Nonlinear Analysis: Modelling and Control, 2007, Vol. 12, No. 3, 359–369

A Comparison of Finite Difference Schemes for Computational Modelling of Biosensors

E. Gaidamauskaitė[1], **R. Baronas**[1,2]

[1]Vilnius University
Naugarduko str. 24, LT-03225 Vilnius, Lithuania
evelina.gaidamauskaite@mif.vu.lt

[2]Institute of Mathematics and Informatics
Akademijos str. 4, LT-08663 Vilnius, Lithuania

Received: 25.03.2007 **Revised:** 08.06.2007 **Published online:** 31.08.2007

Abstract. This paper presents a one-dimensional-in-space mathematical model of an amperometric biosensor. The model is based on the reaction-diffusion equations containing a non-linear term related to Michaelis-Menten kinetics of the enzymatic reactions. The stated problem is solved numerically by applying the finite difference method. Several types of finite difference schemes are used. The numerical results for the schemes and couple mathematical software packages are compared and verified against known analytical solutions. Calculation results are compared in terms of the precision and computation time.

Keywords: numerical simulation, finite difference, reaction-diffusion, amperometric biosensor.

1 Introduction

The interest in biosensors is constantly growing as the range of practical applications of electrochemistry increases. Biosensors are small analytical devices capable of detecting specific compounds and therefore they are often applied in the fields of clinical, industrial, environmental and agricultural analyses [1–3]. A biosensor device is composed of biologically responsive material, mostly enzymes, and the electrode. Enzyme interacts with the target substance yielding the product. This process is usually described by Michaelis-Menten kinetics of the enzymatic reactions [4–6]. Amperometric biosensors are based on the measurement of the Faradaic current when a constant potential is kept. The current on electrode results due to the direct oxidation or reduction of an electroactive spieces.

Analytical solutions for mathematical models of the biosensors are obtainable exclusively in special cases [7,8]. In common case the models have to be solved numerically [9, 10]. Finite difference method is one of the most popular approximation techniques [11]. Numerous types of finite difference schemes can be considered for the solution of non-linear reaction-diffusion systems [4, 11]. Three major factors must be taken into account

when choosing the technique for simulation: the accuracy of the solution, computation time needed to solve the problem and ease of use of the technique. This work is focused on the analysis of several most commonly known finite difference schemes using computer simulation.

2 Mathematical model

An amperometric biosensor can be considered as an amperometric electrode, having a layer of enzyme immobilized onto the surface of the electrode. We assume the symmetrical geometry of the electrode and homogeneous distribution of the immobilized enzyme in the enzyme membrane.

We consider the following enzyme-catalysed reaction

$$\mathrm{S} + \mathrm{E} \rightleftarrows \mathrm{ES} \rightarrow \mathrm{E} + \mathrm{P}. \tag{1}$$

In this scheme the substrate (S) combines reversibly with an enzyme (E) to form a complex (ES). The complex then dissociates into a product (P) and the enzyme is regenerated. Assuming the quasi steady state approximation, the concentration of the intermediate complex (ES) do not change and may be neglected when simulating the biochemical behaviour of biosensors [1, 2]. The scheme (1) reduces to a simplified model of enzyme-catalyzed reaction, where the enzyme (E) binds to the substrate (S) producing the product (P) is considered,

$$\mathrm{S} \xrightarrow{\mathrm{E}} \mathrm{P}. \tag{2}$$

Coupling the enzyme-catalyzed reaction with the one-dimensional-in-space diffusion, described by Fick's second law, leads to the following system of equations [8, 9]:

$$\begin{aligned} &\frac{\partial S}{\partial t} = D_S \frac{\partial^2 S}{\partial x^2} - \frac{V_{max} S}{K_M + S}, \\ &\frac{\partial P}{\partial t} = D_P \frac{\partial^2 P}{\partial x^2} + \frac{V_{max} S}{K_M + S}, \quad 0 < x < d, \;\; t > 0, \end{aligned} \tag{3}$$

where $S(x,t)$ and $P(x,t)$ are the substrate and product concentrations, respectively, t stands for time and x – for space, D_S and D_P are the diffusion coefficients of the substrate and product, respectively, K_M is the Michaelis-Menten constant, V_{max} is the maximal enzymatic rate attainable when the enzyme is fully saturated with substrate, d is the thickness of the enzyme membrane.

Let x = 0 represents the electrode surface, while $x = d$ represents the bulk solution-membrane interface. The biosensor operation starts when some substrate appears on the surface of the enzyme layer,

$$\begin{aligned} &S(x,0) = 0, \quad 0 \le x < d, \\ &S(d,0) = S_0, \\ &P(x,0) = 0, \quad 0 \le x \le d, \end{aligned} \tag{4}$$

where S_0 stands for the concentration of substrate in the bulk solution.

In the case of amperometric biosensors, due to the electrode polarization, the concentration of the reaction product at the electrode surface is being permanently reduced to zero. The substrate does not react at the electrode surface. If the substrate is well-stirred and in powerful motion, then the diffusion layer ($0 < x < d$) remains at a constant thickness of d during the biosensor operation. This is used in the boundary conditions given by:

$$\begin{aligned} &\frac{\partial S}{\partial x}\Big|_{x=0} = 0, \\ &S(d,t) = S_0, \\ &P(0,t) = P(d,t) = 0. \end{aligned} \tag{5}$$

The measured current is accepted as a response of an amperometric biosensor in a physical experiment. The current depends upon the flux of the reaction product at the electrode surface, i.e. at the border x = 0. Consequently, the density $I(t)$ of the anodic current at a time t can be obtained explicitly from Faraday's and second Fick's laws using the flux of the product concentration at the surface of the electrode,

$$I(t) = n_e F D_P \frac{\partial P}{\partial x}\Big|_{x=0}, \tag{6}$$

where n_e is a number of electrons, involved in charge transfer at the electrode surface, and F is the Faraday constant.

We assume that the system (3)–(5) approaches a steady state as $t \to \infty$,

$$I_p = \lim_{t\to\infty} I(t), \tag{7}$$

where I_p is the density of the steady state current.

3 Solution of the problem

The analytical solutions for nonlinear partial differential equations generally do not exist. Equations (3)–(5) describing the action of an amperometric biosensor do not have ones either, so numerical approximation must be used. We applied finite difference technique to solve (3)–(5) the boundary value problem numerically [10, 11].

3.1 Analytical solutions

A non-linear term in equations (3) turns to linear one in special cases of the substrate concentration,

$$\frac{V_{max}S}{K_M + S} \approx \frac{V_{max}}{K_M} S, \quad \text{when } S \ll K_M, \tag{8}$$

$$\frac{V_{max}S}{K_M + S} \approx V_{max}, \quad \text{when } S \gg K_M. \tag{9}$$

The analytical solutions are known for the boundary value problem (3)–(5) in the cases of linear reaction terms [7, 8]. Exact solutions are helpful in testing models and assessing accuracy of the solution.

If inequality $S_0 \ll K_M$ is satisfied, then the biosensor steady state current can be calculated as follows [8]:

$$I_l = n_e F D_P S_0 \frac{1}{d}\left(1 - \frac{1}{\cosh \sigma}\right), \tag{10}$$

where σ^2 is a dimensionless diffusion modulus, Damköhler number,

$$\sigma^2 = \frac{V_{max} d^2}{D_S K_M}. \tag{11}$$

The biosensor response is known to be under mass transport control if the enzymatic reaction in the enzyme layer is faster than the mass transport process [4, 8, 9]. The diffusion modulus essentially compares the rate of enzymatic reaction (V_{max}/K_M) with the diffusion through the enzyme layer (D_S/d^2). If $\sigma^2 \ll 1$ then the enzyme kinetics controls the biosensor response. The response is under diffusion control when $\sigma^2 \gg 1$.

At the high concentration of the substrate ($S_0 \gg K_M$), the biosensor steady state current does not depend on the concentration S_0 of the analyte [7],

$$I_g = \frac{n_e F V_{max} d}{2}. \tag{12}$$

However, in the intermediate concentration cases, i.e. if $S_0 \approx K_M$, the analytical solutions are unknown and numerical methods are used to solve the problem [4, 9, 10].

3.2 Finite difference schemes

We introduce an uniform discrete grid $\omega_h \times \omega_\tau$ to simulate the biosensor using finite difference method,

$$\begin{aligned} &\omega_h = \{x_i :\ x_i = ih,\ i = 0, \ldots, N;\ hN = d\}, \\ &\omega_\tau = \{t_j :\ t_j = j\tau,\ j = 0, \ldots, M;\ \tau M = T\}, \end{aligned} \tag{13}$$

where T stands for the duration of the process analysis.

The differential equations are discretized in that domain assuming the following definitions:

$$\begin{aligned} &S_i^j = S(x_i, t_j), \quad P_i^j = P(x_i, t_j), \quad I_j = I(t_j), \\ &i = 0, \ldots, N; \quad j = 0, \ldots, M. \end{aligned} \tag{14}$$

3.2.1 Explicit finite difference scheme

Using explicit finite difference scheme for the substrate and product concentrations (3) we obtain the following finite difference equations [9–11]:

$$\begin{aligned}
&\frac{S_i^{j+1}-S_i^j}{\tau} = D_S\left(\frac{S_{i+1}^j - 2S_i^j + S_{i-1}^j}{h^2}\right) - \frac{V_{max}S_i^j}{K_M + S_i^j},\\
&\frac{P_i^{j+1}-P_i^j}{\tau} = D_P\left(\frac{P_{i+1}^j - 2P_i^j + P_{i-1}^j}{h^2}\right) + \frac{V_{max}S_i^j}{K_M + S_i^j},\\
&i = 1,\ldots,N-1; \quad j = 1,\ldots,M.
\end{aligned} \tag{15}$$

The initial conditions (4) in numerical model has the following form

$$\begin{aligned}
&S_i^0 = 0, \quad 0 \le i < N,\\
&S_N^0 = S_0,\\
&P_i^0 = 0, \quad 0 \le i \le N.
\end{aligned} \tag{16}$$

For the boundary conditions (5) we obtain:

$$\begin{aligned}
&S_0^j = S_1^j,\\
&S_N^j = S_0,\\
&P_0^j = P_N^j = 0, \quad 1 \le j \le M.
\end{aligned} \tag{17}$$

The formulae for calculation of current density (6) becomes thus ($0 < j \le M$):

$$I_j = n_e F D_P \frac{P_1^j}{h}. \tag{18}$$

We consider the density I_R of the steady state current calculated at the moment T_R

$$I_R = I(T_R) \approx I_p, \quad T_R = \min_{j>0,\, I_j>0}\left\{t_j \colon \frac{I_j - I_{j-1}}{I_j \tau} < \varepsilon\right\}, \quad T \approx T_R. \tag{19}$$

We used $\varepsilon = 10^{-5}$ for the calculations.

One of the most important features of the scheme is the stability [11]. The prerequisite for the stability of the explicit finite difference scheme (15)–(17) is the following condition:

$$\tau \le \min\left\{\frac{h^2}{2D_S}, \frac{h^2}{2D_P}\right\}. \tag{20}$$

Because of these stability conditions, a number of the time steps must be magnified strongly as the number of the space steps is increased. This leads to the inefficient calculations.

3.2.2 Implicit 1 finite difference scheme

Mathematical model of a biosensor can be solved using several implicit finite difference schemes [9–11]. Let us name first of them "Implicit 1 finite difference scheme" and compose its equations.

The model governing equation for the substrate concentration in (3) is replaced by the following finite difference equation:

$$\frac{S_i^j - S_i^{j-1}}{\tau} = D_S\left(\frac{S_{i+1}^j - 2S_i^j + S_{i-1}^j}{h^2}\right) - \frac{V_{max}S_i^{j-1}}{K_M + S_i^{j-1}}. \tag{21}$$

Known concentration values of the substrate at the upper layer can be used for calculation of the product concentration. Hence, governing equation for the product concentration can be approximated with:

$$\frac{P_i^j - P_i^{j-1}}{\tau} = D_P\left(\frac{P_{i+1}^j - 2P_i^j + P_{i-1}^j}{h^2}\right) + \frac{V_{max}S_i^j}{K_M + S_i^j}. \tag{22}$$

The rest equations (4)–(6) take the same form as those of the explicit scheme.

3.2.3 Implicit 2 finite difference scheme

The model equation for the substrate concentration in (3) may be approximated with a more implicit scheme than equation (21). At a numerator of reaction term the concentration of substrate can be used in a upper level,

$$\frac{S_i^j - S_i^{j-1}}{\tau} = D_S\left(\frac{S_{i+1}^j - 2S_i^j + S_{i-1}^j}{h^2}\right) - \frac{V_{max}S_i^j}{K_M + S_i^{j-1}}. \tag{23}$$

Present numerical equation is linear, same as (21). The other equations match those obtained using the explicit scheme.

3.2.4 Crank-Nicolson scheme

Using Crank-Nicolson [6, 11] method the reaction-diffusion equations (3) are approximated by linear finite difference equations,

$$\begin{aligned}
\frac{S_i^j - S_i^{j-1}}{\tau} &= \frac{D_S}{2h^2}\left(S_{i+1}^j - 2S_i^j + S_{i-1}^j + S_{i+1}^{j-1} - 2S_i^{j-1} + S_{i-1}^{j-1}\right) - \frac{V_{max}S_i^{j-1}}{K_M + S_i^{j-1}}, \\
\frac{P_i^j - P_i^{j-1}}{\tau} &= \frac{D_P}{2h^2}\left(P_{i+1}^j - 2P_i^j + P_{i-1}^j + P_{i+1}^{j-1} - 2P_i^{j-1} + P_{i-1}^{j-1}\right) + \frac{V_{max}S_i^j}{K_M + S_i^j}.
\end{aligned} \tag{24}$$

The rest equations (4)–(6) are approximated like in the explicit scheme.

3.2.5 Hopscotch scheme

Using Hopscotch scheme unknown grid points are obtained at two phases [11]. On the first phase, even grid points ($S[e]_i^{j+1}$, $P[e]_i^{j+1}$) are calculated explicitly using known lower layer values (S_i^j, P_i^j),

$$\begin{aligned}\frac{S[e]_i^{j+1}-S_i^j}{\tau} &= D_S\left(\frac{S_{i+1}^j-2S_i^j+S_{i-1}^j}{h^2}\right)-\frac{V_{max}S_i^j}{K_M+S_i^j},\\ \frac{P[e]_i^{j+1}-P_i^j}{\tau} &= D_P\left(\frac{P_{i+1}^j-2P_i^j+P_{i-1}^j}{h^2}\right)+\frac{V_{max}S_i^j}{K_M+S_i^j}.\end{aligned} \tag{25}$$

On the second phase, odd grid points ($S[o]_i^{j+1}$, $P[o]_i^{j+1}$) are calculated using odd values of the lower level and already known even values of the upper layer,

$$\begin{aligned}\frac{S[o]_i^{j+1}-S_i^j}{\tau} &= D_S\left(\frac{S[e]_{i+1}^{j+1}-2S[o]_i^{j+1}+S[e]_{i-1}^{j+1}}{h^2}\right)-\frac{V_{max}S_i^j}{K_M+S_i^j},\\ \frac{P[o]_i^{j+1}-P_i^j}{\tau} &= D_P\left(\frac{P[e]_{i+1}^{j+1}-2P[o]_i^{j+1}+P[e]_{i-1}^{j+1}}{h^2}\right)+\frac{V_{max}S_i^{j+1}}{K_M+S_i^{j+1}}.\end{aligned} \tag{26}$$

Computation continues alternating the calculation order of the odd and even points.

Hopscotch scheme is fully explicit yet unconditionally stable for $V_{max} = 0$ and therefore it can operate with any size of time and space steps.

3.3 Mathematical software packages

The major mathematical software packages provide tools for solving systems of the partial differential equations. However, a greater ease of use and wider range of solvable problems often comes at the expense of lower precision or less efficiency.

We used Maple (Maplesoft, Inc.) version 10 general-purpose solver *"pdsolve"* to find numerical solution for the system of the partial differential equations [13]. This solver uses finite difference method and can be configured with eleven classical schemes, calculation step size and other parameters.

The biosensor action was also simulated with MATLAB (The MathWorks, Inc.) software package [14]. The problem was solved using built-in solver *"pdepe"*, which provides a numerical solution for systems of differential equations in single spatial dimension and time.

4 Results and discussion

Computer simulation was used to compare accuracy and performance of the solution techniques. Since the system of linear algebraic equations is tridiagonal it can be solved efficiently [11]. Calculation results are compared in terms of precision and computation time. We define the relative error E as the absolute difference of the steady state current

density estimated by analytical and numerical solutions divided by the steady state current density of analytical solution.

$$E = \frac{|I_R - I_a|}{I_a}, \quad I_a = \begin{cases} I_l, & S_0 \ll K_M, \\ I_g, & S_0 \gg K_M, \end{cases} \tag{27}$$

where I_R is the numerical solution defined by (19) while I_l and I_g are analytical solutions defined by (10) and (12), respectively.

The following values of the model parameters were employed:

$$\begin{aligned} &K_M = 100\ \mu\text{M},\ S_0 \in \{10^{-3}K_M; 10^3\ K_M\},\ V_{max} = 100\ \mu\text{M/s}, \\ &D_S = D_P = 300\ \mu\text{m}^2/\text{s},\ n_e = 2,\ T = 10\ \text{s},\ d = 100\ \mu\text{m}. \end{aligned} \tag{28}$$

The routines of the finite difference method were implemented in Java programming language [12]. For performance reasons, we executed programs of the mathematical software packages using command-line approach. The experiments were performed on the 2 GHz Intel Core 2 Duo Processor with 1GB of RAM.

As a first test problem, relative errors of the finite difference schemes and mathematical software packages were examined using two known analytical solutions (10) and (12). We applied $M = 10^2$ for the calculations using implicit 1, implicit 2 and Crank-Nicolson schemes and $M = 10^5$ using explicit and Hopscotch schemes because of the stability constraints on the time step. All the considered finite difference schemes yield very similar precision, therefore only explicit scheme results are presented in Fig. 1 as the example. The smallest relative errors are obtained using finite difference schemes. Maple package calculates the steady state biosensor current more accurately than MATLAB. In cases of high substrate concentration (12) Maple's results are as precise as those obtained by explicit scheme. The numbers of steps used in calculations do not influence the accuracy of the MATLAB solution.

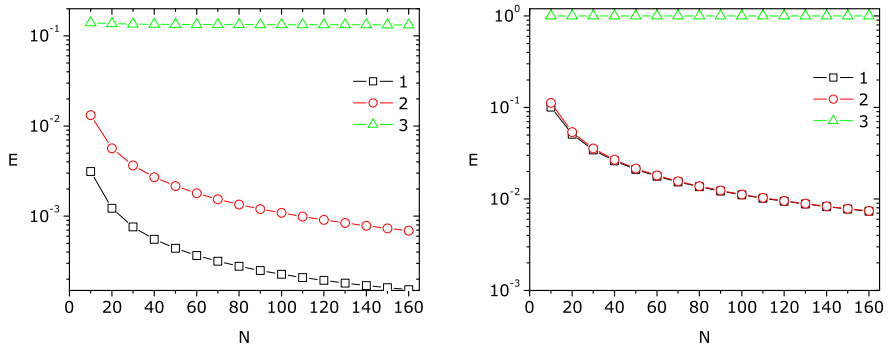

Fig. 1. Dependence of relative error E on the number of space steps N for two values of S_0: $10^{-3}K_M$ (a) and $10^3 K_M$ (b). 1 – explicit scheme ($M = 10^5$), 2 – Maple ($M = 10^2$), 3 – MATLAB ($M = 10^2$).

In Fig. 2 the finite difference schemes are compared to the explicit scheme. The Hopscotch scheme differs very slightly from the explicit scheme in both analytical solution cases, whereas the implicit schemes showed the maximal difference of approximately 0.8 % when the number of space steps N equals to 160 (Fig. 2). The relative errors computed using the analytical solution (12) at $S_0 \gg K_M$ are by a few orders of magnitude larger than the corresponding errors at $S_0 \ll K_M$ calculated using (10). This could be explained by less accuracy of the analytical solution (12) compared to the solution (10) [7, 8]. Considered schemes yield more similar results using analytical solution (12).

In the next test problem, we consider the computation time as a function of the relative error (Fig. 3). Introducing different limits ϵ for the relative error E, the computation

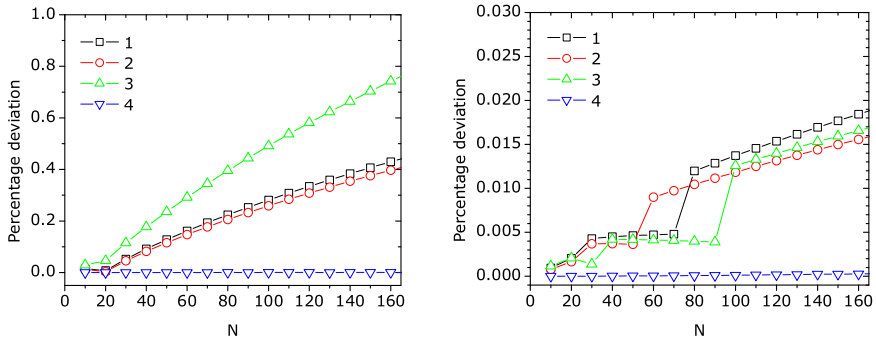

Fig. 2. The percentage ratio of the relative error E of the finite difference schemes to the error E of the explicit scheme for two values of S_0: $10^{-3}K_M$ (a) and $10^3 K_M$ (b). 1 – implicit 1 scheme, 2 – implicit 2 scheme, 3 – Crank-Nicolson scheme, 4 - Hopscotch scheme.

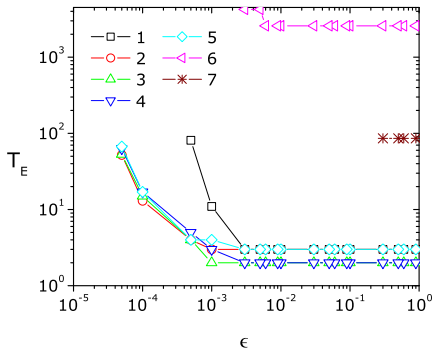

Fig. 3. The computation time T_E versus the relative error ϵ, $N, M \in \{20, 40, 80, 160, 320, 640, 1280, 2560, 5120, 10240\}$. 1 – explicit scheme, 2 – implicit 1 scheme, 3 – implicit 2 scheme, 4 – Crank-Nicolson scheme, 5 – Hopscotch scheme, 6 – Maple, 7 – MATLAB.

time $T_E(\epsilon)$ is given by:

$$T_E(\epsilon) = \min_{N,M} \{T_{N,M} : \ E \leq \epsilon\}, \tag{29}$$

where $T_{N,M}$ is the time of calculation at given numbers of grid steps N and M. $T_E(\epsilon)$ is the minimal time of computation needed to achieve the relative error E not greater than ϵ. The calculations were performed for very different values of space and time steps, $N,\ M \in \{20, 40, 80, 160, 320, 640, 1280, 2560, 5120, 10240\}$.

As one can see in Fig. 3, the implicit and Hopscotch are the fastest schemes to achieve the required relative error. Despite being very computationally intensive, partial differential equation solvers in mathematical software packages cannot accurately calculate the results. The MATLAB solver does not obtain higher precision than 0.1.

Finally, the computation times for different grid steps are reported in Tables 1, 2. The Table 1 shows that there is a very small difference between schemes implicit 2 and Crank-Nicolson and they are the most computationally intensive schemes. Explicit scheme is the fastest computation technique. Mathematical software packages are significantly more computationally intensive, particularly Maple, see Table 2.

Table 1. Computation time [ms] by the finite difference schemes, $N = 100$

M	Explicit	Implicit 1	Implicit 2	Crank-Nicolson	Hopscotch
10000	284	555	996	1064	659
20000	398	1108	1967	2090	1356
40000	755	2232	3974	4205	3062
80000	1480	4450	8080	8853	5563
160000	2957	8866	16542	16829	10623

Table 2. Computation time [s] by the mathematical software packages, $N = 100$

M	Maple	MATLAB
100	695	0.64
200	2480	0.74
400	9379	0.85

5 Conclusions

In this article, several finite difference schemes were applied for modelling an amperometric biosensor. Using all the considered schemes quite satisfactory results were obtained when sufficient number of steps of the discrete grid is employed.

The best accuracy is achieved using implicit calculation and Hopscotch approaches. For the problems where accuracy is not a significant factor but the speed is, the simplest explicit scheme should be used.

General-purpose solvers of Maple and MATLAB are less precise to simulate the biosensor action and need more computation time. Those solvers can be applied for basic problems while taking advantage of the simplicity.

Acknowledgement

The authors express sincere gratitude to prof. Feliksas Ivanauskas and prof. Juozas Kulys for valuable discussions and their contribution into modelling of biosensors.

References

1. F. Scheller, F. Schubert, *Biosensors*, Vol. 7, Elsevier, Amsterdam, 1992.
2. A. P. F. Turner, I. Karube, G. S. Wilson, *Biosensors: Fundamentals and Applications*, Oxford University Press, Oxford, 1987.
3. U. Wollenberger, F. Lisdat, F. W. Scheller, *Frontiers in Biosensorics 2. Practical Applications*, Birkhauser Verlag, Basel, 1997.
4. R. Aris, *The Mathematical Theory of Diffusion and Reaction in Permeable Catalysts. The Theory of the Steady State*, Clarendon Press, Oxford, 1975.
5. R. Baronas, F. Ivanauskas, J. Kulys, The Influence of the Enzyme Membrane Thickness on the Response of Amperometric Biosensors, *Sensors*, **3**, pp. 248–262, 2003.
6. J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type, *Proceedings of the Cambridge Philosophical Society*, **43**, pp. 50–64, 1947.
7. P. W. Carr, L. D. Bower, *Immobilized Enzymes in Analytical and Clinical Chemistry: Fundamentals and Applications*, John Wiley, New York, 1980.
8. J. Kulys, Development of new analytical systems based on biocatalysts, *Enzyme and Microbial Technology*, **3**, pp. 344–352, 1981.
9. T. Schulmeister, Mathematical modelling of the dynamic behaviour of amperometric enzyme electrodes, *Selective Electrode Reviews*, **12**, pp. 203–260, 1990.
10. D. Britz, *Digital Simulation in Electrochemistry*, 3nd ed., Springer-Verlag, Berlin, 2005.
11. A. A. Samarskii, *The Theory of Difference Schemes*, Marcel Dekker, New York-Basel, 2001.
12. J. E. Moreira, S. P. Midkiff, M. Gupta, P. V. Artigas, M. Snir, R. D. Lawrence, Java programming for high performance numerical computing, *IBM Systems Journal*, **39**, pp. 21–56, 2000.
13. W. Gander, J. Hrebicek, Solving Problems in Scientific Computing Using Maple and MATLAB®, 4th ed., Springer-Verlag, Berlin, 2004.
14. W. Y. Yang, W. Cao, T.-S. Chung, J. Morris, Applied Numerical Methods Using MATLAB®, John Willey & Sons, Hoboken, N.J., 2005.