Computer Assisted Mechanics and Engineering Sciences, **10**: 295–319, 2003.

Object-oriented software system that performs FPM simulation in the area with moving boundary, and its application to the blood flow problem

Maciej Paszyński
Institute for Computational Engineering and Sciences ICES,
The University of Texas at Austin
ACES 6.332, 105, Austin, Texas 78712, Campus mail code: C0200
e-mail: maciek@ices.utexas.edu
and
Department of Computer Methods in Metallurgy,
AGH University of Science and Technology
Mickiewicz Avenue 30, 30-059 Krakow

(Received June 13, 2001)

The paper presents the project of object-oriented software system for modeling flows of fluids inside the area with moving boundary. The flow of a fluid is modeled by using Fluid Particle Model. Finite Elements mesh is generated on the boundary of the area, to allow calculations of stresses on the boundary, arisen from interaction of the fluid with the boundary. Here was presented the application of the system, that simulates the phenomenon of energy and mass transport in large arteries in man. The validity of the computational method was established by comparing the numerical results to medical measurements data. The architecture and results of Fluid Particle Model were presented to compare with the architecture and results of Finite Element models for blood flow in arteries, described in other publications.

1. Motivation

The classical approach to the class of problems of fluids flows in the area with moving boundary is based on the Finite Elements Method or Finite Differences Method. The application of the classical models to the problem of blood flow in large arteries in man [26, 37] shows that classical methods produce many problems such as numerical costs, complications of variational formulation, and problem's sensitivity on boundary and initial conditions.

Fluid Particle Model invented by Español [6] is a modern method for simulating both mesoscopic and macroscopic phenomena [5]. The application of the model allows to overcome many disadvantages of continuous models [25].

One of the main disadvantages of the classical methods applied to the class of problems with moving boundary, is that FEM requires generation of 3-dimensional mesh of finite elements at the beginning of each step of calculations in the area with moving boundary. Additional modification of generated mesh are necessary in order to improve precision of solution. FPM is the meshless method. FPM requires a uniform distribution of particles inside the modeled area only at the beginning of the first time step. The method may be successfully used to simulate non-stationary flows in areas with moving boundary.

There are many projects of object oriented software systems that perform calculations of flows of the Newtonian and non-Newtonian fluids by using classical methods such as the Finite Elements Method or the Finite Differences Method [33].

The paper presents the project of object-oriented software system for modeling flows of fluids in the area with moving boundary. The flow of a fluid is modeled by using Fluid Particle Model. Finite Elements mesh is generated on the boundary of the area, to allows calculations of stresses on the boundary, arisen from interaction of the fluid with the boundary. As opposed to FEM realization, only updating of the mesh on the boundary of modeled area is required at the beginning of each time step of calculations.

The architecture of the system is based on some basic concepts of modeling open distributed object-oriented systems [33]. The system was designed as object-oriented opened one, on the basis of the bottom-up paradigm and the conception of generalized classes.

Nowadays, the most promising software development technology is the object-oriented approach. It is a very powerful tool for problem analysis and software design, not only a method of programming. Object-Oriented Analysis (OOA), Object-Oriented Design (OOD) and Object-Oriented-Programming (OOP) may constitute the most efficient instruments for large systems development [2].

There are, in general, two possible approaches to the design process of the software system: the bottom-up approach and the top-down one. Within the top-down approach we start the design process by creating composed objects and then we decompose them into many simple basic ones. On the contrary, within the bottom-up approach we first create simple basic objects and then more composed objects are created as a composition of many basic ones. The bottom-up approach in the process of designing composed engineering software systems, seems to be the more effective way in comparison with the top-down approach [33]. Within the top-down approach it is necessary to define and solve complex problems on the level on big composed objects. The bottom-up approach, as opposed to the top-down one, decomposes complex problems into many simple ones, solved on the level on simple basic objects. It is the more natural way of solving many composed problems, that is easier to implement.

The crucial rule in the system that performs Fluid Particle Model calculations is played by the following classes: Generalized Particle and Generalized Interparticle Forces. The application was designed as an open one, it means that it is possible to define new types of particles and interparticle forces classes with additional functionality. That's why Generalized Particle and Generalized Interparticle Forces classes were defined, which use the mechanism of virtual methods that allows to operate on many objects of different types.

The software system may be used for wide variety of problems from the class of fluid flows problems in the area with moving boundary. Blood flow problem in central part of the circulation system in man is an example of such problem.

To confirm the validity of the software system, interesting simulation of the phenomenon of energy and mass transport in large arteries has been performed.

2. Overview on the Fluid Particle Model

Fluid Particle Model invited by Español [6] is a modern method for simulating both mesoscopic and macroscopic phenomena [5]. The method consists in defining mesoscopic objects (particles – "droplets" of fluid) and establishing physical forces between them. The fluid area is divided using Voronoi tessellation scheme into the set of N separate particles with a given initial position and velocity, covering the entire fluid area.

The coarsening is performed through Voronoi tesselletion, which allows one to divide physical space into a set of nonoverlapping cells that cover all the space in a well-defined manner. Given a discrete set of points (that can be distributed at random) the Voronoi tessellation assign to each point (called "cell center") that region of space that surrounds it and that is nearer to this point than to any other point of the set.

2.1. Basic steps of Fluid Particle Model

The simulation of the flow of the fluid by using FPM consists in updating of the state of each particle in consecutive time steps of simulation, on the basis of its interactions with neighboring particles distanced up to R_{cut} border distance. Each time step of simulation consists in performing stages 1, 2, 3 described in Table 1.

Table 1. Three stages of one time step of FPM discrete simulation

STAGE 1 Initial state (positions, linear velocities and angular velocities) (r_i, v_i, ω_i) of each cell $i = 1, ..., N$ is established
STAGE 2 On the basis of interparticle forces, the change of state of each particle is calculated: $\Delta r_i = v_i \Delta t$ $\Delta v_i = \frac{1}{m} \sum_{j, j \neq i} F_{ij} \Delta t$ $\Delta \omega_i = I^{-1} \sum_{j, j \neq i} N_{ij} \Delta t$ where F_{ij} is the force which particle j exerts on particle i, and the torque N_{ij} is defined as $N_{ij} = \frac{1}{2} r_{ij} \times F_{ij}$. For each particle it is necessary to calculating the force that neighbouring particles exerts on the particle. For each particles only its neighbouring particles distanced far to R_{cut} are considered.
STAGE 3 The state of each particle is updated: $r_i(t + \Delta t) = r_i(t) + \Delta r_i$ $v_i(t + \Delta t) = v_i(t) + \Delta v_i$ $\omega_i(t + \Delta t) = \omega_i(t) + \Delta \omega_i$

After performing the discrete FPM simulations, we get accurate results on the mesoscopic level. Obtained results may be analyzed using suitable statistical methods.

It is possible, for example, to trace particular particles and establish how chaotic is theirs motion.

Let us focus on the problem of modeling of pulsatile blood flow in large arteries in man.

Because of pulsating character of the blood flow in central circulation system, the transition of the blood flow from laminar to turbulent is possible [21, 34].

The modeling of turbulent flow using FEM is very difficult if possible. As opposed to classical methods, FPM may be easily used for modeling the turbulent flow.

Once the turbulence occured, the motion of particular particles becomes more chaotic, and it would be easily recognized by using suitable statistical method.

2.2. Interparticle forces

Interparticle forces are divided into classical contributions $F^C + F^T + F^R$ and connected corresponding stochastic contribution $\widetilde{F}$ (1)

$$F_{ij} = F_{ij}^C + F_{ij}^T + F_{ij}^R + \widetilde{F}_{ij}. \tag{1}$$

Stochastic noise is represented in the model by independent increments of the Wiener process ΔW_{ij} [1]. The first contribution to classical forces is assumed to arise from a conservative potential $V(r)$ that depends on the separation distance between particles (2).

$$F_{ij}^{C} = -V'(r_{ij})\, e_{ij}. \tag{2}$$

The second contribution is a friction force that depends on relative translational velocities of the particles i, j, with positions r_i, r_j, and velocities v_i, v_j (3).

$$F_{ij}^{T} = -\gamma m T(r_{ij})\, v_{ij}, \tag{3}$$

where $r_{ij} = r_i - r_j$ is the vector connecting both particles, e_{ij} is the versor of r_{ij}, $v_{ij} = v_i - v_j$ means relative velocity, γ is some friction parameter.

The last classical contribution is a rotational force (4)

$$F_{ij}^{R} = -\gamma m T(r_{ij}) \left(\frac{r_{ij}}{2} \times (\omega_i + \omega_j) \right). \tag{4}$$

$T(r_{ij})$ is defined as the following matrix (5)

$$T(r_{ij}) = A(r_{ij})\, I + B(r_{ij})\, e_{ij} e_{ij}, \tag{5}$$

where $\Re \ni \|r_{ij}\| \to A(r_{ij}) \in \Re$ and $\Re \ni \|r_{ij}\| \to B(r_{ij}) \in \Re$ are scalar functions that parameterizes classical forces.

Stochastic force is defined in the following way (6)

$$\widetilde{F}_{ij} \Delta t = \sigma m \left(\widetilde{A}(r_{ij})\, \overline{\Delta W}_{ij}^{S} + \widetilde{B}(r_{ij})\, \frac{1}{D} tr(\Delta W_{ij}) + \widetilde{C}(r_{ij})\, \Delta W_{ij}^{A} \right), \tag{6}$$

where σ is a parameter governing the overall noise amplitude, $\Re \ni \|r_{ij}\| \to \widetilde{A}(r_{ij}) \in \Re$, $\Re \ni \|r_{ij}\| \to \widetilde{B}(r_{ij}) \in \Re$ and $\Re \ni \|r_{ij}\| \to \widetilde{C}(r_{ij}) \in \Re$ are scalar functions that parametrizes stochastic force, $D = 3$ is the spatial dimension. $\overline{\Delta W}_{ij}^{S}$ and ΔW_{ij}^{A} are the traceless symmetric and antisymmetric random matrices [6]. Each one of three contributions to the stochastic force is connected with one of three contributions to the classical forces. Stochastic forces are "heating-up" the system, to ensure energy conservation. The system works like a thermostat.

If we consider the system in the equilibrium state, where the Gibbs equilibrium ensemble is the unique stationary solution of the model dynamics [6], we get the following relations between scalar functions $A(R)$, $B(R)$ defining classical forces, and scalar function $\widetilde{A}(R)$, $\widetilde{B}(R)$, $\widetilde{C}(R)$ defining stochastic forces (7), (8)

$$A(r_{ij}) = \frac{1}{2} \left(\widetilde{A}^2(r_{ij}) + \widetilde{C}^2(r_{ij}) \right), \tag{7}$$

$$B(r_{ij}) = \frac{1}{2} \left(\widetilde{A}^2(r_{ij}) - \widetilde{C}^2(r_{ij}) \right) + \frac{1}{D} \left(\widetilde{B}^2(r_{ij}) - \widetilde{A}^2(r_{ij}) \right) \tag{8}$$

and additionally the interdependence between γ friction parameter and σ noise amplitude (9)

$$\gamma = \frac{\sigma^2 m}{2 K_B T}, \tag{9}$$

where K_B is the Boltzmann constant, T is the temperature of the system in the equilibrium state.

On the basis of theory of stochastic differential equations and kinetic theory of gases, it is possible to derive Navier–Stokes equations from equations describing the evolution of FPM particles in phase space [6, 18]. The theory shows that the solution of the problem using FPM simulations is equivalent to the calculation of the solution of Navier–Stokes equations.

3. Composition of the application

The section presents the project of object-oriented application that performs FPM simulation in one part of central circulation system in man. The application was designed by using Booch diagram notation [2] and Unified Modeling Language [3]. Class diagrams were designed by using Visual Modeler tool, that allows to generate C++ [36] source code from modeled diagrams. Generated code was then developed with Visual C++.

Within the particular modules, the names of the classes were adjusted to the problem of blood flow in large arteries in man, but they may be easy generalized and used to model any other problems from the class of flow problems in the area with moving boundary.

The application were divided into the following modules: Mathematical Module, Vessel Module, Flow Calculation Model Module, Data Base Access Module, Boundary Conditions Module, User Interface Module and Arterial Wall Calculation Module (Fig. 1). All modules were implemented as dlls, and User Interface Module was implemented as exe file. The relations presented on the diagram translate themselves into linkage dependencies between modules. In the following sections particular modules will be described in details.

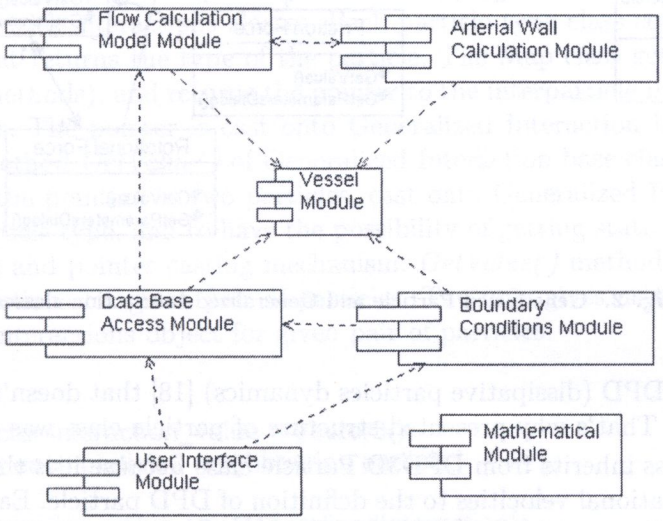

Fig. 1. Composition of the application

3.1. Mathematical Module

The module constitutes the library of mathematical classes, used by the other calculation modules including the following classes: Vector, Point, Matrix with defined operations on the classes. The library is intensively used by the other modules. Additionally were defined the classes of scalar and vector functions with their formulas kept as Visual C++ CString objects, and methods to calculate its values using the ONP mechanism. It gives us the possibility to dynamically define the function formula without the necessity of recompilation.

3.2. Flow Calculation Model Module

The module includes the implementation of FPM method applied to calculations of the problem of energy and mass transport in one part of the artery (Fig. 2).

The module defines classes of Particles and Interparticle Interactions that will be described in details. FPM requires to define positions, velocities and rotational velocities of each particle. FPM

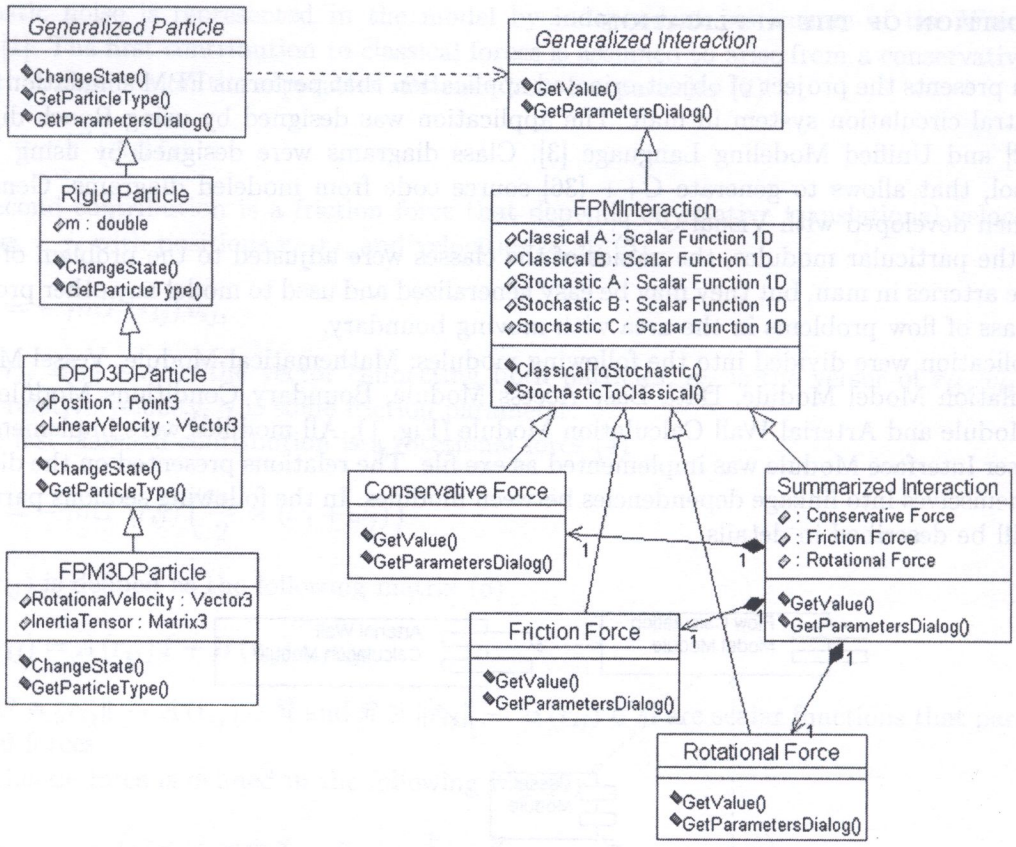

Fig. 2. Generalized Particle and Generalized Interaction classes

is the generalization of DPD (dissipative particles dynamics) [18] that doesn't require the definition of rotational velocities. That's why presented structure of particle class was defined.

FPM3D Particle class inherits from DPD3D Particle class because it is the generalization of the class, that is adding rotational velocities to the definition of DPD particle. Each particle has its own mass, that's why Rigid Particle class was defined. FPM 3D Particle class requires also the definition of the inertia tensor I because it describes rotational velocities.

The application was designed as an open one, it means that it is possible to define new types of particles with additional functionalities. That's why Generalized Particle class was defined, which uses the mechanism of virtual methods that allows us to operate on many objects of different types of particles.

In the similar way it is possible to define new types of interparticle interactions classes, derived from Generalized Interaction base class (Fig. 2). For 3 dimensional simulation of Fluid Particle Model the FPM3D Interaction class was defined.

During each step of FPM simulation, it is necessary to calculate interparticle interactions of each particle and its neighboring particles. Interparticle forces are divided onto three contributions: conservative forces, dissipative forces (depends on translational velocities) and rotational forces. Each of these forces is also divided into classical and stochastic contributions. Interparticle forces are parametrized by some scalar functions (scalar functions that parametrizes classical forces $A(r_{ij})$, $B(r_{ij})$, and connected scalar functions that parametrizes stochastic forces $\widetilde{A}(r_{ij})$, $\widetilde{B}(r_{ij})$ and $\widetilde{C}(r_{ij})$).

There are some relations between classical and stochastic contributions of the forces, presented in the first part of the paper. It is possible to define the first classical contribution, and then define stochastic one by using relations formulas.

That's why FPM Interaction base class contains scalar functions that define classical and stochastic forces, together with the methods that allows us to translate classical forces parameters easily into stochastic one. Classes of particular parameters inherits from the class, and defines itself on the basis of general parameters. Summary Interaction class keeps together the three mentioned classes of three types of interparticle interactions.

The Generalized Interaction class contains the method *GetValue( )* which calculates interparticle force value for given pair of particles (that's why Generalized Interaction class uses Generalized Particle class – to get data of both particles). The method *GetValue( )* from Summarized Interactions class calls the methods from 3 interparticle force classes, and summarize their values. The Generalized Particle class contains the method *Change State( )*, which changes state of the particle on the basis of interparticle interaction value.

The application has been designed so as to give the possibility of maintaining particles of different types during simulation. It was possible to do by using virtual functions mechanism. During each step of simulation, for each particle its neighboring particles with different types are found. Interparticle interactions classes are created for each pair of particles with different types.

An example of the code that performs interparticle interactions calculations is presented in Fig. 3. The applications keep the Map class, that is mapping from pair of types of particles into the object of interparticles interactions classes. Each particle type class contains a virtual method *GetParticleType( )* that returns the type of the particle. The Map class gets types of two particles (by using its virtual methods), and returns the pointer to the interparticle interaction class for given pair of particles types. The pointer is cast onto Generalized Interaction base class. By using the pointer, the virtual method *GetValue( )* of Generalized Interaction base class is called. The method gets as an argument the pointers to two particles, cast onto Generalized Particle base class, to be independent from particle type, and to have the possibility of getting state of the particles by using some virtual methods and pointer casting mechanism. *GetValue( )* method is calculating the value of interparticle interactions for given pair of particles. The map gives us the possibility of finding quickly interparticle interactions object for given pair of particles.

```
Vector interaction_value = Vector3();
For each particle2 neighbouring particle1
{
interaction_value += (GeneralizedInteraction*)
InteractionsCollection (
    (GeneralizedParticle*)particle1->GetParticleType(),
    (GeneralizedParticle*)particle2->GetParticleType()
    )
    ->GetValue(
        (GeneralizedParticle*)particle1,
        (GeneralizedParticle*)particle2);
}
(GeneralizedParticle*)particle1->ChangeState(interaction_value)
```

Fig. 3. An example of the code that performs interparticle interaction calculations

For each particle neighboring given particle, we get its interaction value. All interaction values are summarized to obtain general force value that all neighbouring particles exerts on given particle. Then, *ChangeState( )* virtual method of given particle is called, to actualize its state after neighboring particles interactions. Such implementation gives us the possibility of expanding the model on new types of particles and its interactions. The only thing to do is to define new type of particle class, and classes of interactions of new particle with all other particles types, and to add the classes to the map.

3.3. Arterial Wall Calculation Module

The module is responsible for calculating stresses on the arterial walls. There are some interactions between Arterial Wall Calculation Module and Flow Calculation Model Module (Fig. 1). Flow Calculation Model Module will calculate the forces that blood exerts on arterial walls. Arterial Wall Calculation Module will calculate arterial wall deformation on the basis of two information: (1) the interaction of blood particles onto the arterial wall, (2) the wave of wall deformation produced by the heart during its eruption of blood. For more details see subsection *Boundary Conditions Module* below.

3.4. Boundary Conditions Module

The boundary conditions for the Fluid Particle Model simulations are the following:

- Particles velocities distributions at the inflow of the modelled area, in consecutive time steps of simulation.

In consecutive time steps of simulations, the boundary condition of the pressure gradient produced by the heart eruption may be realized by adding new particles with given velocities on the boundary of modeled area.

- Boundary conditions for the Arterial Wall Calculation Module.

The version of the software system used to simulate the phenomenon of energy and mass transport in artery, assumes rigid tube vessel walls. However our model doesn't include vessel elastic walls, the definition of interparticle interactions within FPM allows to simulate transport of energy by exchange of particles momentum.

Within the software system, it is possible to define simplified or precise arterial walls – modeled fluid interactions.

Within the simplified one, it is expected to define the shape of arterial wall deformation translocating along to the modeled area from the inflow to the outflow in consecutive time steps of the simulation.

On the basis of the medical measurements (see sub-section *Medical measurements*) it is possible to establish the speed of the pulse wave propagation, as well as the shape of the deformation at the inflow and the outflow of modeled area, in consecutive time steps of the simulation.

For the simulation purposes measurements should be performed at the inflow of modeled area. To perform simplified arterial wall – modeled fluid interactions, it is reasonable to assume that measured deformation will translocate from the inflow to the outflow of the modeled area, with measured speed of the pulse wave propagation.

One can say that the absence of the arterial wall dynamic model is an inadmissible simplification. There are comprehensive works concerning blood flow in large arteries in man, based on the classical finite elements method, considering arterial tree as a system of rigid tubes filled with incompressible fluid [37]. Also the author has performed classical FEM simulation without including of the wall dynamic model, with satisfactory results [26]. But the inclusion of the arterial wall dynamics model is necessary to improve the precision of the solution.

For the definition of precise arterial wall – modeled fluid interactions, it is expected to include the mathematical model of two-dimensional arterial wall dynamic to the Arterial Wall Calculations Module.

The soft tissues of the vessel walls consists of different fibers, elastin, smooth muscle and water [29]. Due to this complex structure, it is difficult to provide a synthetic mathematical description of the mechanical behaviour of vessel walls.

But there are several one-dimensional or two-dimensional attempts to describe mathematical model of the arterial wall dynamic [14, 29]. The models require two dimensional finite elements mesh to be generated on the boundary of modeled area.

It is possible to extend our application easily by including such a model within the Arterial Wall Calculation Module. The Vessel class consists of triangular finite elements, to allow to perform vessel wall dynamic calculation within Arterial Wall Calculation Module.

Data required for the vessel wall dynamic model are (1) the interaction of blood particles onto the arterial wall, (2) measured shape of the pulse wave deformation at the inflow and the outflow of modeled area.

3.5. Vessel Module

The module includes classes responsible for defining the shape of the modeled vessel. The initial shape of the vessel is defined within the Vessel Geometry class. In general, particular curve shape of the axle of the vessel may be modeled by defining the series of points, with 2 perpendicular vectors, defining ellipse with center at one axle point.

Vessel Geometry class may be created in other external application, for example modeled within AutoCAD. The application may keep definition of the model of entire central circulation system, with possibility of selecting one part of the system, to perform simulation on selected vessel (Fig. 4).

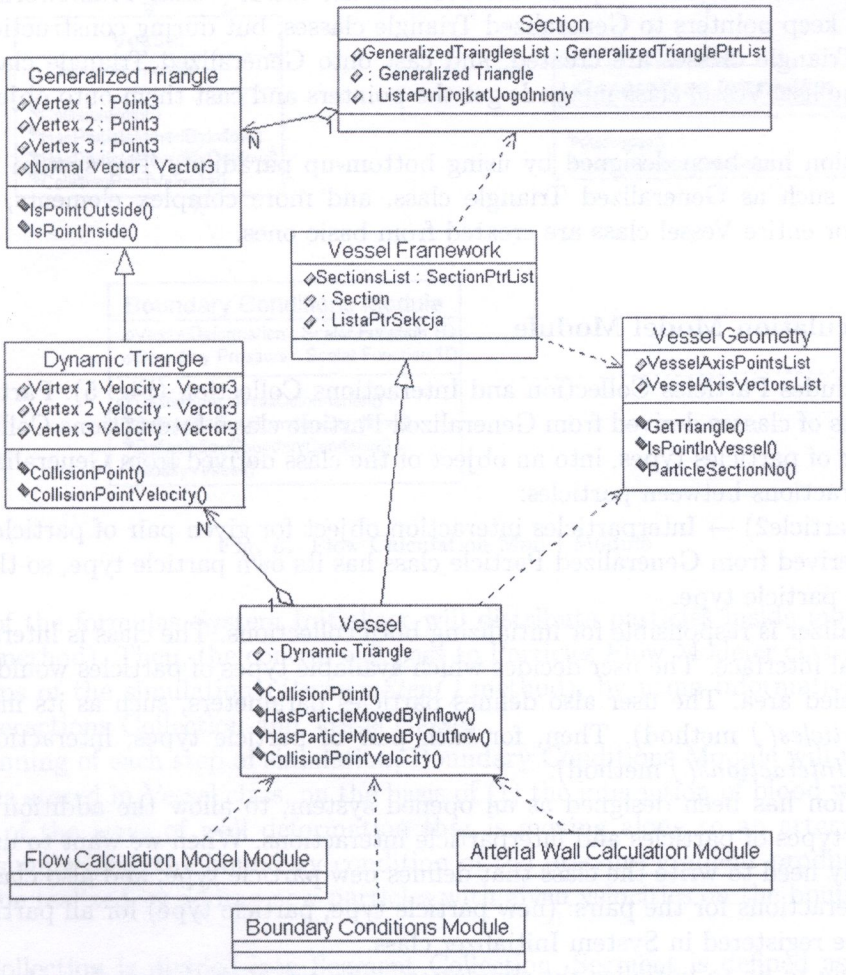

Fig. 4. Vessel module classes

Vessel class is created on the basis of the information from Vessel Geometry class. Vessel Geometry class defines the initial state of the vessel geometry. Within the Vessel class, the geometry of

the vessel may change. Vessel class during simulation keeps information about current geometrical configuration of the Vessel. Additionally, Vessel class keeps $2d$ finite element on the boundary of modeled area, for Arterial Wall Calculation Module.

Vessel class inherits from Vessel Framework class, that includes pure topological data (how the vessel is divided into sections, and how sections are divided into triangles). Vessel class includes vessel dynamics data. Constructor of the Vessel class gets reference to Vessel Geometry class, and pass the reference to its Vessel Framework base class. Vessel Framework on the basis of Vessel Geometry class data is constructing topology of the vessel. The vessel is divided into sections, and boundaries of the sections are divided into triangular elements. Vessel Framework class keeps the list of the pointers to Vessel Section classes. Each Vessel Section class keeps the list of the pointers to objects of Generalized Triangle class. There are two classes: Generalized Triangle and Dynamic Triangle.

Generalized Triangle class keeps static data (concerning positions of vertexes of a triangle, and the orientation of a triangle). Dynamic Triangle class consists of dynamical information (concerning velocities of vertexes of a triangle). The methods of Vessel Framework class works on static data and uses Generalized Triangle class. The methods of Vessel class works on dynamic data and uses Dynamic Triangle class.

It is important to see that triangle objects are not duplicated. There is only one list of triangle objects for particular section – kept in the sections list inside Vessel Framework class. Lists of triangle objects keep pointers to Generalized Triangle classes, but during construction of particular list, Dynamic Triangle classes are created, and cast onto Generalized Triangle classes when they are stored on the list. Vessel class methods get the pointers and cast them onto objects of Dynamic Triangle class.

The application has been designed by using bottom-up paradigm. The system is created from basic elements, such as Generalized Triangle class, and more complex elements, such as Vessel Section classes or entire Vessel class are created from basic ones.

3.6. Flow Calculation Model Module

The module includes Particles Collection and Interactions Collection (Fig. 5). Particles Collection is a set of objects of classes derived from Generalized Particle class. Interactions Collection class is a map from a pair of particles types, into an object of the class derived from Generalized Interaction, that define interactions between particles:

(particle1, particle2) $\rightarrow$ Interparticles interaction object for given pair of particles type).

Each class derived from Generalized Particle class has its own particle type, so the class may be identified by its particle type.

System Initializer is responsible for initializing both collections. The class is interacting with the user by graphical interface. The user decides which available types of particles would be distributed inside the modeled area. The user also defines particles parameters, such as its mass and inertia tensor (*EditParticles()* method). Then, for each pair of particle types, interactions objects are initialized (*EditInteractions()* method).

The application has been designed as an opened system, to allow the addition of new classes that define new types of particles and interparticle interactions. When we want to use new types of particles, we only need to write the class that defines new particle type, and also classes that define interparticle interactions for the pairs: (new particle type, particle type) for all particle types. New classes should be registered in System Initializer class.

The next thing to do to initialize the simulation, is to establish the number of particles of each type, and also initial conditions for linear and rotational velocity fields. It is done by giving the formulas of 2 functions of the following form (10):

$$\Re^3 \ni (x_1, x_2, x_3) \rightarrow f(x_1, x_2, x_3) = (f_1(x_1, x_2, x_3), f_2(x_1, x_2, x_3), f_3(x_1, x_2, x_3)) \in \Re^3. \tag{10}$$

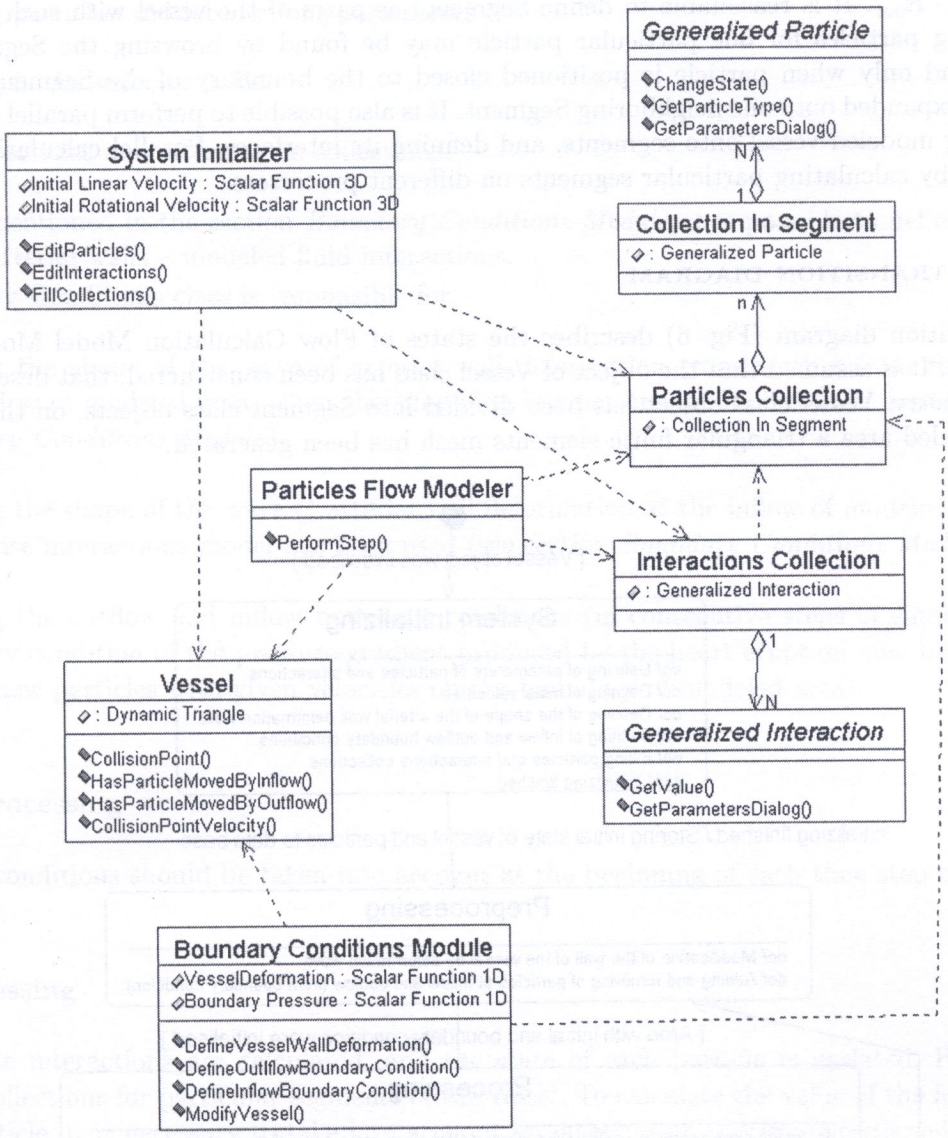

Fig. 5. Flow Calculation Model Module

On the basis of the formulas System Initializer will distribute particles inside modeled area (*FillCollections( )* method). Then, the control will pass to Particles Flow Modeler class, which performs succeeding steps of the simulation (*PerformStep( )* method), by using information from Particles Collection, Interactions Collection and Vessel classes.

At the beginning of each step of simulation, Boundary Conditions Module will modify geometry of modeled area stored in Vessel class, on the basis of (1) the interaction of blood with arterial wall, (2) the shape of the wave of wall deformation that is moving along to an artery. In consecutive time steps of simulations, the boundary condition of the pressure gradient produced by the heart eruption may be realized by adding new particles with given velocities on the boundary of modeled area.

Particles Collection is divided into Segment Collection. Segment is defined as one part of the vessel. For example it is possible to define Segments as equal to Vessel Sections. Such division of Particles Collection is connected with decreasing calculation time. During each time step of calculations for each particle it is necessary to calculate the interactions of the particle with all its neighboring particles, distanced up to R_{cut} distance. It would be very time consuming to browse the entire vessel to find all neighboring particles. It is necessary to browse one part of the vessel, with

diameter $2 \cdot R_{\text{cut}}$. It is reasonable to define Segments as parts of the vessel with such a diameter. Neighboring particles for one particular particle may be found by browsing the Segment of the particle, and only when particle is positioned closed to the boundary of the Segment, searching should be expanded onto one neighboring Segment. It is also possible to perform parallel calculations by dividing modeled vessel onto segments, and defining its interfaces. Parallel calculations may be performed by calculating particular segments on different processors.

4. State transition diagram

State transition diagram (Fig. 6) describes the states of Flow Calculation Model Module during simulations. It is assumed that the object of Vessel class has been constructed, that describes initial vessel geometry. Vessel class object has been divided into Segment class objects, on the boundary of the modeled area a triangular finite elements mesh has been generated.

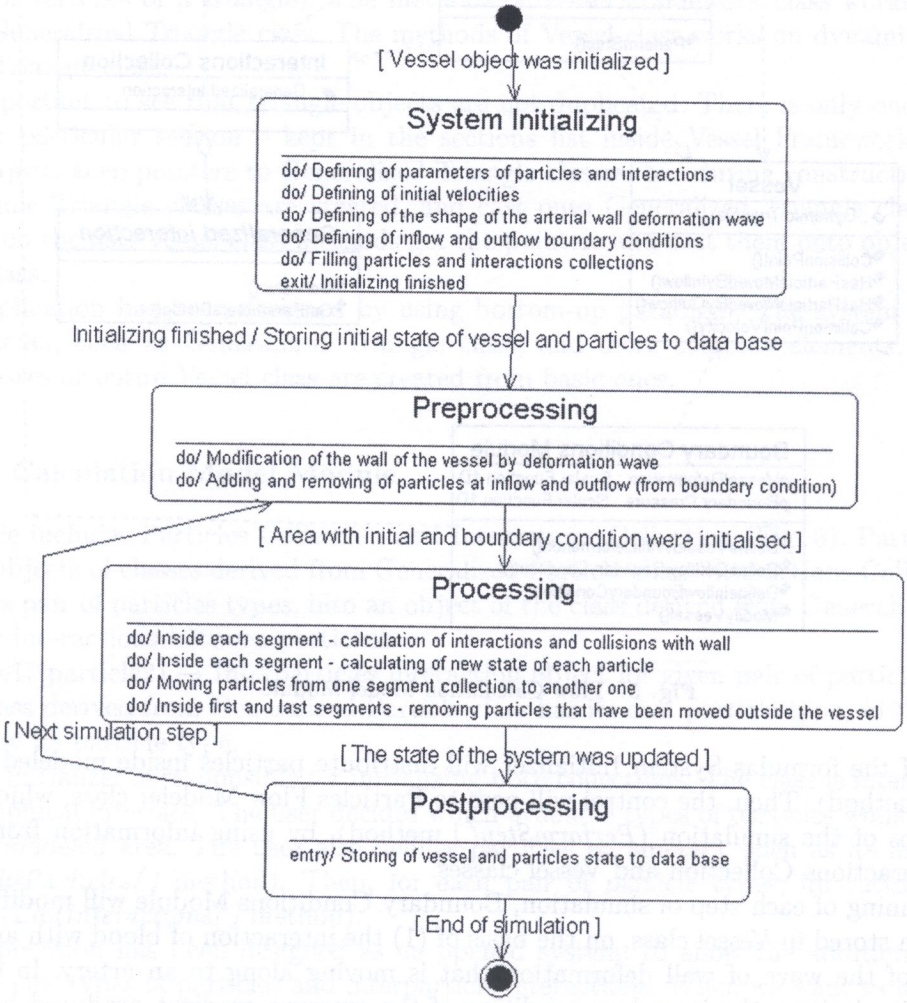

Fig. 6. State transition diagram

4.1. System initializing

In this state the following classes are working: System Initializator (that uses Particle Collection and Interaction Collection classes), Boundary Conditions (that uses Vessel class). System Initializator class is responsible for:

- Defining particles and interactions parameters.
- Defining initial velocity fields.
- Filling particles and interactions collections.

As it was mentioned in the section *Boundary Conditions Module*, it is possible to define simplified or precise arterial walls – modeled fluid interactions.

Boundary Conditions class is responsible for

- Defining the shape of the wave of arterial wall deformation translocating from the inflow to the outflow of modeled area, when the simplified interactions model has been used (see section *Boundary Conditions Module*).
- Defining the shape of the wave of arterial wall deformation at the inflow of modeled area, when the precise interactions model has been used (see section *Boundary Conditions Module*).
- Defining the outflow and inflow boundary conditions (in consecutive steps of simulations, the boundary condition of the pressure gradient produced by the heart eruption may be realized by adding new particles with given velocities on the boundary of modeled area).

4.2. Preprocessing

Boundary conditions should be taken into account at the beginning of each time step calculations.

4.3. Processing

Interparticle interactions are calculated, and the state of each particle is updated. Particles are stored in collections for particular segments of the vessel. To calculate the value of the force exerted on one particle, it is necessary to take into account its neighboring particles positioned far to R_{cut} boundary distance. For particles from one segment

- If the particle's distance from the boundary of the segment is superior to R_{cut}, it is enough to browse only particles from the segment, and take into account these particles that are distances from the particle less than R_{cut}.
- If the particle's distance from the boundary of the segment is inferior to R_{cut}, it is necessary to browse particles from the segment and from neighboring one, and take into account particles distances less than R_{cut}.

After calculating values of interactions between one particle and all of its neighboring particles, the entire force value exerted on the particle is given as summarized interactions value. The state of the particle is then actualized.

For the particle it is also necessary to check if the particle was deflected from the wall of modeled vessel, and if it was, the state of the particle should be modified. Arterial wall – particles interactions should be taken into account. At the boundary segments, it is also possible that particle was moving outside of the vessel. In such a situation, it is necessary to remove particle from the collection.

4.4. Postprocessing

New configuration of the vessel, modified by Boundary Conditions class, is saved to Data Base. The graphical interface of User Interface Module allows the user to browse simulations results from calculated up to now time steps of simulations, while performing the simulation. It is also to possible to calculate different necessary statistics from simulation results stored at Data Base.

5. The application of the software system to the problem of energy and mass transport in large arteries in man

5.1. Introduction

Human beings are warm-blooded. A high level of oxygen metabolism is essential in maintaining this condition. This enables the human organism to be active for the whole year, independent of the climate zone. That is why a quick transport of oxygen, nutritional substances and heat throughout the body as well as the removal of wastes is crucial for the living tissues. Otherwise malfunctions and even the necrosis of tissues may occur.

The cardiovascular system can be divided into active and passive parts. The heart is the active part. It works in a cyclic way, generates pulse waves which carry the energy necessary to pump the blood into the capillaries and replenish the outlay of blood in the arterial part of the cardiovascular system [17]. Arteries, arterioles, capillaries and veins form the passive part of the cardiovascular system [15, 20]. As the heart works in cycles, the arterial response is cyclic too [4, 30]. During each cycle the heart produces a pulse wave which is moved along the arterial tree and is reflected back to the heart [30]. The average speed of the pulse wave propagation is about 3 to 15 m/s [19]. The pulse wave should be distinguished from the blood flow, which has a compound pulsatile character and whose maximum velocity of flow is about 1 m/s [19].

It must be emphasized that single spurt of blood (about 70 ml) occupies about 17 cm of the beginning of the aorta [22] and this quantity of blood will reach capillaries after several heart evolutions. Otherwise pulse wave because it travels with much higher velocity and achieves arterioles before next heart evolution begins.

As it mentioned earlier, pulse wave reflects in peripheral part of arterial tree, and travels backward to the heart. In physiological conditions pulse wave achieves the beginning of aorta when aortic valve is closed, and boosts coronary blood flow [31].

The simulation of the phenomenon of energy and mass transport has been performed on one part of human carotid artery, but valid results may also be obtained on aorta or any large artery (on the large vessels, with diameter from a few mm to a few cm, belonging to the arterial component of the circulation system).

In order to understand the normal and the pathologic behavior of the human vascular system, detailed knowledge of blood flow and the response of blood vessels is required. This is possible with the help of numerical models of blood flow in arterial part of the circulation system. In recent years, computational techniques have been increasingly used by researchers seeking to understand vascular hemodynamics.

Numerous one-dimensional analyses have been performed whereby arteries have been approximated as branched channels [24]. These approximations, although constituting a necessary first step in the development of the computational framework, are largely inapplicable to arterial flow [37].

The example of two-dimensional modeling is the work [35]. The peristaltic blood flow was represented by two layers of axisymmetric 3D fluid model in cylindrical coordinates system, where the core region with erythrocytes was assumed to be Casson fluid and peripheral plasma region was established as Newtonian fluid. Analytical results presented in [35] assume that there is a zero radial pressure gradient and the problem is axial symmetrical. Variational formulations of analytical models formulated in cylindrical coordinate system can be singular. Another disadvantage of that

model is that information about the locations of surface which separate different kinds of flows are necessary. It makes that model impractical.

In contrast to the large number of two-dimensional solutions, few solutions to the three dimensional, transient flow equations governing blood flow have been performed. The work of Perktold [27] and the work of Ethier [7], who are investigating pulsatile flow in three-dimensional models of bifurcations and bypass grafts, are of particular interest. The most interesting three-dimensional model governing the entire arterial part of the circulation system is the work of Hughes [37]. Also, the author has performed classical FEM calculations of the pulsatile blood flow in arteries [26].

Three-dimensional models were concentrated on the classical continuous methods such as Finite Element Method or Finite Differences Method.

Classical methods used for the issue of blood flow in arteries produce many problems such as numerical costs, complications of variational formulation, and problem's sensitivity to boundary conditions. The appearance of the problems during classical methods was described in publications [29, 37] and [26]. Many problems which arose during the investigations have inclined the author to try to overcome the disadvantages of continuous models, by using Fluid Particle Model invented by Español [6].

5.2. The aim of modeling

The aim of modeling is to confirm the hypothesis that heart produces pressure gradient during blood eruption, that is initializing spontaneous pulse wave propagating along the arterial tree and transporting energy to propel blood flow in capillaries and refills arterial tree with blood volume expended to maintain tissue blood supply.

Pulsatile blood flow in arteries may be understood as an assembling of two physical phenomena: (1) transport of energy, (2) mass transport with much slower velocity [10, 30].

The phenomenon of the transport of energy could be divided into two parts. The first, propagation of transversal wave along the elastic walls, and the second, the propagation of longitudinal wave along the fluid filling the vessel.

However fluids are not compressible and pressure wave in a stiff pipe is very fast, when the tube walls are elastic the wave velocity is much slower because fluid mass is moving temporarily transversal and simultanously potential energy is cumulated in strained vessel wall and then it is converted again in momentum spread in fluid in terms of rush interchange between molecules. That's why the transports of energy and mass are divided. The pulse wave conveys energy with velocity 3–15 m/s [19] and during heart ejection it reaches peripheries of the arterial system and after reflection comes back to ascending aorta. Whereas the mass transport takes place with maximum speed of about 1 m/s [19]. The ejected blood takes up about 17 cm of the beginning of aorta [22] and reaches capillaries after several ejections of the heart, so that heart function is dual: first it produces pulses of energy propagating very fast and driving flow in capillaries and second it refills arterial tree with blood.

5.3. Medical measurements

Medical measurements has been performed in case of the pulsatile blood flow in human carotid artery.

The aim of measurements was to establish

- The shape of the arterial wall deformation induced by transversal wave contribution to the pulse wave.
- The blood flow velocity distribution.
- The speed of pulse wave propagation.

It should be emphasized that all data needed for the simulation could be obtained by non invasive technique as Doppler flowmetry, applanation tonometry and cuff manometry.

For clinical measurements ultrasound device Aloka SSD-110 equipped with pulsed wave system for blood velocity was used. Sector probe 7.5 MHz was applied on the carotid skin and duplex presentation was used to place Doppler sample within carotid artery and the angle between ultrasound beam and long axis of vessel was taken into account during velocity estimation. More information about the use of ultrasound in pulse-wave and blood velocity estimation can be found in [13, 23].

Figure 7 presents pulse wave shape measurements made by using ultrasonographic station Aloka SD-1100. On the left side of the picture there is an ultrasonographic view of the human carotid artery. The artery is directed perpendicular to the picture area. On the right side there is a sequence of snap-shots presenting the artery cross-sections made in regular time intervals. The pulse wave shoulder is denoted by a square. If we denote by ds – the length of time interval and V – pulse wave propagation speed, it follows that the length of this shape would be equal to $s = V \cdot dt$. Also the diameter of the vessel may be easily established from the measurements.

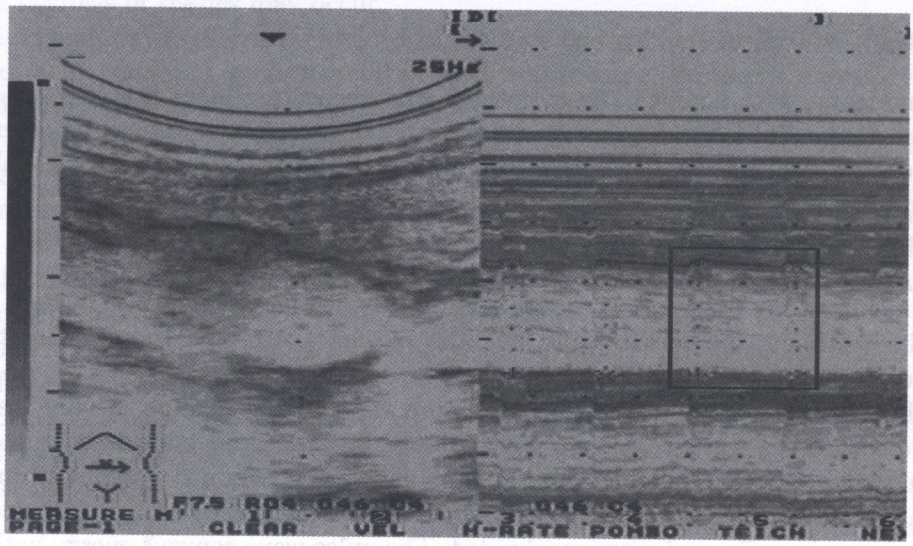

Fig. 7. Peristaltic wave shape measurement

Figures 8, 9 presents blood flow velocity distributions measured on human carotid artery again by using ultrasonographic station Aloka SSD-1100. On the left side of the picture there is an ultrasonographic view of the human carotid artery. The artery is directed parallel to the picture area. The placement and size of the range gate are determined by the user, and this determines the time instance for the sampling operation. The range gate length determines the area of investigation and sets the length of the emitted pulse [13]. On the right side there is a sequence of snap-shots presenting the blood flow velocity distributions made in regular time intervals. Horizontal axis denotes time, vertical axis denotes components of the velocity distribution.

In the Fig. 8, the range gate is set to measure velocity distributions from the central part of the carotid artery.

In the Fig. 9, the range gate is set to measure velocity distributions close to the arterial wall.

Pulse wave velocity can be estimated using two sensors, usually one is placed over carotid artery and second over femoral or radial artery [23]. Pulse wave velocity is a result of division of distance by time of pulse lag. Pulse wave velocity was not measured here, its value was taken from [19]. For the carotid artery usually it ranges from 8 to 12 m/s.

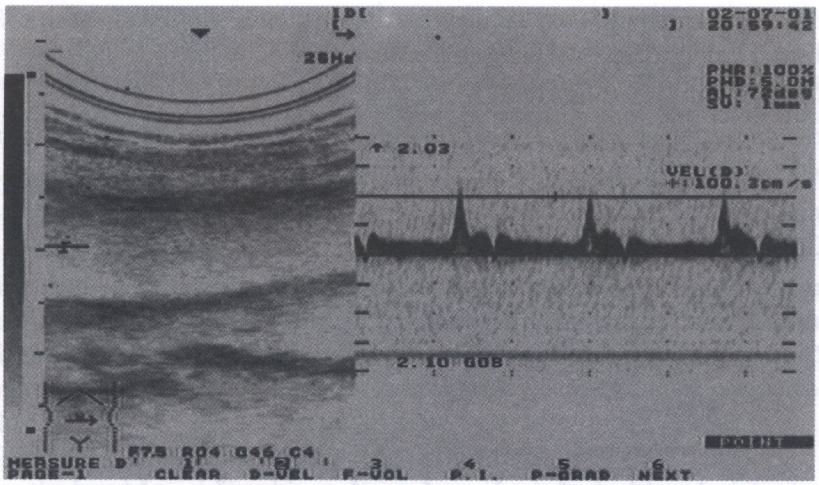

Fig. 8. Blood flow velocity distribution measurement at the central part of the carotid artery

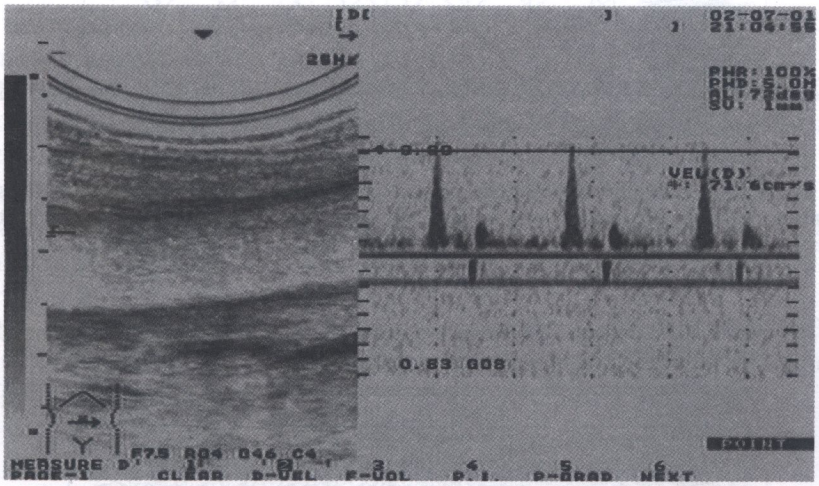

Fig. 9. Blood flow velocity distribution measurement close to the carotid artery wall

5.4. Simulation results

The software system that performs simulation of flows by using Fluid Particle Model has been used for simulating the phenomenon of energy and mass transport on one part of the carotid artery in man.

As it was described in the section *Boundary Conditions Module*, the software system allows to define simplified or comprehensive arterial walls – modeled fluid interactions. The version of the application used to simulate the phenomenon of energy and mass transport in artery doesn't include any mathematical description of the arterial wall dynamics. Also the simplified interactions have not been used – the modeled vessel contains rigid tube shape wall. However our model doesn't include vessel elastic walls, the definition of interparticle interactions allows to simulate transport of energy by exchange of particles momentum.

FPM simulation has been performed on the tube shape area with diameter of 6 mm and length of 1 cm. Number of particles on 1 cm length tube area was established as $N = 10^4$.

The simulation was performed with the length of time step $\Delta t = 10^{-4}$. The application of the software system to the blood flow problem may be developed by performing simulation of the flow on entire arteria part of the circulation system in man, for the time interval equal to entire heart

cycle. Because FPM algorithms are easy to distribute [5], such a simulation seems to be possible by performing distributed calculations within the network.

R_{cut} value has been established as equal to $R_{\text{cut}} = 10^{-3}$.

The simulation was performed by using relatively slow, Pentium 133 MHz computer. The simulation has governed 10 time steps, so the length of modeled time interval was equal to 10^{-3} s.

The interparticle forces have been established so as to match rheological properties of the blood, understood as a Casson fluid [8].

The pictures in the Figs. 10, 11 that present particles in the vessel, have been obtained by using User Interface Module. Also the statistics presented in the sub-section *Comparison of simulation results to medical measurements data* have been calculated within the User Interface Module.

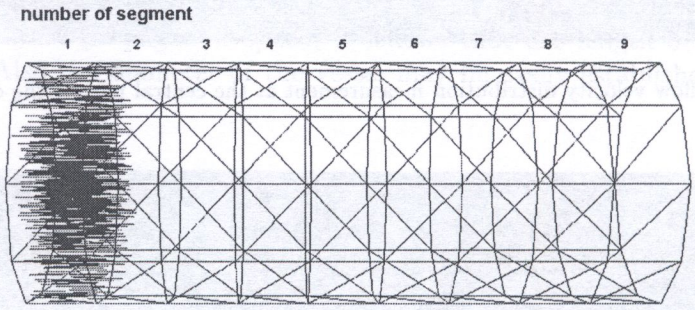

Fig. 10. Initial particles velocity profile. On the picture, only the moving particles (particles with non-zero velocity vectors) are indicated by drawing they velocity vectors that were scaled proportionally to they lengths. The diameter of the modelled vessel is equal to 6 mm. The length of the vessel is equal to 1 cm

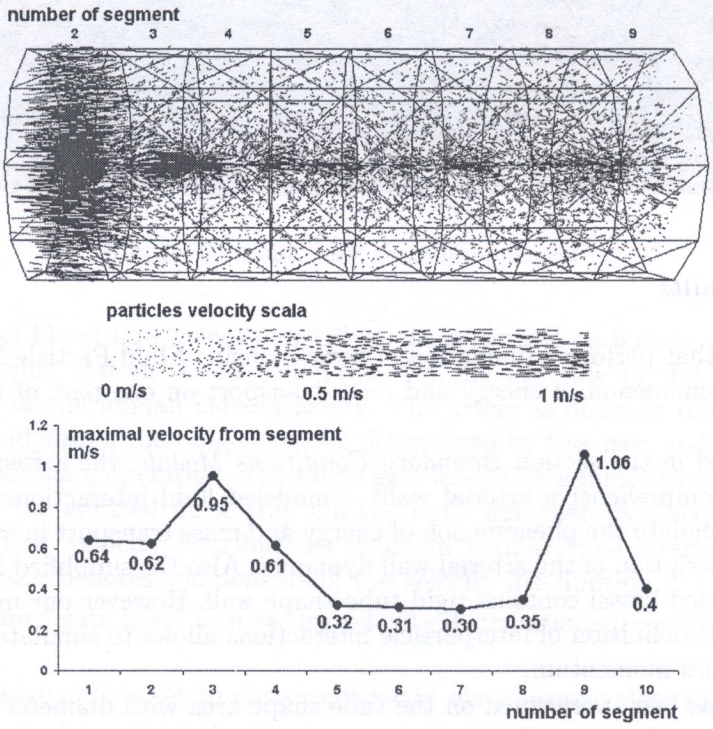

Fig. 11. Pulse wave propagation along the artery – results after 10 time steps of the simulation. On the picture, only the moving particles (particles with non-zero velocity vectors) are indicated by drawing they velocity vectors that were scaled proportionally to they lengths. The diameter of the modelled vessel is equal to 6 mm. The length of the vessel is equal to 1 cm

To simulate the phenomenon of energy and mass transport in artery, there should be an initial velocity distribution that produces pressure gradient at the inflow of the modeled area. It was assumed that the velocity of the particles filling the first segment of the vessel was equal to 1 m/s directed along to the artery axis (see Fig. 10). The remaining particles from other segments were quiescent.

In the Fig. 10 and Fig. 11 only the moving particles (particles with non-zero velocity vectors) are indicated by drawing they velocity vectors that were scaled proportionally to they lengths. The distribution of particles within entire modeled area remains almost uniform during the simulation. The particles with the velocities about 0 m/s were not indicated.

As a result of interparticle interactions, in consecutive time steps of the simulation, momentum transfer occured, thus creating particles velocity profile.

It should be emphasized that the momentum exchange between particles overtakes particles with the maximal velocity.

The maximal particles velocity is of the order of 1 m/s.

Time step length, has been defined so as to ensure the translocations of particles with maximal velocity so that they were small in comparison with particles and vessel dimensions.

The particles moving with the velocity of the order 1 m/s will translocate up to 0.1mm during one time step of the simulation.

After a few time steps of the simulation the primary uniform velocity distribution in the first segment became as presented in Fig. 11.

The Figure 11 presents results obtained after 10 time steps, with the initial velocity profile from Fig. 10. At the top of Fig. 11 there is a simulation domain (part of the carotid artery) filled with the particles. The vessel has been divided into 10 segments. The first segment is not shown on the pictures. At the right side of Fig. 11 there is a grayscale that helps to read particles velocities from the pictures above. At the bottom of the Fig. 11 there is a diagram that shows maximal values of the particles velocities from particular segments of the vessel, measured after 10 time steps of the simulation.

The Figure 12 shows velocity distribution from segments number 3, 4 and 7, after 8 time steps of the simulation. The Figure 13 shows velocity distribution from segments number 6, 7 and 9, after 10 time steps of the simulation.

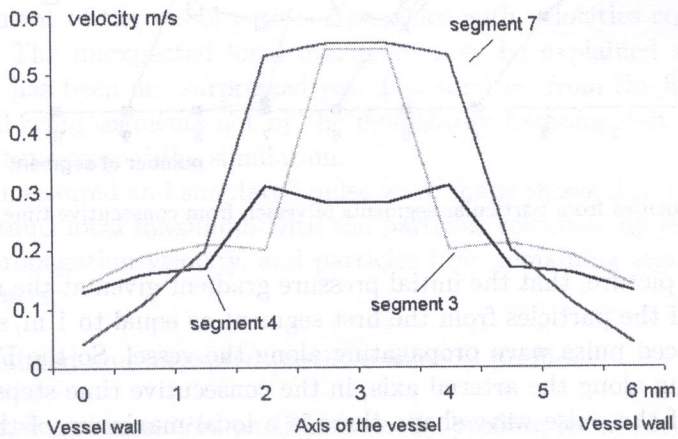

Fig. 12. Velocities profiles from 3, 4 and 7 segment of the vessel, after 8 time steps of the simulation

In the general, the velocities profiles are as follows: close to the arterial axis, the velocities have maximal values, then they are decreasing along the artery radius to be of the order of a few cm/s when close to the arterial wall.

The calculated velocity distribution was created spontaneously even though a plane velocity distribution at the entry of the artery had been assigned. It should be emphasized that obtained velocity profile was created only by momentum exchange, and not by translocations of particles.

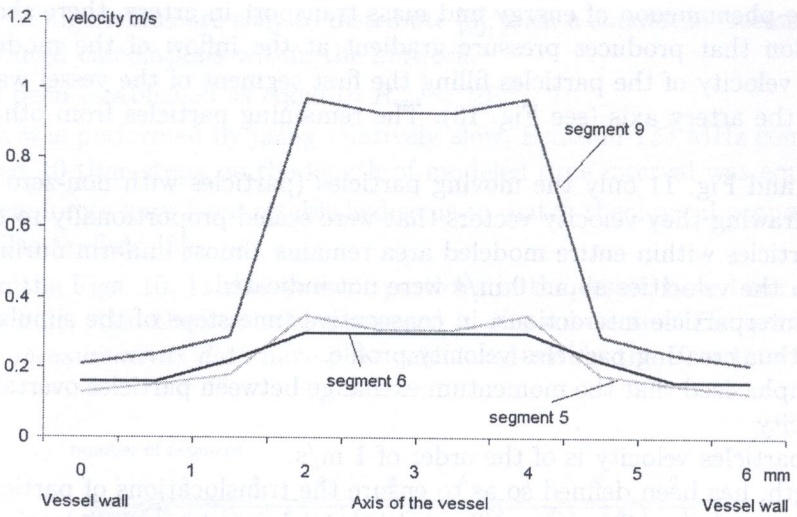

Fig. 13. Velocities profiles from 6, 7 and 9 segment of the vessel, after 10 time steps of the simulation

In the Fig. 14 there are maximal values of the particles velocities (maximum from the velocity profile in the segment, usually located at the central part of the segment, at the axis of the vessel) from particular segments of the vessel, measured after 2, 4, 6, 8 and 10 time steps of the simulation.

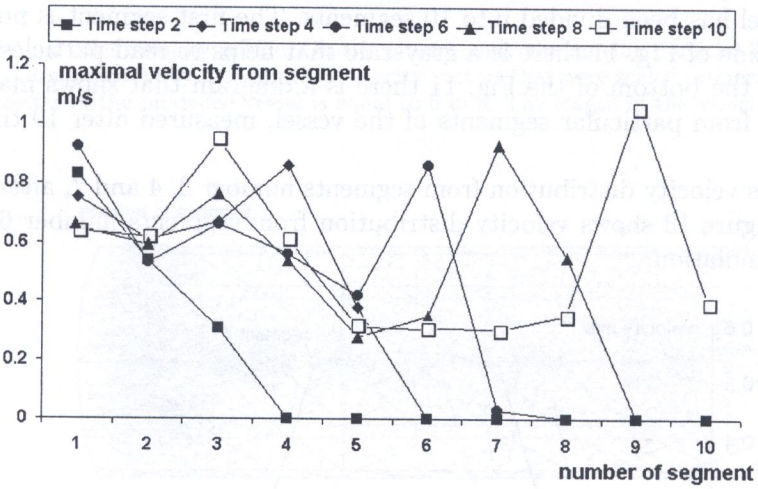

Fig. 14. Maximal velocities from particular segments of vessel, from consecutive time steps of the simulation

It is clear from the picture, that the initial pressure gradient given at the inflow of the vessel (by establishing velocity of the particles from the first segment as equal to 1 m/s and directed along to vessel axis) has produced pulse wave propagating along the vessel. So the Fig. 14 shows the pulse wave shape propagating along the arterial axis, in the consecutive time steps of the simulation.

At the beginning of the pulse wave shape there is a local maximum of the order of 1 m/s (the same as the initial pressure gradient), that is translocating along to the vessel with the velocity of order 10 m/s ($v = \frac{ds}{dt} = \frac{10^{-2}}{10^{-3}} = 10$ m/s, ds = the length of modeled vessel, dt = 10 time steps 10^{-4}s $= 10^{-3}$). After the local maximum, particles velocities are decreasing to the value of about 30 cm/s. At the end of the pulse wave shape, there is an another local maximum – it is the initial pressure gradient that has not been suppressed yet.

It should be emphasized again that the pulse wave propagation speed equal to the velocity of the momentum exchange between particles (of order 10 m/s) is faster than particles with the maximal velocity (of order 1 m/s).

5.5. Comparison of simulation results to medical measurements data

- Comparison of pulse wave propagation velocities

To compare simulated and measured velocities of pulse wave propagation, let us focus in Fig. 14 and Fig. 8. From the Fig. 8 it follows that the shape of pulse wave propagating along to the artery contains local maximum with the particles velocities up to 1 m/s. From the Fig. 14 it follows that the simulated pulse wave shape (created spontanously from assumed pressure gradient at the inflow of the artery), contains also local maximum of the particles velocities, at the beginning of the shape.

We identify pulse wave propagation velocity with the velocity of propagation of local maximum from the pulse wave shape, where the particles are moving with the velocity up to about 1 m/s.

From medical measurements it follows that the pulse wave propagation velocity in the carotid artery usually ranges from 8 to 12 m/s (see sub-section *Medical measurements*). From the simulation results it follows that the local maximum is propagating with the velocity of order 10 m/s (see sub-section *Simulation results*).

The comparison of measured pulse wave velocity (from 8 to 12 m/s) to simulated pulse wave velocity (of order 10 m/s) confirms the validity of obtained propagation velocity results.

- Comparison of simulated and measured pulse wave shapes

To compare simulated and measured shapes of pulse wave, let us focus in Fig. 8 and Fig. 11. From the Fig. 8 it follows that the shape of pulse wave propagating along to the artery contains local maximum with the particles velocities up to 1 m/s, and in the remaining part of the pulse wave, the particles are moving with the velocities of order of 30 cm/s.

From the Fig. 11 it follows that the simulated pulse wave shape measured after 10 time steps of the simulation, also contains local maximum with the particles velocities up to 1 m/s within the segments number 9 and 10. In the remaining part of the pulse wave, the particles velocities are decreasing to the velocity of order of 30–35 cm/s, in the segments number 5, 6, 7 and 8. The only difference is another local maximum near the segment number 3, with the particles velocities up to 1 m/s.

The simulation results has been obtained from assumed initial pressure gradient, that was created by filling the first segment of the vessel with the particles with velocities equal to 1 m/s, directed along the vessel axis. The unexpected local maximum may be explained as remaining of initial pressure gradient that has been not surpressed yet. The particles from the first segment have been translocated to the following segments not by the momentum exchange but by translocating small distances within each time step of the simulation.

The comparison of measured and simulated pulse wave shape shows that the simulated as well as the measured one contains local maximum with the particles velocities up to 1 m/s that is moving with the pulse wave propagation velocity, and particles from remaining segments are moving with the velocity about 30 m/s.

- Comparison of simulated and measured particles velocity profiles

To compare simulated and measured particles velocity profiles, let us focus in Fig. 8, Fig. 9 and Fig. 12 with Fig. 13. Let us compare velocity profiles close to the local maximum of the pulse wave, and from the remining part of the pulse wave.

From the Fig. 8 it follows that close to the local maximum of the pulse wave, in the central part of the vessel, particles velocities are of order of 1 m/s, and particles velocity distribution contains only velocities close to the value. It follows that in the central part of the vessel, particles are moving only with the velocities close to 1 m/s and not smaller.

From the Fig. 9 it follows that close to the local maximum of the pulse wave, close to the arterial wall, particles velocity distribution contains values from 0 m/s to 1 m/s. It follows that close to the arterial wall, particles velocities are decreasing from the value of 70 cm/s to 0 m/s.

The Figure 12, contains particles velocities distributions from the segments number 3, 4 and 7 measured after 8 time steps of the simulation. The segment 7 contains velocity distribution close to the local maximum of the pulse wave (see Fig. 14). It follows that in the central part of the vessel, particles velocities within the segment 7 are close to 1 m/s. Local minimum at the central part of the velocity profile from segment 7 is small, compared with the velocity distribution of the particles from the local maximum of the pulse wave. From the distance of 2 mm from the arterial wall, the velocities begin decreasing to the value of 20 cm/s close to the arterial wall. The result is comparable to the measured decreasing of the velocities close to the arterial axis.

From the Fig. 8 it follows that in the remaining part of the pulse wave, far from the local maximum, close to the arterial axis, particles velocity distribution contains values from 0 cm/s to 30 cm/s. From the Fig. 9 it follows that in the remaining part of the pulse wave, far from the local maximum, close to the arterial wall, particles velocity distribution contains values up to 10 m/s.

Let us focus again in the Fig. 12. The segments 3 and 4 contain velocity distributions in the remaining part of the pulse wave, far from the local maximum. It follows that in the central part of the artery, in the segment 4, particles velocities are increasing up to 30 cm/s. Within the segment 3, in the central part of the artery, particles velocities are increasing up 50 cm/s, and it is the effect of unsuppressed initial pressure gradient. Close to the arterial wall, within the segment 4, as well as the segment 3, particles velocities are of the order of 10 cm/s.

The similar comparison may be performed with respect to the Fig. 13 with Fig. 8 with Fig. 9.

- Summary

The conclusions from the comparison between results of FPM simulation and the medical measurements data, are presented in Table 2.

Table 2. The comparison between results of FPM simulation with the medical measurements data

	Medical measurements	FPM simulation
Pulse wave velocity	8–12 m/s	10 m/s
Maximal velocity of particles	1 m/s	1 m/s
Pulse wave shape	presented in Fig. 8	valid
Particles velocity profile, at the arterial axis, close to the local maximum of the pulse wave shape	presented in Fig. 8 1 m/s	1 m/s
Particles velocity profile, at the arterial axis, in the remaining part of the pulse wave shape	presented in Fig. 8 30 cm/s	30 cm/s
Particles velocity profile, close to the arterial wall, close to the local maximum of the pulse wave shape	presented in Fig. 9 0–70 cm/s	30 cm/s
Particles velocity profile, close to the arterial wall, in the remaining part of the pulse wave shape	presented in Fig. 9 10 cm/s	20 cm/s

Presented here comparison of simulation results to medical measurements data confirm validity of numerical results. Only one unexpected effect is an unsuppressed initial pressure gradient.

6. Conclusions

- The software system that performs simulation of flows by using Fluid Particle Model has been used for modeling the physical problem of blood flow in large arteries in man. Obtained simulation results confirms the validity of object-oriented project of the system, and shows that the software system may be successfully used for wide variety of problems from the class of fluid flows problems in the area with moving boundary. Blood flow problem in large arteries is an example of such problem.

- The idea of generalized classes, such as Generalized Particle and Generalized Interaction allows to enlarge the system by addition of new particles and interparticle interactions classes. Thus, the system was designed as object-oriented, opened one.

- Bottom-up paradigm assumption that was taken into account during design process of the software system, confirms its validity and usefulness.

- The simulation of the flow problems by using presented object-oriented application with database access, is easy to perform and to verify. Data Base Access Module allows to store simulation results to the data base and to verify results from previous time steps of the simulation during the calculations of the following steps.

- The definition of the Vessel class as the central part of the system allows to perform parallel calculations of the fluid flow within Flow Calculation Module and arterial wall stresses within Arterial Wall Calculation Module.

- The problem of pulsatile blood flow in arteries is hard to solve by using classical methods such as FEM [26, 37]. The propagation of the pressure wave along the artery was successfully simulated by using FPM application created on the basis of presented object-oriented design.

- Numerical costs of the classical FEM algorithm applied to the problem of blood flow in artery consumes many hours of super computer time [37]. Accurate calculations of the blood flow requires generating and solving up to 250000 equations for 200–500 time steps [37]. Numerical costs of FPM simulations are connected with the establishment of the state of each particle by calculating its interactions with neighbouring particles, in consecutive time steps. One step of FPM simulation with 10^4 particles takes about 1 hour on Pentium 133 MHz, and gives reasonable precision of results. Particle algorithms are perfect both for parallel computation and for distribution within the network [5]. The quantity of particles within the FPM model is limited by the calculation power of the fastest parallel computers and is equal to 10^9 [5].

- The simulation has been performed on relatively slow Pentium 133 MHz computer. The application of the software system to the blood flow problem may be developed by performing simulation of the flow on entire arterial part of the circulation system in a man. Such a simulation seems to be possible to perform by using faster computer, or by performing distributed calculations within the network.

- Because of pulsating character of the blood flow in central circulation system, the transition of the blood flow from laminar to turbulent is possible [21, 22, 34]. The modeling of turbulent flow using FEM is very difficult, if possible. As it was outlined in the sub-section *Basic steps of Fluid Particle Model* FPM, as opposed to classical methods, may be easily used for modeling the turbulent flow.

- As it was described in the *Boundary Conditions Module*, the software system allows to define simplified or comprehensive arterial walls – modeled fluid interactions. The version of the application used to simulate the phenomenon of energy and mass transport in artery doesn't include any mathematical description of the arterial wall dynamics (see sub-section *Simulation results*).

The application of the model to the blood flow problem may be developed by the introduction of simplified or comprehensive arterial walls – modeled fluid interactions.

For the definition of the comprehensive interactions, it is necessary to include the mathematical model of two-dimensional arterial wall dynamic to the Arterial Wall Calculations Module.

It would be interesting work to describe within the mathematical model, the fluid-solid body interactions as a biofeedback.

The biofeedback may be explained as a composition of mechanisms of optimal matching blood supply to local metabolic demand. Some examples of them are

(1) Passive interaction between blood pressure and wall distension.

(2) Active flow mediated vasodilatation. As blood flows through a conduit vessel, the vessel dilates in proportion to the increment in endothelial shear stress [11, 16]. This phenomenon requires endothelial integrity, and is therefore known as a flow-mediated endothelium-dependent vasodilatation [12, 28]. Fluid flow across the endothelial surface induces the release of EDRF (nitric oxide previously was known as epithelium derived relaxing factor, EDRF [9]) which is thought to be largely responsible for the vasodilatation in conduit vessels [16, 32].

(3) Negative feedback between pressure and smooth muscles tension of the vessels walls.

7. Acknowledgments

The author wishes to thank Robert Schaefer D. Sc. Eng., for priceless numerical consultations, and Dr. Andrzej Pelc for making ultrasound measurements, and for medical consultations.

References

[1] L. Arnold. *Stochastic Differential Equations: Theory and Applications.* R. Oldenbourg Verlag, Munich, 1973.

[2] G. Booch. *Object-oriented analysis and design with applications.* The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1994.

[3] G. Booch, I. Jacobson, J. Rumbaugh. *The Unified Modeling Language User Guide.* Adison-Wesley Pub.Co., 1998.

[4] C. G. Caro, T. J. Pedley, R. C. Schroter, W. A. Seed. *The mechanics of the circulation.* Oxford University Press, Oxford, 1978.

[5] W. Dzwinel, W. Alda, J. Kitowski, D. A. Yuen. Using discrete particles as a natural solver in simulating multiple-scale phenomena. *UMSI 99/131*, July 1999.

[6] P. Español. Fluid particle model. *Physical Review E*, **57**: 3, 2930–2948, 1998.

[7] D. A. Steinman, C.R. Ethier, X. Zhang, S. R. Karpik. The effect of flow waveform on anastomotic wall shear stress patterns. *ASME Adv. Bioengrgr.*, 173–176, 1995.

[8] Y. C. Fung. *Biomechanics.* Springer-Verlag, New York, 1984.

[9] R. F. Furchgott, J. V. Zawadzki. The obligatory role of endothelial cells on the relaxation of arterial smooth muscle by acetylcholine. *Nature*, **288**: 373–376, 1980.

[10] W. Harvey. *On the circulation of blood.* Blackwell, Oxford, 1957.

[11] S. M. Hilton. A peripheral arterial conduction mechanism underlying dilation of the femoral artery and concerned in functional vasodilation in skeletal muscle. *J. Physiol.*, **149**: 93–111, 1959.

[12] J. Holtz, U. Forstermann, U. Pohl, M. Geisler, E. Bassenge. Flow-dependent endothelium-mediated dilation of epicardial coronary arteries in conscious dogs: Effects of cyclo-oxygenase inhibition. *J. Cardiovasc. Pharmacol.*, **6**: 1161–1169, 1984.

[13] J. A. Jensen. *Estimation of blood viscosities using ultrasound.* Cambridge University Press, Cambridge, 1996.

[14] A. Kabała, P. Kalita. Application of thin shell theory for modelling dynamics of arterial wall. *Proceedings of Seventh National Conference on Application of Mathematics in Biology and Medicine*, 2001.

[15] S. Konturek. *Fizjologia człowieka, tom II, Układ krążenia.* wydanie VII poprawione, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków, 2000.

[16] M. Lie, O. M. Sjersted, F. Kill. Local regulation of vascular cross section during changes in femoral arterial blood flow in dogs. *Circ. Res.*, **27**: 727–737, 1970.

[17] R. C. Little. *Physiology of the heart and circulation*, Year Book Medical Publishers, Chicago, 1985.

[18] C. A. Marsh, G. Backx, M. H. Ernst. Static and dynamic properties of dissipative particle dynamics. *Physical Review E*, **56**(2): 1676–1691, 1997.
[19] D. A. McDonnald. *Blood Flow in Arteries*, Edward Arnold, London, 1974.
[20] R. L. Memmler, D. L. Wood. *Structure and function of the human body.* J.B. Lippincott Company, Philadelphia, 1987.
[21] R. M. Nerem, W. A. Seed. An in vivo study of aortic flow disturbances. *Cardiovascular Research*, **6**: 1–14, 1972.
[22] R. M. Nerem, W. A. Seed, N. B. Wood. An study of the velocity distribution and transition to turbulence in the aorta. *J. Fluid Mech.*, **52**: part 1: 137–160, 1972.
[23] A. Nowicki. *Podstawy ultrasonografii doplerowskiej*, PWN, Warszawa, 1995.
[24] M. S. Olufsen. Structured tree outflow condition for blood flow in large systemic arteries, *Am. J. Physiol.*, Vol. 276, Heart Circ. Physiol., **45**: H257–H268, 1999.
[25] M. Paszyński. Calculations of blood flow in arteries – comparison of Fluid Particle Model with Finite Elements Methods. *Proceedings of Seventh National Conference on Application of Mathematics in Biology and Medicine*, Zawoja, 2001.
[26] M. Paszyński, A. Pelc. Numerical analysis of peristaltic blood flow in arteries. *CAMES*, **8**: 4, 2001.
[27] K. Perktold, M. Resch, R. Peter. Three-dimensional numerical analysis of pulsatile flow and wall shear stress in the carotid artery bifurcation, *J. Biomech.*, **24**(6): 409–420, 1991.
[28] U. Pohl, J. Holtz, R. Busse, E. Bassenge. Crucial role of the endothelium in the vasodilator response to increase flow in vivo. *Hypertension*, **8**: 37–44, 1986.
[29] A. Quarteroni, T. Massimiliano, A. Veneziani. Computational vascular fluid dynamics: problems, models and methods. *Comput Visual Sci.*, **2**: 163–197, 2000.
[30] M. F. Rourke, M. E. Safar. *Arterial vasodilation, mechanisms and therapy*, Edward Arnold, London, 1993.
[31] M. F. Rourke, R. P. Kelly. Wave reflection in the systemic circulation and its implications in ventricular function, *Journal of Hypertension*, **11**: 327–337, 1993.
[32] G. M. Rubanyi, J. C. Romero, P. M. Vanhoutte. Flow-induced release of endothelium derived relaxing factor. *Am. J. Physiol*, **250**: H1145-9, 1986.
[33] R. Schaefer, J. Krok, P. Leżański, J. Orkisz, P. Przybylski. Basic concepts of an open distributed system for cooperative design and structure analysis. *CAMES*, **3**: 169–186, 1996.
[34] W. A. Seed, N. B. Wood. Velocity patterns in the aorta. *Cardiovascular Research*, **5**: 319–330, 1971.
[35] V. P. Srivastava, M. Saxena M. A two-fluid model of non-Newtonian blood flow induced by peristaltic waves. *Rheol Acta*, **34**: 406–414, 1995.
[36] B. Stroustrup. *The C++ Programming Language*, Second Edition, Reading, Addison-Wesley Publishing Company, Massachusetts, 1991.
[37] C. A. Taylor, T. J. R. Hughes, C. K. Zarris. Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering*, **158**: 155–196, 1998.