DYNAMIC VISUALISATION OF THE COMBUSTION PROCESSES IN BOILERS

Marek Gayer[1], František Hrdlička[2] and Pavel Slavík[3]

Department of Computer Science and Engineering
Czech Technical University, Karlovo náměstí 13
121 35 Prague 2
Czech Republic
xgayer@fel.cvut.cz, slavik@cslab.felk.cvut.cz http://www.cgg.cvut.cz/~xgayer/

ABSTRACT

This paper focuses on the simulation and visualisation of coal combustion in the pulverised coal boilers. It is important to find optimal boiler configurations (both for the ecological and economical reasons), determine appropriate combustibles, optimize process of combustion, etc. These tasks are typically solved using traditional Computational Fluid Dynamics (CFD) methods that are in general computationally very expensive. Our work is based on a different approach. We use simplified methods for determining direction and speed of air stream in particular places in the boiler. Further we use simplified methods for the simulation of combustion processes and heat transfer as well. A particle system is used to simulate and visualise the behaviour of the coal particles and air streams in voxelized boiler space. We developed concept of virtual particles – they represent certain amount of coal, air, ash and other materials in a voxel under investigation.

Keywords: FLUENT, visualisation, fluid, CFD, combustion, pulverized coal

1. INTRODUCTION

This paper describes our current work of visualisation of combustion processes in pulverized coal boilers [Hrdl96]. The goal is to improve design of the boilers - to reduce pollution, find ways of preparing fuel, determine particle sizes and quantity, speed etc. In engineering practice, it is very difficult to investigate the combustion processes of various kinds of combustibles directly in the boiler. Rather than constructing real boilers and trying to check and improve these characteristics „on the fly“, computers are used to experiment with models of the boiler.

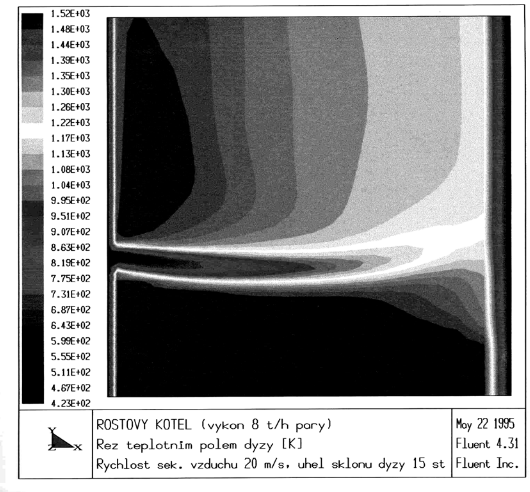
Therefore the efficient boiler design is based on simulation models. The models that simulate combustion processes are of various types. One type of models deals with simulation and visualisation of behaviour of flames. These models use several approaches like cellular automata [Takai95] or diffusion processes [Stam95]. Models of this type are more concentrated on the visualisation part of the combustion process. The resulting pictures can be used in applications where the quality of visual effect plays decisive role (e.g. movies etc.). In our case we investigated the approach that simulates and visualises the combustion process from the point of parameters of the combustion process. These parameters can be: temperature achieved in various parts of the boiler, speed of air and gases in the boiler during the combustion process and similar aspects that would help the boiler designer during the boiler design.

Some well-known algorithms and technologies for solving this problem based on CFD [Anderson95] have been already developed. CFD is a sophisticated analysis technique. It not only predicts fluid flow behaviour, but also heat transfers,

---

[1] Department of Computer Science and Engineering
[2] Faculty of Mechanical Engineering, Department of Thermal and Nuclear Power Plants
[3] Department of Computer Science and Engineering

mass, phase change (such as in freezing or boiling), chemical reaction (such as combustion), mechanical movement (such as an impeller turning), and stress or deformation of related solid structures (such as a mast bending in the wind). Using current CFD methods, we are able to solve only some specific cases with simplified boundary conditions. Nevertheless they sufficiently cover our needs.



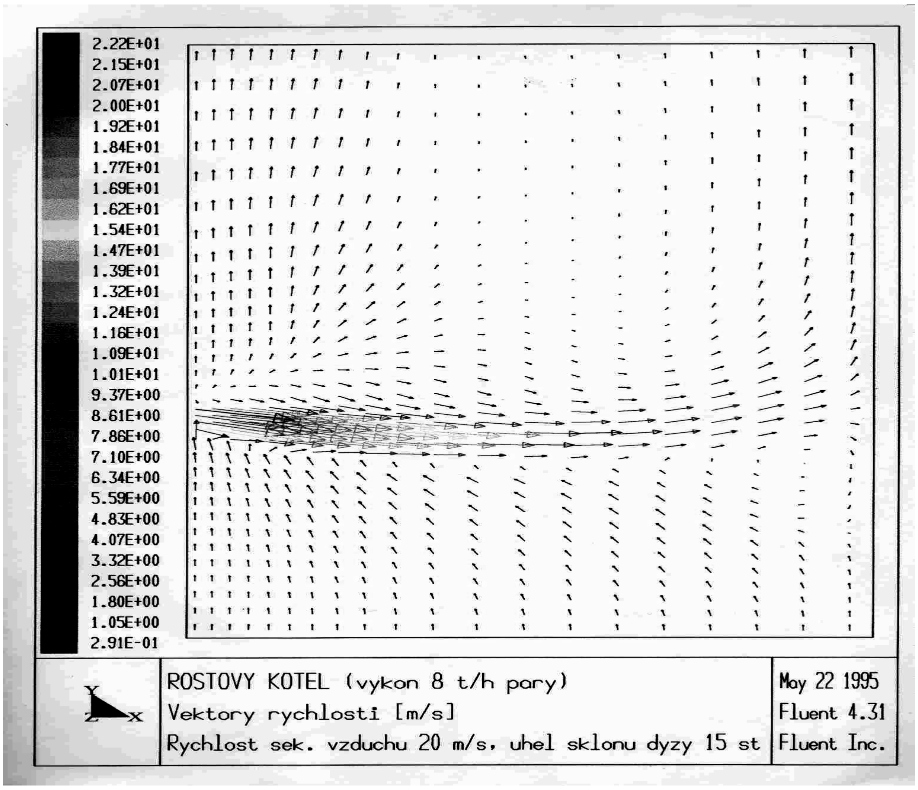
Slice of the temperature array in a boiler
Figure 1

2. FLUENT

FLUENT is the most known and respected universal CFD application for modelling fluid flow and heat transfer in complex geometries. Currently, FLUENT is one of the most used professional systems for CFD.

Geometry modelling in FLUENT is based on constructing a mesh for object (e.g. boiler). Supported are 2D and 3D meshes. The second step (after defining a correct mesh) is to define boundary conditions – walls, inlets, outlets, and physical properties and models of used materials and environments. The FLUENT offers excellent ways of visualisation of computed results. Various conditions such as temperature arrays, mass tracks and heat flux could be displayed (see Fig. 1. and Fig. 2.). Using a special pre-processor – PREPDF – the system can be applied for solving coal combustion computation tasks.

We use FLUENT results as reference ones to verify our results.



Sample visualisation of the vectors of the air stream speed in a boiler
Figure 2

3. OUR WORK

Current CFD effort is based on solving complex differential equations (such as the Navier-Stokes equations). Computation time needed for solving non-trivial tasks is counted in hours and days, even on a very powerful system.

In no way, current CFD methods can be used for dynamic real-time computations and visualisation with ability to change boundary conditions online. These real-time simulations are often needed to determine dynamic characteristics of the boiler for the transition to the stable state, synoptically display the flow of fuel and air etc.

The aim of our research is to develop a much faster system (though less accurate), based on a completely different approach, than the computation of the differential equations. It should even allow dynamic visualisation of the combustion process in the real-time.

4. PRINCIPLES

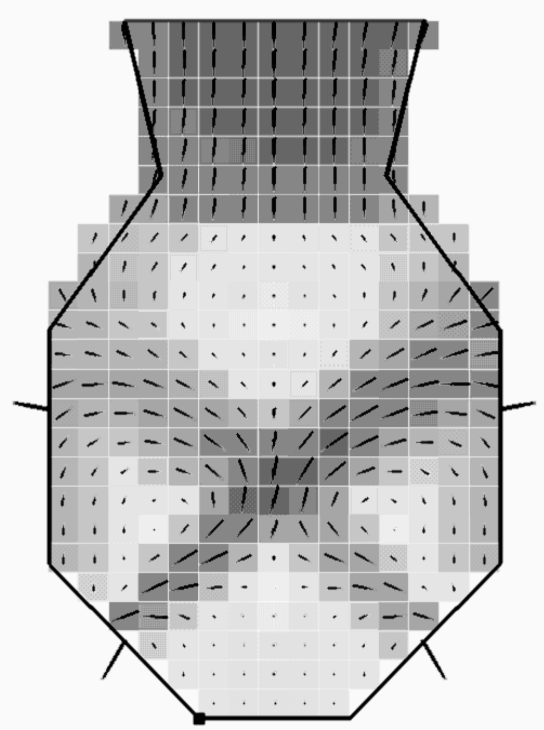
The main principles, by means of which our methodology enables dynamic visualisation, could be summarised as follows:

Particle system – is a common method for visualisation of fuzzy objects (e.g. clouds, water, and fire) in computer graphics. It is also used for industrial technology [Rhodes98]. In our case, the application of particle system represents a real technological problem. The simulation and visualisation using particle system is divided into separated steps. For simplicity and maximum computation speed, this part and all the other parts of our system are implemented only in the 2D space.

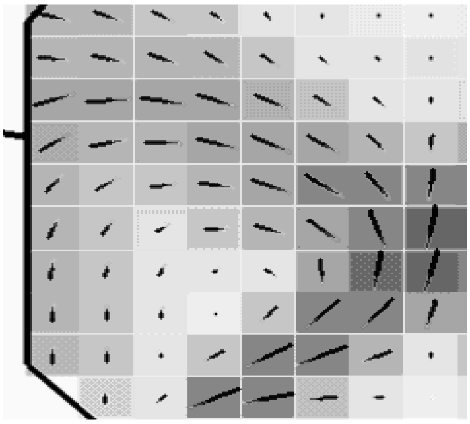
Pre-calculated vectors of motion - (flow array) – (See Fig. 3 and Fig. 4) dramatically increases visualisation and computation speed. The particles are moved only on the pre-calculated trajectories. These trajectories are computed only once at the beginning of simulation. They are represented as a floating-point data structure – called Flow array. The size of the flow array in our case is typically calculated and visualised 32 x 32 elements. This array divides the area of the boiler into mesh of squares (voxels in 3D interpretation see Fig. 4).

5. INPUT

Our system does not depend on some specific tasks and boiler configurations. We use small flat text files to describe geometry representation of the boiler and to configure air and coal jets. This allows us to solve boilers of different shape. It allows us to develop some visual editor. It could be useful for constructing and editing tasks without need of changing current source codes of our system later on.



Visualisation of Flow array in the test boiler
Figure 3



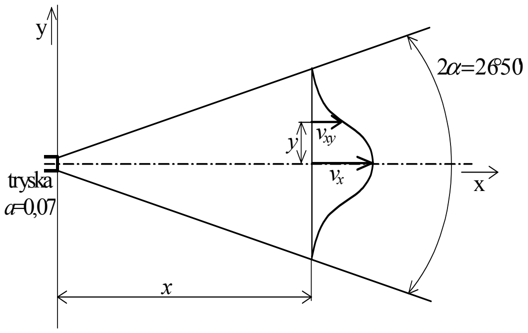
Velocity and direction of the vectors of velocity in the Flow array
Figure 4

6. FLOW ARRAY GENERATION

There are many ways to obtain flow array. The classical way is based on the differential equations. Since we try to reach the maximum simulation speed, we do not use this approach.

Instead of that, we use isotherm, loose flow that runs from a circle jet. The air stream flows through jets to the boiler. The solution of the streams in a limited area in the boiler is quite complex, especially for the non-isothermal flow. That is why we calculate it as an isothermal free stream that flows from the circle jet. Our solution is based on the G.N.Abramovič's idea that can be found in the [Cihe69].

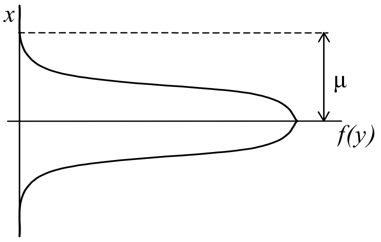
The air stream forces to move surrounding air under the influence of the turbulence. This approach allows us to speed up considerably the calculation of flow array. The stream can be considered as a cone. The top angle 2α depends on the level of the turbulence of the stream in the jet, see Fig 5.



Isotherm free stream flowing from the jet
Figure 5

For any distance x from the input of the stream, the maximal speed in the x-axis and y-axis is decreasing to zero with the Gauss distribution (see Fig. 6) described as (Eq. 1):

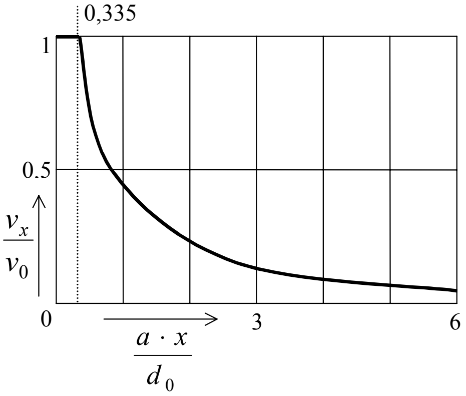
$$f(y)=\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{\frac{-(y-\mu)^2}{2\sigma^2}} \qquad (1)$$



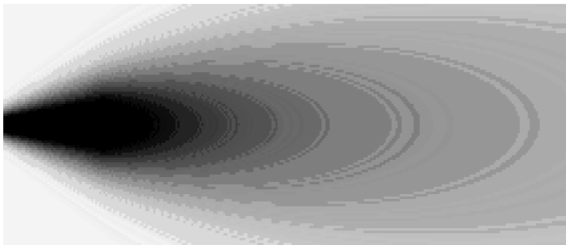
Graph of the Gauss distribution
Figure 6

The x-axis speed gradually decreases (see Fig. 7) and is determinate by the Eq. 2. See Fig. 8 for an illustration of all the above given approach.

$$v_x = v_0 \frac{0{,}48}{\frac{a\cdot x}{d_0}+0{,}145} \qquad (2)$$



Curve of the axial speed v_x
Figure 7



Distribution of the speed of the air stream in the space, see [Faltyn99]
Figure 8

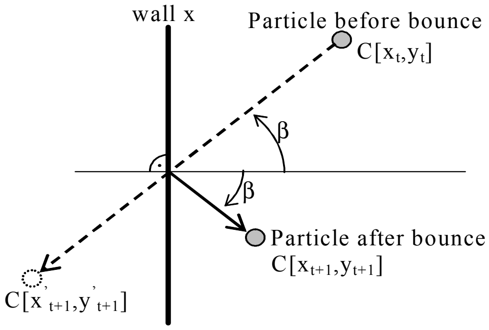
However, there may appear a situation, when a stream collides with the wall and/or there may occur a collision with another air stream. In such a case, other virtual jets are added to handle this situation and to match the real situation. Detail description of this situation exceeds the scope of this paper.

7. THE PARTICLE SYSTEM AND VIRTUAL PARTICLES

For our work, the particle system allows us computation and visualisation of mass elements in the boiler. The particles displayed and calculated do not correspond to the real coal particles in the boiler. Instead of that, they represent some corresponding mass of coal in the voxel under investigation. Various particle types represent proper amounts of air, gas, ash and other materials in the boiler space. Therefore, we call them virtual particles. Thus, one virtual coal particle carries many real coal particles. The quality and speed of simulation and visualisation could be altered by increasing or decreasing the amount of these virtual particles.

The movement of the virtual particles is strongly determined by the flow array. For each particle, the new x and y position, according to air speed current in the current voxel is computed. The magnitude of the speed is time dependent.

There are some factors that could change their motion. For coal particles, we cannot omit the force of gravity. The coal particles are attracted to the bottom of the boiler. The acceleration is determined by the weight of the particle and by the surrounding environment. Before moving a particle to the predicted destination, we must check for possible collisions with the wall. For each voxel, we have list of walls the voxel interferes with it. We first determine, in which voxel the particle is located, and according to that we check for possible collisions. If this is the case, the particle track is mirrored and bounced from the wall (See Fig. 9). Particles are generated from the jets (usually installed in the walls of the boiler).



A particle bouncing from the wall
Figure 9

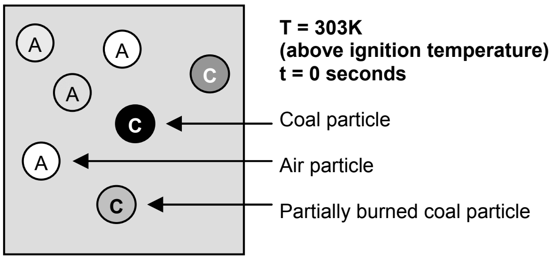
Currently, we are using a simplified model of particle system, because we ignore collisions between each single particle, from which the final motion is calculated. Sample visualisation of our particle system can be found in Fig. 10

8. COMBUSTION AND HEAT TRANSFER

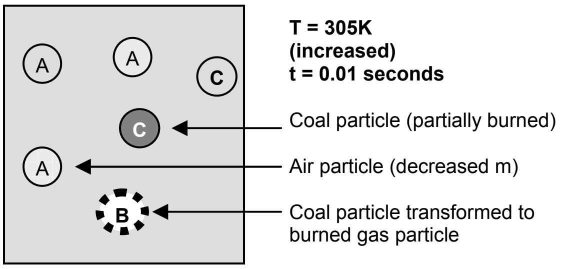
Combustion process of the coal particles is in fact a quite complex problem [Dibble96]. Again we use some simplifications due to the need for the fast computation. In each step we compute a temperature array. It contains weighted average of the particle temperatures for all the voxels. To start combusting coal, two conditions must be satisfied: in the voxel there must be at least some minimal combustion temperature (which is defined - in our example we use 300 K), and a proper mass of coal and air (represented by virtual particles) that is to be burned.

Particles flowing from a jet
Figure 10

Depending on the current temperature, weight and proportion of the coal, the coal particles are being burned. For air particles, we just decrease their appropriate mass. If the mass of the air particle reaches some minimal value, we remove this particle from the system. For coal particles, we decrease the amount of the combustible part of the particle, and increase the amount of the gas burnt. If the mass of the combustible part reaches some minimal value, we assume that the coal particle is burned out and we change it to the burnt gas particle. This process is shown in the Fig. 11 and Fig. 12.



The situation in a voxel before the combustion process start
Figure 11



The situation in a voxel after the end of the combustion process
Figure 12

Between these processes, depending on reaction heat transfer, the released heat is transferred to all the particles, which are present in the current voxel. Therefore, the temperature of the voxel increases.

Because of the dynamic processes in the boiler, the heat is distributed by the moving particles to the other voxels, thus increasing the temperature and making possible to start another combustion reactions. We also count with the heat radiation between the walls and the mass in the voxels. The heat transferred from the given surface *F* during the time *dt* is comparable to differences of the temperatures of the wall and the voxel (power of four) [Dibble96]. We also need to determine the coefficient of the radiation C_{12}. These ideas are summarized in Eq. 3.

$$Q = F \cdot C_{1.2} \cdot \left\{ \left(\frac{T_1}{100} \right)^4 - \left(\frac{T_2}{100} \right)^4 \right\} \cdot dt \qquad (3)$$

We assume, for the sake of the simplicity, that the temperature of the walls is constant (typically bellow the minimal combustion temperature). In general, it is possible to say that our model is based on mutual reactions of virtual particles of various types in each voxel in the boiler space.

9. INITIAL CONDITIONS AND INITIALIZATION

To start combustion again, there must be already some appropriate condition in the boiler. It is necessary, because air and coal, which is coming through the jets, are not usually heated enough. Their temperature is under the ignition point. Thus, there must be a way to start combustion. We assume in the current implementation, that there are already some air particles warmed up above the ignition point, which allows the ignition.

10. IMPLEMENTATION

All the parts of our system have been implemented in the standard, ANSI C language. Visualisation is based on the OpenGL graphics interface. Windowing interface is maintained by the GLUT library [Mason 98]. Thanks to this, our system is easily and fully portable to other systems. No problems should occur porting to Linux systems or even SGI workstations, although it was originally developed on the Windows NT/2000 platform. We have tested our system modelling a boiler with real dimensions, characteristics and parameters. The behaviour and results gained from our system were well comparable with a situation in a real boiler. Thus, the current implementation is correct, although, there are still many things to improve.

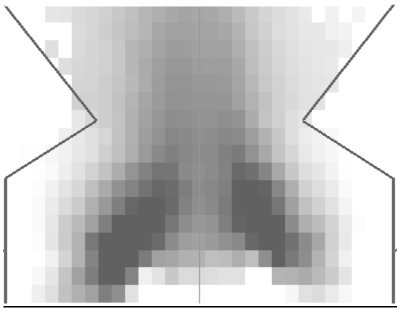
11. VISUALISATION

To maintain the reliable and fast visualisation, our system uses industry standard OpenGL platform. Thanks to this, nowadays, our system could be used on a standard, even a cheap graphics accelerator. There is no big lack of speed in particle visualisation even when using 10 000 of particles. Furthermore, our system uses MGL graphics library on the backend to maintain easier visualisations of common OpenGL primitives [Gayer00]. That it is an OpenGL based library optimised for visualisation of common 2D graphics and graphical user interface (supports images and fonts directly). We use it to easy implement an easy user interface, in common 2D coordinates. Therefore, such tasks are much easier to program than in a native OpenGL.

The boiler walls and outlets are approximated by the straight lines. The particles are displayed using standard OpenGL pixels. For the examples of our current graphics output, see Fig. 3, 4 and 10. The selected local characteristics in the voxel, such as the total temperature, mass storage, the wattage, and heat flux state and/or changes can be in the real-time visualised (see Fig. 13). The particle tracks can be easily determined by the fast particle system animation. Currently, the characteristics in a voxel are simply visualised by the quads. Although the quality of the visualisation could be improved by choosing smaller voxel sizes, we plan to implement contours to bring smoother graphics output.

12. RESULTS

The current research brings promising results. On a test boiler (dimensions 6m x 13.7m) we have simulated and visualised combustion processes. We have discussed the obtained results with the experts from the Faculty of Mechanical Engineering of CTU with positive response. To compare our results with current CFD methodology, we used the FLUENT solver.



Sample visualisation of the total voxel temperature in the test boiler
Figure 13

The global parameters, which could be easily compared, match well overall design and implementation of our ideas, see Table 1.

Parameter	Our system	FLUENT
Average temperature	1029 °C	1158 °C
Outlet temperature	1151 °C	1384 °C
Maximum temperature	2360 °C	2753 °C
Average stream velocity	23 m/s	17 m/s
Average outlet velocity	28 m/s	21 m/s
Wattage	192 w/m^3	232 w/m^3
Mass total	21.1 kg	21.3 kg
Time needed to converge solution	**12 seconds**	**7 hours**

Global parameters results in the test boiler
Table 1

Next we compared the images of the temperature and velocity maps, which summarize local characteristics. We found they are visually similar.

13. CURRENT IMPLEMENTATION SPEED

The current implementation is very fast. We tested it on different systems. We measured the number of the frames (images) which our system was able to compute and display per second (FPS). On each system we generated 10.000 particles, and we measured the FPS value, see table 2.

System	Frames per sec.
Celeron 300, no hardware OpenGL accelerator, 64 MB RAM	2
Celeron 400, S3 Savage 4, 128 MB RAM	26
AMD Athlon 1333, nVidia Geforce 2 MX 400, 256 MB RAM	64

Simulation speed on different systems
Table 2

On an average system (Intel Celeron 400, 128 MB RAM we can reach 26 frames per second with 5000 particles computed and visualised. Note that on the Celeron 300 system the speed of the simulation rapidly decreased due to lack of the OpenGL accelerator. However, it should not be a problem, because almost every new computer system is equipped at least with a cheap graphics accelerator (such as Savage 4), which is sufficient for our system.

14. ADVANTAGES AND DISADVANTAGES

As it is obvious from the previous text, the main advantage is the speed of computation. The speed is far beyond of reach offered by the traditional methods. Thus, the developers of the boilers could test many configurations and modifications of the boiler with the immediate response. This results in the possibility to get a very good preview of the dynamics of combustion processes in a boiler. This is not available in the traditional approaches. Thus, our system could also be used for experimentations and educational purposes in the field of the combustion processes.

Although it gives fast and reliable results, there still would be necessary to test and compare more deeply the gained results with results gained from CFD systems. So again, the main advantage against less accuracy and features offered by the CFD systems is the speed of computation. The next advantage is the possibility to visualise the state of the particles during the combustion process.

15. FUTURE WORK

The future work will be concentrated on:

- We plan to implement more accurate heat distribution.
- We will develop a methodology for verification of our results with the CFD computations.
- We plan to simulate and monitor additional characteristics (pressure, turbulence, etc.)
- Much interest we will probably give to the computation of the Flow array. While this

is computed only once, with no influence to the speed of the other computation, there could be used some more complex algorithms to compute. We plan to include influence of the mode of combustion also.

- Some advantages could be gained through the visualisation itself. We plan to use linear approximations to convert attributes from the singular points to the continual array. It would be useful for visualisation of the temperature map.

16. CONCLUSION

For combustion system visualisation, current investigation brings an interesting alternative to the classical CFD applications. We implement a brand new way based on the fast computation of the flow arrays. We use simplified model of combustion process and light-speed visualisation using OpenGL graphics interface. Current implementation is very fast even on average systems. The behaviour and results gained from our system were comparable with a situation in the real boiler and FLUENT CFD system. Therefore, our system can be used for dynamic simulation or preview of dynamic processes of coal combustion in boilers. This may be used in the CFD process for the fast and efficient design for the boilers. The system could also be used for education purposes in order to give students idea about the behaviour of boilers under various conditions.

REFERENCES

[Anderson95] Anderson, J.: *Computational Fluid Dynamics – The basics with the applications, McGraw Hill*, 1995.

[Dibble96] Dibble, W., Maas U.: *Combustion*, *Springer*, 1996.

[Cihe69] Cihelka, J.*: Vytápění a větrání. SNTL, Praha*, 1969.

[Faltyn99] Faltýn, R.: *Diploma work / Diplomová práce, ČVUT FEL Praha*, 1999.

[Gayer00] Gayer, M.: *Graphical library MGL, part of the diploma thesis Programové knihovny pro grafické akcelerátory (in Czech), ČVUT FEL Praha*, 2000. *http://sgi.felk.cvut.cz/~xgayer/mgl/*

[Hrdl96] Hrdlička,F., Janeba,B.: Fluid combustion, *Energie* (Vol 1, pp. 50), 1996.

[Mason98] Mason, W.: OpenGL Architectural Review Board, Jackie, N., Davis, T., Shreiner, D.: *The OpenGL Programming Guide. Addison-Wesley*, 1998.

[Rhodes98] Rhodes, M.: *Introduction to particle technology, John Wiley & Sons Ltd.*, 1998.

[Stam95] Stam J., Fiume, L.: Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes, *Proceedings of SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series, pp. 129-136*, 1995.

[Takai95] Takai, Y., Ecchu, K. Takai, N.: A cellular automaton model of particle motions and its applications, *The Visual Computer, 11(5), pp. 240-252,* 1995.