10a}$$

$$m_e(x,0) = \begin{cases} m_0, & x = 0, \\ 0, & 0 < x \le d_e, \end{cases} \tag{10b}$$

$$e_{ox}(x,0) = e_0, \ e_{red}(x,0) = e_S(x,0) = 0, \quad 0 < x < d_e, \tag{10c}$$

$$s_d(x,0) = \begin{cases} 0, & d_e \le x < d_e + d_d, \\ s_0, & x = d_e, \end{cases} \tag{10d}$$

$$m_d(x,0) = p_d(x,0) = 0, \quad d_e \le x \le d_e + d_d, \tag{10e}$$

where e_0 is the concentration of the enzyme in the enzyme membrane, m_0 is the concentration of the mediator at the boundary between the electrode and the enzyme layers, s_0 is the concentration of the substrate in the bulk solution.

The boundary and matching conditions are defined as follows[1] ($t > 0$):

$$s_d(d_e + d_d, 0) = s_0, \quad m_d(d_e + d_d, 0) = p_d(d_e + d_d, 0) = 0 , \tag{11a}$$

$$D_{Se} \left. \frac{\partial s_e}{\partial x} \right|_{x=0} = 0, \ m_e(0,t) = m_0, \ p_e(0,t) = 0 , \tag{11b}$$

$$D_{Se} \left. \frac{\partial s_e}{\partial x} \right|_{x=d_e} = D_{Sd} \left. \frac{\partial s_d}{\partial x} \right|_{x=d_e}, \ D_{Me} \left. \frac{\partial m_e}{\partial x} \right|_{x=d_e} = D_{Md} \left. \frac{\partial m_d}{\partial x} \right|_{x=d_e}, \ D_{Pe} \left. \frac{\partial p_e}{\partial x} \right|_{x=d_e} = D_{Pd} \left. \frac{\partial p_d}{\partial x} \right|_{x=d_e}, \tag{11c}$$

$$s_e(d_e,t) = s_d(d_e,t), \ m_e(d_e,t) = m_d(d_e,t), \ p_e(d_e,t) = p_d(d_e,t) . \tag{11d}$$

The anodic or cathodic current is measured as a response of an amerometric biosensor in a physical experiment. The density i of the current at time t is

$$i(t) = n_e F D_{Pe} \left. \frac{\partial p_e}{\partial x} \right|_{x=0}, \tag{12}$$

where n_e is a number of electrons involved in the electrochemical reaction, and F is Faraday's constant ($F = 96486$ C/mol)[16].

In order to define the main governing parameters of the model the corresponding dimensionless mathematical models was derived for the detail and the quasi-state assumption-based mathematical models.

The digital simulation was carried out using the finite difference technique[17]. The software was developed by implementing the detail and the quasi-state based mathematical models. C++ programming language was chosen as it is the most commonly supported language in computational grids.

3.2 The effect of the thickness of the diffusion layer

We normalize the steady-state current[2] because of high sensitivity of the maximal biosensor current to the thickness of the enzyme layer:

$$I_N(Bi) = \frac{I(Bi)}{I(\infty)}, \quad Bi = \frac{d_e / D_{Se}}{d_d / D_{Sd}} = \frac{D_{Sd} d_e}{D_{Se} d_d}, \tag{13}$$

where Bi is the Biot number expressing the ratio of the internal mass transfer resistance to the external one, $I(Bi)$ is the steady-state current of the detail mathematical model, calculated at given Biot number Bi. $I(\infty)$ corresponds to the biosensor response of the detail mathematical model for zero thickness of the external diffusion layer, $d_d = 0$.

The thickness d_d of the diffusion layer is inversely proportional to the intensity of stirring. The more intense stirring corresponds to the thinner diffusion layer. This diffusion layer, also known as the Nerst layer, practically does not depend upon the enzyme membrane thickness d_e. The Nerst layer may be practically minimized up to $d_d = 2\mu$m, but no zero thickness can be achieved[18]. The influence of two diffusion modules (Damköhler numbers), σ_{ox} and σ_{red},

$$\sigma_{red}^2 = k_{red} Q = \frac{k_1 k_2}{k_{-1} + k_2} \frac{E_0 d^2}{D_S}, \quad \sigma_{ox}^2 = k_3 Q = k_3 \frac{E_0 d^2}{D_S}, \tag{14}$$

were investigated.

The normalized steady-state current versus the Biot number Bi was calculated at the following six values of the thickness d_e of the enzyme layer: 17.3, 24.4, 34.6, 48.9, 69.2, 97.9 μm and six values of the enzyme concentration e_0 in the enzyme membrane: 10, 50, 250 nM, 1.25, 6.25, 3.125 μM. At each value of d_e as well as e_0 the Biot number Bi was changed from 0.1 up to 100. At these values of d_e and e_0 keeping other parameters unchanged the diffusion modules (σ_{ox} and σ_{red}) change in five orders of magnitude. Thus, the behaviour of the biosensor response was calculated at different limitations of the response. The following values of the model parameters were constant in the numerical simulation:

$$k_{-1} = 10^4\,\mathrm{s}^{-1},\ k_1 = 1.1\times 10^5\,\mathrm{M}^{-1}\mathrm{s}^{-1},\ k_2 = 10^3\,\mathrm{s}^{-1},\ k_3 = 10^7\,\mathrm{M}^{-1}\mathrm{s}^{-1},$$

$$m_0 = 10\,\mu\mathrm{M},\quad s_0 = 100\,\mu\mathrm{M},\quad n_e = 1, \tag{15}$$

$$D_{Se} = D_{Me} = D_{Pe} = 300\,\mu m^2 s^{-1},\ D_{Sd} = 2D_{Se},\ D_{Md} = 2D_{Me},\ D_{Pd} = 2D_{Pe}.$$

The results are displayed in Figure 1.

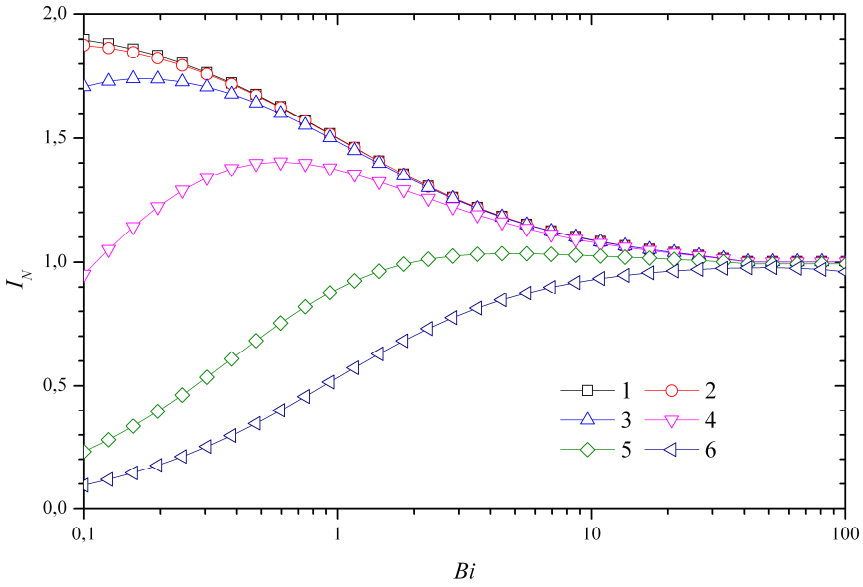

Figure 1. The normalized steady-state current I_N versus the Biot number Bi at six values of the diffusion module σ^2_{red}: 10^{-4} (1), 10^{-3} (2), 10^{-2} (3), 0.1 (4), 1 (5), 10 (6), $\sigma^2_{ox} = 1000\,\sigma^2_{red}$.

One can see that the steady-state biosensor current I_N is a monotonous increasing function of the Biot number Bi when $\sigma^2_{red} \geq 1$. I_N is non-monotonous when $\sigma^2_{red} \leq 1$. The normalized steady-state current only slightly varies for different diffusion modules when $Bi > 10$.

3.3 The accuracy of the quasi-state approximation

Using numerical simulation based on the detail mathematical model (8)-(11), the peculiarities of the biosensor action were investigated at different values of the model parameters. The simulation results were compared with the results obtained by using the corresponding model simplified by the QSSA[1].

We introduced the relative error E_{QSSA} of the biosensor response arose due to the QSSA,

$$E_{QSSA} = \left| \int_0^\infty i_{Det}(t)dt - \int_0^\infty i_{QSSA}(t)dt \right| \Big/ \int_0^\infty i_{Det}(t)dt\,, \qquad (16)$$

where i_{Det} is the density of the biosensor's current calculated by the detailed mathematical model of the CME, and i_{QSSA} is the density of the biosensor's current calculated by the corresponding model derived from the detailed model by applying QSSA. The error E_{QSSA} can also be called the relative error of QSSA.

The dependence of error E_{QSSA} on the dimensionless Biot number Bi is shown in Figure 2. The responses were calculated by changing the values of d_e and e_0 in the same way as in Figure 1. There can be seen an effect of the dimensionless Biot number Bi on the modeling error E_{QSSA}. The larger Biot number corresponds to the larger error of QSSA calculations. The error of the QSSA may be neglected with all the values of diffusion module except the case of small diffusion module for relatively large Biot

number (curve 1). When $\sigma_{red}^2 = 10^{-4}$ and $Bi > 30$ the error of QSSA calculations is bigger than 1% with even bigger E_{QSSA} for larger Biot numbers.

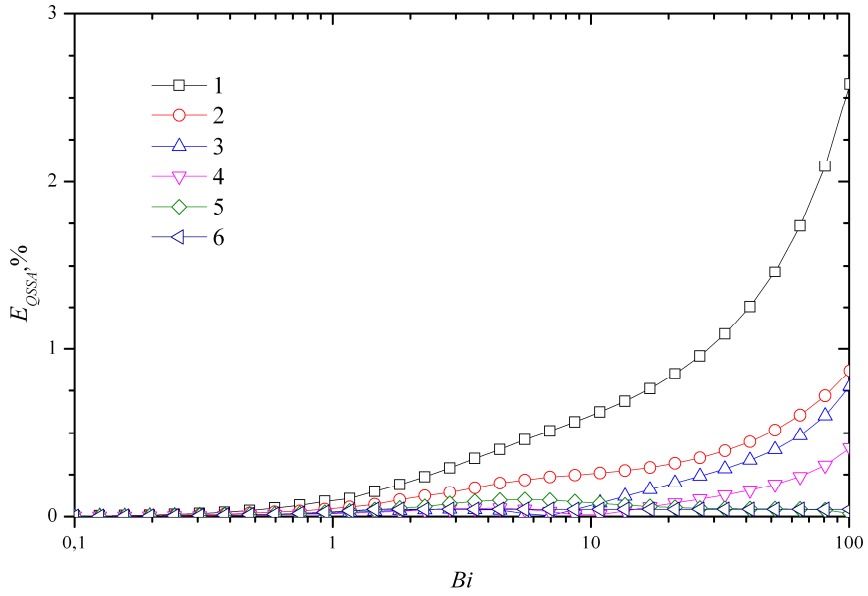

Figure 2. The error E_{QSSA} versus the dimensionless Biot number Bi at six values of the diffusion module σ_{red}^2: 10^{-4} (1), 10^{-3} (2), 10^{-2} (3), 0.1 (4), 1 (5), 10 (6), $\sigma_{ox}^2 = 1000\,\sigma_{red}^2$.

The parameters affecting the error of the quasi-state-based mathematical model were calculated by applying the developed technique. The same problem with different sets of parameter-values was solved more than 1000 times by comparing two models in the computational grid. This analysis was solved more than 5 times faster by using the computational grid than it would be solved on a local computer. An adaptation of the grid computing to the modeling of biosensors is especially perspective in the case of two-dimensional and three-dimensional models.

4 TECHNIQUE EFFICIENCY

By applying the developed technique to model chemically modified biosensors, the computing time required to complete specific simulation locally was compared with the time consumed in the *BalticGrid*. The recommendations for an efficient and convenient usage of the grid computing were developed.

4.1 Efficiency of computational Grid usage

The difference between simulations efficiency in a local computer and in a grid environment was evaluated during the performed investigation. Local simulations were carried out on a computer with the processor of 2 GHz speed and a memory of 1 GB of RAM. The acceleration R of simulations in the computational grid is described as follows:

$$R = (M_{PQ}\, T_S - T) / T \times 100\ \%, \tag{17}$$

where T_S stands for duration of simulation with single parameter queue in local computer, T is the duration of calculations with all parameter queues in the computational grid. T was measured as time between submission of groups of tasks G and the receiving of the results.

Comparisons were carried out by changing the amount of total tasks N_G submitted to the grid between 2 and 15 as well as with different durations M_{PQ} T_S = 30, 40, 60, 120, 240 and 480 minutes. Accelerations R dependency on N_G is displayed in Figures 3 and 4. The efficiency of simulations in the computational grid was measured by submitting the same tasks and by calculating the average durations. In some cases, values of T were much larger than the calculated average and it caused some non-monotonicity as it is seen in the Figure 3. This can be attributed to the constantly changing load of the grid environment.

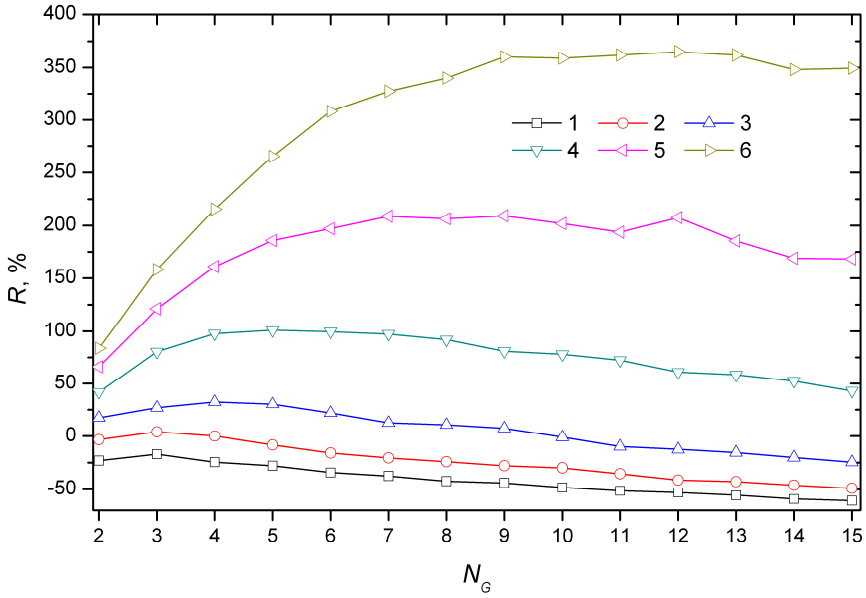

Figure 3. Acceleration R dependence on total count of tasks N_G at six values of calculations durations on local computer M_{PQ} T_S = 30 (1), 40 (2), 60 (3), 120 (4), 240 (5) and 480 (6) min.

As one can see in Figure 3, the efficiency of the developed technique of computational grid usage depends on the volume of the task. The smallest task efficiently solved in *BalticGrid* was with a duration of M_{PQ} T_S = 40 minutes. Calculations were faster than in the local computer but the task was divided into three parts and three remote computers were used instead of one local computer. By dividing the same task into more parts calculations were even slower than in the local computer because the time required to assign concrete task to its node increases more that decreases the time of calculations. The task in local computer solved in 120 minutes was efficiently solved in *BalticGrid* environment with all analyzed total count of tasks N_G. The task of 480 minutes was also solved efficiently with all analyzed total number N_G of tasks, but the most efficiently when N_G is between 9 and 13 (curve 6).

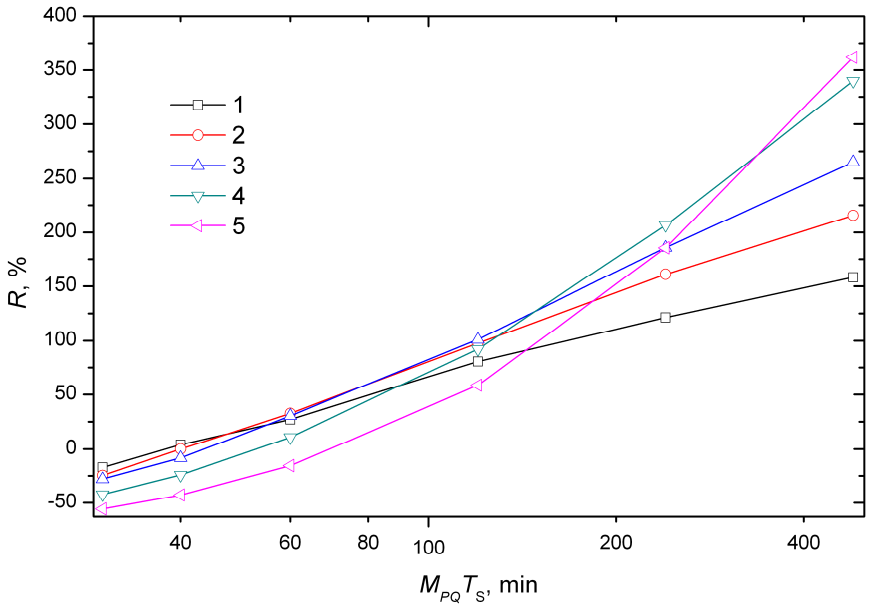

Figure 4. Acceleration R dependence on calculations durations on local computer M_{PQ} T_S at five total count of tasks N_G = 3 (1), 4 (2), 5 (3), 8 (4) and 13 (5).

Figure 4 shows that the larger is the task M_{PQ} T_S the larger becomes optimal count of tasks submitted to the grid N_G. 60 minutes duration task was solved most efficiently when N_G is 3, 4 and 5. Larger, 8 hours task, was solved most efficiently when N_G is 13. Accelerations differ slightly for the tasks with durations between 1 and 4 hours, when N_G is between 3 and 8 (curves 1–4).

Relatively small division size N_G and acceleration R can be attributed to the fact that resource broker used in *BalticGrid* environment is centralized. The centralized resource brokers can cause a noticeable delay. It was noticed during the studies that the delay is caused mainly by two reasons:

- The time required for middleware to assign concrete task to its computational node. This delay remains roughly constant during the experiments and is caused by configuration of the computational grid and the middleware itself.
- Random time when one or more computational tasks are finished later than the others. This delay depends on the load of the computational grid and the difference in capacities of computational nodes.

4.2 Recommendations for efficient usage of computational Grids

Our experience gathered during the process can be summarized by following recomendations.

It is required to determine if the usage of computational grid is meaningful at all before using grid computing to solve biosensors behavior. The smallest computational task solved in the grid environment faster than in local computer is required to be determined. The minimal effective duration T_{min} can be found by the following steps:

1. The duration T_S of calculations in a local computer with one parameter queue $PQ_1 = p_1, p_2, ..., p_N$ is measured.
2. In the initial state we assign $N_G = 2$ and $M_{PQ} = N_G$, where N_G is a total count of submitted tasks to the grid and M_{PQ} is the parameter queue length in parameter sweep.
3. A group of tasks $G = \{M, PS_i\}$ is sent to computational grid, where PS_i contains M_{PQ} / N_G parameter queues.

4. The duration T of calculations in the computational grid is measured.
5. If T is more than $M_{PQ}\ T_S$ then M_{PQ} is increased by N_G and steps beginning step number 3 are repeated. If T is less than $M_{PQ}\ T_S$ then $T_{min} = T$.
6. Steps beginning step number three can be repeated by increasing N_G by 1. This might be done in effort to find lesser T_{min} then is already determined in step number 5, but it is not necessary.

In our case the smallest task which is solved in computational grid faster than in the local computer was approximately $T_{min} = 40$ minutes and $N_G = 3$.

A total number of submitted tasks N_G should be chosen respectively to the volume of whole task. The larger is the task $M_{PQ}\ T_S$ the larger becomes optimal count of tasks submitted to the grid N_G.

5 CONCLUSIONS

The technique presented in this paper for an efficient usage of the computing grids can be applied to investigate the peculiarities of the biosensor response. The technique is bounded to parameterized computer modeling but not concrete biosensors, so it can be used for any parameter sweep-related simulations.

The technique was validated by investigating the biochemical behavior of the biosensor based on a chemically modified electrode. The biosensors action was analyzed in stirred and non-stirred solutions. The parameters affecting the error of quasi-steady-state modeling were determined.

The developed technique highly depends on the amount of the calculations and the count of parameter queues. The technique is more efficient for larger calculations rather than smaller ones.

REFERENCES

[1] R. Baronas and J. Kulys, Modelling amperometric biosensors based on chemically modified electrodes, *Sensors*, **8**, pp. 4800–4820 (2008).

[2] R. Baronas, F. Ivanauskas and J. Kulys, *Mathematical modeling of biosensors*, Springer (2010).

[3] C. P. Hong, *Computer modelling of heat and fluid flow in materials processing*, Taylor and Francis (2004).

[4] J. Binder, Supercomputers: Where are they now? *Aerospace America,* **43**, pp. 20–22 (2005).

[5] A.P.F. Turner, I. Karube and G.S. Wilson, *Biosensors: fundamentals and applications*, Oxford University Press (1987).

[6] F. W. Scheller, F. Schubert and J.Fedrowitz, *Frontiers in biosensorics*, Birkhauser Verlag, **Vol. II** (1997).

[7] V. Berstis, *Fundamentals of grid computing*, IBM Redbooks Paper, IBM (2002).

[8] I. Foster, What is the Grid? - a three point checklist. *GRIDtoday*, **1,** pp. 22–25 (2002).

[9] D. Abramson, J. Giddy and L. Kotler, High Performance Parametric Modelling with Nimrod/G: Killer Application for the Global Grid? In proceedings of the

14th International Parallel and Distributed Processing Symposium, IPDPS 2000, Cancun, Mexico, pp. 520–528 (2000).

[10] H. Casanova and F. Berman, Parameter Sweeps on the Grid with APST. In: F. Berman, G. Fox and T. Hey, *Grid Computing: Making the Global Infrastructure a Reality*, John Willey & Sons, pp. 773–788 (2003).

[11] M. Yarrow, K. M. Mccann, R. Biswas and R. F. Van der Wijungaart, An advanced user interface approach for complex parameter study process specification on the Information Power Grid (IPG), *Lect. Notes Comput. Sc.*, **1971**, pp. 146–157 (2000).

[12] D. Abramson, T. Peachey and A. Lewis, Model Optimization and Parameter Estimation with Nimrod/O. *Lect. Notes Comput. Sc.*, **3991**, pp. 720–727 (2006).

[13] D. Abramson, Applications development for the computational grid. *Lect. Notes Comput. Sc.*, **3841**, pp. 1–12 (2006)

[14] W. Sudholt, K. K. Baldridge, D. Abramson, C. Enticott and S. Garic, Applying Grid Computing to the Parameter Sweep of a Group Difference pseudopotential, *Lect. Notes Comput Sc*., **3036**, pp.148–155 (2004).

[15] F. Scheller and F. Schubert, *Biosensors*, Elsevier (1992).

[16] T. Schulmeister, Mathematical modelling of the dynamic behaviour of amperometric enzyme electrodes. *Select. Electr Rev*., **12**, pp. 203–260 (1990).

[17] A. A. Samarskii, *The theory of difference schemes*, Marcel Dekker (2001).

[18] J. Wang, *Analytical electrochemistry* (2^{nd} edn), Wiley (2000).