FLUIDS6 : CFD MULTIPROCESSING, NUMERICAL PROBLEMS, NEW FEATURES

Stanislaw Raczynski
Universidad Panamerciana
Augusto Rodin 498, Mixcoac, 03920 Mexico City
E-mail: stanracz@stanr.com

KEYWORDS:
fluid dynamics, fluid flow simulation, Navier Stokes

ABSTRACT

A new simulation tool for 3D simulation of fluid dynamics for liquids and gases is presented. The new version of Fluids6 supports multiprocessing, which considerably accelerates the simulation. The program includes en editor for 3D ducts with obstacles, holes, moving walls and internal temperature sources (in gas case). The external pressure pulse can be applied. Fluids6 provides various forms of result representation, mainly the images of the velocity, temperature and pressure distributions in user-defined sections of the duct, and 3D images of flow lines. One of the aims of such simulation is to visualize shock waves in liquids and gases. The presented simulations are dynamic. We are looking for the fluid movement and not for a steady solution.

INTRODUCTION

Computational Fluid Dynamics (CDF) is perhaps one of the most difficult fields of research, involving numerical methods for partial differential equations and related topics. As the name (CFD) suggests, the fluid dynamics must be simulated as it is, namely in the dynamic way. Consequently, the steady-state solution is not the target result of Fluids6. Moreover, our point is that in a real physical fluid flow the steady state is hardly reached, except simple academic examples.

The main mathematical model is the Navier-Stokes (N-S) equation. It is a nonlinear partial differential equation that describes fluid dynamics. The N-S equation is complemented by the continuity equation. The model has four unknown variables assigned to each point of the 3D region: the pressure and the three components of the particle velocity. In the case of gas both temperature and density are also unknown variables. Recall that the governing equations of flow in the liquid case are as follows.

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{1}$$

$$\begin{cases} \rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=-\frac{\partial p}{\partial x}+\mu\nabla^2 u+\rho f_x \\ \rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=-\frac{\partial p}{\partial y}+\mu\nabla^2 v+\rho f_y \\ \rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial p}{\partial z}+\mu\nabla^2 w+\rho f \end{cases} \tag{2}$$

$$\nabla^2=\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}$$

x, y, z – *space coordinates*

(u, v, v) – *the velocity vector*

ρ – *density* μ – *viscosity*

p – *pressure* f – *external force*

The gas flow the equations are similar, except for the continuity equation that is no longer valid for gases and the viscosity term, which normally is ignored. The density for gas is not constant and becomes a dependent variable, so an additional equation is required. The system equations are closed by adding the equation for the conservation of thermal energy. The governing equations for gas (in vector notation) are as follows.

$$\begin{cases} \frac{\partial p}{\partial t}+\nabla\cdot(\rho\,\mathbf{u})=0 \\ \rho\frac{\partial \mathbf{u}}{\partial t}+\rho\,(\mathbf{u}\cdot\nabla)\mathbf{u}=-\nabla p \\ \rho\frac{\partial e}{\partial t}+\rho\,(\mathbf{u}\cdot\nabla)e=-p\nabla\cdot\mathbf{u}+\nabla\cdot(k\,\nabla T) \\ \mathbf{u}=(u,v,w) \quad \text{gas velocity} \\ \nabla=\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right) \\ e-\text{gas internal energy} \quad \rho-\text{density} \\ \text{gas temperature } dT=de/C_V \end{cases} \tag{3}$$

Proceedings 25th European Conference on Modelling and Simulation ©ECMS Tadeusz Burczynski, Joanna Kolodziej Aleksander Byrski, Marco Carvalho (Editors)
ISBN: 978-0-9564944-2-9 / ISBN: 978-0-9564944-3-6 (CD)

NUMERICAL PROBLEMS AND METHODS

There are two main methods that are commonly used: The finite element method and the methods based on the uniform discretization in space and time. The original N-S equation has been derived considering a small volume element and then passing with the element size to zero. In a numerical method with space and time discretization we assume a finite size of the element. Then, we assume that the solution we obtain is similar to the real fluid flow. This may be valid for highly dissipative processes like the heat or diffusion equations. But, in the case of liquid or gas movement this assumption, commonly taken as an obvious fact, is at least doubtful. What makes us to be sure that if we fix, even small, volume element, the flow inside the element is nearly uniform? The flow may change within any small volume, and there is no reason to ignore turbulence inside the element that can make the discrete version of the N-S equation completely false. In other words, if we consider a sequence of solutions with the element volume approaching zero, we can hardly be sure that this sequence converges to the real solution. Despite this pessimistic comment, the reality is that many numerical algorithms and software tools for the N-S equations have been developed and seem to work correctly, providing a quite good illusion of the real flow, compared to the physical experiments.

The turbulence in a fluid flow is difficult to simulate using a finite grid of points. The artificial viscosity or dissipation introduced by many numerical algorithms makes it difficult to obtain turbulence. The common approach is based of the vorticity equations that are simulated in parallel with the original movement equations. Many examples can be found in the literature and on the Internet, but most of them only show 2D simulations. The general issues and theoretical background can be found, for example, in Currie (1974), Landau and Lifshitz (1988) or Serrin (1959). There are many different ways to simulate fluid flow and turbulence. Stam and Fiume (1993) use spectral analysis to simulate small turbulences. Other way is to use particle-based simulations. As we recently can simulate N-body systems of hundreds of millions of particles, this may be a feasible alternative in fluid dynamics (see Reeves, 1983, Hockney and Eastwood, 1988). A frequent approach to turbulence simulation is the use of *vortex particles* and the vortex equation (see Christiansen, 1973, Noronha 2003). In the present work, we do not use any vortex equations. The rotation of the fluid and turbulence results in a "natural way" from the original equations, solved in the 3D space. From the experiments it can be seen that turbulence form complicated 3D structures, so any attempt to simulate them in 2D is rather academic and unrealistic.

Replacing the derivatives in the original equation by their finite difference approximations, we obtain a set of difference equations that can be solved by a computer. In the case of the N-S equation, this results in the differencing scheme that can be integrated forward in time, taking into account the imposed initial and boundary conditions. The simplest algorithm, which seems to be the most exact and logically correct, is the Forward Time Centered Space (FCTS) scheme. In this scheme, the solution for the next time instant is calculated using the approximation of the partial derivatives based on the difference between the neighbor grid points. Unfortunately, the FCTS scheme is always unstable and, consequently, of very limited usefulness. There are some ways to cure the FCTS instability. One of them is a simple change in the differentiation scheme, due to Lax (Harten, Lax, Van Leer, 1983). Namely, we replace the values of the variable in the numerical time-derivative term by their averages taken from the neighbor points. This makes the algorithm stable within certain range of the time step. However, this is achieved sacrificing the validity of the model. It can be shown that using the Lax method we add an artificial viscosity to the original model. This is called numerical dissipation or numerical viscosity. In other words, the numerical algorithm based on the Lax method results in an invalid model. The Lax method consists in adding the increment of the dependent variable not to the original previous value, but to its spatial average.

We will not discuss here the details of numerical methods for the N-S equation. There is a huge research being done in this field. What we are discussing here is the solution to the dynamic problem and the transient process simulation. There are other algorithms used to obtain steady solutions. Some CFD programs first linearize the model, and then solve it with certain algebraic methods. Repeating this procedure we can obtain final steady solution, provided the whole algorithm converges. The algorithm applied in the Fluids6 program is similar to the Lax approach. However, it is not exactly the Lax algorithm. The averaging operation (numerical dissipation) is reduced to a necessary level, to maintain the algorithm stable. This makes the influence of numerical dissipation as small as possible, preserving the stability of the algorithm. It should be noted that if we allow greater dissipation, then the solutions seem to be very nice and “regular”. However, such models are false, and hardly show the turbulence. Moreover, by averaging or smoothing the velocity we change the total kinetic energy of the system (dissipation).

THE SIMULATION TOOL - FLUIDS6 PROGRAM

Fluids6 solves the Navier-Stokes (N-S) equation for a gas or an incompressible liquid that flows through 3D channels with obstacles. The channel and the obstacles can be arbitrary 3D shapes. Optionally, the channel may be ax-symmetric. Fluids6 handles non-uniform grid of points in the space-discretization. The grid is uniform in the interior of the simulated volume, while its resolution is two times greater (smaller point-to-point spacing) for

the points near to the duct walls. For a summary of Fluids6 the reader may consult
http://www.raczynski.com/pn/fluids.htm

The flow is considered to occur in an ax-symmetrical or irregular channel that has an inlet and outlet. However, the user can define any other configuration, like, for example, an open region where some internal points have fixed pressure, being sources of the flow (holes in the duct). The boundary conditions are defined as sets of points with fixed pressure, temperature or velocity. The pressure at the inlet/outlet points can be defined by the user, as an external excitation.

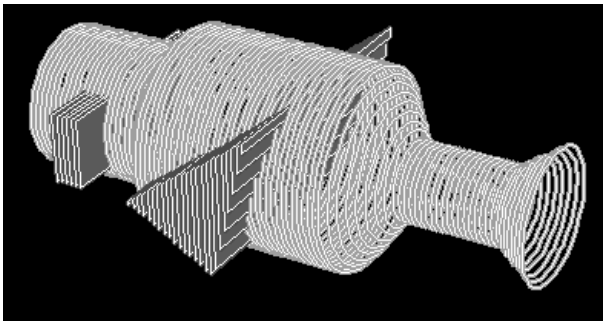
Fluids6 program includes a 3D duct editor. The duct shape is defined graphically. For the ax-symmetric duct the user draws the duct projection (side view). For the arbitrary duct the shape is defined by sections (layers) on the X-Y plane, which are given consecutive Z coordinate values. Then, the program creates the set of grid points to discretize the problem in space. A normal channel needs about 100,000 points, while the reasonable number should be no more than 1,500,000. For each point the pressure, the temperature (gas case) and the three velocity components are calculated.

The following result images are provided.
Velocity in a vertical (Y-Z), horizontal (X-Z) or X-Y projections. The velocity is marked at each grid point as a section which direction shows the particle velocity direction. The length of the velocity indicator may be constant or proportional to the absolute velocity, and the color changes with the velocity.
Pressure marked at each point with a corresponding color.
Rotation marked with a corresponding color. The rotation is the sum of the absolute values of the components of the rotation tensor.
Flow lines in X-Z, Y-Z and X-Y projections, and the 3D image of the flow lines, starting from random points inside the duct. This image can be seen from different angles, providing an illustrative image of the flow behavior.

The time-plots for pressure and/or velocity in selected points are also provided. The frequency analysis can be performed. Finally, the animation of the velocity changes can be generated in order to quickly see the evolution of the velocity field.

Multiprocessing. New PCs are frequently equipped with two, four or eight or more processors, so it is important that the new software can use this advantage. Fluids6 detects automatically the number of available processors, and performs the flow calculations on the maximal number of processors. As the result, you get a considerable speedup, approximately N times, with N processors. On the other hand, if you define a higher resolution model with thick duct, then the number of grid point in 3D space may be rather big, which slows down the simulation. The reasonable number of (fine) grid points is up to 1,500,000. Note that to simulate big models with Fluids6 **you need a fast machine with multiple processors**. The minimal recommended platform is double or quad processor, 2.4 GHz, 2GB of memory, at least 10 GB of hard disk space available. The model files are also big, normally more than 100 MB.

Even more multiprocessing can be achieved using the ATI technologies graphic cards, that include much more processors



.Figure 1. A 3D duct with two obstacles. The parts of the obstacles outside the main duct are ignored.

Let see some examples of fluid analyses done with Fluids6. Figure 1 shows a 3D duct with two obstacles edited by the Fluids6 editor. In this model the medium was a liquid (water), with pressure fixed on the left side (inlet) and the outlet (rightmost side) open. The parts of the obstacles that go out of the duct are ignored, because only the flow inside the duct is simulated. Figure 2 shows the velocity field inside the duct. This is a vertical cross section of the duct, crossing the main duct axis. The whole model has 54201 grid points, but the figure only shows the velocity in the points that belong to one section of the duct (its 2D projection). On this and on the following figures, the Fluids6 colors have been suppressed. On the original Fluids6 screens different velocities are marked with corresponding colors.

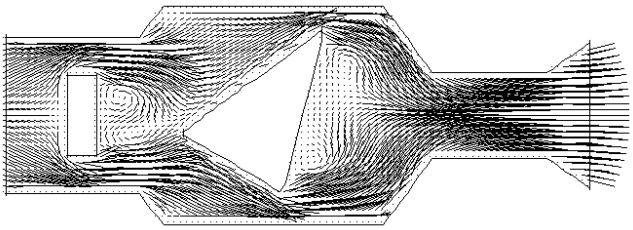


Figure 2. Velocity field for the model of fig. 1. Vertical cross-section.

An interesting result of Fluids6 simulation is the set of flow lines. One of the options of flow line display is to show flow lines that start at randomly generated points in the whole volume. Figure 3 shows an example of a set of such lines. The rotation is clearly shown. This is a 3D image, and the user can change the view angle. Moving the image in this way one can see the spatial 3D shapes of the lines, better than on a static drawing. One

of the observations that can be made looking at the lines is that the turbulence is a complicated 3D structure and can hardly be obtained as a 2D velocity field. This makes any 2D simulation of fluid flow doubtful, because the 2D and 3D models can provide qualitatively different results.

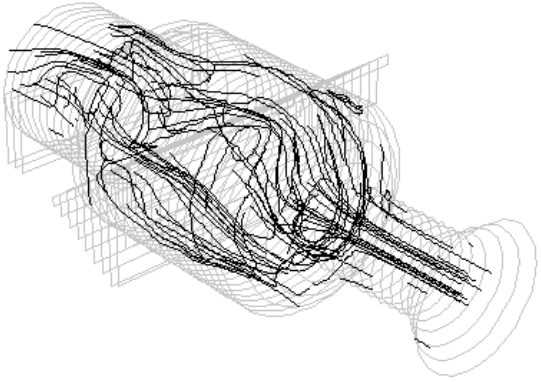


Figure 3. Flow lines with random starting points for the model of fig.1.

OSCILLATING GAS FLOW

One of the important issues discussed here is the simulation of a pneumatic systems, which are excited by a step external pulse (non-oscillating constant pressure), and commence to oscillate spontaneously. There are two reasons for experimenting with such models.

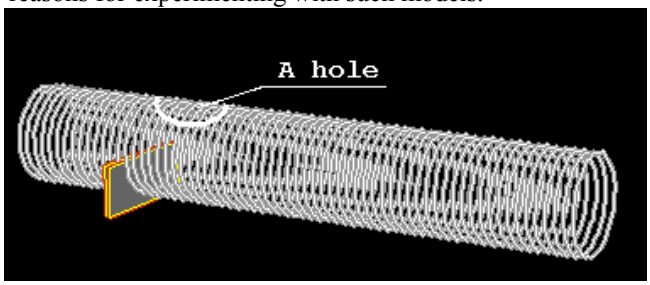


Figure 4. A self-exciting duct.

First, it is a good test for any fluid simulation algorithm or program. As stated before, the original FCTS method is always unstable. There are many ways to stabilize the solution, but, all of them introduce some artificial viscosity or dissipation. This makes the solution stable and reliable, but, in fact, if we apply such tricks, then the model is no longer valid. This may result in difficulties while simulating turbulence and while simulating a resonance and self-exiting pneumatic systems. In particular, the gas oscillations that arise in the real physical system do not appear or are rapidly dumped in models with artificial dissipation. From our experiments we can conclude that the algorithm should work near the limit of the numerical stability. The problem is that if we work with a narrow margin of stability, then it is difficult to create robust software and to distinguish between numerical instability and the instability of the original system

. An example of a self-exciting flow is shown on the figures 4 and 5 (something like a musical instrument).

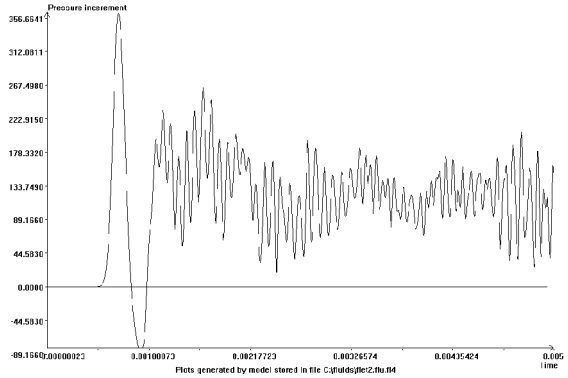


Figure 5. An oscillating flow

The model of figure 4 consists of an open tube with an.obstacle and a hole. Figure 5 shows the pressure oscillations near the right side (duct outlet). A constant pressure is applied to the left side (200 Pascal).
Pressure oscillations for the shape of figure The pressure near the outlet.

Figure 5 shows the pressure oscillations, composed of at least three frequencies: the frequency of the resonance of the left short cylinder, the frequency of the right longer part and the frequency of the perpendicular waves, rebooting between the duct walls.

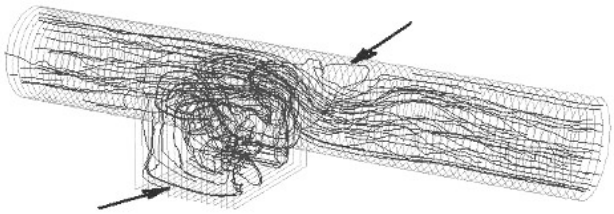


Figure 6. Two regions of counter-flow

Other equally important applications are simulations of shock waves and oscillations in valves, around the wings of a plane or various medical applications, like air turbulence in the human trachea.

MORE EXAMPLES

Figure 6 shows the flow lines in a tube with an appendix (a box added at the lower part of the duct). The air flows from left to right. The lower arrow indicates the counter flow in the box-shaped appendix. Other, somewhat unexpected counter flow appears at the upper part of the duct (the upper arrow). On figure 7 we can see a section of the velocity field for the flow of air in a symmetrical duct. Obviously, the resulting flow needs not to be symmetrical.

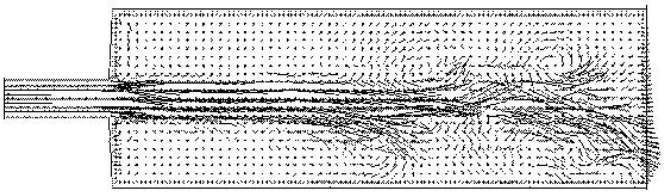


Figure 7. Air flow in a symmetrical duct.

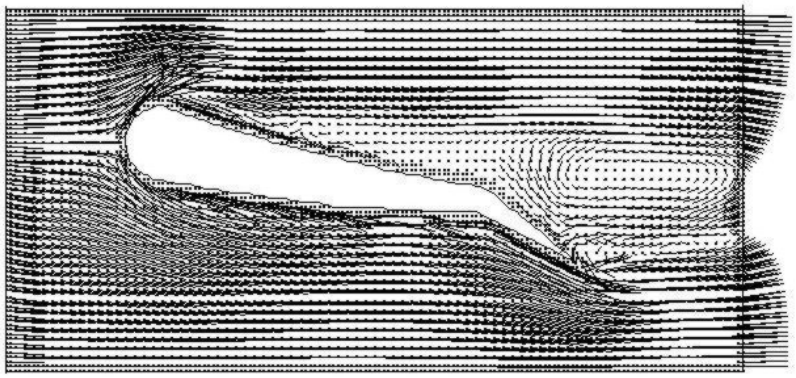


Figure 8 A wing, 0.6 Mach

On figure 8 we can see the air flow around a wing. This flow hardly reaches any steady state. All the above images have been edited to be black and white. On the original Fluids6 screens they look better, with the velocity elements drawn in colors.

CONCLUSIONS

Fluids6 works satisfactory enough to provide, like any other finite-element or grid method, an approximation of the real fluid flow. The flow should be simulated in three-dimensional space, because the flow lines both in turbulent and non-turbulent flows are always complicated 3D structures. Anyway, the real world is 3- and not 2-dimensional.

Simulation of self-exciting pneumatic systems is an interesting challenge, which may be applied in the space-aeronautics industries where a system (nozzle, hydraulic duct, flow around a wing, etc.) may oscillate and provoke unwanted effects, like backflow or strong shock waves.

Multiprocessing is an important feature available even on small machines. CFD is a perfect field of parallel computing, because the problem can be easily decomposed into parallel tasks.

REFERENCES

Ames, W.F. 1977. "Numerical Methods for Partial Differential Equations", *2nd ed., New York, Academic Press.*

Christiansen J.P. 1973. "Numerical Simulation by the Method of Point Vortices", *Journal of Computational Physics*, vol. 13, pp. 363-379.

Currie I.G. "Fundamental Mechanics of Fluids", *McGraw-Hill,New York.*

Harten, A., Lax, P.D. and Van Leer, B. 1983. "On Upstream differencing and Godunov-type schemes for hyperbolic conservation laws", *SIAM Review,* vol.25, pp.36-61.

Hockney, R.W., Eastwood J.W. 1988. "Computer Simulation using Particles", *IOP Publishing, Bristol, 1988.*

Landau L.D., Lifshitz, E.M. 1963. "Fluid Mechanics", *Pergamon Press, London*, Paris 1963.

Leonard, A. 1980. "Vortex Methods for Flow Simulation", *Journal of Computational Physics*, vol. 37, pp. 289-335.

Noronha Gamito, M. 2004. "Simulation of Turbulent Flow Using Vortex Particles", *Revista Electronica, Instituto de Engenharia de Sistemas e Computadores* http://virtual/inesc.pt/virtual/tr/gamito95/artigo.html

Reeves, W.T. 1983. "Particle Systems - A technique for Modeling a Class of Fuzzy Objects". *ACM Computer Graphics,* vol. 17., no.3, pp. 359-376, (Proceedings SIGGRAPH '83)

Stam, J. and Fiume, E. 1993. "Turbulent Wind Fields for Gaseous Phenomena", *ACM Computer Graphics*, vol. 27, no. 4., pp. 369-376, (Proceedings SIGGRAPH '93)

Serrin, J. 1959. "Mathematical Principles of Classical Fluid Mechanics", in: Handbuch der Physik, vol.8, no.1, Springer-Verlag, p 148.

AUTHOR BIOGRAPHY

Stanislaw Raczynski received his master degree (1964) from the Academy of Mining and Metallurgy (AGH) in Krakow Poland, Electrical Engineering Department, his Ph.D. (1969) and Habilitation Degree (1977) from the same Academy, in the area of control theory and optimization methods. In 1964 Dr.Raczynski joined the Institute for Automatics and Industrial Electronics of the Academy of Mining and Metallurgy in Krakow. From 1971 through 1972, he was head of the Computer Center of the AGH. Between 1973 and 1976 he worked as a researcher in the International Research Group in Moscow, USSR. In 1976 Dr.Raczynski became head of the Systems Analysis Group at the Academy of Mining and Metallurgy in Krakow. From 1980 through 1983 he participated in the activities of the European Workshop on Industrial Computer Systems. Between 1983 and 1986 he was a visiting professor of the National University of Mexico. In 1986 Dr.Raczynski joined the Panamericana University in Mexico City, Engineering Department. His didactic activities include courses on control theory, electronics and computer simulation.

Between 1996 and 2000 and then between 2002 and 2004 Dr. Raczynski had been the International Director of The Society for Computer Simulation. In 2003-2004 is the international co-director of the McLeod Institute for Simulation Sciences (part of The Society for Computer Simulation in San Diego, California). He wrote two books on computer simulation and has more than 70 articles and papers published in professional journals and conference proceedings. Books: 1. "Simulacion por computadora" (in Spanish), LIMUSA - Grupo Nowiega, Mexico, 2: "Modeling and Simulation: The Computer Science of Illusion", Wiley, England, April 2006.

Personal page: http//www.raczynski.com/rac/rac.htm