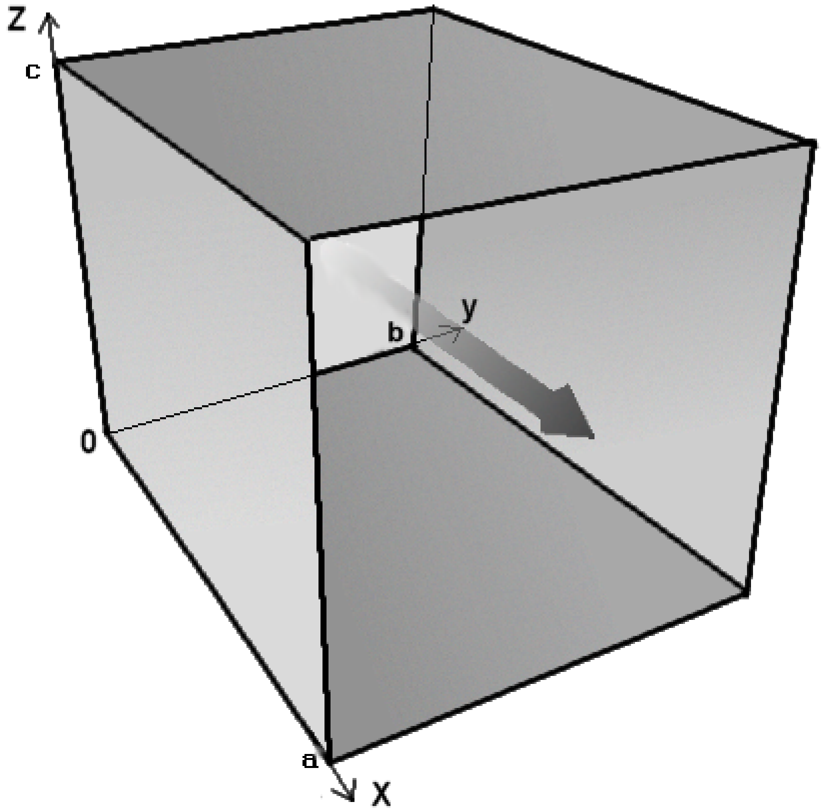
1 Description

The objective is an average field of turbulent flow in 3D area.

There is a rectangular area $\Omega = [0, a] \times [0, b] \times [0, c]$ where the flow is examined. A compressible viscous liquid flows into this area trough the bound at the plane OYZ. The liquid can freely flow out through the other bounds.



1.1 Boundary conditions

Boundary conditions correspond to the inflow of the liquid on the boundary where x = 0.

$$u\big|_{x=0} = U(y,z); \qquad v\big|_{x=0} = 0; \qquad w\big|_{x=0} = 0;$$
$$p\big|_{x=0} = P(y,z); \qquad \rho\big|_{x=0} = \rho(y,z); \qquad T\big|_{x=0} = T(y,z);$$

The other part of Ω boundary is marked as Σ. There are "free" boundary conditions on Σ:

$$\left.\frac{\partial u}{\partial n}\right|_{\Sigma} = 0; \qquad \left.\frac{\partial v}{\partial n}\right|_{\Sigma} = 0; \qquad \left.\frac{\partial w}{\partial n}\right|_{\Sigma} = 0;$$
$$\left.\frac{\partial p}{\partial n}\right|_{\Sigma} = 0; \qquad \left.\frac{\partial \rho}{\partial n}\right|_{\Sigma} = 0; \qquad \left.\frac{\partial T}{\partial n}\right|_{\Sigma} = 0;$$

1.2 Initial data

At the start time all variables are equal to the constants everywhere inside the area Ω.

$$u = U_{in}; \qquad v = 0; \qquad w = 0;$$
$$p = P_{in}; \qquad \rho = \rho_{in}; \qquad T = T_{in};$$

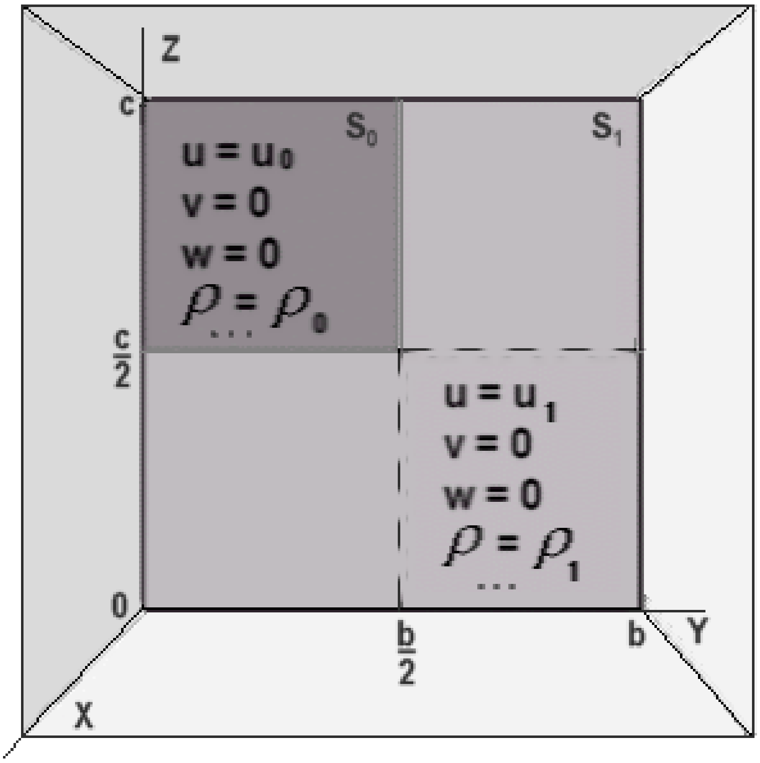
1.3 Modeling boundary conditions

The boundary conditions used in computations are described below

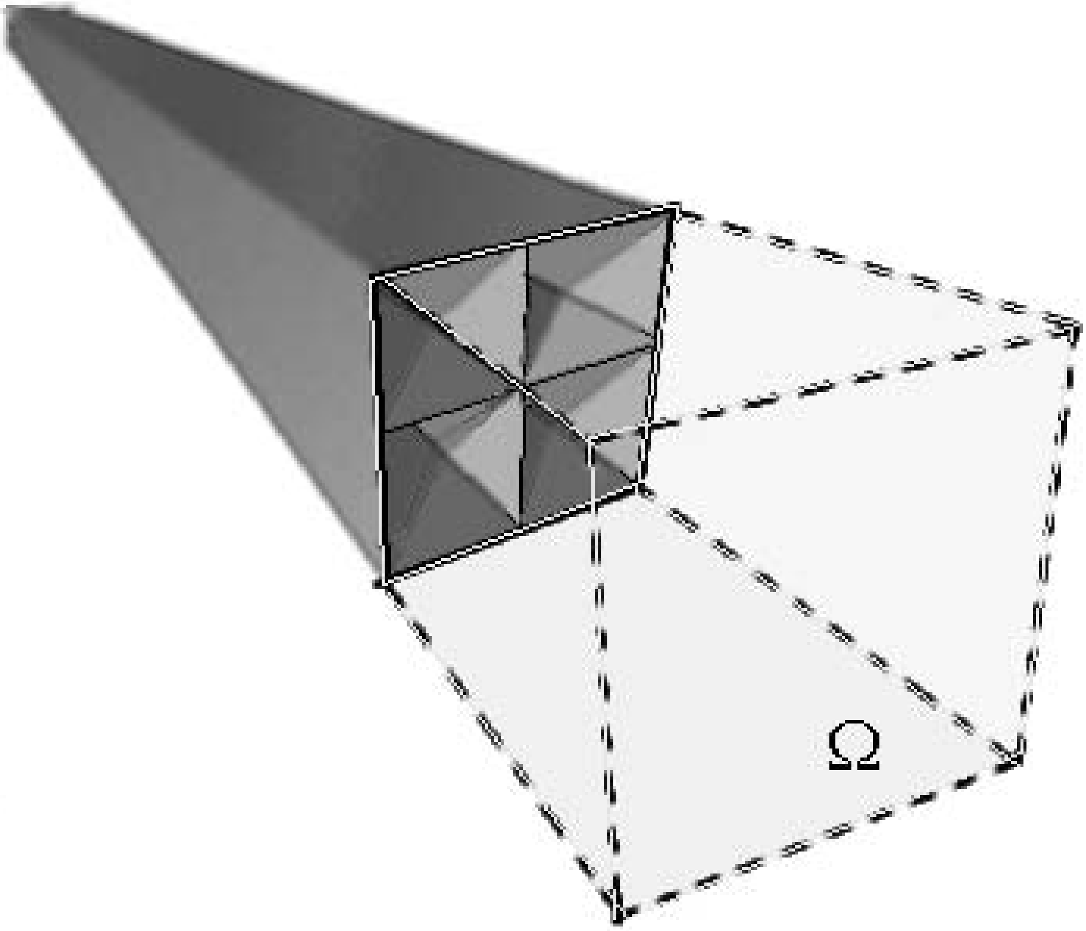
The boundary where liquid inflows is the rectangle with sides b and c. This rectangle is divided into four equal parts by lines that link the middles of it's sides. All variables are equal to the constants everywhere inside this parts. The values inside of three of this parts are equal. So this rectangle is divided into two areas - S_0 and S_1, where S_1-larger

$$u|_{S_0} = U_0; \qquad v|_{S_0} = 0; \qquad w|_{S_0} = 0;$$
$$p|_{S_0} = P_0; \qquad \rho|_{S_0} = \rho_0; \qquad T|_{S_0} = T_0;$$

$$u|_{S_1} = U_1; \qquad v|_{S_1} = 0; \qquad w|_{S_1} = 0;$$
$$p|_{S_1} = P_1; \qquad \rho|_{S_1} = \rho_1; \qquad T|_{S_1} = T_1;$$



This conditions may be interpreted as if our area Ω is sided to the edge of square tube with thin partitions inside.

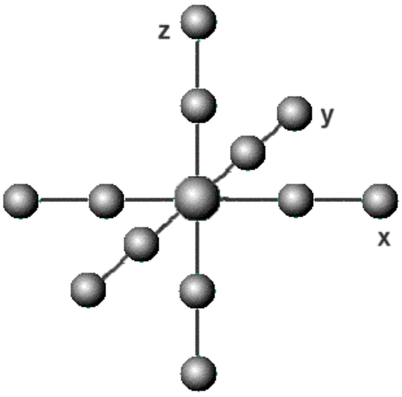


Ω

4 Difference scheme

The scheme used in this computations is the implicit Roe scheme. This program can use another implicit schemes provided in additional modules.

Template of the scheme include 13 nodes:



The description of this scheme is given below for the linear transport equation only.

Linear transport equation (mesh with constant step)

$$\frac{\partial Q}{\partial t}+\frac{\partial F(Q)}{\partial x}=0 \qquad F(Q)=aQ$$

$$\frac{\hat{Q}-Q}{\Delta t}+\frac{W_{i+1/2}\left(Q^{+}_{i+1/2},Q^{-}_{i+1/2}\right)-W_{i-1/2}\left(Q^{+}_{i-1/2},Q^{-}_{i-1/2}\right)}{\Delta x}$$

$$W_{i+1/2}=\frac{1}{2}\left[F^{c}\left(Q^{+}_{i+1/2}\right)+F^{c}\left(Q^{-}_{i+1/2}\right)\right]-\frac{1}{2}\left[F^{d}\left(Q^{+}_{i+1/2}\right)-F^{d}\left(Q^{-}_{i+1/2}\right)\right]$$

$$F^{c}(Q)=F(Q)=aQ \qquad F^{d}(Q)=|a|Q$$

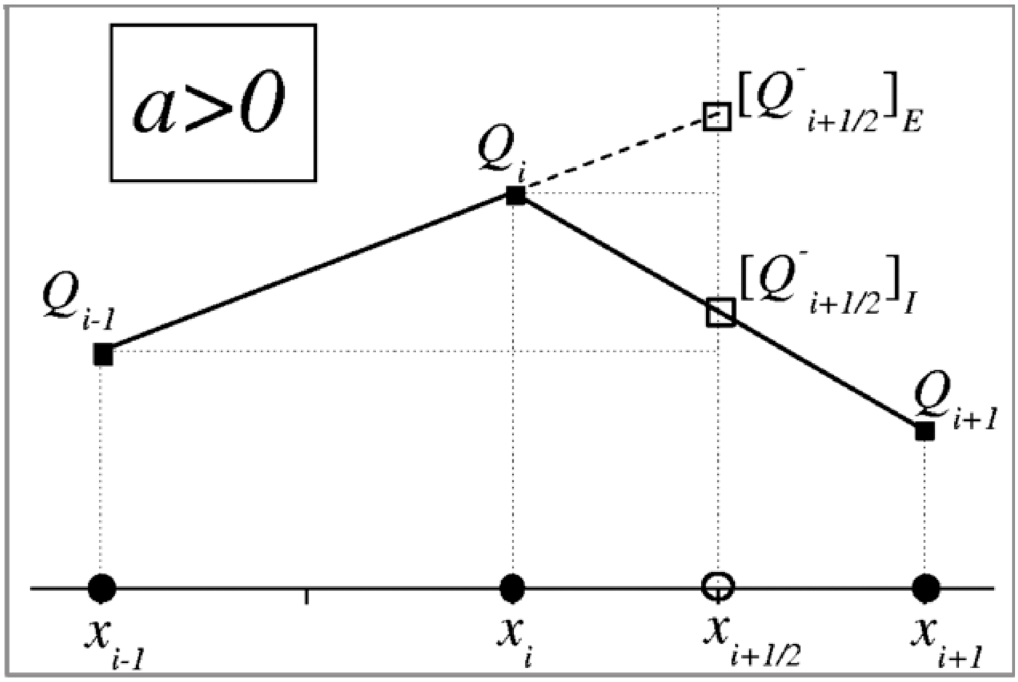
Constant approximation inside of mesh’s cell gives the first order

$$Q^{+}_{i+1/2}=Q_{i+1}, \quad Q^{-}_{i+1/2}=Q_{i}, \quad Q^{+}_{i-1/2}=Q_{i}, \quad Q^{-}_{i-1/2}=Q_{i-1}$$

Piecewise-linear approximation inside of cell:

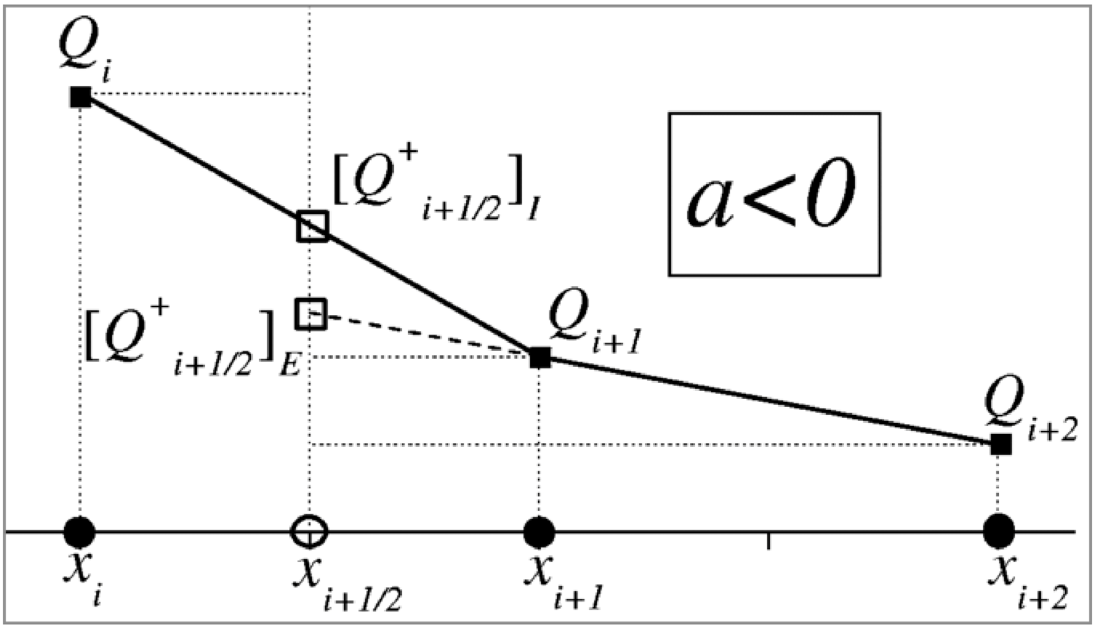
If a>0

$$\begin{cases}\left[Q^{-}_{i+1/2}\right]_{E}=Q_{i}+\dfrac{Q_{i}-Q_{i-1}}{2}\\[2ex]\left[Q^{-}_{i+1/2}\right]_{I}=Q_{i}+\dfrac{Q_{i+1}-Q_{i}}{2}\end{cases}$$



if a<0

$$\begin{cases} \left[Q^{+}_{i+1/2}\right]_E = Q_{i+1} - \dfrac{Q_{i+2} - Q_{i+1}}{2} \\ \left[Q^{+}_{i+1/2}\right]_I = Q_{i+1} - \dfrac{Q_{i+1} - Q_{i}}{2} \end{cases}$$



$$Q^{\pm}_{i+1/2} = \beta \left[Q^{\pm}_{i+1/2}\right]_E + (1-\beta)\left[Q^{\pm}_{i+1/2}\right]_I$$

5 Mesh

5.1 Generic description

The mesh used in this program is structured with variable step and rectangular cells. Our rectrangular area is $\Omega = [0,a]\times[0,b]\times[0,c]$. There are n nodes on the interval $[0,a]$ on the *X* axis, m on the interval $[0,b]$ on the Y axis and k on the interval $[0,c]$ on the Z axis. The bounds of this intervals are included in this sets of nodes. $\overline{[0,a]_n}$ -the set of nodes on the X axis, $\overline{[0,b]_m}$ - on the Y axis, $\overline{[0,c]_k}$ - on the Z axis.

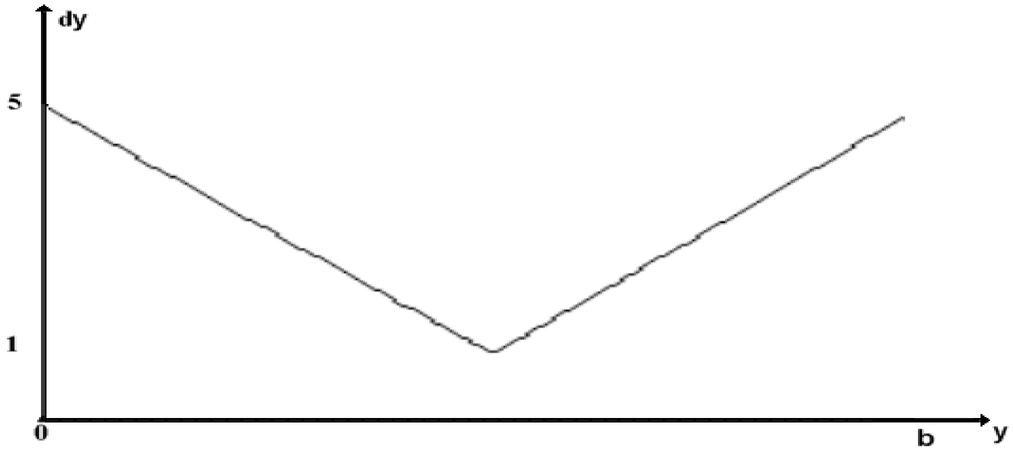
$\tilde{\Omega} = \overline{[0,a]_n} \times \overline{[0,b]_m} \times \overline{[0,c]_k}$ - all the nodes of the mesh.

5.2 The mesh used in computations

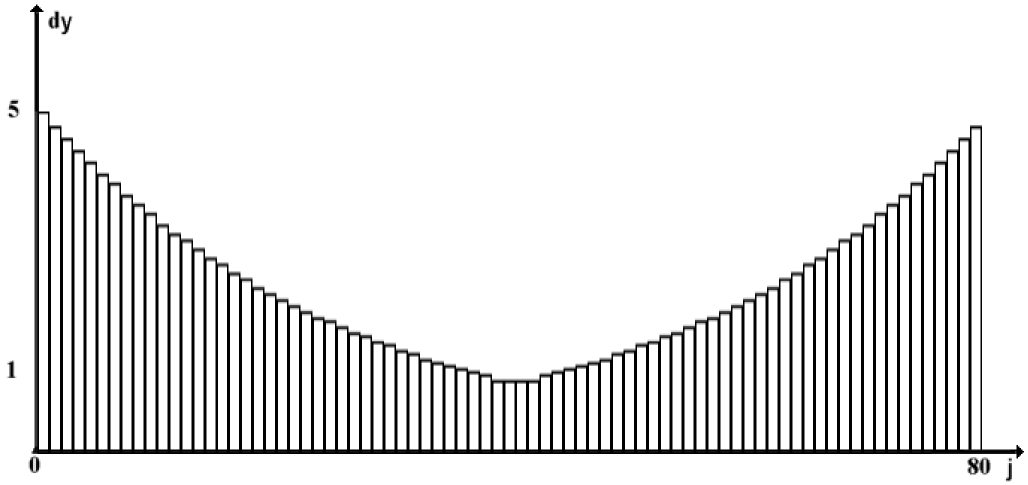
Step on Y axis is getting smaller when approaching a middle of the interval $[0,b]$ where mixing of the flows is taking place. Ratio between the maximal step and the minimal step on the Y axis is 5:1. The same step variation is on the Z axis.

The chart of step variation (depend upon the Y coordinate)

.



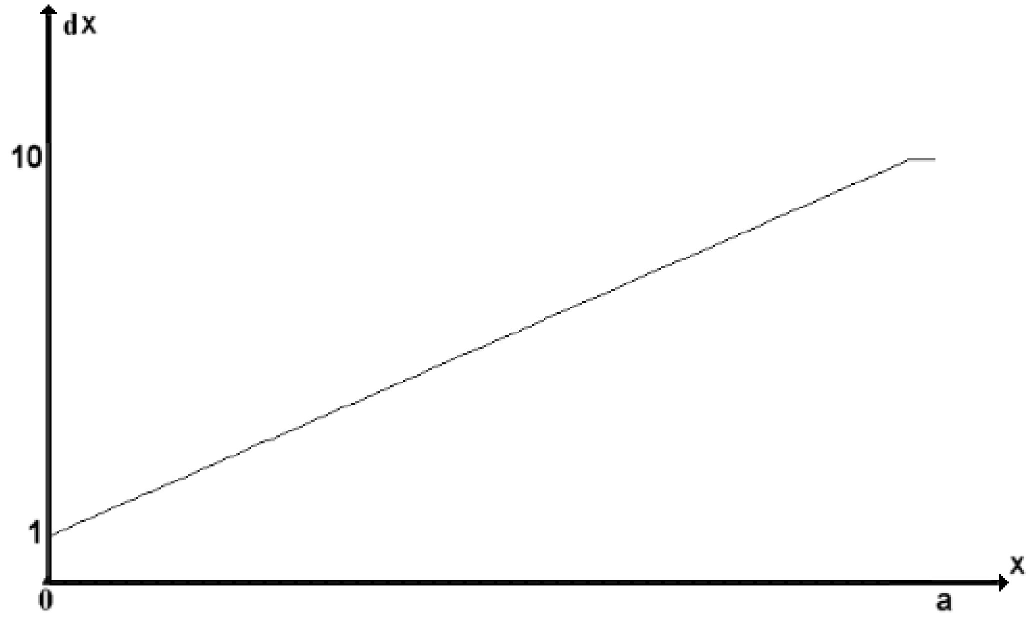
The chart of the step variation (depend upon the number of a node)



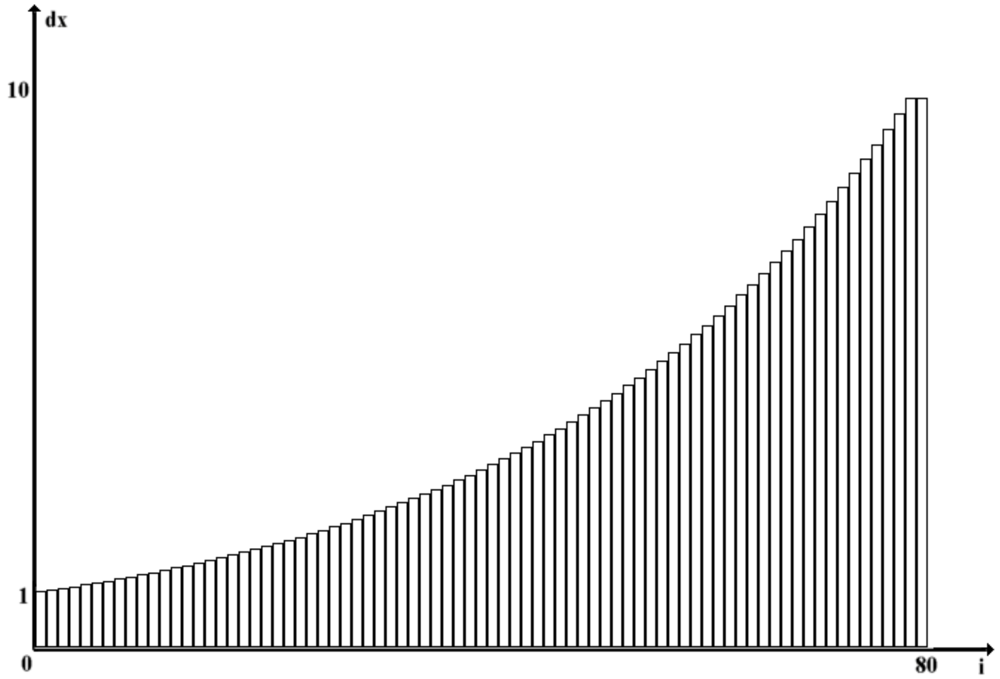
Step on the X axis is getting smaller near the bound x = 0 where the liquid inflows.

Ratio between the maximal step and the minimal step on the X axis is 10:1.

The chart of this step variation (depend upon the X coordinate)



The chart of the step variation (depend upon the number of a node)

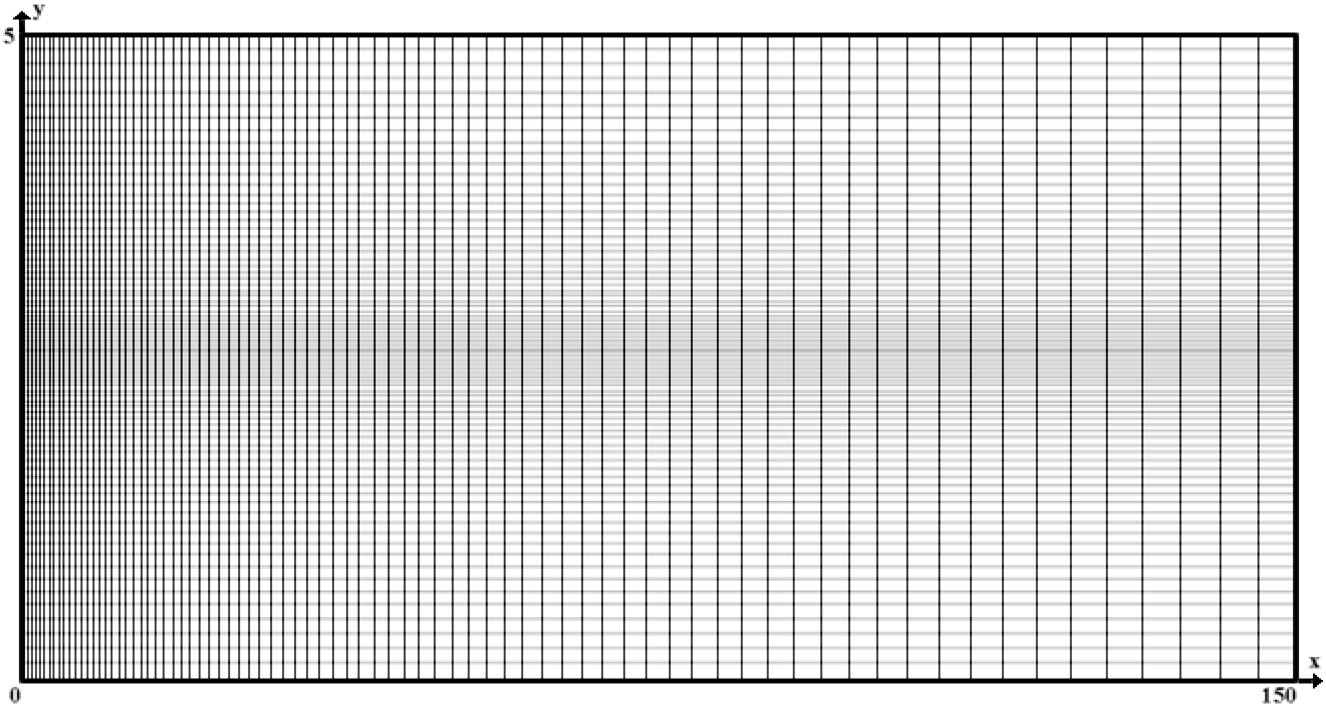


The ratio between the length of the interval [0,a] and the length of the interval [0,b] is 150:1.

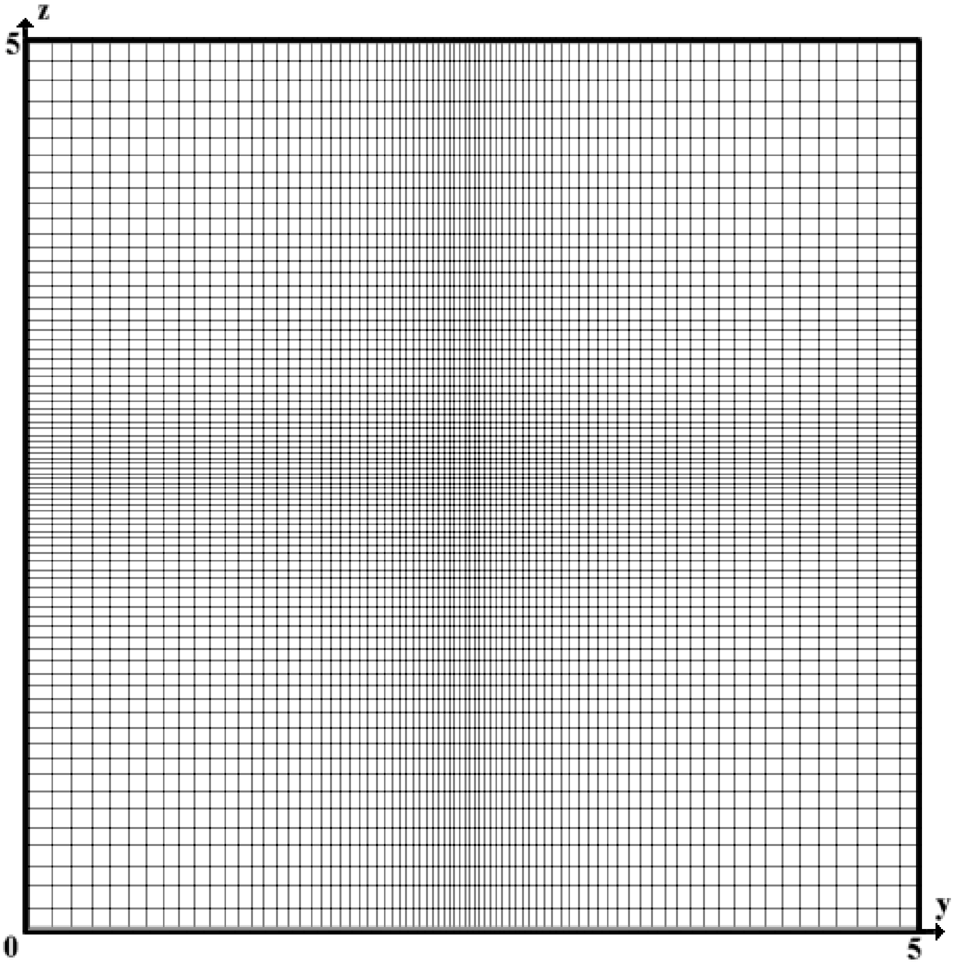
The length of the interval [0,b] is equal to the length of the interval [0,c]

The visual illustrations of the mesh structure is given below. *(The nodes is the points of concurrence of lines)*

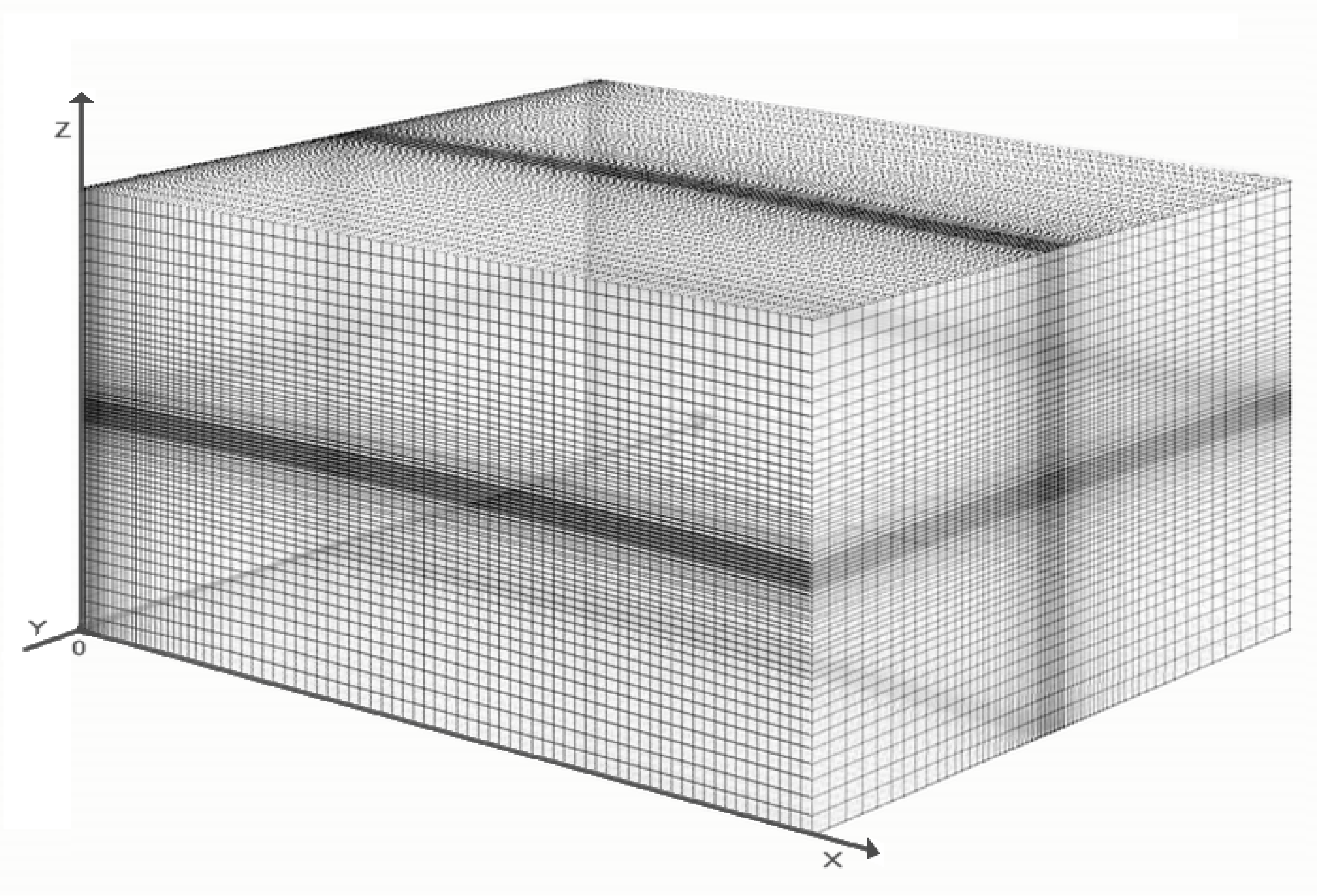
View of the mesh in the cross-sectional parallel to the OXY plane (OXZ – the same)



View of the mesh in the cross-sectional parallel to the OYZ plane



3D view of the mesh

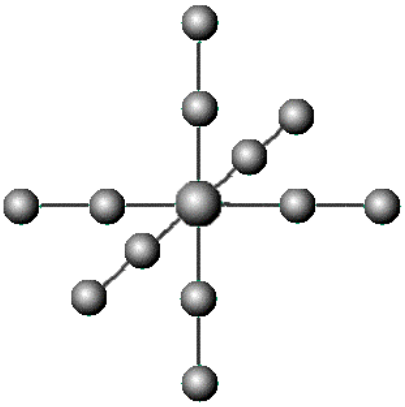


6 Area fragmentation

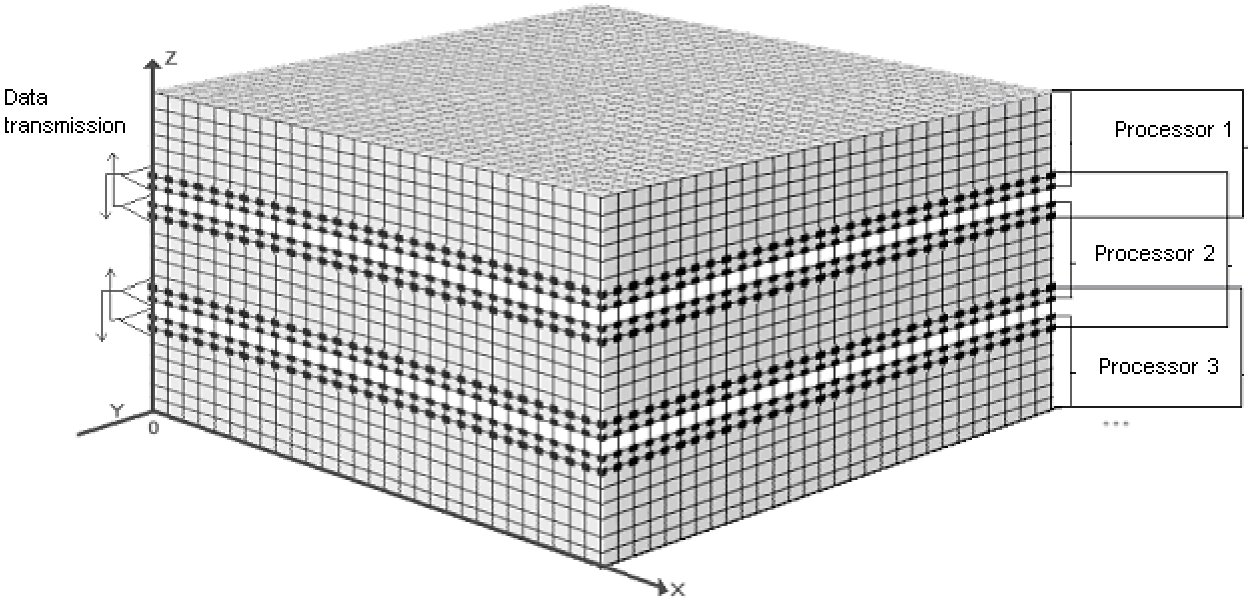
The whole mesh is fragmented into the beds of the equal length at right angle to Z axis.

(the bed contains few layers of the mesh) This fragmentation method is chosen according to allocation of data in computer memory. The number of the beds is equal to the number of processors used.

The difference scheme used in this program is based on 13-node template:



That is why each processor needs data from the neighbor beds to perform the computation in its own bed. Each processor sends the data from two bound layers of its bed on each direction on Z axis to it's neighbors. So the beds allocated in the memory is wider then initial beds.



The process #i is performing computations for the nodes of the bed #i. i = 1,..., N

Number of the first node on Z of the bed #i is specified as given below

$$Beg_i = i\frac{K}{Number \quad of \quad processors} + 1$$, where K –number of the nodes on Z axis

The last node's number is

$$End_i = Beg_i + \frac{K}{Number \quad of \quad processors} - 1$$, if the bed is not #N

and $End_i = K$, if the bed is #N.

The layers allocated in memory of the first processor have numbers: $1,2,..., End_1, End_1 + 1, End_1 + 2$,

where $End_1 + 1 = Beg_2, \quad End_1 + 2 = Beg_2 + 1$.

The layers allocated in memory of the last processor have numbers: $Beg_N - 2, Beg_N - 1, Beg_N, ..., End_N$,

where $Beg_N - 1 = End_{N-1}, \quad Beg_N - 2 = End_{N-1} - 1, \quad End_N = K$, N – the number of processors.

The layers allocated in memory of the processor # i (1 > i > N) have numbers:

$$Beg_i - 2, Beg_i - 1, Beg_i, ..., End_i, End_i + 1, End_i + 2$$, where

$$Beg_i - 1 = End_{i-1}, \quad Beg_i - 2 = End_{i-1} - 1,$$

$$End_i + 1 = Beg_{i+1}, \quad End_i + 2 = Beg_{i+1} + 1$$

7 Passing data

The passing of data is needed on every time step. All the data from two bound layers from both sides of each bed must be transmitted. (If bed is the bound bed then data transmission is needed from only one side)

There are 9 values of variables to send for each node - $u, v, w, \rho, p, t, E, p_k, \varepsilon$.

So the size of the data to transmit from only one side of the bed is 2*2*80*80*9*8 = 1834200 bytes if the mesh size is 80x80x80. There is a special buffer into which all the data is copied to perform transmission as one MPI message.

8 Computation on the cluster

The MPI parallel computation technology is used in this program.

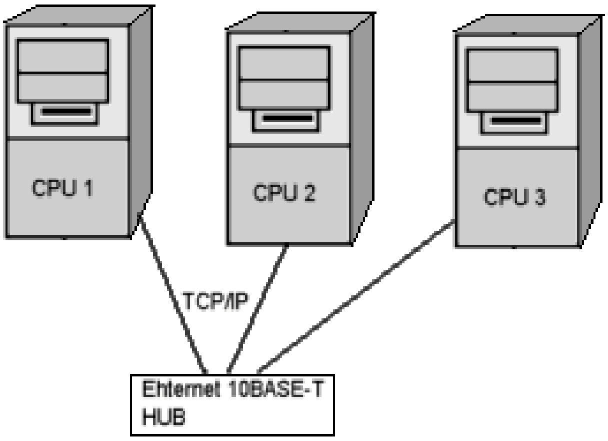
(*MPI - message passing interface – library of functions which is designed to support functioning of parallel processes in the terms of message transmission.)*

The functions MPI_Send and MPI_Recv are used for data communication.

There can be more than one data transmission during computations of one time step. It depends on the chosen scheme order. The function MPI_ALLGATHER is used for broadcast communications.

Testing and debugging of the program were performed at home on the self-made parallel computer with distributed memory. This computer was based on the LAN Ethernet 10BASE-T and consisted of three personal computers. (Pentium III 1100MHz, Pentium III 550MHz, Pentium II 330 MHz)

Structure of this computer



There were OS Windows 98 and MPI WMPI 1.3 library installed on each PC.

The TCP/IP network protocol was used for communications between the parallel processes.

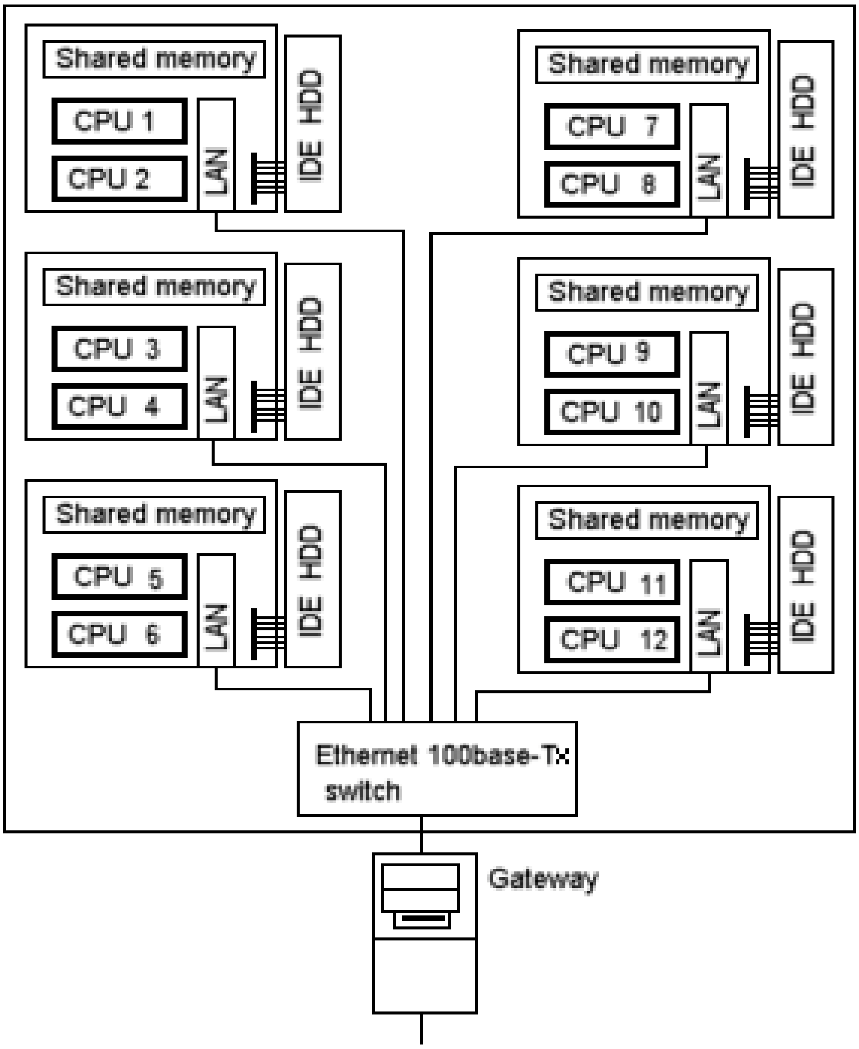
WMPI 1.3. also may be used to simulate multi-processor mode one the PC with single processor. So it was possible to test the program on more than 10 processors. The files of results of the parallel and original program were compared byte to byte to make sure that the results are correct.

Main computations were performed on the 12-processor computer at the IMM.

This computer has a cluster architecture. It consists of six computational nodes which are linked by the LAN Ethernet 100. Each node consists of two processors Pentium III 600 MHz installed on the dual-processor motherboard. Each node also has it's own IDE disk.

This computer is controlled by the Red Hat Linux 7.0 OS and the MPI MPICH library.

Structure of computer

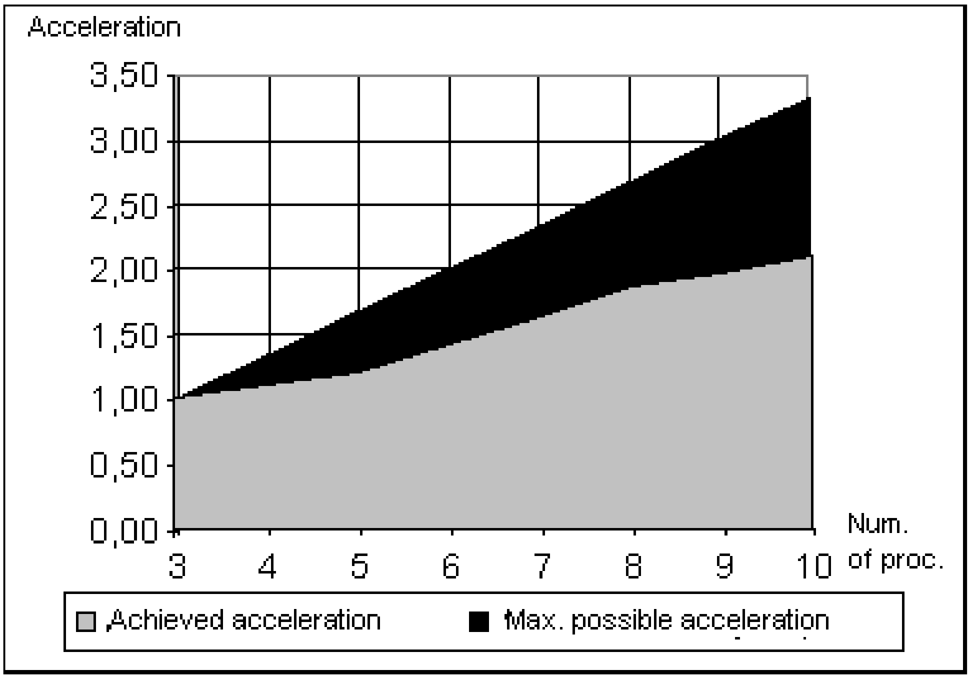


Efficiency of the parallelization

The duration of computation of ten time steps had been measured for different number of processors to find out the efficiency. Results are listed in the tables below:

Test with mesh 80x80x80

Number of Processors	Duration of computations, sec.	Acceleration, times	Efficiency
3	151	1	1
5	125	1.2	72%
8	81	1.86	69%
10	72	2.1	63%



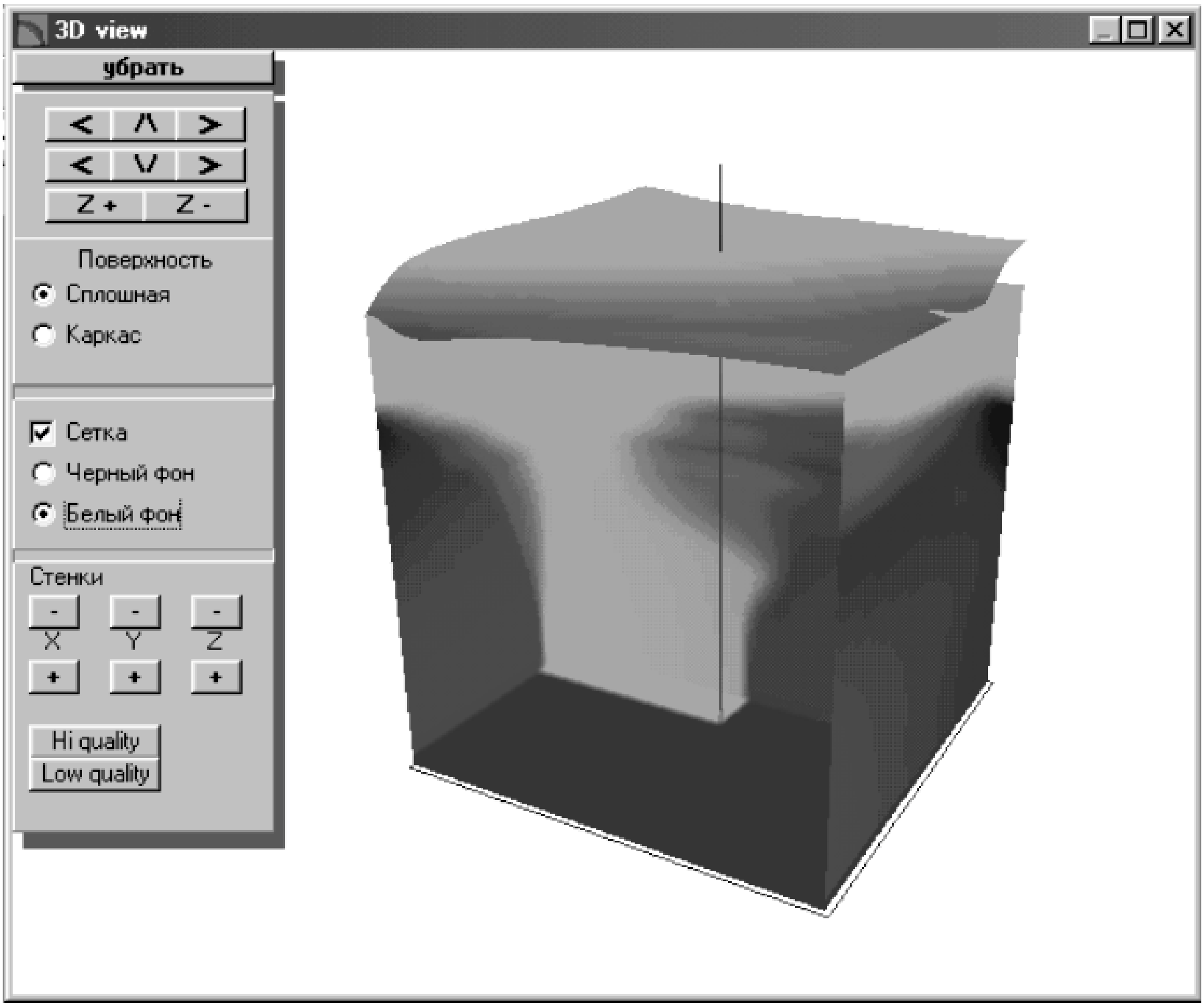
Efficiency is the ratio between the achieved acceleration and the maximum possible acceleration, multiplied on 100%. The duration of computation using three processors is the unit of acceleration. This is because the data cannot fit into memory when less then three processors are used and it causes huge reduction of performance.

Visualization of the results

8.1 Program description

The program that builds the 3d model using values in the nodes of the mesh was created. This program uses OpenGL graphic library.

Example of the program output

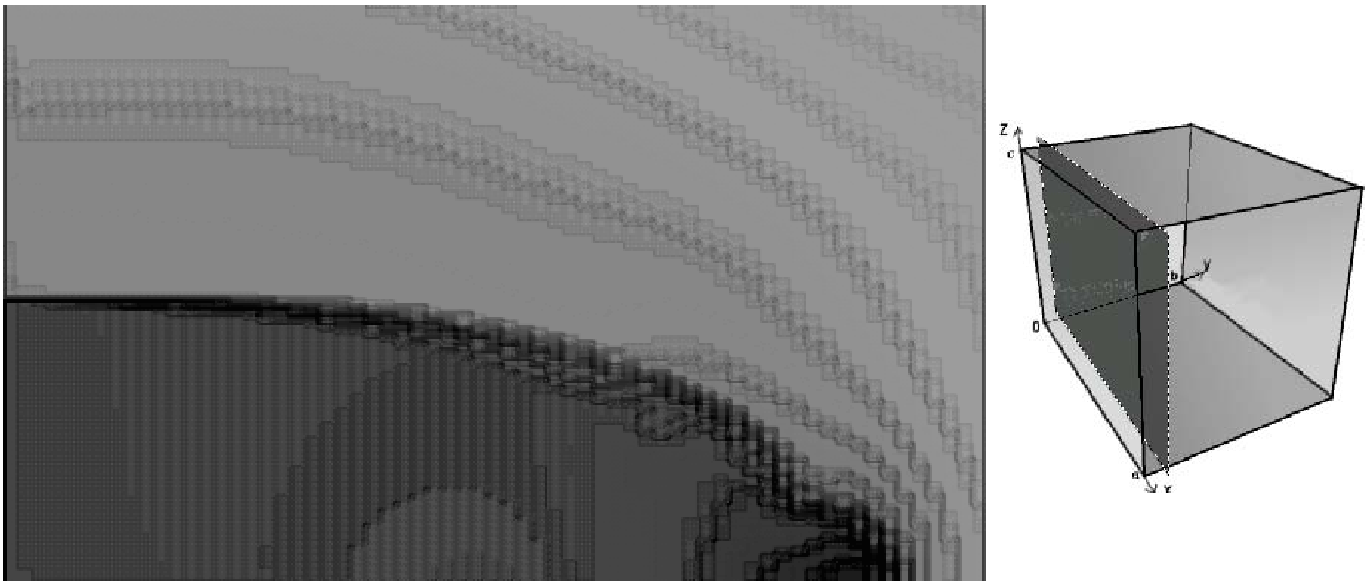


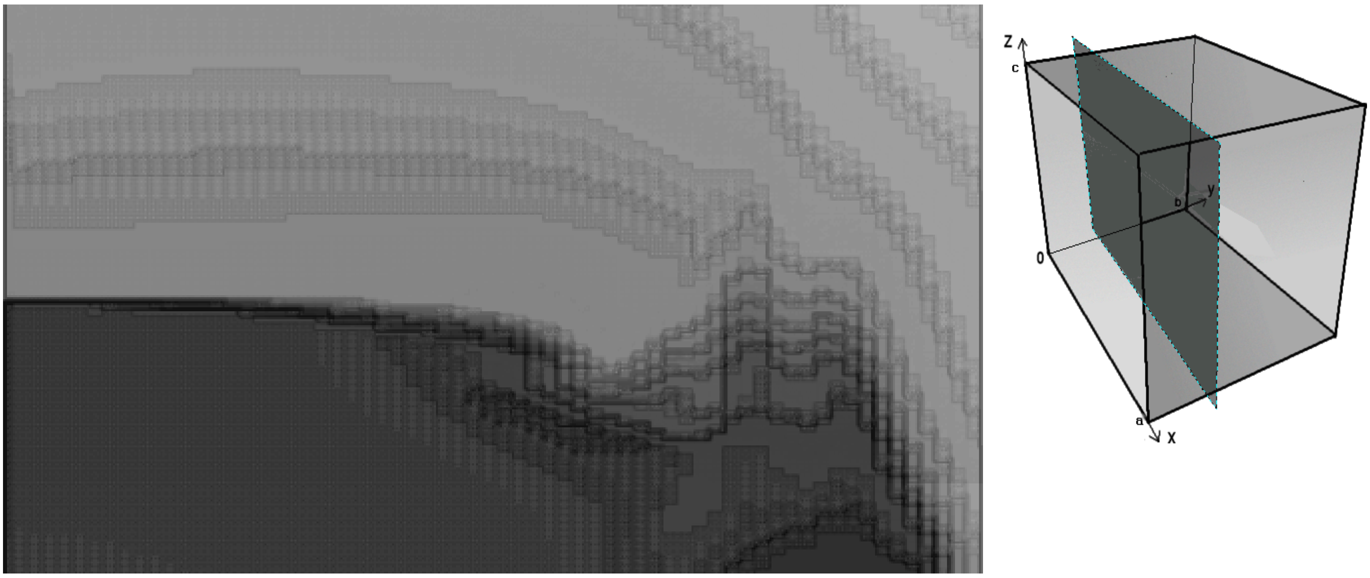
Поверхность
● Сплошная
○ Каркас

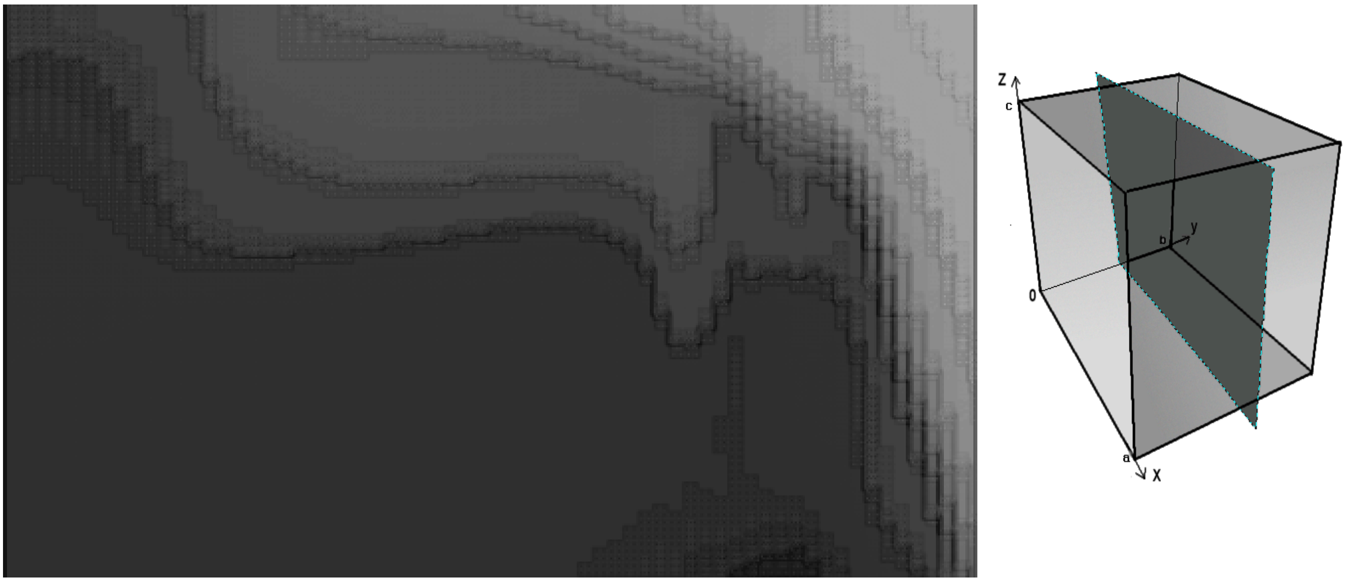
☑ Сетка
○ Черный фон
● Белый фон

Variable ρ

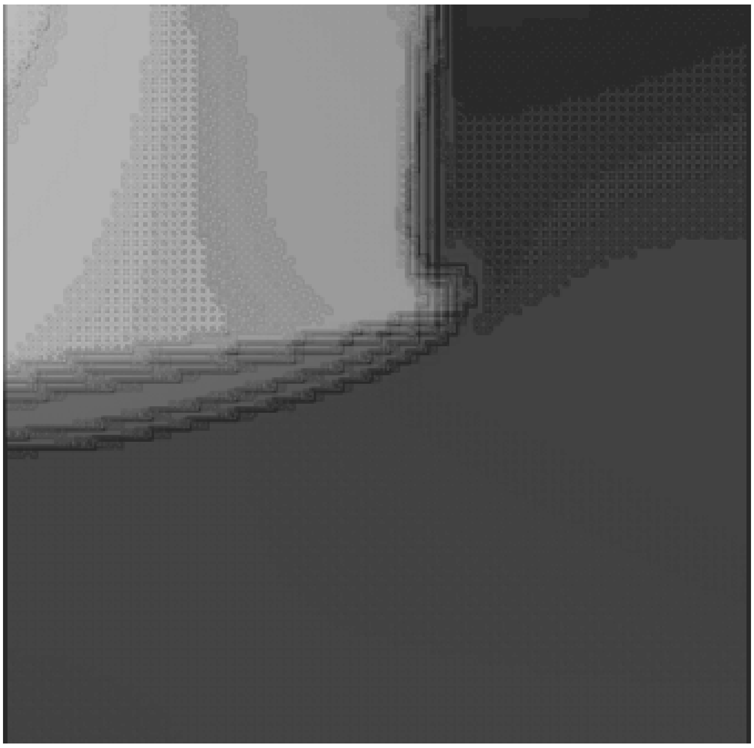
Plane OXY

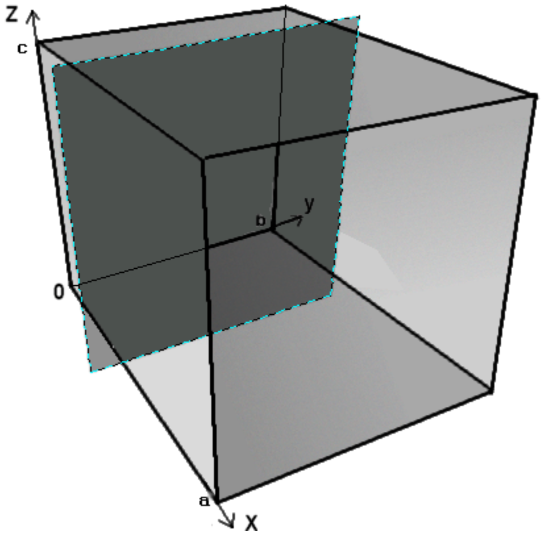


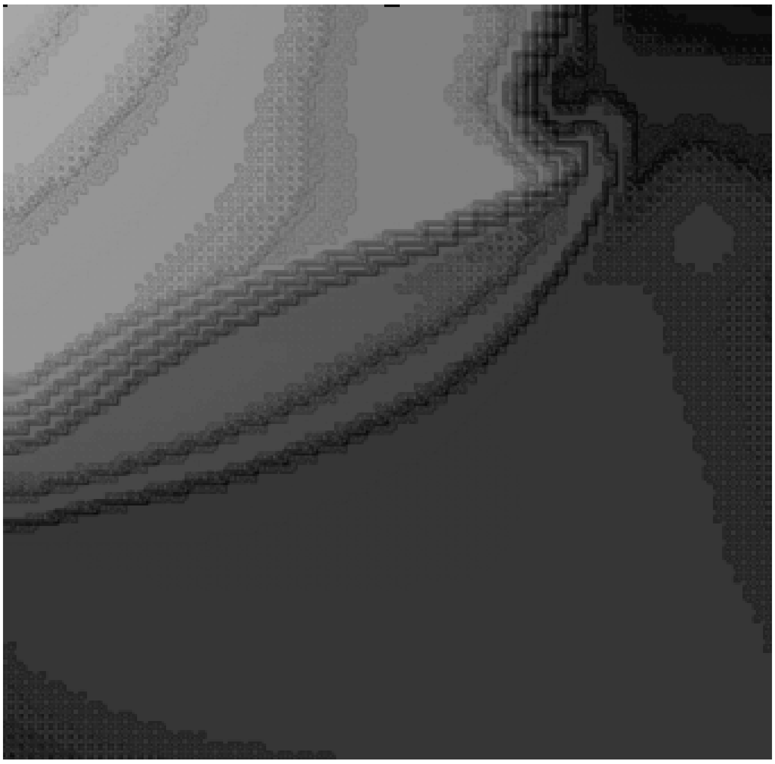


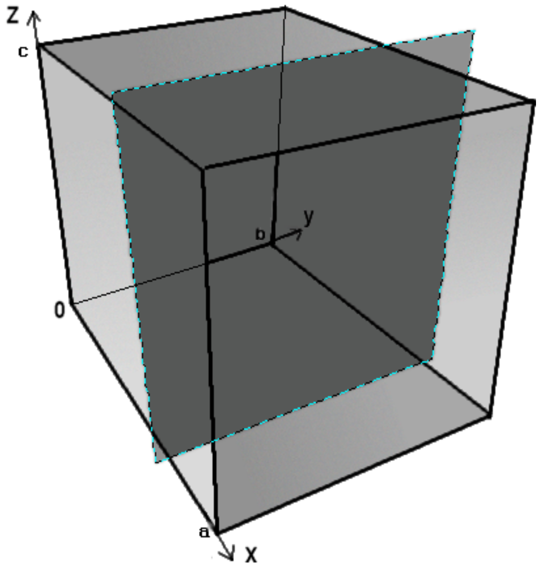


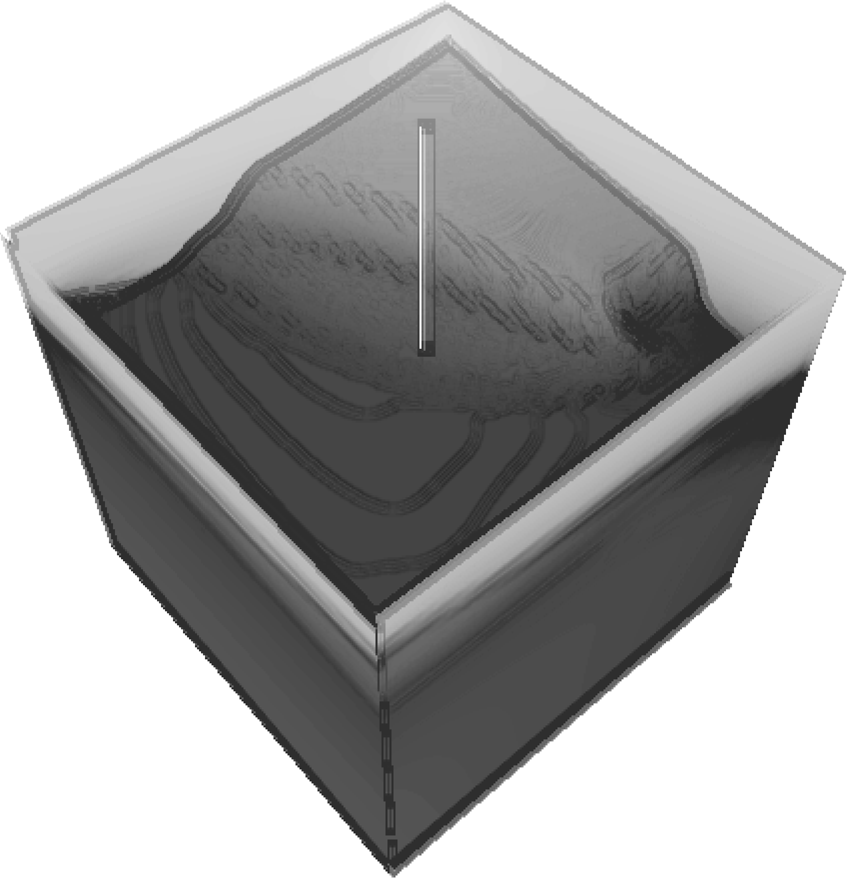
Plane OYZ

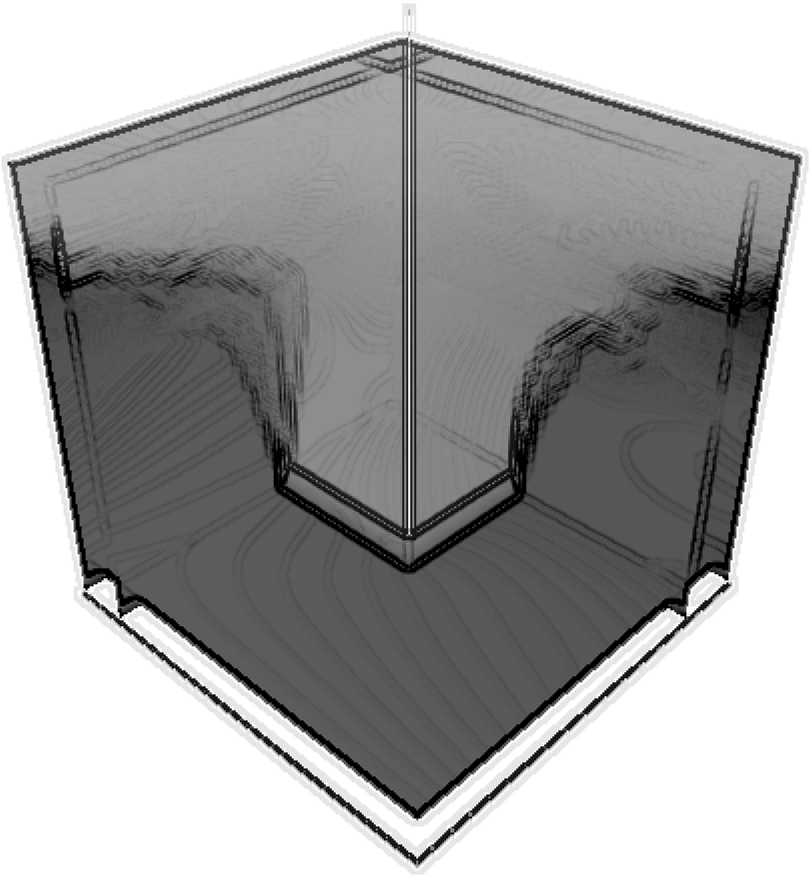












Results

- The parallel computational program was created
- Application of the parallel computations for the problems of this kind was tested
- The system of 3D visualization for 3D area was created.

This program may be used for testing of various implicit difference schemes, rectangular meshes and boundary conditions of inflow.

The system of visualization is applicable for representation of 2D and 3D functions. It loads data from file of the simplest format.