

Possibilities of using Neural Networks to Blood Flow Modelling

Katarína Buzáková¹^a, Katarína Bachratá¹^b, Hynek Bachratý¹^c and Michal Chovanec²

¹Department of Software Technology, Faculty of Management Science and Informatics, University of Žilina, Žilina, Slovakia

²Tachyum, s.r.o., Bratislava, Slovakia

Keywords: Convolutional Neural Networks, Microfluidic Devices, Red Blood Cells Trajectory Prediction.

Abstract: Computer simulation of the flow of blood or other fluid is beneficial to reduce the variety of costs necessary for biological experiments in microfluidics. It turns out, that as biological experiments, even the simulations have limitations. However, data from both types of experiments can be further processed by machine learning methods in order to improve them and thus contribute to the optimization of microfluidic devices. This article describes the possibilities of using neural networks to blood flow modelling. In this paper, we focus mainly on the prediction of red blood cells movement. We propose other possibilities of using neural networks with regard to the needs of further research in simulation modelling.


1 INTRODUCTION


Nowadays, the blood flow in microfluidic devices is studied in many biological and medical researches, see (Chen et al., 2012; Guo et al., 2017) for example. The aim of the Cell in Fluid, Biomedical Modelling & Computation Group is to optimize microfluidic devices to capture and sort cancer cells from other solid components of blood. The reason is to use these devices for early diagnosis of cancer from a blood sample. The starting point of our research group's work a few years ago was that the design of microfluidic devices and the optimization of their performance by real testing puts high demands on time, cost and equipment. The solution was to create an extensive simulation model in which experiments can be performed in silico. For the description of the simulation model see (Cimrák et al., 2012; Cimrák et al., 2014; Cimrák and Jančígová, 2018).


Current development of our work has resulted in situations where the simulations are becoming too complex and demanding in terms of time and computational power. This complexity results in particular from the geometry and the size of the simulated device, the number of modelled RBCs, the quantity of watched and recorded measurements and also the duration of the simulation. One simulation run often several days or weeks. This is restrictive for repeating

or expanding simulation experiments with the same or slightly altered parameters. On the other hand, each of the simulations contains a huge amount of output data, of which often only a small part is needed to evaluate the specific phenomenon under investigation. From studies (Bachratá et al., 2017a; Bachratá et al., 2017b; Bachratý et al., 2017; Bachratý et al., 2018) turn out that these output data can be comprehensively described and characterized the course of individual experiments using various statistical methods. We also used statistical methods to compare and evaluate the quality of simulation experiments. Because we have large output data from simulations and neural networks can find hidden features in the data, this led us to the idea of complex processing of the output data of simulation experiments using neural networks. This should allow us to use them to extend and obtain further results without the need to perform new simulations. We do not use data from video recordings of biological experiments because they do not provide sufficiently large and accurate data. In our research, we decided to use convolutional neural networks because they can successfully capture the spatial and temporal dependence in the image using appropriate filters. In this case, the images represent the locations of the cells in the channel in the individual time steps of the simulation.

This is also indicated by studies in other fields using machine learning methods, where simulations are significantly limited, see (Exl et al., 2019; Gusenbauer et al., 2020).

^a  <https://orcid.org/0000-0001-7615-0038>

^b  <https://orcid.org/0000-0002-5510-5585>

^c  <https://orcid.org/0000-0003-1378-488X>

1.1 Machine Learning for Blood Flow Modelling

In this section, we introduce our results of using machine learning to simulate the blood flow. Since red blood cells (RBCs) form a major part of haematocrit, the correct modelling of their behaviour is decisive in these simulations. The first results of machine learning approach were presented in (Bachratý et al., 2018). The foundation of this study is the use of extensive and detailed data outputs of simulations describing RBC trajectories in the examined channel. These were used as a training and testing input for learning algorithms that created radial basis function network based on Kohonen's self-organized maps. In (Chovanec et al., 2019; Chovanec et al., 2019), convolutional neural networks (CNN) predict the RBC center velocity vector in the channel. This allow us to:

- virtually extend or artificially create new RBC trajectories,
- estimate the impact of RBC motion on cancer cell behaviour at all examined points in the channel points in the channel,
- improve RBC tracking when processing videos of real experiments.

These papers describe ways of CNN learning, the accuracy of trajectory predictions and their dependence on neural network architecture, type of input parameters and methodology of verification and selection of appropriate neural network experiments.

1.2 Prediction of Red Blood Cells Trajectory

Here we describe how neural networks predict red blood cells movement. In (Chovanec et al., 2019) we created a framework for predicting RBCs trajectories in microfluidic channels using CNN. (In what follows we will call it the prediction model and neural networks (NN) experiments as prediction experiments.) In this model, the network learns to predict the velocity vector of a cell's center from the temporal sequence of its previous positions. This information comes from the simulation outputs. More information about simulation outputs and their processing for the model are in the work (Chovanec et al., 2019). The velocity of the cell in the microchannel is affected by various factors, for example the previous movement of the cell, the motion of other particles, the topology of the channel, the speed of the liquid which is different in the channel slits than in the areas free

from obstacles. For the trajectory prediction itself, we determine cells positions from their predicted velocities. By repeating this procedure, predicted positions are determined from predicted velocities for all time steps. Finally, we obtain the predicted trajectories from the initial trajectories of all modelled cells. (Figure 1). Note that the obtained RBC trajectory is a set of discrete points, which represents positions of the cell's center in all time steps of the prediction experiment.

2 DATASETS

Datasets for the prediction experiments come from simulations. As the output of a simulation experiment, we obtain miscellaneous characteristics about cells. From these data, we extract the coordinates of the centers of the cells and their velocity vectors. Then we use this information as an entry for the prediction experiment.

For a given prediction experiment, we use data from simulations, which vary in the initial seeding of RBCs. Hence, the cell trajectories in these experiments are different from each other. However, there should be similarities in the trajectories, since all other setups of these simulations are the same. It includes elastic parameters of cells, fluid parameters and geometry of microchannel. These data are divided to training and testing sets for neural network. The training set consist of extracted data from one simulation experiment and data for the testing set comes from the second simulation experiment output.

2.1 Simulation Experiment Designs

Simulation experiments use Open-source software ESPResSo (Arnold et al., 2013). The fluid is modelled using the Lattice-Boltzmann method, see (Ahlrichs and Dünweg, 1998). RBCs and other elastic objects are immersed in the fluid (Cimrák et al., 2014).

For prediction experiments, we use simulations of blood flow in two microfluidic devices with different topology described below. The RBC model is the triangulation of its surface. Both simulations use RBC model with 374 nodes and the size of RBC in a relaxed state is $7,8\mu\text{m} \times 7,8\mu\text{m} \times 2,56\mu\text{m}$. The direction of the blood flow is from left to right along the horizontal x -axis. The channels are periodic (in this direction) in a sense that the cell which leaves the simulation channel at one end reenters the channel on the other end.

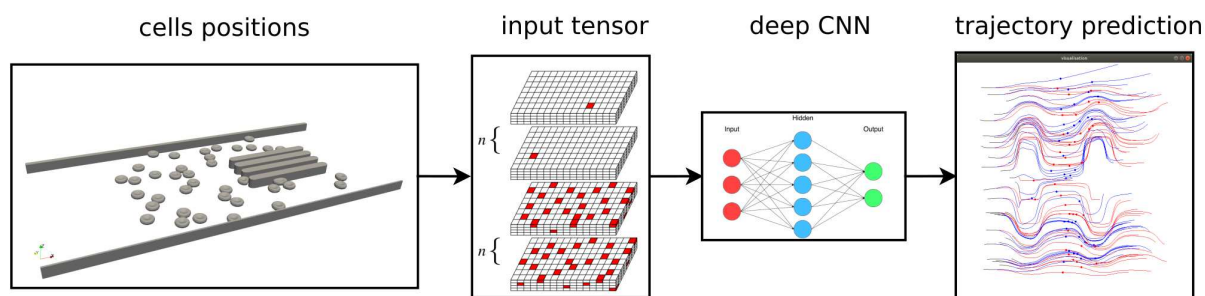


Figure 1: Red blood cells trajectory prediction using CNN.

2.1.1 Channel with Narrow Slits

The simulation of this channel is based on the laboratory experiment described in (Tsai et al., 2016). This paper studies the correlation between RBC velocity and its deformation in narrowings of the microfluidic channel formed by obstacles. The internal dimensions of the simulation channel are $208\mu\text{m} \times 98\mu\text{m} \times 3.5\mu\text{m}$. There are 4 longitudinal obstacles in the channel, see Figure 2. To fit the haematocrit used in the laboratory experiment, the number of blood cells in the simulation experiment is 38. At the beginning of the simulation, all cells were in the left part of the channel. A more detailed description of the simulation can be found in (Kovalčíková et al., 2019). We will refer to the dataset and the channel from this simulation as dataset A and simulation channel A.

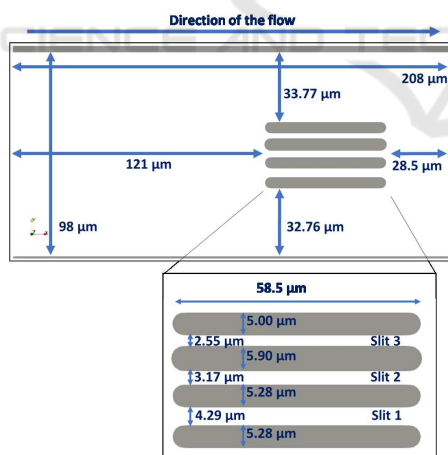


Figure 2: Simulation channel A with narrow slits.

2.1.2 Channel with Cylindrical Obstacles

Simulations of the blood flow in the channel with cylindrical obstacles were an important tool to design and test most of the RBC characteristics used in the simulation model, for example the absolute cell velocity, the position, the slope and the rotation of RBC or a periodic behaviour in the channel, see (Bachratá

et al., 2017b; Bachratý et al., 2017; Bachratá et al., 2017a). The channel is shown in Figure 3. It is a cuboid of size $100\mu\text{m} \times 50\mu\text{m} \times 30\mu\text{m}$ and it contains 5 cylindrical obstacles with a diameter of $5\mu\text{m}$. An element of each cylinder is parallel to the z -axis and the height of each cylinder is equal to the height of the simulation channel. The amount of RBCs in experiments is 100. The initial seeding of the cells is random in the whole space of the simulation channel. We will refer to the dataset and the channel from this simulation as dataset B and simulation channel B.

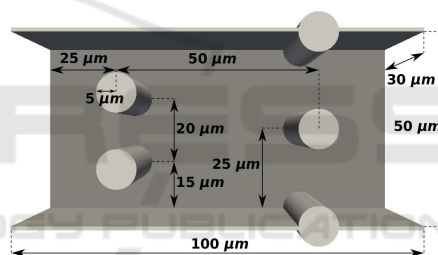


Figure 3: Simulation channel B with cylindrical obstacles.

The calibration of elastic coefficients of the cell's model was made by stretching experiment. The detailed explanation of the calibrating process is explained in (Tóthová et al., 2015). The obtained elastic parameters for simulation models are summarised in Table 1. Another type of parameters which had to be set separately for each cell model are the interaction parameters between cells. These parameters prevent cell collisions. They are summarised in Table 2. The numerical parameters of simulation liquid are depicted in Table 3.

3 NEURAL NETWORK MODEL

In this section we first introduce neural network input. Then we describe the networks architectures, hyperparameters that are used and other details.

Table 1: Elastic parameters of the cell model used in simulation experiments, established by simulation of stretching experiment.

	simulation A		simulation B	
	LB units	SI units	LB units	SI units
Radius	$3.91Lm$	$3.91 \cdot 10^{-6}m$	$3.91Lm$	$3.91 \cdot 10^{-6}m$
Stretching coefficient k_s	$6 \cdot 10^{-3}LN/Lm$	$6 \cdot 10^{-6}N/m$	$5 \cdot 10^{-3}LN/Lm$	$5 \cdot 10^{-6}N/m$
Bending coefficient k_b	$8 \cdot 10^{-3}LNLm$	$8 \cdot 10^{-18}Nm$	$3 \cdot 10^{-3}LNLm$	$3 \cdot 10^{-18}Nm$
Coefficient of local area conservation k_{al}	$1 \cdot 10^{-3}LN/Lm$	$1 \cdot 10^{-6}N/m$	$2 \cdot 10^{-2}LN/Lm$	$2 \cdot 10^{-4}N/m$
Coefficient of global area conservation k_{ag}	$0.9LN/Lm$	$9 \cdot 10^{-4}N/m$	$0.7LN/Lm$	$7 \cdot 10^{-4}N/m$
Coefficient of volume conservation k_v	$0.5LN/Lm^2$	$5 \cdot 10^2N/m^2$	$0.9LN/Lm^2$	$9 \cdot 10^2N/m^2$
Membrane viscosity	$0Lm^2/Ls$	$0m^2/s$	$0Lm^2/Ls$	$0m^2/s$

Table 2: Parameters of inter-cellular interactions.

	simulation A		simulation B	
	LB units	SI units	LB units	SI units
a	$2 \cdot 10^{-3}(-)$	$2 \cdot 10^{-3}(-)$	$1 \cdot 10^{-3}(-)$	$1 \cdot 10^{-3}(-)$
n	$1.5Lm$	$1.5 \cdot 10^{-6}m$	$1.2Lm$	$1.2 \cdot 10^{-6}m$
cutoff	$0.4(-)$	$0.4(-)$	$0.5(-)$	$0.5(-)$
offset	$0(-)$	$0(-)$	$0(-)$	$0(-)$

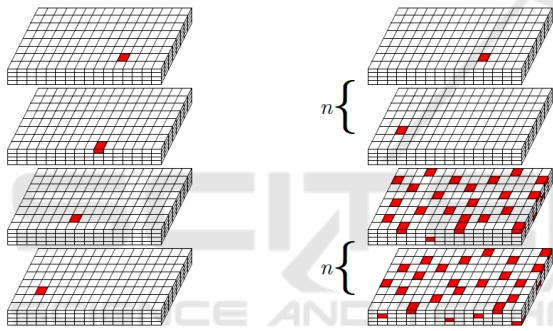


Figure 4: Input tensor based on spatial discretization of the channel.

3.1 CNN Input

Inputs to the neural network are tensors. They are based on the discretization of the simulation channel to a three-dimensional rectangular network in which the position and movement of cells is described by their occupation evolving over time (Figure 4). For more detailed description of the input see (Chovanec et al., 2019; Chovanec et al., 2019). In (Chovanec et al., 2019) we compared this input format with another input format based on the numerical expression of the cell's center positions. For the input based on discretization, the maximum error of the experiments performed was 3.26%. For the other input type, the error of the performed experiments was in the range of approximately 10% to 40%.

In (Chovanec et al., 2019) we tested the impact of various modifications of CNN input on the accuracy of the prediction experiment. In both of these studies, we used the dataset A from the simulations of the

channel with narrow slits.

3.2 Networks Architectures

In prediction experiments, we use 3 different CNN architectures *net 0*, *net 1* and *net 2* for both dataset A and dataset B. We chose these architectures based on the accuracy of NN experiments in our previous studies (Chovanec et al., 2019; Chovanec et al., 2019) and also with regard to the latest improvements in the field of neural networks. In these models we use the activation function ELU (Clevert et al., 2015). For dataset B, we also use *net 6*, which we used for dataset A in (Chovanec et al., 2019). Networks architectures are shown in Figure 4. In networks *net 0* and *net 1*, there are alternating convolution layers with max pooling layers. At the end, there are 2 fully connected layers with 256 and 3 neurons, respectively. Network *net 1* differs from network *net 0* by adding spatial attention layers (Vaswani et al., 2017). In network *net 2*, compared to *net 1*, 4 dense blocks (Huang et al., 2017) are used instead of 3×3 convolution kernels.

3.2.1 Hyperparameters

All CNN architectures have the following hyperparameters: weights are initialized using *xavier* (Glorot and Bengio, 2010); the bias is set to 0; minibatch size is 32 (Li et al., 2014).

Regularization parameters are: dropout = $2 \cdot 10^{-2}$; $L_1 = L_2 = 1 \cdot 10^{-6}$. We also use feature pooling with 1×1 kernels to prevent overfitting.

Table 3: The numerical parameters of simulation liquid.

	simulation A		simulation B	
	LB units	SI units	LB units	SI units
Density	1Lkg/Lm ³	1*10 ³ kg/m ³	1.025Lkg/Lm ³	1.025*10 ³ kg/m ³
Kinematic viscosity	1Lm ² /Ls	1*10 ⁻⁶ m ² /s	1.3Lm ² /Ls	1.3*10 ⁻⁶ m ² /s
Friction coeff.	1.15(-)	1.15(-)	1.41(-)	1.41(-)

Table 4: Networks architectures.

layer	net 0	net 1	net 2	net 6
0	conv 3x3x32	conv 3x3x32	dense conv 3x3x8	dense conv 3x3x8
1	max pooling 2x2x1	spatial attention	dense conv 3x3x8	dense conv 3x3x8
2	conv 3x3x32	max pooling 2x2x1	dense conv 3x3x8	dense conv 3x3x8
3	max pooling 2x2x1	conv 3x3x32	dense conv 3x3x8	dense conv 3x3x8
4	conv 3x3x64	spatial attention	conv 1x1x32	conv 1x1x16
5	max pooling 2x2x1	max pooling 2x2x1	spatial attention	dense conv 3x3x8
6	conv 3x3x64	conv 3x3x64	max pooling 2x2x1	dense conv 3x3x8
7	max pooling 2x2x1	spatial attention	dense conv 3x3x8	dense conv 3x3x8
8	fc 256	max pooling 2x2x1	dense conv 3x3x8	dense conv 3x3x8
9	fc 3	conv 3x3x64	dense conv 3x3x8	conv 1x1x16
10		spatial attention	dense conv 3x3x8	dense conv 3x3x8
11		max pooling 2x2x1	conv 1x1x32	dense conv 3x3x8
12		fc 256	spatial attention	dense conv 3x3x8
13		fc 3	max pooling 2x2x1	dense conv 3x3x8
14			dense conv 3x3x8	conv 1x1x32
15			dense conv 3x3x8	fc 3
16			dense conv 3x3x8	
17			dense conv 3x3x8	
18			conv 1x1x64	
19			spatial attention	
20			max pooling 2x2x1	
21			dense conv 3x3x8	
22			dense conv 3x3x8	
23			dense conv 3x3x8	
24			dense conv 3x3x8	
25			conv 1x1x64	
26			spatial attention	
27			max pooling 2x2x1	
28			fc 256	
29			fc 3	

3.3 Networks Training

In all prediction experiments, we use training algorithm ADAM (Kingma and Ba, 2015) and learning

rate is set to $2 \cdot 10^{-4}$. We minimize the loss function:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n},$$

where y_i are target values, \hat{y}_i are predicted values and n is the dataset size.

For both datasets and for the nets *net 0*, *net 1* and *net 2*, the networks did not trained enough. We suppose, it is due to the fully connected layer with 256 neurons. For *net 6* and for dataset B, the value of loss function was significantly lower at the end of the training. Note, that we used this network and networks with similar architectures in (Chovanec et al., 2019). Thus, we suspected good training results for this network. Moreover, there is bigger decrease of loss function from the beginning to the end of the training than for the dataset A trained on *net 6* in (Chovanec et al., 2019).

4 DATASET DAMAGING

In this section, we propose a method of using neural networks to detect and eliminate possible errors and inaccuracies of simulation experiments. During our simulation experiments, we encountered inaccuracies in the simulation outputs several times. These errors may be due to the improper calibration of the simulation parameters, numerical errors caused by computational algorithms or measurement errors with respect to data obtained by processing laboratory experiment records. For example, in the simulation model, inaccuracies are encountered in the calculation of velocity of cells center and nodes. Figure 5 shows graphs comparing the y - and z -coordinates of the cell's center velocity obtained from the simulation, and the same coordinates computed from the cell's center positions (determined by the simulation). This simulation was used in (Bachratý et al., 2018). This could serve as a tool to correct the simulation experiments, since to correct the simulation by using a prediction experiments is faster then to run the simulation again with slightly altered setup.

Our aim is to find out at what extent of the damage we can still predict the movement of blood cells with sufficient accuracy. To do this we intentionally damage a part of the training data. The data damage can be described using three parameters:

1. type,
2. percentage,
3. data corruption level.

The parameter type says what kind of data is damaged. In our prediction experiments, it can be cell's center positions or velocities. The second parameter determines the percentage of damaged data. Finally, data corruption level is the degree of inaccuracy of a damaged value compared to the actual value.

For the percentage $p\%$ of the cell's positions damage with the $d\%$ data corruption level, we randomly damage $p\%$ of positions as follows:

$$c_{damaged} = (1 - 0.01 \cdot d)c + 0.01 \cdot d \cdot rand(-1, 1),$$

where c corresponds to the value of individual coordinates x , y and z normalized to the interval $(0, 1)$ as is common in neural networks, and $rand(-1, 1)$ is a random value from the range $(-1, 1)$.

Damaged value in coordinate c is then

$$c_{damaged} = \begin{cases} c_{damaged}, & \text{if } c_{damaged} \in (0, 1), \\ 0, & \text{if } c_{damaged} < 0, \\ 1, & \text{if } c_{damaged} > 1. \end{cases}$$

5 PREDICTION BASED ON LOCAL AREA INFORMATION

The study (Chovanec et al., 2019) shows that the accuracy of the prediction experiment depends on the input data format. A format based on the discretization of the channel to a three-dimensional rectangular network (see 3.1) seems to be clearly more appropriate.

The discretization of the channel affects the size of the input tensor for the neural network. It is limited in the z -axis by the depth of the neural network. Due to the computational complexity, we can only use discretizations smaller than 9 in this direction, which is not sufficient for deeper microfluidic channels. Instead, we can only look at the close local area of the monitored RBC, and use a finer discretization of the situation there (see Figure 6, the cell of an interest is in the red rectangle, and for this rectangle we use a finer discretization). This discretization is three-dimensional, captures the position of the cell of an interest and other objects in the neighborhood, that is, other cells, channel walls and obstacles. It describes positions of cells more precisely, hence it could be used to predict other characteristics of cells movement, such as the rotation and slope. One of the most important purposes of this local prediction model should be the prediction of cells movement in channels with different topology. It means that topology of a channel used for training is different from the topology of a simulation channel used in testing. This is of great interest for simulating process because simulations of large channels are computationally very difficult or even impossible to run. An interesting question is if a network trained on these local data will be able to predict the RBC behaviour globally across the channel with sufficient accuracy.

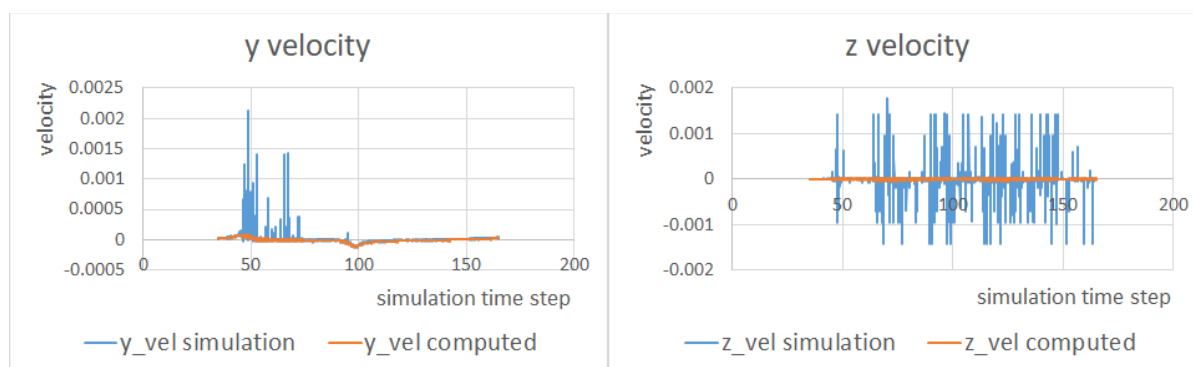


Figure 5: Cell center velocity at y and z coordinates. The blue curve shows the speed calculated by the simulation. The orange line represents the speed computed from the cell center positions.

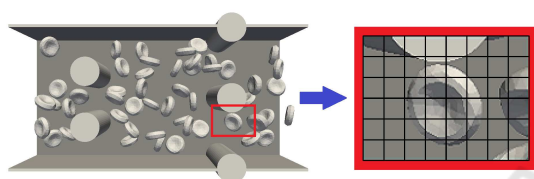


Figure 6: Discretization of local area of the simulation channel.

6 CONCLUSIONS

This paper presents the possibilities of the use of neural networks to optimize simulation and biological experiments of blood flow in microfluidic devices. It deals mainly with the prediction of the trajectory of red blood cells, which can significantly help in the tracing of red blood cells from video recordings of real experiments. This is also useful in simulating blood flow where simulations are limited by computational complexity. Furthermore, we point out the ways of improving prediction experiments and propose their further use. This would be especially useful for predicting the movement of blood cells in devices, for which, due to their topology, current simulation experiments cannot be performed. Further, this can be used to detect possible inaccuracies of simulation outputs, and to investigate other characteristics of red blood cells needed for the proper simulation model.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic (contract No. VEGA 1/0643/17) and by Operational Program "Integrated Infrastructure" of the project "Integrated strategy in the development of personalized medicine of selected malignant tumor dis-

eases and its impact on life quality", ITMS code: 313011V446, co-financed by resources of European Regional Development Fund.

REFERENCES

- Ahlrichs, P. and Dünweg, B. (1998). Lattice-boltzmann simulation of polymer-solvent systems. *International Journal of Modern Physics C (IJMPC)*, 09(08):1429–1438.
- Arnold, A., Lenz, O., Kesselheim, S., Weeber, R., Fahrenberger, F., Roehm, D., Kosovan, P., and Holm, C. (2013). Espresso 3.1 — molecular dynamics software for coarse-grained models. In *Meshfree Methods for Partial Differential Equations VI*, volume 89.
- Bachratá, K., Bachratý, H., and Kovalčíková, K. (2017a). The sensitivity of the statistical characteristics to the selected parameters of the simulation model in the red blood cell flow simulations. In *2017 International Conference on Information and Digital Technologies (IDT)*, pages 344–349.
- Bachratá, K., Bachratý, H., and Slavík, M. (2017b). Statistics for comparison of simulations and experiments of flow of blood cells. *EPJ Web of Conferences*, 143:02002.
- Bachratý, H., Bachratá, K., Chovanec, M., Kajánek, F., Smiešková, M., and Slavík, M. (2018). *Simulation of Blood Flow in Microfluidic Devices for Analysing of Video from Real Experiments*, pages 279–289. Springer International Publishing.
- Bachratý, H., Kovalčíková, K., Bachratá, K., and Slavík, M. (2017). Methods of exploring the red blood cells rotation during the simulations in devices with periodic topology. In *2017 International Conference on Information and Digital Technologies (IDT)*, pages 36–46. IEEE.
- Chen, J., Li, J., and Sun, Y. (2012). Microfluidic approaches for cancer cell detection, characterization, and separation. *Lab on a Chip*, 12(10):1753–1767.
- Chovanec, M., Bachratý, H., Jaseňáková, K., and Bachratá, K. (2019). Influence of cnn input modification for red blood cells trajectory prediction in blood

- flow. In *2019 IEEE 15th International Scientific Conference on Informatics*, pages 469–476.
- Chovanec, M., Bachratý, H., Jasenčáková, K., and Bachratá, K. (2019). Convolutional neural networks for red blood cell trajectory prediction in simulation of blood flow. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 284–296. Springer.
- Cimrák, I., Gusenbauer, M., and Jančígová, I. (2014). An espresso implementation of elastic objects immersed in a fluid. *Computer Physics Communications*, 185(3):900–907.
- Cimrák, I., Gusenbauer, M., and Schrefl, T. (2012). Modelling and simulation of processes in microfluidic devices for biomedical applications. *Computers & Mathematics with Applications*, 64(3):278–288.
- Cimrák, I. and Jančígová, I. (2018). *Computational Blood Cell Mechanics: Road Towards Models and Biomedical Applications*. CRC Press.
- Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *Under Review of ICLR2016 (1997)*.
- Exl, L., Mauser, N., Schrefl, T., and Suess, D. (2019). Learning time-stepping by nonlinear dimensionality reduction to predict magnetization dynamics.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256.
- Guo, Q., Duffy, S., Matthews, K., Islamzada, E., and Ma, H. (2017). Deformability based cell sorting using microfluidic ratchets enabling phenotypic separation of leukocytes directly from whole blood. *Scientific Reports*, 7.
- Gusenbauer, M., Oezelt, H., Fischbacher, J., Kovacs, A., Zhao, P., Woodcock, T., and Schrefl, T. (2020). Extracting local nucleation fields in permanent magnets using machine learning. *npj Computational Materials*, 6:1–10.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kovalčíková, K., Cimrák, I., Bachratá, K., and Bachratý, H. (2019). *Comparison of Numerical and Laboratory Experiment Examining Deformation of Red Blood Cell*, pages 75–86. Springer International Publishing.
- Li, M., Zhang, T., Chen, Y., and Smola, A. J. (2014). Efficient mini-batch training for stochastic optimization. In *KDD*, pages 661–670.
- Tóthová, R., Jančígová, I., and Bušík, M. (2015). Calibration of elastic coefficients for spring-network model of red blood cell. In *2015 International Conference on Information and Digital Technologies*, pages 376–380. IEEE.
- Tsai, C.-H., Tanaka, J., Kaneko, M., Horade, M., Ito, H., Taniguchi, T., Ohtani, T., and Sakata, Y. (2016). An on-chip rbc deformability checker significantly improves velocity-deformation correlation. *Micromachines*, 7.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.