A Web-Based Non-invasive Estimation of Fractional Flow Reserve (FFR): Models, Algorithms, and Application in Diagnostics

Yuri Vassilevski, Timur Gamilov, Alexander Danilov, German Kopytov, and Sergey Simakov

1 Introduction

Evaluation of stenosis severity is an important part of the decision-making process for coronary lesions treatment. Fractional flow reserve (FFR) is a golden standard for evaluating hemodynamic importance of coronary stenosis [1]. It is defined to be the ratio of the average pressure distal to the stenosis to the average aortic pressure, during maximum hyperemia. Computed (virtual) FFR [2–4] has emerged as an effective computational tool for non-invasive FFR evaluation. Virtual FFR evaluation provides a nontraumatic and cheap diagnostic tool for patients with the ischemic heart disease.

In this work we present a new web-based computational technology for non-invasive estimation of FFR based on patient-specific data. In Sect. 2 we briefly

Y. Vassilevski (✉)
INM RAS, Moscow, Russia

Sirius University of Science and Technology, Sochi, Russia

T. Gamilov · A. Danilov
Sechenov University, Moscow, Russia

Sirius University of Science and Technology, Sochi, Russia
e-mail: timaperm@yandex.ru

G. Kopytov
Baltic Federal University, Kaliningrad, Russia

INM RAS, Moscow, Russia
e-mail: gkopytov@yandex.ru

S. Simakov
MIPT, Dolgoprudny, Russia

Sirius University of Science and Technology, Sochi, Russia
e-mail: simakovss@ya.ru

R. P. Mondaini (ed.), *Trends in Biomathematics: Modeling Epidemiological, Neuronal, and Social Dynamics*, https://doi.org/10.1007/978-3-031-33050-6_18

present the 1D mathematical model of coronary haemodynamics. In Sect. 3 we describe algorithms for contrast enhanced CT (ceCT) data segmentation and construction of a 1D core network of individual coronary vasculature. For important details of the algorithms we refer to [5, 6]. In Sect. 4 we present our novel web-based software. It provides 3D visualization and a graphical user interface. We also discuss and describe a web protocol for efficient data exchange and processing. Finally, in Sect. 5 we demonstrate performance of our technology and discuss remaining issues and a future work.

2 Mathematical Model and Its Numerical Implementation

This section describes a coronary hemodynamics model which underlies our software. To calculate FFR, the model uses a 1D mesh on a vascular graph extracted from patient's CT scans. The model describes unsteady axisymmetric flows of Newtonian viscous incompressible fluid (blood) through a network of elastic tubes representing patient's vasculature. Below we present a brief summary, for details we refer to [3, 5, 7].

The flow in every vessel (graph edge) is described by the mass and momentum balance equations

$$\frac{\partial \mathbf{V}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{V})}{\partial x} = \mathbf{G}(\mathbf{V})\,, \tag{1}$$

$$\mathbf{V} = \begin{pmatrix} A \\ u \end{pmatrix},\ \mathbf{F}(\mathbf{V}) = \begin{pmatrix} Au \\ u^2/2 + p(A)/\rho \end{pmatrix},\ \mathbf{G}(\mathbf{V}) = \begin{pmatrix} 0 \\ -8\pi\mu\dfrac{u}{A} \end{pmatrix},$$

where t is the time, x is the distance along the vessel counted from the vessel's junction point, $\rho = 1.060\,\mathrm{kg/m^3}$ is the blood density, $A(t, x)$ is the vessel cross-section area, p is the blood pressure, $u(t, x)$ is the linear velocity averaged over the cross-section, and $\mu = 2.5\,\mathrm{mPa} \cdot s$ is the dynamic viscosity of the blood.

Relation between the pressure and the cross-sectional area is defined by the constitutive equation

$$p(A) = \rho_w c^2 \left(\exp\left(\frac{A}{\tilde{A}} - 1 \right) - 1 \right), \tag{2}$$

where $\rho_w = 1.060\,\mathrm{kg/m^3}$ is the density of the vessel wall material, c is the velocity of small disturbance propagation in the material of the vessel wall (velocity of the pulse wave propagation), and $\tilde{A}$ is the rest cross-sectional area of the vessel at zero transmural pressure and zero flow.

At the vessel's junction points, we impose the mass conservation condition and the total pressure continuity (Bernoulli's law):

$$\sum_{k=k_1,k_2,\ldots,k_M} \varepsilon_k A_k\left(t,\tilde{x}_k\right) u_k\left(t,\tilde{x}_k\right) = 0, \tag{3}$$

$$p_k\left(t,\tilde{x}_k\right) + \frac{\rho u^2\left(t,\tilde{x}_k\right)}{2} = p_{k+1}\left(t,\tilde{x}_{k+1}\right) + \frac{\rho u^2\left(t,\tilde{x}_{k+1}\right)}{2}, k = k_1, k_2, \ldots, k_{M-1}, \tag{4}$$

where k is the index of the vessel, M is the number of connected vessels, $\{k_1, \ldots, k_M\}$ is the range of indices of the connected vessels, $\varepsilon = 1, \tilde{x}_k = L_k$ for incoming vessels, and $\varepsilon = -1, \tilde{x}_k = 0$ for outgoing vessels.

The boundary conditions at the aortic root simulate the blood flow from the heart, which is set as a function $Q_H(t)$:

$$u(t,0)\,A(t,0) = Q_H(t), \tag{5}$$

$$Q_H(t) = \begin{cases} SV\dfrac{\pi}{2\tau}\sin\left(\dfrac{\pi t}{\tau}\right), & 0 \leqslant t \leqslant \tau, \\ 0, \quad \tau < t \leqslant T, \end{cases} \tag{6}$$

where SV is the stroke volume of the left ventricle, T is the period of the cardiac cycle, and τ is the duration of the systole. Parameters $SV(\tau)$, $T(\tau)$ and τ can be extracted from patient's data [8].

The boundary condition at the outlet of terminal vessel k involves constant outflow pressure $P_{out} = 25$ mmHg and resistance R_k:

$$P_t - P_{out} = Q_t / R_k. \tag{7}$$

where P_t and Q_t are pressure and flow at the terminal point. R_k is set according to the Murray's law through an iterative algorithm described in [3].

The computational domain (graph) consists of the aortic root (represented by a graph edge), the aorta (a graph edge), and two patient-specific graphs for the left coronary artery (LCA) and the right coronary artery (RCA) with their branches. We imitate a stenosis of coronary arteries as a separate vessel with decreased diameter according to the patient's data. Section 3 describes an algorithm for automatic generation of the computational graph and evaluation of the stenosis geometry. FFR is computed as the ratio between the averaged (in time) pressure distal to stenosis and the averaged (in time) aortic pressure during hyperemia. Hyperemia is imitated by 70% reduction of R_k in (7) which increases coronary flow by 150–250% [1].

Equation (1) at the internal (for each vessel or graph edge) grid nodes is discretized and solved by the explicit grid-characteristic method [9]. The boundary conditions at the junction nodes involve compatibility conditions of the hyperbolic system (1) along characteristics leaving the integration domain. Together with Eqs. (3) and (4), the boundary condition yields a system of algebraic nonlinear equations solved by the Newton method. Numerical method allows us to split calculations for internal mesh points and boundary points, processing each branch and each junction separately. Such splitting grants efficient parallelization.

3 Algorithms for ceCT Image Processing

Given a contrast enhanced CT (ceCT)[1] image of coronary vessels, we perform the 1D vessel network reconstruction in three major steps: vessel segmentation, thinning-based extraction of centerlines, and graph reconstruction. We utilize semi-automatic user-supervised algorithms for segmentation of the aorta and coronary arteries. Skeletonization and graph reconstruction are performed in an automatic way. The methods and algorithms are presented in details in our previous work [6].

Once the 1D vessel network graph is constructed, the user may define stenotic regions in order to estimate their hemodynamic significance. In our framework of the web-based simulation, all the steps are performed on the server side and are supervised and guided by the user on the client side.

3.1 Segmentation

The client uploads the input data in an anonymized format to the server. The user may crop and/or resample the image in order to reduce the computational cost of the image segmentation.

The first stage of the segmentation is detection of the aorta on the transversal ceCT slices through Hough circleness transform [6]. This algorithm is used to detect the largest bright disk corresponding to contrast enhanced blood inside the circular-shaped aorta (Fig. 1, stage 2). The user may override the detected slice and/or location of the aorta circle. The center of the aorta circle is used as a seed point in the isoperimetric distance trees (IDT) method for aorta segmentation [6, 10] (Fig. 1, stage 3). Depending on the quality of the ceCT image, the user may tune the default value of the ratio coefficient used in the IDT method in order to improve the quality of the segmentation. Since the actual 3D shape of the aorta is not used in the virtual FFR computation, the final quality of the aorta segmentation is not crucial.

Once the rough shape is captured, the user is advised to proceed to the next stage (Fig. 1, stage 4) where additional morphological operations [11] are applied in order to hide pulmonary arteries and smooth the aorta. After that, the Frangi vesselness filter [12] is used to segment the coronary arteries [6] (Fig. 1, stage 5). Depending on the quality of the input ceCT images, the user may tune the default values of

[1] More precisely, coronary CT angiography (cCTA) with contrast.

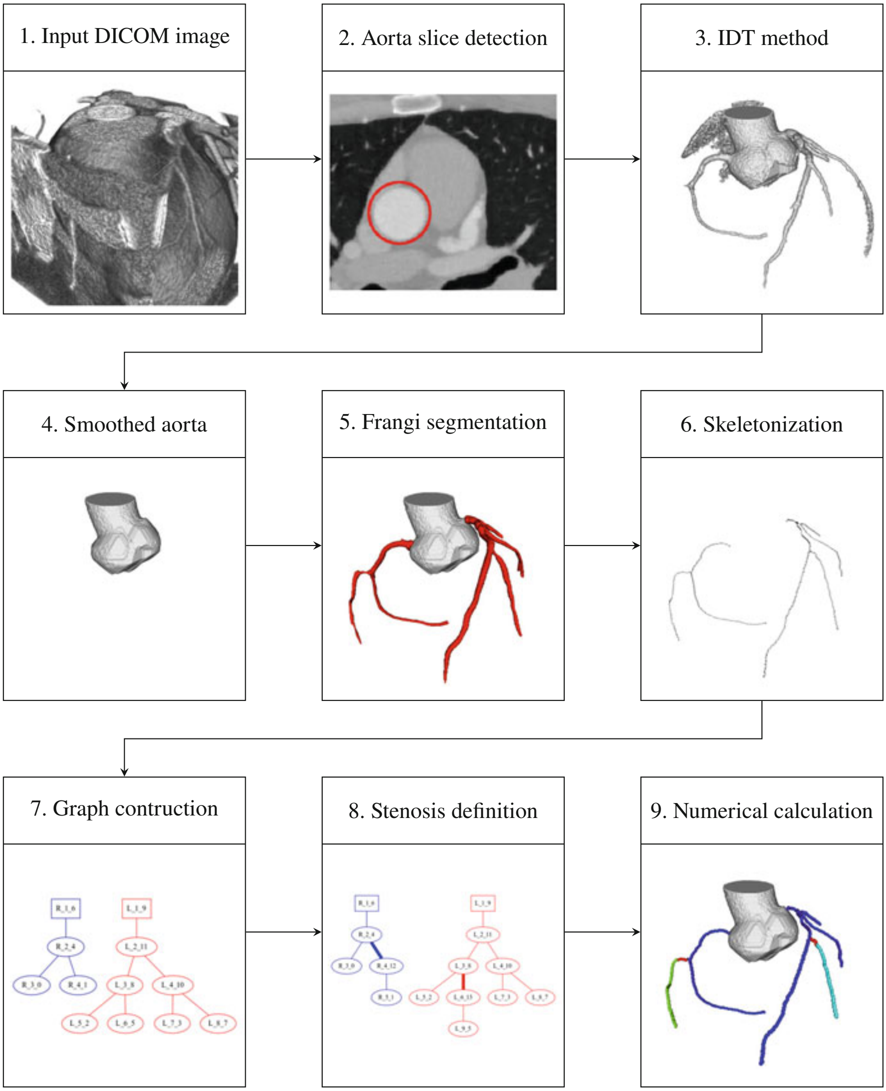

Fig. 1 Flowchart of the segmentation and modelling pipeline stages: (1) input ceCT data as DICOM image, (2) aorta slice detection by Hough transform filter, (3) aorta segmentation by IDT method, (4) smoothing of aorta by mathematical morphology, (5) coronary arteries segmentation by Frangi filter, (6) skeletonization of coronary arteries, (7) initial 1D graph network, (8) modified graph with marked stenoses, and (9) virtual FFR calculation

coefficients used in the Frangi filter in order to improve the quality of the coronary artery segmentation. However, in practice the default values are good enough for almost all ceCT images.

3.2 Skeletonization and Graph Construction

Once the ceCT image is segmented, the centerlines are extracted from the coronary arteries binary masks through skeletonization process with the help of the distance-ordered homotopic thinning [13] modified method [6] and through false twig elimination post-processing [6] (Fig. 1, stage 6). These methods reduce the diameter of the vessels down to one voxel while preserving the topology of the network. Minor false twigs occurring at the first step are identified and removed during the second step. Once the skeletonization is done, all voxels in the resulting skeleton are classified in three types: endpoints, midpoints, and junctions. The endpoint voxels have only one neighbor in the skeleton. The midpoint voxels have exactly two neighbors in the skeleton. The junction voxels have more than two neighbors in the skeleton.

This voxel classification is used to reconstruct a graph representation of the vessel network. The endpoint voxels become endpoints in the graph. The groups of junction voxels become bifurcation (junction) points. The groups of midpoint voxels become edges of the graph (Fig. 1, stage 7).

3.3 Processing of Stenoses

The user defines a stenotic region basing on a vessel profile histogram which represents the vessel diameter along its centerline. For vessel diameter calculation, we use a method of inflating ball. If voxels of the ceCT image are cubic, we locate the center of the ball with the initial diameter of three voxels at the centerline of the processed vessel. This ball is assumed to contain only voxels masked as belonging to the vessel. Then we inflate the ball until it catches a background (non-masked) voxel closest to the ball center. The diameter (in millimeters) of the inflated ball is the vessel diameter. We note that ceCT images are given in the DICOM format, and Z-spacing between DICOM slices can differ from XY-spacing so voxels are not exactly cubic. For example, for DICOM data with XY-spacing 0.405 mm and Z-spacing 1 mm, the cube in the Euclidean space is represented by a set of $7 \times 7 \times 3$ voxels. To take into account the geometric anisotropy of the voxel grid, we compute the Euclidean distance from each voxel center to the ball center.

4 User's Interface

The browser-based application *virtual FFR* allows for non-invasive estimation of FFR on the basis of the conventional CT angiography (cCTA) with contrast.

4.1 General Concept

The general idea is to use a browser as a visualization and interaction tool for scientific modeling systems. In other words, a browser becomes a front end or a user interface (UI) for a remotely executed program. This does not mean using a browser for trivial visualization but rather utilizing browser capabilities for rich 3D visualization and interaction with the user. This approach has an obvious advantage that there is no need to install a remote program on the user's computer so the user can run it faster and can access from everywhere where you can find a browser. This fits ideally to the software as a service (SaaS) concept.

In particular, our web application *virtual FFR* does not need to be installed on a local machine (user's computer) and requires only a browser to get full access to application capabilities. DICOM data is uploaded from the local machine to a remote server and processed there, and results of *virtual FFR* calculation are visualized in a browser at a local machine. All data are anonymized before transferring to a remote server.

This approach requires solution of three tasks: (1) how to visualize in a browser, (2) how to interact with a remote program, and (3) how to make a remote program capable of interacting with a browser. For the first task, we suggest using THREE.JS[2] library which is a de facto standard for developing web-based applications with sophisticated visualization.

The second task can be accomplished in different ways. One can use HTTP protocol as a natural way of communication with a browser, or use existing frameworks like Google's gRPC, or build custom mechanisms. The text-based nature and simple request-response schemes of HTTP protocol do not allow for more elaborated schemes like client-server streaming interactions. Google's gRPC provides a variety of interaction types including streaming interactions, but its deploying requires significant efforts both on client and server sides. By that reason we decided to develop our own transport mechanism based on web sockets and custom data serialization similar to gRPC Protocol Buffers. The mechanism was tailored for transferring and processing of DICOM data although the result turned out to be equally applicable for any type of interaction between a browser and a remote program.

[2] https://threejs.org/.

To accomplish the third task, we use docker multistage builds and CRADLE containers, a special technique of compiling and linking service executable with no external dependencies. We also use web socket stubs for implementing remote procedure call (RPC) interface via web sockets.

4.2 Transport Protocol and Data Serialization

Data exchange between a browser and a remote program occurs via web socket, a two-way communication channel on top of TCP/IP connection. Like HTTP protocol, the web socket protocol WS uses the same port 80 that eliminates a problem of traffic being blocked by a firewall. The web socket supports both text and binary messages. *Virtual FFR* application usually uses the text mode for sending short messages with a few input or output parameters. Binary messages are sent only if there is a big payload in a message. The web socket protocol easily distinguishes between binary and text mode so there is no confusion. The web socket binary format supports sending all types of integers (signed and unsigned), floats and doubles. On top of that a serialization protocol was developed that introduces new data types of one-, two-, and three-dimensional arrays of basic types. Strings are represented as 1D arrays of UTF-8 characters. All data transferred in messages are tagged with one-byte data descriptor which allows for unambiguous parsing. Besides that, serialization protocol introduces three new types tailored specifically for DICOM data. These types are *bit mask*, *mask with ROI*, and *bit mask with ROI*.

Type *bit mask* is a 3D Boolean array packed as a bit string. Type *mask with ROI* is a 3D Boolean array that contains additional indices of a region of interest (ROI). ROI is a subvolume that encompasses foreground (masked) voxels. Type *bit mask with ROI* is a packed version of the second type. Voxel masks (volumes containing 0 or 1) often occur in DICOM processing. More often voxel masks are sparse when regions with "ones" occupy only a small part of the original volume. Introducing new types of data allows us to reduce greatly (up to 50 times) the amount of transferred data and storage data. Packed/unpacked data is a trade-off between memory and speed of data access, and only a programmer decides what is better.

Transport protocol implements both simple RPC and server streaming RPC calls. The latter are used for long-lasting calculations like coronary arteries segmentation for reporting back a percentage of operation execution which is visualized in UI as a waiting cursor or progress bar. So far there was no need in client streaming or bidirectional RPC calls.

4.3 Software, Algorithms, and Libraries

Virtual FFR application is written on pure JavaScript and uses Google Closure Compiler (GCC) for JS minification and bundling. Custom web components are built upon Google Closure Library. GCC is apparently the best JS minifier so far and produces a bundle two times less than Webpack and UglifyJS. GCC library has a variety of components including sliders, dialogues, splitters, etc. but requires low-level programming when using. For rendering of DICOM data, the application uses AMI Toolkit developed by radiology research Center FNNDSC (Fetal-Neonatal Neuroimaging Developmental Science Center) at Boston Children's Hospital. AMI Toolkit is based on THREE.JS library.

The application uses NODE.JS[3] both for building WEB UI and server part of the application. To avoid "dependency hell" problem when different services require different versions of libraries, we use docker multistage builds for compiling and linking service executables. Every service has a pre-configured docker container called CRADLE with all compilers and libraries installed. For example, CRADLE container for *virtual FFR* application is based on Ubuntu 20.04 with specific versions of ITK and LWS libraries installed. It has a footprint of 1.13 Gb.

The application building system uses CRADLE container for compiling and linking static service executable with no external dependencies that can be executed in any lightweight container such as alpine Unix. Later we found that a no-external-dependencies requirement allows for service execution even on a SCRATCH docker container, i.e., on a bare Unix kernel with 77 byte footprint. Such extremely lightweight service containers can be distributed via dockerhub and executed on any docker-enabled computer rather than a dedicated server.

5 Application Example

In this section we present a short illustration of the stenosis markup process and *virtual FFR* application user interface (Fig. 2). The user uploads DICOM data from the local machine to the remote server and gets back aorta slice located by the application.

Then the application performs aorta and coronary vessel segmentation using segmentation threshold and other parameters given by the user if necessary. For stenosis markup one has to select consecutive histogram bars on vessel profile histogram and input stenosis percentage (vascular occlusion factor (VOF)) before sending the request to the remote server to calculate the *virtual FFR*. The resulting FFR is visualized for the coronary tree (Fig. 3).

The developed approach was tested on a variety of clinical cases. It was discovered that the main factors affecting the virtual FFR values are the degree (VOF) and the length of each stenosis followed by the hyperemia model, the cardiac output, the vessel's elasticity, and the outflow pressure. With proper parameters identification, this technology provided deviation of the computed FFR from invasive FFR measurement about 0.05 (around 6% of FFR value) [14]. This

[3] https://nodejs.org/.

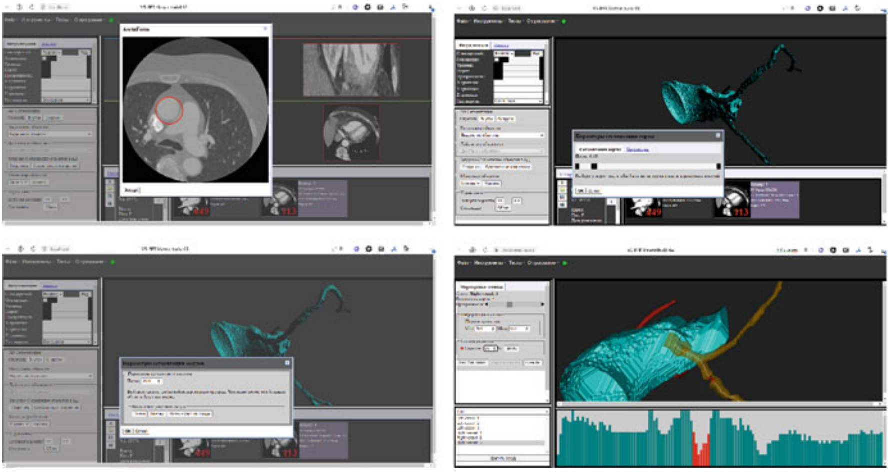

Fig. 2 Various stages of segmentation process. Top-left, detection of the aorta; top-right, segmentation of the aorta and coronary arteries; bottom-left, extraction of coronary arteries and centerlines; bottom-right, stenoses demarcation

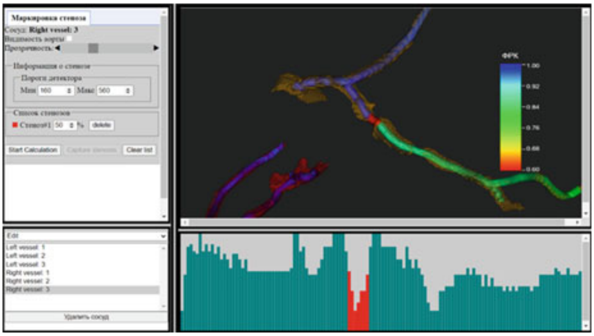

Fig. 3 Calculated FFR along coronary arteries

accuracy is good enough for most of the cases, but stenoses with FFR values close 0.8 may be misdiagnosed for stenting.

6 Discussion and Future Research

The developed software provides a user-friendly interface which allows for the manual control of the data preparation process, computations, and analysis. The web-based architecture grants easy installation in any medical center on any known operating system.

The proposed methodology provides sufficient accuracy for patient-specific non-invasive FFR estimation at computational time acceptable for clinical usage (few minutes). Our approach was compared with computed FFR based on 3D simulations

[4]. We demonstrated that 1D and 3D approaches provide similar sensitivity and specificity. Correlation coefficient between 1D and 3D models were rather high, while the demand for computational resources for 1D models was significantly lower. These results correspond to findings of other scientific groups [15]. However, some cases require additional computational analysis of various pathophysiological factors including multivascular stenosis [16], increased heart output due to a physical load, mental stress, [17], blood viscosity [18], variable heart rate [8, 19], and myocardial perfusion damage [20].

There exist other hemodynamic indices (CFR, iFR) which are important for analyzing coronary stenosis severity as well. Their virtual counterparts have been studied by computational models in [3, 21–23]. These computational models should be clinically tested and implemented in future releases of our *virtual FFR* application.

The use of neural networks and machine learning technique may decrease substantially the computational time and increase robustness of the software in detecting stenoses [24] and evaluating FFR [25].

Acknowledgment The work was supported by Russian Science Foundation Grant No.21-71-30023.

References

1. Gould, K.L. Kirkeeide, R.L., Buchi, M.: Coronary flow reserve as a physiologic measure of stenosis severity. J. Am. Coll. Cardiol. *15(2)*, 459–474 (1990)
2. Morris, P.D., van de Vosse, F.N., Lawford, P.V., Hose, D.R., Gunn, J.P.: “Virtual” (Computed) Fractional Flow Reserve: Current Challenges and Limitations. JACC Cardiovasc Interv. *8(8)*, 1009–1017 (2015) https://doi.org/10.1016/j.jcin.2015.04.006
3. Simakov, S., Gamilov, T., Liang, F., Kopylov, P.: Computational Analysis of Haemodynamic Indices in Synthetic Atherosclerotic Coronary Netwroks. Mathematics. *9(18)*, 2221 (2021) https://doi.org/10.3390/math9182221
4. Gognieva, D. G., Pershina, E. S., Mitina, Yu. O., et.al.: Non-Invasive Fractional Flow Reserve: a Comparison of One-Dimensional and Three-Dimensional Mathematical Modeling Effectiveness. Cardiovascular Therapy and Prevention. *19(2)*:2303 (2020) (In Russian) https://doi.org/10.15829/1728-8800-2020-2303
5. Vassilevski, Yu., Olshanskii, M., Simakov, S., Kolobov, A., Danilov, A.: Personalized Computational Hemodynamics: Models, Methods, and Applications for Vascular Surgery and Antitumor Therapy. Academic Press (2020)
6. Danilov, A., Ivanov, Yu., Pryamonosov, R., Vassilevski, Yu.: Methods of graph network reconstruction in personalized medicine. Clinical implications of dysregulated cytokine production. Int. J. Numer. Method. Biomed. Eng. (2015) https://doi.org/10.1002/cnm.2754
7. Simakov, S.: Spatially averaged haemodynamic models for different parts of cardiovascular system. Russian J. of Num. Anal. and Math. Mod. *35(5)*, 285–294 (2020)
8. Gamilov, T., Kopylov, P., Serova M., et.al.: Computational Analysis of Coronary Blood Flow: The Role of Asynchronous Pacing and Arrhythmias. Mathematics. *8(8)*, 1205 (2020) https://doi.org/10.3390/math8081205
9. Magomedov, K. M., Kholodov, A. S. Grid-characteristic numerical methods. Nauka, (2018) [in Russian].

10. Grady, L.: Fast, quality, segmentation of large volumes – Isoperimetric distance trees. Comput. Vis. – ECCV 2006, **3953**, 3711–3723 (2007)
11. Yang, G., Kitslaar, P., Frenay, M., Broersen, A., Boogers, M.J., Bax, J.J., Reiber, J.H.C., Dijkstra, J.: Automatic centerline extraction of coronary arteries in coronary computed tomographic angiography. Int. J. Cardiovasc. Imaging, **28**, 921–933 (2012)
12. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. Medical Image Computing and Computer-Assisted Intervention – MICCAI98, *1496*, 130–137 (1998)
13. Pudney, C.: Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images. Comput. Vis. Image Underst., **72**, 404–413 (1998)
14. Carson J.M., Pant, S., Roobottom, C., et.al.: Non-invasive coronary CT angiography-derived fractional flow reserve: A benchmark study comparing the diagnostic performance of four different computational methodologies. Int. J. Numer. Method. Biomed. Eng., *35(10)*, e3235, (2019) https://doi.org/10.1002/cnm.3235
15. Blanco, P.J., Bulant, C.A., Müller, L.O., et.al.: Comparison of 1D and 3D Models for the Estimation of Fractional Flow Reserve. Sci Rep *8*, 17275 (2018). https://doi.org/10.1038/s41598-018-35344-0
16. Simakov, S.S., Gamilov, T.M., Koylov, F.Yu., Vasilevkii, Yu.V.: Evaluation of Hemodynamic Significance of Stenosis in Multiple Involvement of the Coronary Vessels by Mathematical Simulation. Bulletin of Experimental Biology and Medicine, *162(1)*, 111–114, (2016).
17. Gamilov, T., Simakov, S.: Blood Flow Under Mechanical Stimulations, Advances in Intelligent Systems and Computing, *1028*, 143–150 (2020)
18. Simakov, S., Gamilov, T.: Computational Study of the Effect of Blood Viscosity to the Coronary Blood Flow by 1D Haemodynamics Approach. Smart Innovation, Systems and Technologies, *214*, 237–248 (2021)
19. Ge, X., Simakov, S., Liu, Y., Liang, F.: Impact of Arrhythmia on Myocardial Perfusion: A Computational Model-Based Study. Mathematics, *9(17)*, 2128 (2021)
20. Simakov, S., Gamilov T., Liamf, F., et.al.: Numerical evaluation of the effectiveness of coronary revascularization. Russian J. of Num. Anal. and Math. Mod., *36(5)*, 303–312 (2021)
21. Ge, X., Liu, Y., Tu, S., et. al.: Model-Based Analysis of the Sensitivities and Diagnostic Implications of FFR and CFR Under Various Pathological Conditions. Int. J. Numer. Method. Biomed. Eng., *37(11)*, e3257, (2021) https://doi.org/10.1002/cnm.3257
22. Ge, X., Liu, Y., Yin, Z., et. al.: Comparison of Instantaneous Wave-Free Ratio (iFR) and Fractional Flow Reserve (FFR) With Respect to Their Sensitivities to Cardiovascular Factors: A Computational Model-Based Study. Journal of Interventional Cardiology, *2020*, 40941421 (2020) https://doi.org/10.1155/2020/4094121
23. Carson, J.M., Roobottom, C., Alcock, R., Nithiarasu, P.: Computational Instantaneous Wave-Free Ratio (IFR) for Patient-Specific Coronary Artery Stenoses Using 1D Network Models, Int. J. Numer. Method. Biomed. Eng., *35(11)*, e3255, (2019) https://doi.org/10.1002/cnm.3255
24. Jones, G., Parr, J., Nithiaasu, P., Pants, S.: Machine Learning for Setection of Stenoses and Aneurysms: Application in a Physiologically Realistic Virtual Patient Database. Biomechanics and Modeling in Mechanobiology, *20(6)*, 2097–2146 (2021)
25. Itu, L., Rapaka, S., Passerini, T., et.al.: A Machine-Learning Approach for Computation of Fractional Flow Reserve From Coronary Computed Tomography. J. Appl. Physiol., *121*, 42–52 (2016)