# Odorant Transport Modeling

*E. B. Rudnyi*

IMTEK, Freiburg University, 2006

The text below is part of the joint work in the framework of GOSPEL http://gospel-network.org under the jointly executed research Biomimetic Measurement Systems.

## Introduction

The document describes results of simple 2D simulation for the convection-diffusion transport within a sensor array. Geometry and numerical values of the parameters are chosen rather arbitrary. The goal is rather to demonstrate what typical simulation results look like and how one can speed simulation up considerably by means of modern model reduction. We will start with the description of a case study, then we will present model reduction and discuss what computational advantages it brings forward and after that we will show how the concentration profile depends on the consumption rate at sensors and flow velocity.

## Case Study

Fig. 1 shows a 2D computational domain. It consists from a rectangle of 1 mm by 12 mm with 4 sensors. The rectangle is open from the left. The gas species diffuses from the left to the right over four sensors and additionally the carrier gas can have flow velocity $v$. Initially the species concentration within the rectangle is assumed to be zero.

We solve the convection-diffusion transport equation

$$dc/dt = D\nabla^2 c - v\nabla c \tag{1}$$

with initial and boundary conditions as follows.

As was mentioned above, the initial conditions are assumed to be zero:

$$c(\mathbf{r}, t=0) = 0. \tag{2}$$

The boundary conditions on the left side of the rectangle (red arrows in Fig. 1) are the Dirichlet boundary conditions:

$$c(\mathbf{r} = left\,side, t) = 1, \tag{3}$$

that is, the concentration in the opening is fixed and does not change with time. This condition can be relaxed by introducing a function in time $c(\mathbf{r} = left\,side, t) = f(t)$. In any case, this corresponds to the case when species supply outside of the computational domain is very fast and it is assumed that the concentration at the opening is controlled externally. The concentration units could be arbitrary as Eq (1) is linear. Please use any appropriate units.

The boundary conditions on the sensors model the consumption of the species (the mixed boundary conditions):

$$species\,flux(\mathbf{r} = sensor, t) = kc(\mathbf{r} = sensor, t), \tag{4}$$

that is, the consumption rate is proportional to the current concentration.

With exception of the left side and sensors surfaces, the rectangle is considered to be non-penetrable, that is, the species flux at other locations on the boundary was assumed to be zero (the zero Neumann boundary conditions).

The numerical values of parameters are listed in Table 1. It should be mentioned that the flow profile was approximated by a rectangular. This is quite a rough approximation and has been used only to demonstrate the principal possibility to treat the flow. First, we change the consumption rate for a pure diffusion problem and then we change the flow velocity for a convection-diffusion problem.

Table 1. Parameters used in numerical simulations.

| *Code* | *Diffusion coefficient (Eq 1) in $cm^2/s$* | *Flow velocity (Eq 1) in cm/s* | *Consumption rate (Eq 4) in cm/s* |
|---|---|---|---|
| V0K001 | 1 | 0 | 0.01 |
| V0K01 | 1 | 0 | 0.1 |
| V0K1 | 1 | 0 | 1 |
| V0K10 | 1 | 0 | 10 |
| V2K1 | 1 | 2 | 1 |
| V10K1 | 1 | 10 | 1 |
| V20K1 | 1 | 20 | 1 |

ANSYS has been used to solve Eq (1) with boundary conditions (2) to (4) by the finite element method. As the diffusion equation (1) is equivalent to the heat transfer equation, the latter was employed in ANSYS. This means that TEMP should be read as concentration on the ANSYS plots.

The computational domain has been meshed with quadratic PLANE77 elements when the element size was set to 0.025 mm. After meshing, there were 19200 elements and 58641 nodes. In our experience, quadratic elements give us the best accuracy. Unfortunately, PLANE77 can solve a pure diffusion problem only and the linear element PLANE55 was employed for simulations with nonzero flow velocity.

### Model Reduction as Fast Solver

The discretization in space of Eq (1) to (4) produces a system of linear ordinary differential equations as follows:

$$E\dot{c} + Kc = Fu(t) \tag{5}$$

where $E$ and $K$ are the system matrices, $c$ is the state vector representing concentrations at the nodes introduced during the discretization, $F$ is referred to as the load vector and $u(t)$ is the input function. In our case, the input function was modeled as the step function but it can be any function in time corresponding to the species concentration at the inlet of the sensor array controlled externally.

The idea of model reduction [1][2] (see also http://ModelReduction.com ) is to find a low dimensional subspace that accurately captures the dynamic

$$c \approx Vz \tag{6}$$

and then project the original system onto this subspace

$$\hat{E}\dot{z} + \hat{K}z = \hat{F}u(t) \tag{7}$$

As the main computational cost to integrate Eq (5) in time comes from the high dimension of the state vector, model reduction allows us to speed up transient simulation considerably.

Software mor4ansys [3] (http://www.imtek.uni-freiburg.de/simulation/mor4ansys/ ) allows us to employ modern model reduction directly to ANSYS models. It reads ANSYS binary files to build the system (5) and then performs implicit moment matching via the Arnoldi process to find the low-dimension subspace (6). We demonstrate its work with a case study V0K1 (see Table 1).

Fig 2 to 5 shows the concentrations at four sensors (see Fig 1) obtained by integrating in time the full (58641 equation) and reduced models (30 equations). In figures, the difference is within the line thickness (you cannot see the red line at all). This shows that the low dimensional subspace found by mor4ansys do capture the transient dynamics of the original system. In our experience, the dimension of the reduced model of 30 is enough for many thermal and mechanical models. The method to determine an optimal dimension for a reduced model in the general case is presented in ref [4].

Below we report timing measured at Sun Ultra-80 with a 450 MHz Sparc processor and 4 Gb of RAM. Modern computers are faster by a factor 4-5 and the time should be treated accordingly.

The minimal computational time required for a finite element model is for a stationary solution, which corresponds to a single solution of a system of linear equations. In our case, it took 41 s in ANSYS from which the solution of a linear system was about 20 s. The time for transient simulation is much larger as it is necessary to solve a system of linear equation for each timestep. In our case, with the use of adaptive method to integrate in time, ANSYS has made 121 timesteps for 2320 s. On the other hand, mor4ansys has generated a reduced system of dimension 30 for 32 s and the reduced system was integrated in time in Mathematica for less than 1 s. As such, we can state that model reduction allows us to have transient simulation for the computational cost comparable with that for a static solution. This is consistent with our analysis described in [3].

Note that the model reduction process does not depend on the input function $u(t)$ in Eq (5). This means that the same reduced model can be used for any input function. However, when geometry or system properties change, then system matrices in Eq (5) change as well and model reduction process should be repeated.

Still as was shown above, model reduction is computationally advantageous even in the case when the reduced model is employed once. As a result, we can consider model reduction as a fast solver for transient simulation.

There are new results [5] that allow us to preserve some parameters in the symbolic form during model reduction. For example, in the case considered in this report one can write for the system matrix $K$

$$K = K_o + \sum_i p_i K_i \tag{8}$$

where $p_i$ is the scalar parameter to preserve (consumption rate or flow velocity) and $K_i$ is the contribution to the system matrix related to the parameter $p_i$. The method described in [5] preserves $p_i$ in the symbolic form. This means that in principle one can change consumption rates and flow velocity at the level of the reduced model. However, the software to use the new method is under development.

## Simulation Results

Transient simulation has been performed with the use of model reduction (ANSYS + mor4ansys) for all seven cases listed in Table 1.

For each simulation, results are presented in two plots. First is a contour plot for the stationary solution of Eq (1) (the derivative in time is zero). Please note that the color map on each contour plot is different from each other. This means that the same color on different plots means different concentration range. Please consult the color map on the right side of the plot.

The second plot displays the concentration over four sensors as a function of time. The concentration units are the same as in Eq (2) and the time is in seconds.

The simulation results in Fig 6 to 19 are in agreement with intuition. The higher consumption rate, the stronger becomes the concentration gradient along the rectangle (see Fig 6 to Fig 13). On the other hand, the flow velocity reduces the concentration gradient (see Fig 14 to Fig 19). However, even when the species concentration is close to uniform for the stationary solution, transient response in the beginning is quite different for different sensors.

## Conclusion

The goal of the report was to demonstrate the possibilities of modern finite element software to solve a convection-diffusion problem and stress that model reduction can reduce the computational cost considerably. The combination of ANSYS and mor4ansys allows us increase the number of computational experiments and hence to play "what-if".

mor4ansys is available for download from the mor4ansys site (http://ModelReduction.com ). Note that we also have a development environment to solve an optimization problem based on model reduction [6].

It is also possible to use parametric model reduction [5]. However, one should keep in mind that this technique is still under development.

## References


[1] Athanasios C. Antoulas, Approximation of Large-Scale Dynamical Systems. Society for Industrial and Applied Mathematic, 2005, ISBN: 0898715296.

[2] E. B. Rudnyi, J. G. Korvink, Review: Automatic Model Reduction for Transient Simulation of MEMS-based Devices. Sensors Update v. 11, p. 3-33, 2002.

[3] E. B. Rudnyi, J. G. Korvink. Model Order Reduction for Large Scale Engineering Models Developed in ANSYS. Lecture Notes in Computer Science, v. 3732, pp. 349–356, 2005.

[4] T. Bechtold, E. B. Rudnyi and J. G. Korvink, Error indicators for fully automatic extraction of heat-transfer macromodels for MEMS. Journal of Micromechanics and Microengineering 2005, v. 15, N 3, pp. 430-440.

[5] E. B. Rudnyi, C. Moosmann, A. Greiner, T. Bechtold, J. G. Korvink. Parameter Preserving Model Reduction for MEMS System-level Simulation and Design. 5th MATHMOD, Proceedings. Volume 1: Abstract Volume, p. 147, Volume 2: Full Papers CD, 8 pp, February 8 - 10, 2006, Vienna University of Technology, Vienna, Austria. ISBN 3-901608-30-3.

[6] J. S. Han, E. B. Rudnyi, J. G. Korvink. Efficient optimization of transient dynamic problems in MEMS devices using model order reduction. Journal of Micromechanics and Microengineering 2005, v. 15, N 4, p. 822-832.

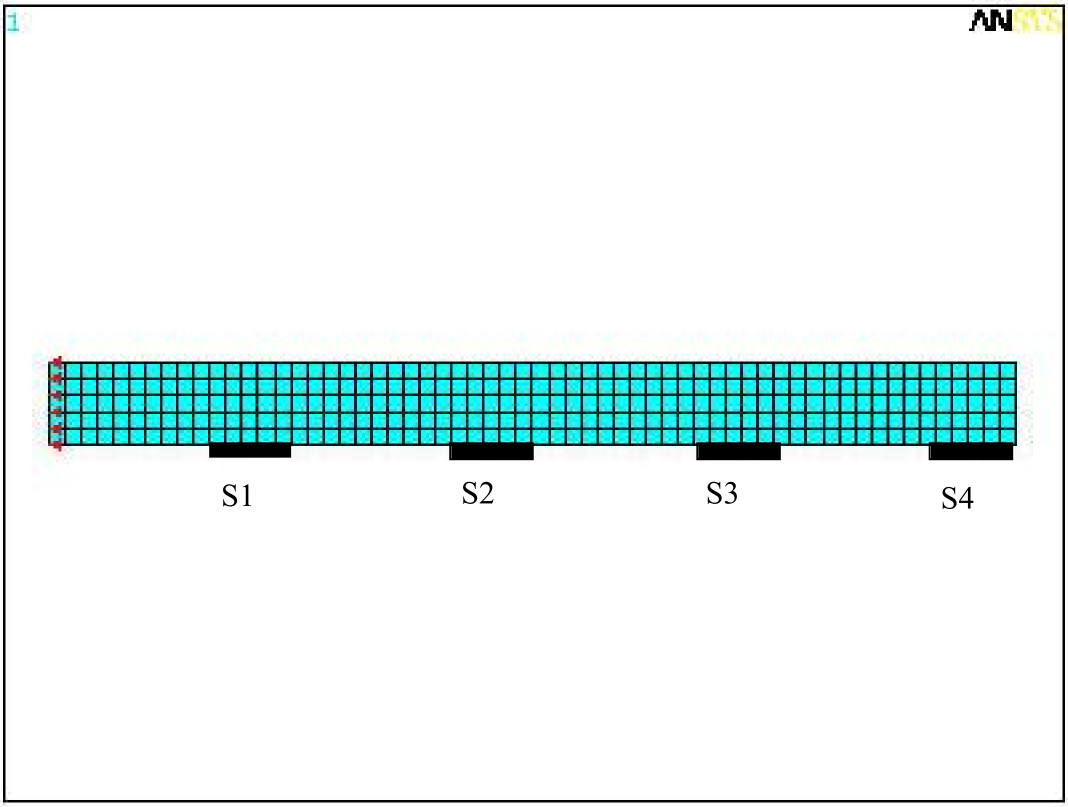


Fig. 1. Computational domain. Real mesh used in the simulation is much finer than shown in the figure (see text).

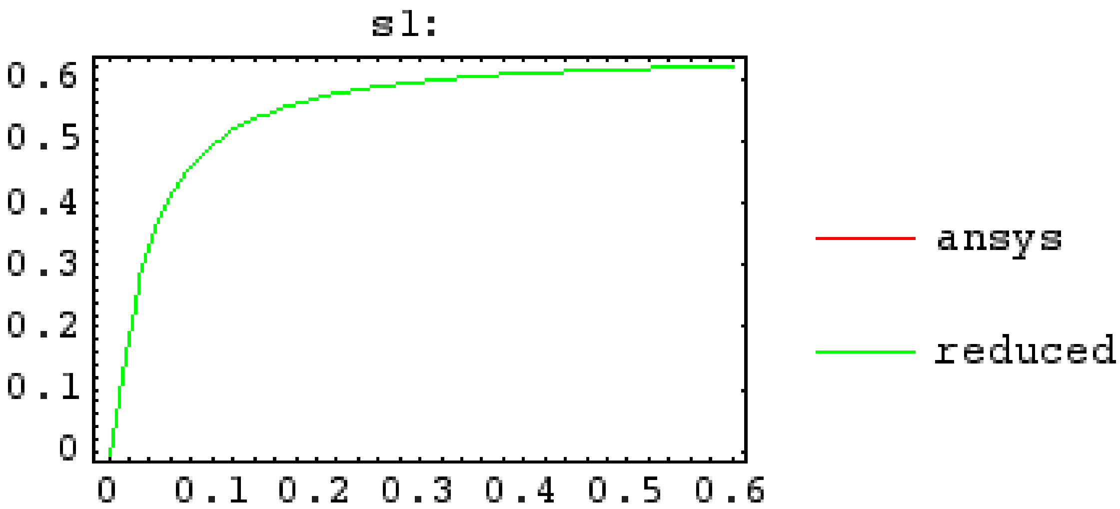


Fig. 2. V0K1. Concentration over sensor 1 as a function of time. Red line is for ANSYS simulation (58641), green line is for the reduced model (30 equations).

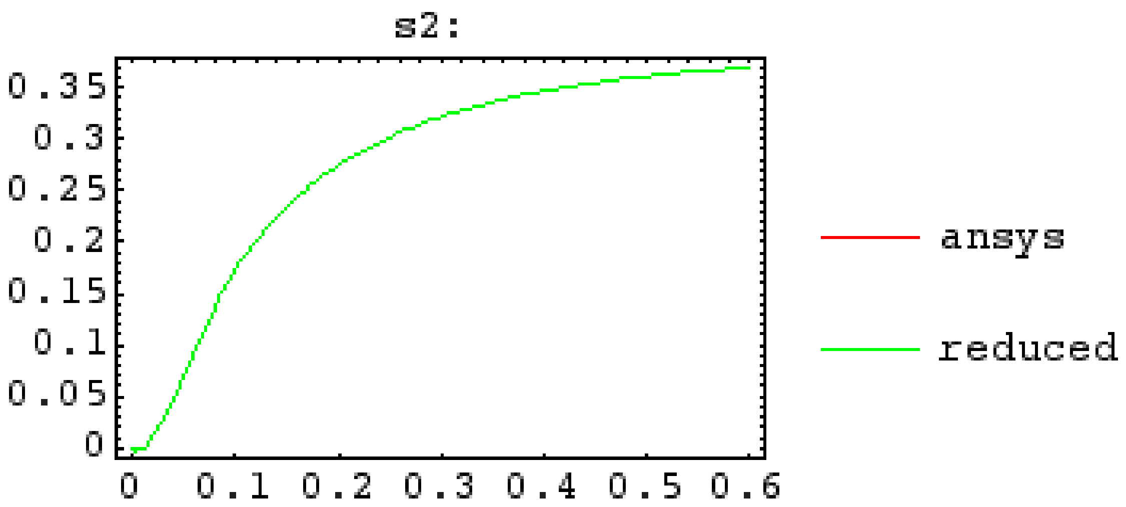


Fig. 3. V0K1. Concentration over sensor 2 as a function of time. Red line is for ANSYS simulation (58641), green line is for the reduced model (30 equations).

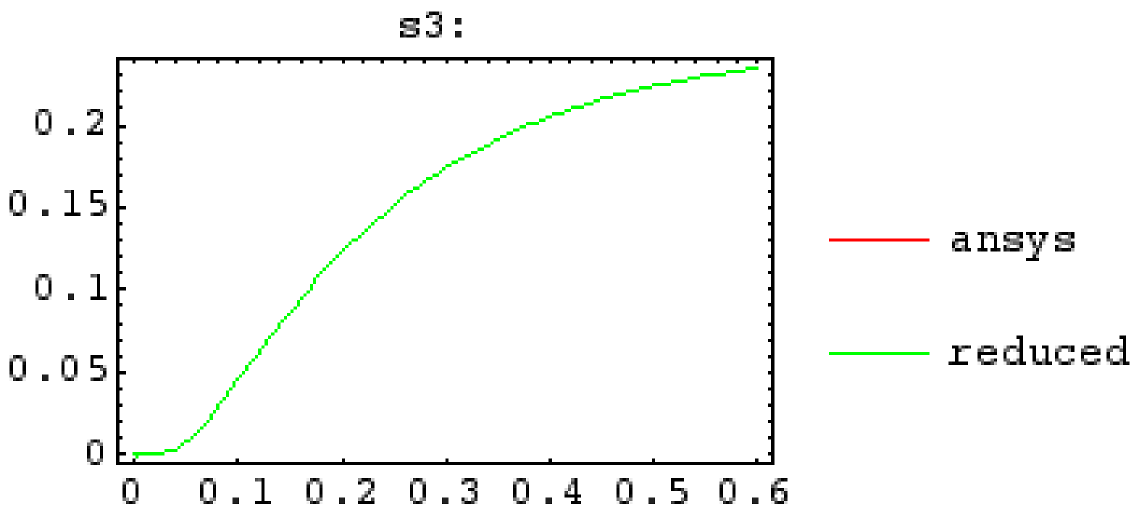


Fig. 4. V0K1. Concentration over sensor 3 as a function of time. Red line is for ANSYS simulation (58641), green line is for the reduced model (30 equations).

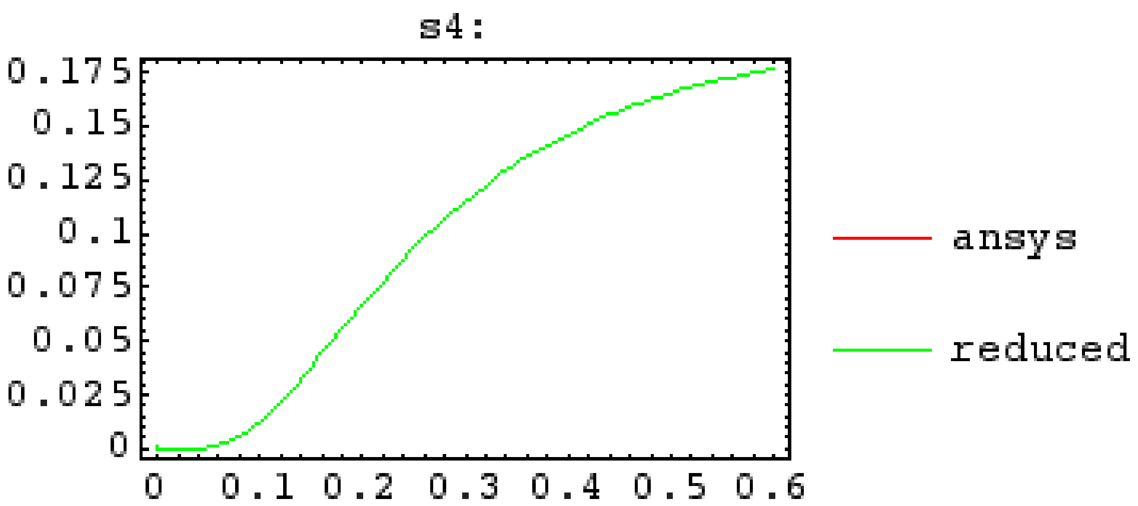


Fig. 5. V0K1. Concentration over sensor 4 as a function of time. Red line is for ANSYS simulation (58641), green line is for the reduced model (30 equations).

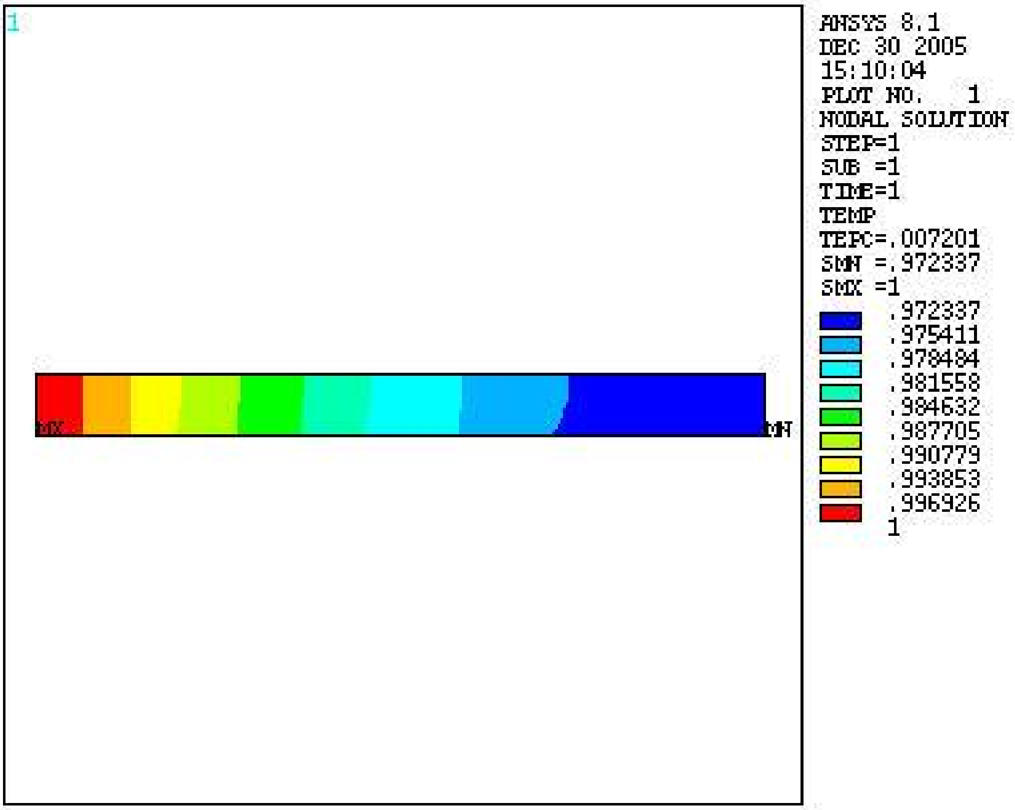


Fig. 6. V0K001. Contour plot for the stationary solution. **Color map is different from other similar pictures.**

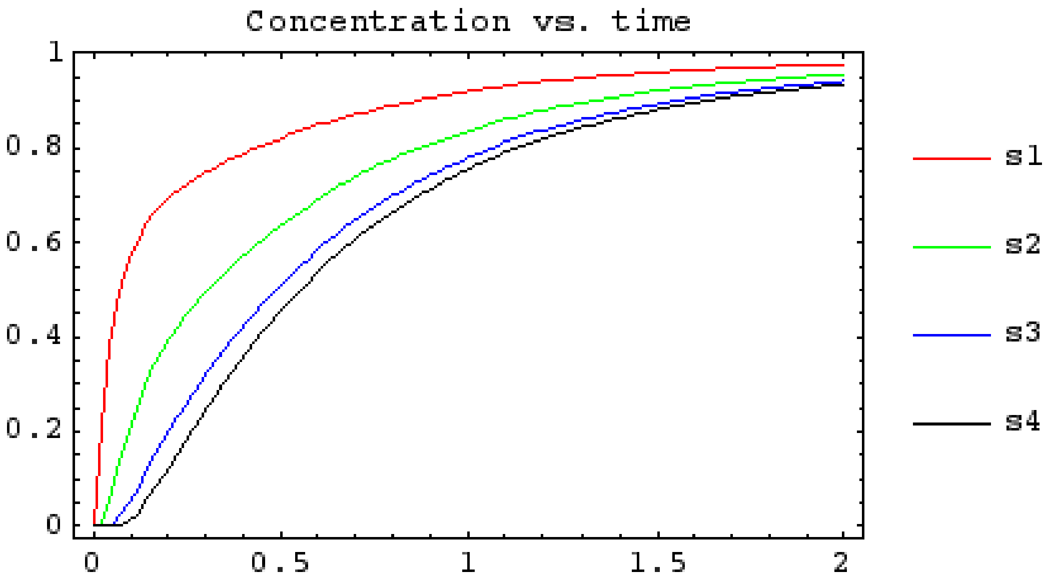


Fig. 7. V0K001. Concentration over different sensors as a function of time.

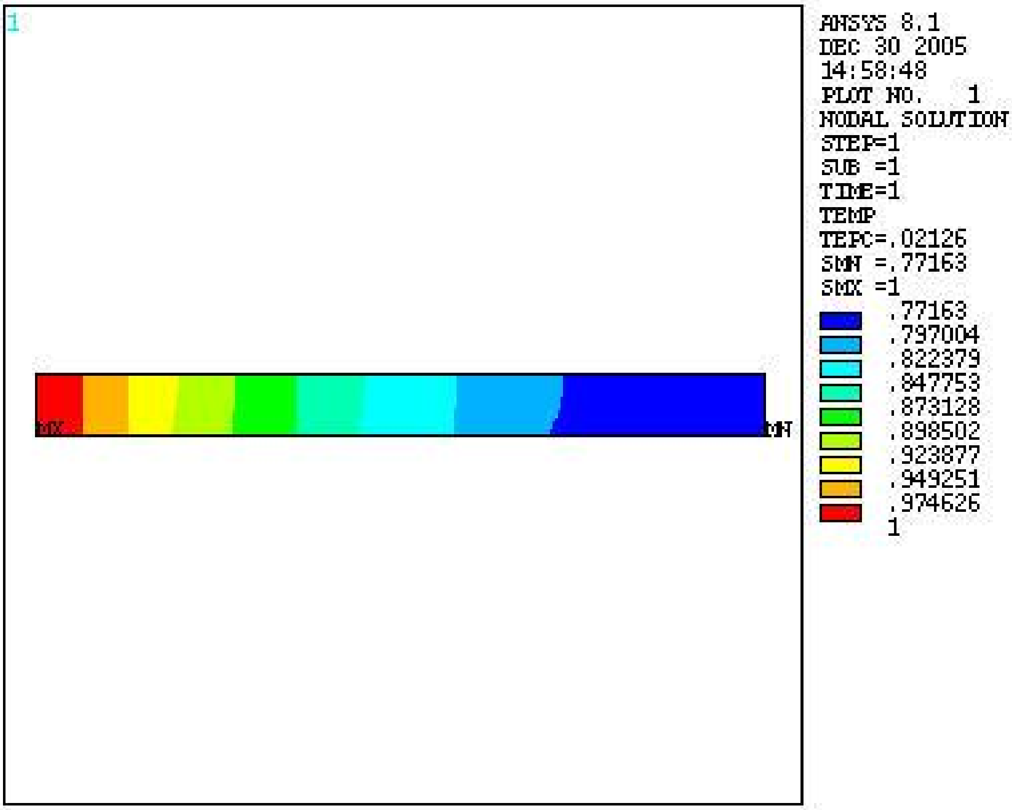


Fig. 8. V0K01. Contour plot for the stationary solution. **Color map is different from other similar pictures.**

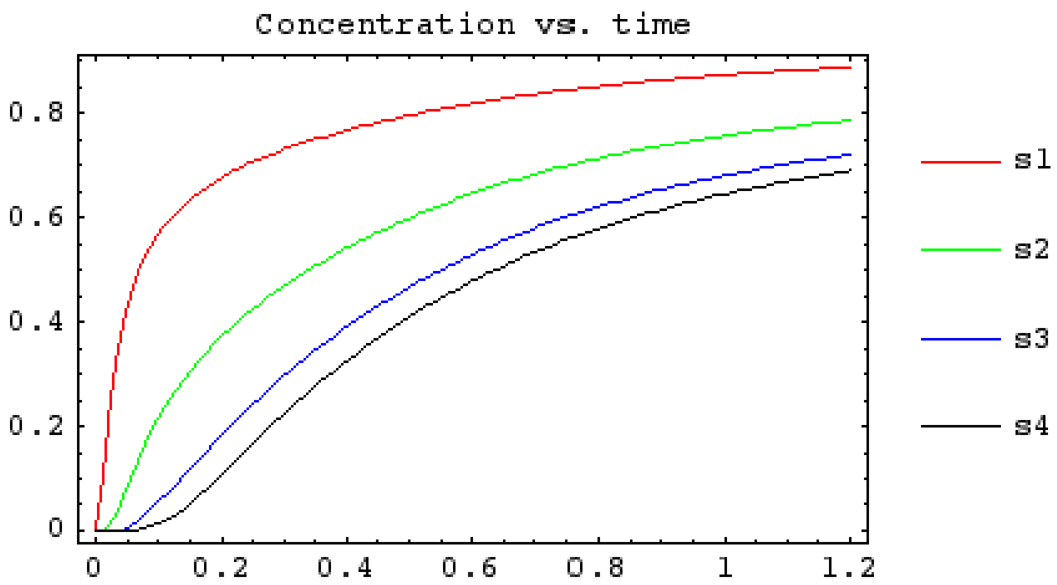


Fig. 9. V0K01. Concentration over different sensors as a function of time.

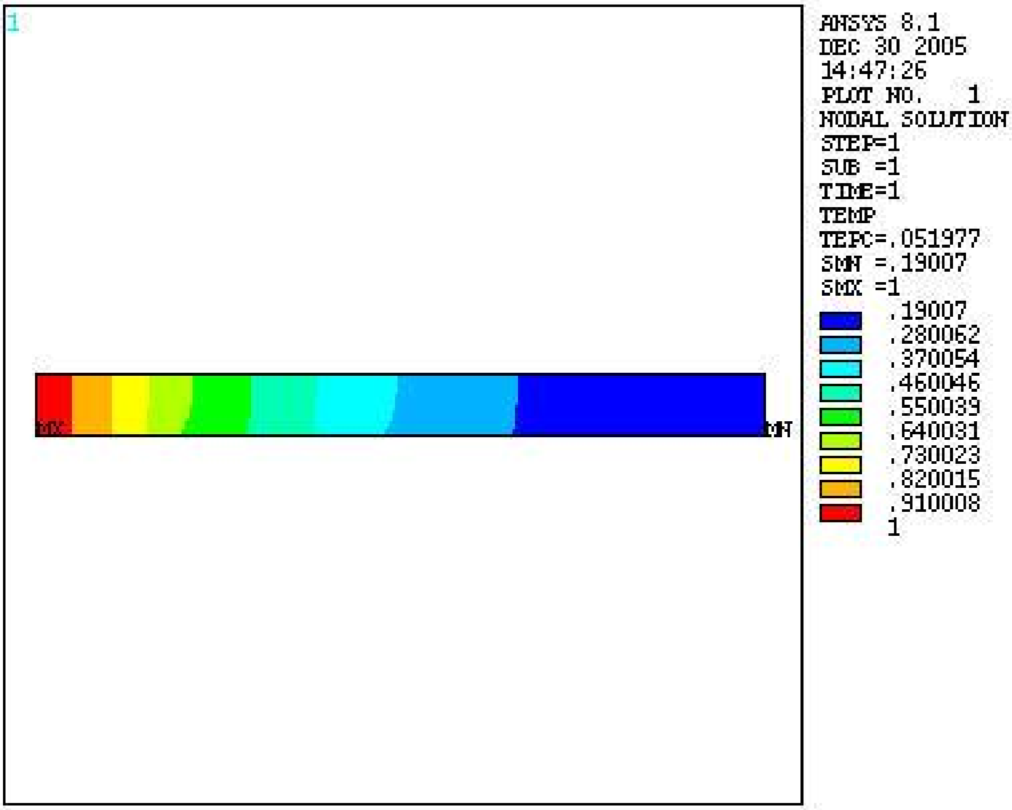


Fig. 10. V0K1. Contour plot for the stationary solution when. **Color map is different from other similar pictures.**

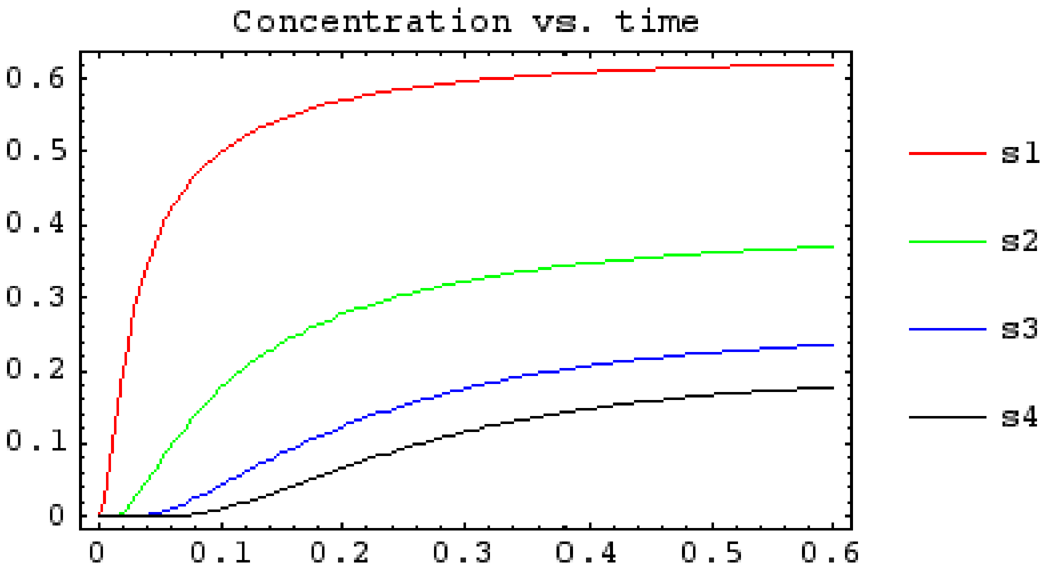


Fig. 11. V0K1. Concentration over different sensors as a function of time.

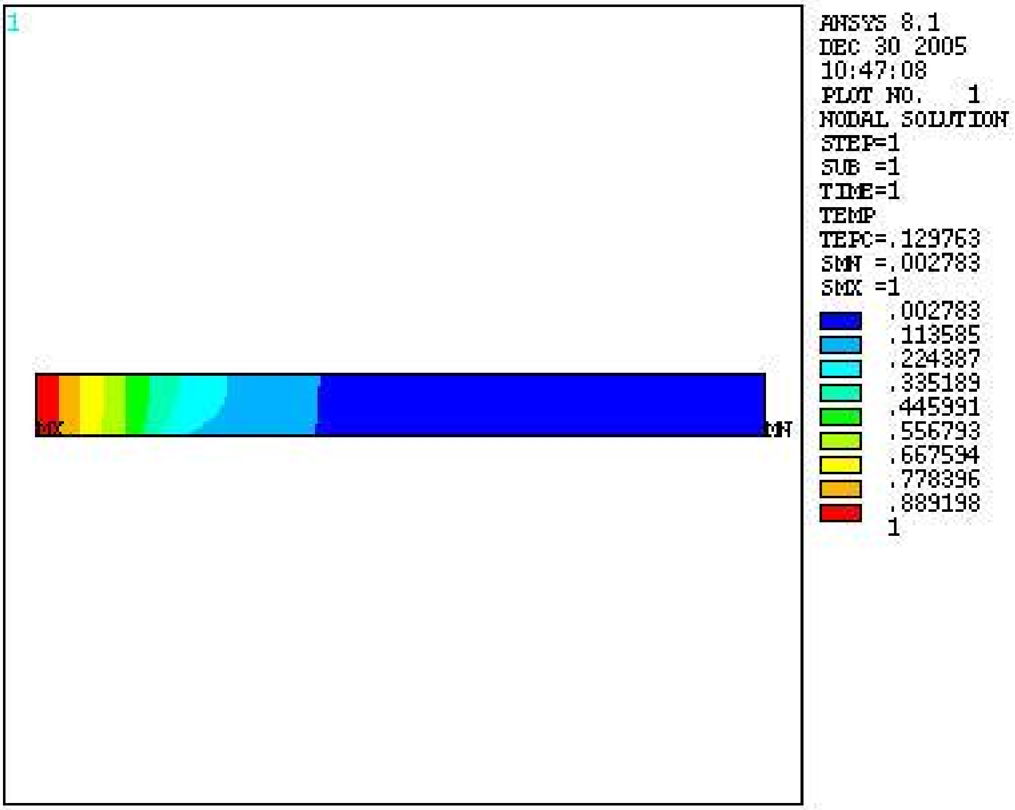


Fig. 12. V0K10. Contour plot for the stationary solution. **Color map is different from other similar pictures.**

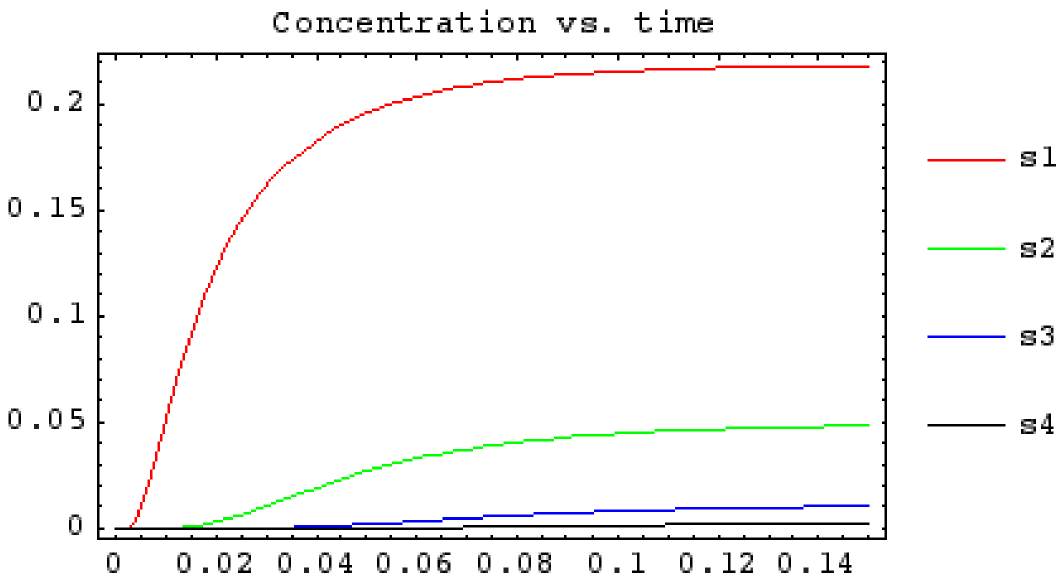


Fig. 13. V0K10. Concentration over different sensors as a function of time.

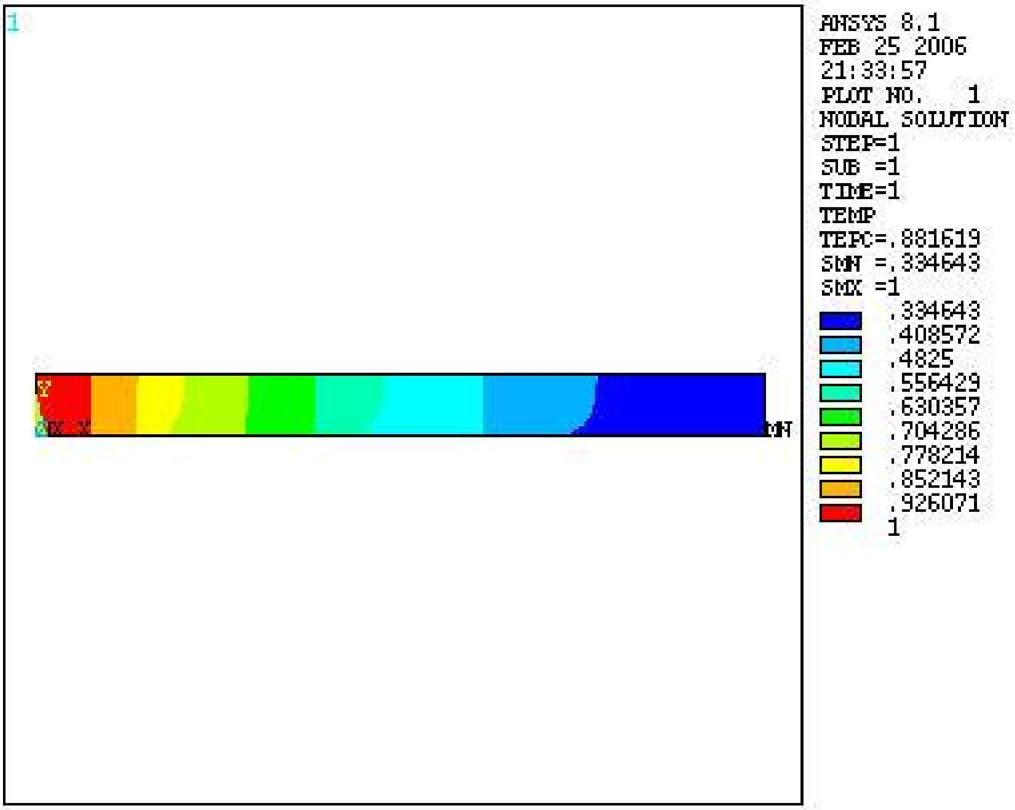


Fig. 14. V2K1. Contour plot for the stationary solution. **Color map is different from other similar pictures.**

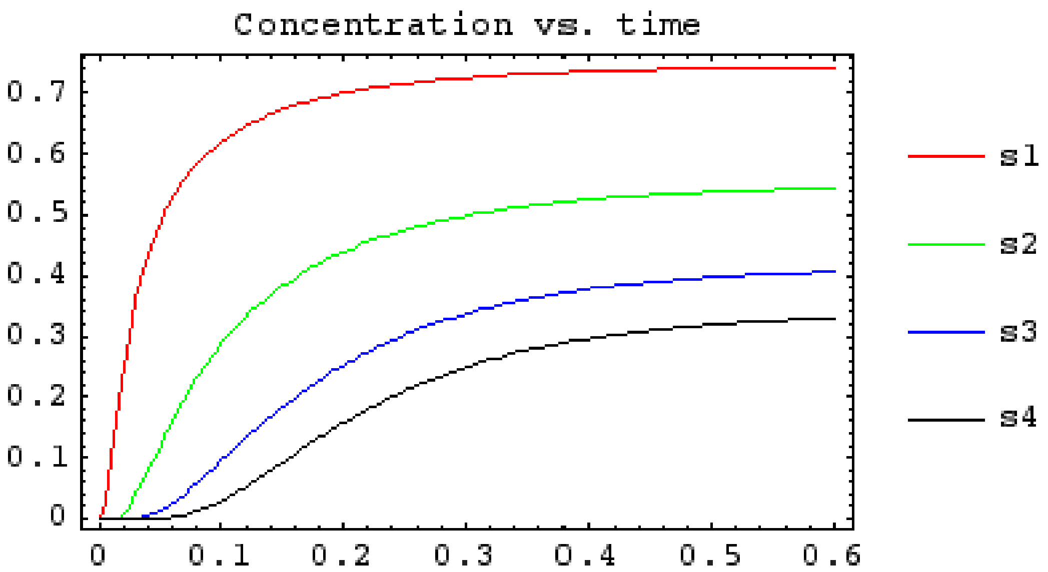


Fig. 15. V2K1. Concentration over different sensors as a function of time.

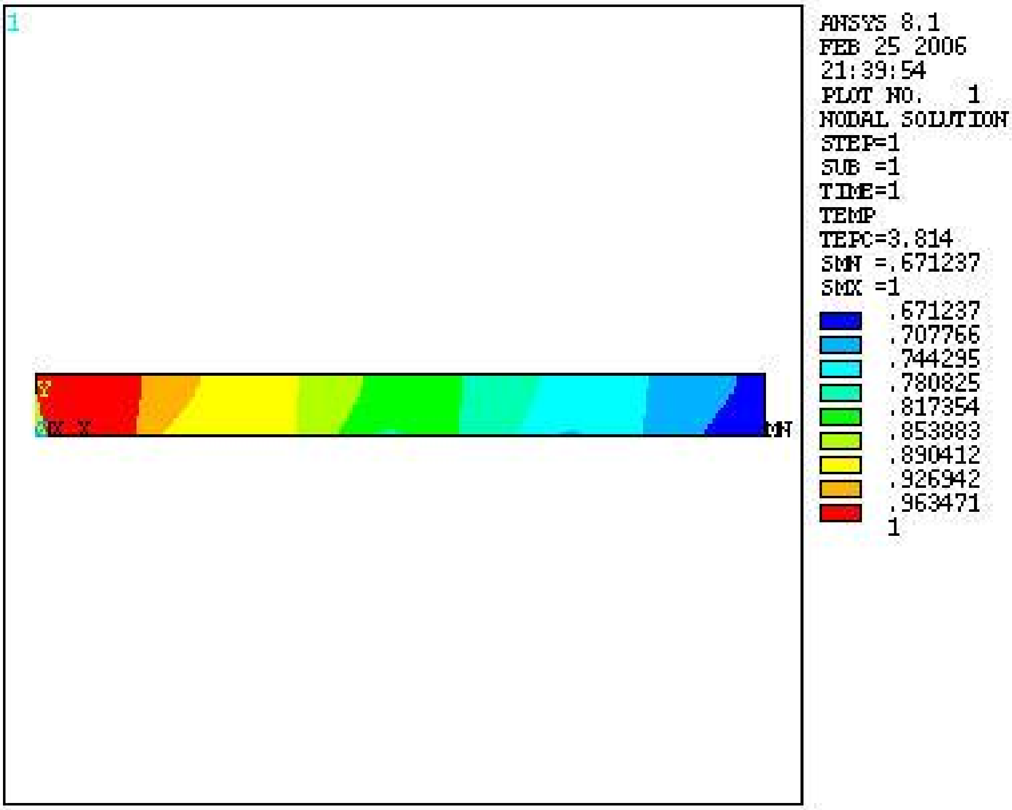


Fig. 16. V10K1. Contour plot for the stationary solution. **Color map is different from other similar pictures.**

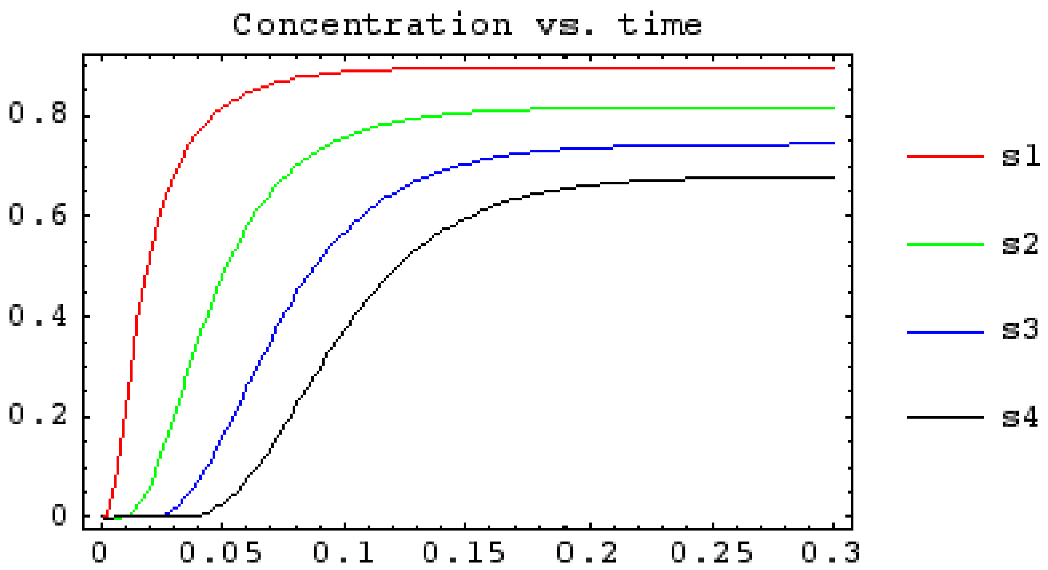


Fig. 17. V10K1. Concentration over different sensors as a function of time.

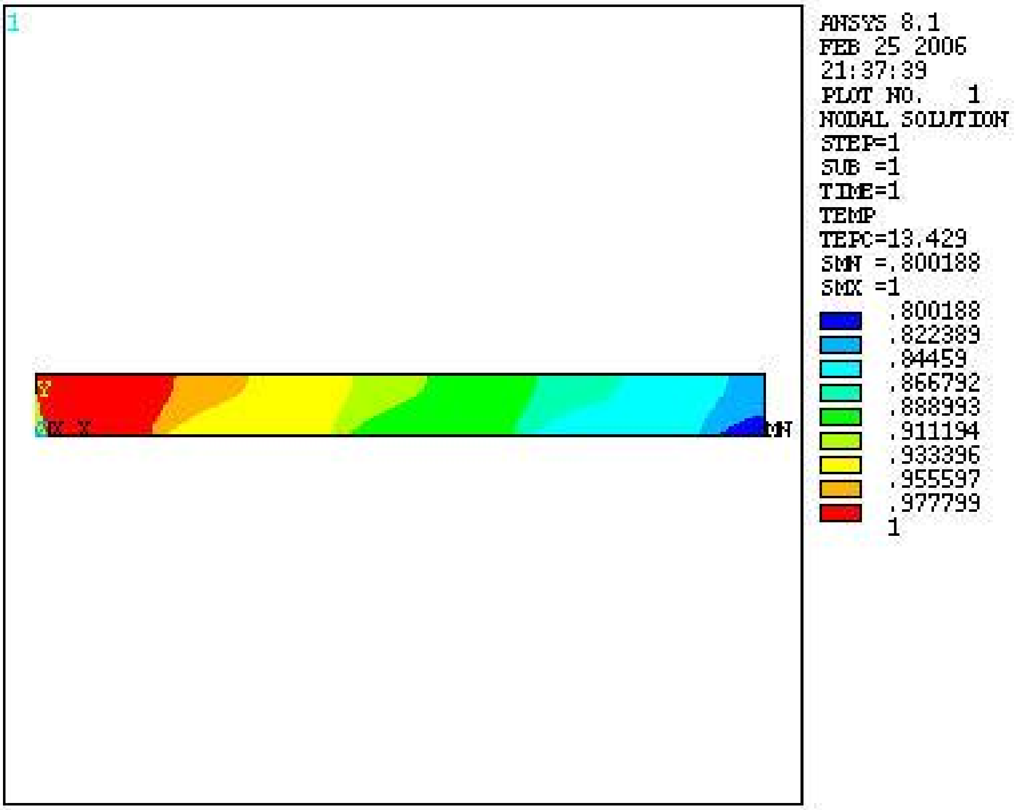


Fig. 18. V20K1. Contour plot for the stationary solution. **Color map is different from other similar pictures.**

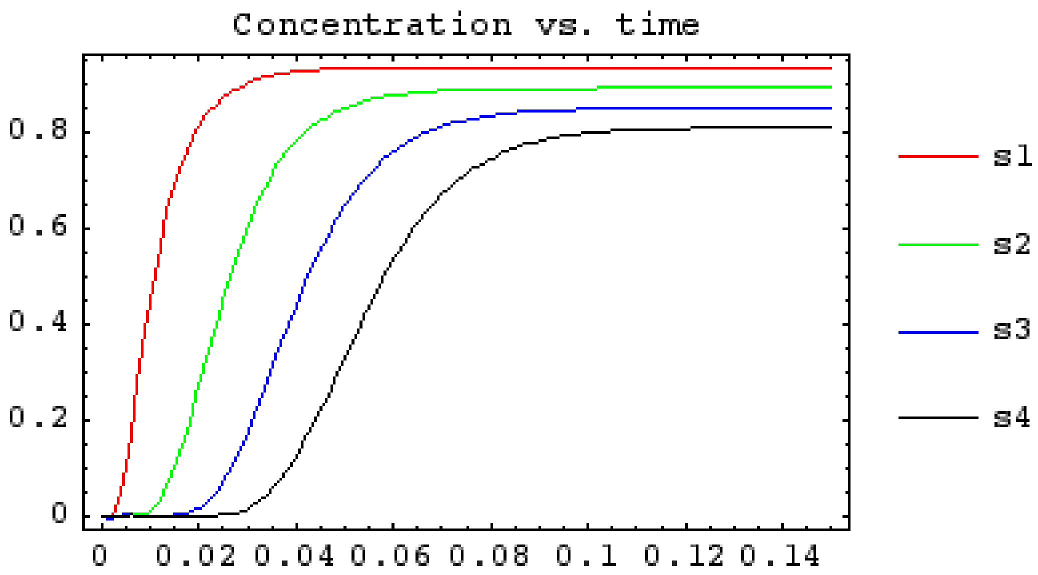


Fig. 19. V20K1. Concentration over different sensors as a function of time.