

Simulation of Engine Internal Flows Using Digital Physics

J. Halliday¹, C. Teixeira¹ and C. Alexander¹

¹ Exa Corporation, 450 Bedford Street, Lexington, MA 02175 - United States
e-mail: jackie@exa.com

Résumé — **Simulation des écoulements dans les moteurs avec la technique « Digital Physics »** — Cet article présente la simulation des écoulements dans les conduits d'admission et dans les cylindres avec le code PowerFLOW. La technique numérique sur laquelle repose PowerFLOW, appelée « Digital Physics », est basée sur une théorie statistique de la cinétique. Elle est numériquement stable. La technique « Digital Physics » utilise un nombre élevé de mailles de calcul avec un modèle de turbulence simple procurant une indépendance du maillage et une précision élevée. De plus, la méthodologie est explicite en temps, permettant la simulation des transitoires. Des simulations d'écoulements dans les conduits d'admission et dans les cylindres sont présentées.

Abstract — **Simulation of Engine Internal Flows Using Digital Physics** — This paper presents simulations of engine intake port and cylinder flows performed using PowerFLOW software. The numerical technique behind PowerFLOW, called Digital Physics, is based on statistical kinetic theory and it is numerically stable, so divergence does not occur during calculations. Digital Physics uses large numbers of computational cells with a simple turbulence model, giving grid-independence and high levels of accuracy. In addition, the technique is explicit in time, and so a transient simulation is always obtained. The paper outlines the numerical technique and presents details of an engine port and cylinder simulation.

INTRODUCTION

Recent advances have been made in numerical simulation techniques based upon the kinetic theory of gases, in particular, the development of the Digital Physics® methodology for performing fluid flow simulations. This paper presents the Digital Physics approach, as used in PowerFLOW® fluid flow simulation software [1], and its application to the flow through an engine intake port and cylinder.

The conceptual approach used to solve fluid flow problems in Exa's PowerFLOW code is significantly different from the approach used in all other commercial CFD codes. The main difference is that while traditional CFD approaches start with a mathematical description of a fluid at the continuum level, i.e. the Navier-Stokes equations, Digital Physics simulates fluids at a more fundamental or microscopic level.

There is a strong motivation to simulate a fluid this way. First of all, by using a microscopic prescription, the modeled

physics is simpler and more general than what is captured in the Navier-Stokes equations, a complex set of non-linear partial differential equations (PDEs) that describe the macroscopic behavior of a fluid. The physics is simpler since we can restrict ourselves to capturing the kinetic behavior of particles or collections of particles (i.e. movement and interactions between particles) as opposed to attempting to solve non-linear PDEs, which is very difficult. The microscopic prescription is also more general since by augmenting the particle interactions at this level, more complex fluid physics, such as multi-phase or multi-species flows, can be modeled effectively. For complex fluid flows, a first-principles continuum description may not even exist.

A complete reproduction of all aspects of microscopic particle dynamics would be much too expensive to follow computationally. However it is possible to construct a simplified form of the microscopic dynamics that still contains the sufficient physics to allow accurate recovery of the desired macroscopic behavior.

In essence by simplifying the physics, the microscopic approach becomes computationally feasible. This is the idea underlying the Digital Physics method used in PowerFLOW.

1 NUMERICAL METHOD

Digital Physics is based upon the kinetic theory of gases [2] in which the fluid properties of a small volume of gas are described as a statistical distribution of the mass, momentum and energy of gas molecules. This is a simplified microscopic approach using statistical “particles” rather than molecules, which move and interact in space and time.

We are not concerned with individual molecules, rather a distribution function of molecules lying within a volume element d^3x about \mathbf{x} , and lying in a velocity-space element d^3c about \mathbf{c} . The number of molecules, N , can be expressed as a function of space, velocity and time.

$$N = \int f(\mathbf{x}, \mathbf{c}, t) d^3c d^3r \quad (1)$$

Where the volume elements d^3c , d^3x have finite volume containing a very large number of particles, approximately 10^5 to 10^9 . If the volume elements are small enough that the density does not change rapidly between neighboring elements, then the function in (1) may be considered continuous. Thus a gas can be completely described by its distribution function. The evolution of this distribution follows the well-known Boltzmann transport equation [2, 3].

This microscopic description of the fluid contains enough physical detail that the macroscopic flow behavior is well represented. Indeed, standard kinetic theory applied to a fluid continuum allows the Navier-Stokes equations describing fluid flow to be derived exactly [2].

Digital Physics is a hybrid of “lattice-gas” [4-7] and “lattice-Boltzmann” [3, 8] methods. It solves the Boltzmann transport equation discretely on a regular cubic grid by moving and colliding statistical distributions of particles using the correct microscopic physics.

This means that the Digital Physics computations are exactly equivalent to the full Navier-Stokes equations in describing a flow field [9]. Unlike the Navier-Stokes equations, however, Digital Physics does not directly solve partial differential equations (pde’s), and so the computational problems of solving pde’s, such as numerical instability and divergence, do not arise in the Digital Physics approach.

In Digital Physics, each discrete timestep consists of a complete set of dynamics, so the approach is always time-varying as the fluid evolves from time (t) to time ($t + 1$). The particles have discrete velocities, denoted \mathbf{c}_{ji} where the speed is indicated by the discrete integer, j , and the direction is indicated by the discrete integer, i . Thus the kinetic energy of a particles is described as:

$$\epsilon_j = 1/2 m (\mathbf{c}_{ji} \cdot \mathbf{c}_{ji}) \quad (2)$$

The set of discrete velocity vectors, \mathbf{c}_{ji} , are determined to ensure that the macroscopic fluid dynamics are not dependent on the structure of the underlying grid, which is Cartesian, and are described in Chen *et al.* [4]. This simple fluid grid is combined with a proprietary method for creating surface cells by intersecting the fluid grid with a faceted representation of the surface geometry. As a result, extremely complex geometries can be gridded automatically.

The dynamics computed at each timestep involve two stages, particle movement and collision. The movement stage, “advection” is a linear process in Digital Physics, which makes it highly efficient computationally. The number of particles within a cell with energy j , moving in direction i at location \mathbf{x} and time t , is represented by $N_{ji}(\mathbf{x}, t)$. During the advection stage, this value is updated as follows:

$$N_{ji}(\mathbf{x}, t) \rightarrow N_{ji}(\mathbf{x} + \mathbf{c}_{ji}, t + 1) \quad (3)$$

The collision phase modifies the particles further, and can be represented by:

$$N_{ji}(\mathbf{x}, t) \rightarrow N_{ji}(\mathbf{x}, t) + \Omega_{ji} \{N_{ji}(\mathbf{x}, t)\} \quad (4)$$

Where Ω_{ji} is the “collision operator” that exactly conserves local mass, momentum and energy. The exact form of the collision operator is what determines the transport coefficients in the model, such as viscosity [2]. PowerFLOW uses a very simple collision operator, the so-called single-time relaxation form, also known as the “BGK” form [10] that allows the Maxwell-Boltzmann distribution function to relax to the equilibrium distribution for the fluid, N_{ji}^{eq} , at a rate dictated by the relaxation parameter, ω , as follows:

$$\Omega_{ji} = -\omega (N_{ji}(\mathbf{x}, t) - N_{ji}^{eq}(\mathbf{x}, t)) \quad (5)$$

Thus the update equation for each state can be represented as:

$$N_{ji}(\mathbf{x} + \mathbf{c}_{ji}, t + 1) = N_{ji}(\mathbf{x}, t) + \Omega_{ji} \{N_{ji}(\mathbf{x}, t)h\} \quad (6)$$

That is, in Digital Physics this simple equation is solved, which leads to a hydrodynamics solution indirectly. The fluid properties are derived by taking appropriate moments of N_{ji} . Mass is obtained through the summation of the particles in each cell:

$$\rho(\mathbf{x}, t) = \sum_{ji} N_{ji}(\mathbf{x}, t) \quad (7)$$

Typically each cell in Digital Physics dynamics has approximately 10^5 to 10^6 particles, so that particle discreteness is eliminated.

Momentum is obtained through:

$$\rho \mathbf{u}(\mathbf{x}, t) = \sum_{ji} \mathbf{c}_{ii} N_{ji}(\mathbf{x}, t) \quad (8)$$

Zero momentum, for example, is achieved when the particles moving with the same speed in opposite directions cancel each other out in the summation. If the particles do not cancel each other out, then there is a net momentum at that cell. Since there are more than 10^5 particles per cell, the net momentum is a sum of a significant statistical sample, and so none of the discreteness of particle speeds and directions remains in the net momentum.

Energy at a cell is obtained from the equation:

$$E(\mathbf{x}, t) = \sum_{ji} \varepsilon_i N_{ji}(\mathbf{x}, t) \quad (9)$$

In addition, other higher order properties of the fluid that contain mean-flow gradient information can be obtained from local operations on the distribution function, such as the stress tensor, $\mathbf{S}(\mathbf{x}, t)$ and the energy flux $Q(\mathbf{x}, t)$ as:

$$\mathbf{S}(\mathbf{x}, t) = \sum_{ji} \mathbf{c}_{ii} \mathbf{c}_{ii} N_{ji}(\mathbf{x}, t) \quad (10)$$

$$Q(\mathbf{x}, t) = \sum_{ji} \varepsilon_j \mathbf{c}_{ii} N_{ji}(\mathbf{x}, t) \quad (11)$$

The hydrodynamics equations are derived by taking the mass, momentum and energy moments of the update equation, Equation (6). It can be shown that the continuum equations of hydrodynamics are obtained exactly [4]. As a result, Digital Physics produces accurate and physical predictions of flow.

Through further extension of lattice-type approaches, Digital Physics can robustly be applied to high Reynolds number turbulent flows by using both wall and fluid modeling [11]. This makes it appropriate for simulating the flow speeds typically seen in engineering applications.

The nature of the calculations performed, that is, movement of particles between cells, makes Digital Physics highly efficient computationally. As a result, very large numbers of cells can be used to achieve detailed flow information and grid-independent solutions. A simple turbulence model is then used to model the unresolved turbulence on a local level within each cell. An algebraic turbulence model, the Smagorinsky model [12], is used here.

To summarize, the characteristics of a flow simulation using the Digital Physics techniques are:

- the simulation is always time-varying (transient);
- the computations are numerically stable;
- high computational efficiency which allows large numbers of cells to be used;
- the process gives excellent scalability on parallel processor machines.

These characteristics mean that PowerFLOW is very well suited to many aspects of engineering development, allowing rapid, accurate assessment of a component.

2 GRID GENERATION TECHNIQUE

The user starts with a representation of the surface of the component in the form of a surface mesh of triangles. The surface mesh is imported into PowerFLOW's pre-processor as a STL, Ansys or Nastran file. The user then defines a box that surrounds the component completely. This box, or "simulation volume", contains a grid of cubic cells. Fluid cells and solid cells are defined automatically. Where the surface facets cut the cubic cells, the surface is automatically sub-divided into surface elements, so that the exact representation of the component surface is maintained.

During simulation, the interaction between particles in fluid cells and surface elements is determined, as the fluid is reflected back from the surface into the fluid volume. Since the surface is completely defined by this approach, highly complex surfaces require no more user effort than do simple surfaces.

PowerFLOW allows the user to define regions of additional, higher grid resolution, in order to make the simulation even more computationally efficient without any loss in accuracy. These additional resolution regions can be defined as simple boxes or as more complex shapes that may be imported as facetized geometries. The resolution regions are nested so that many resolution regions can be defined. PowerFLOW automatically creates the computational grid defining fluid and solid cells, surface elements and increasing or decreasing resolution. As a result the setup time for a case is independent of the surface complexity and grid resolution.

3 PORT AND CYLINDER SIMULATION

PowerFLOW was used to simulate the flow through a helical intake port and cylinder, the geometry is shown in Figure 1.



Figure 1
Port and cylinder geometry.

The simulation replicated steady-state flow-bench tests where air enters the intake port and discharges to the atmosphere through an extended cylinder. Simulations were performed for three valve lifts; 7 mm, 9 mm and 11 mm. A stagnation pressure boundary condition was applied to the inlet of the intake port and a static pressure boundary condition was applied to the outlet of the cylinder. The case setup took 30 min of user time.

The simulation volume contained cubic cells 1.7 mm in length. Two box-shaped additional resolution regions were added around the valve region, which gave a cell size of 0.42 mm in the region of the valve. This configuration resulted in a grid containing 3.9 million fluid cells.

The simulation was performed on a 4-processor, 250 MHz, Sun Ultra workstation taking 22 h to complete. The case initial conditions were results from a low-resolution simulation, which took 7 h to complete.

4 RESULTS

Results from the simulations can be seen visually in Figures 2 and 3.

The flow structure predicted is a complex interaction of vortices, with a major vortex forming in the cylinder that precesses about the cylinder axis.

The volume flow rate through the system was compared to experimental data. The difference between experimental and PowerFLOW flow rates were 3.2%, 2.3% and 6.0% for the 7 mm, 9 mm and 11 mm valve lift simulations respectively.

In addition, the swirl moment was derived from the PowerFLOW results. A swirl meter was used in the

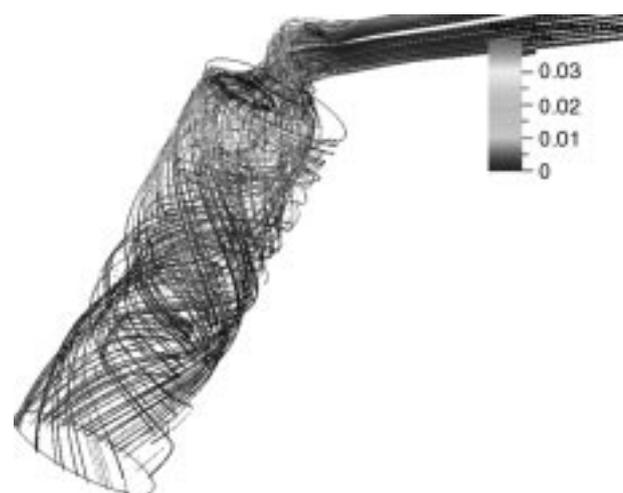


Figure 2
Streamlines showing flow structure.

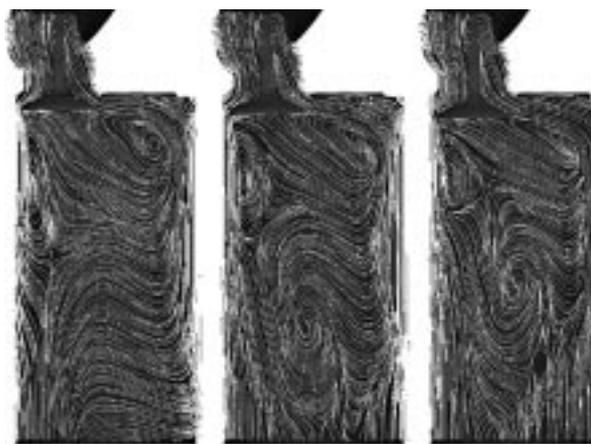


Figure 3
Cross-sections showing flow through valve and cylinders.

experiment to measure the swirl moment. The swirl meter is an intrusive device, however, causing small changes to the flow field. Therefore it is not appropriate to compare the absolute measurements of swirl moment between experiment and PowerFLOW. Instead, the swirl moment obtained for the 9 mm valve lift case was used to normalize the swirl moment in the 7 mm and 11 mm valve lift cases. This gives an indication of the relative change in swirl between like-configurations.

The swirl moment ratio predicted by PowerFLOW differed from experiment by 5.7% for the 7 mm valve lift case and by 5.9% for the 11 mm valve lift case. From all comparisons of predicted data to experiment, PowerFLOW simulations gave excellent correlation giving both qualitative and quantitative information.

CONCLUSIONS

A highly accurate simulation of engine port and cylinder flow was achieved using PowerFLOW software. The simulation was grid-independent, as confirmed by another study to be published. The simulation required very little engineer time to prepare, which significantly impacts the resources that can be dedicated to performing flow simulations. Indeed with little user setup time required, engineers are free to focus on interpreting results and performing more simulations, hence working very effectively towards product design and development.

ACKNOWLEDGMENTS

The authors would like to thank Dave Hill of *Exa Corporation* for his assistance in the preparation of this paper.

REFERENCES

- 1 *PowerFLOW User Guide Release 3.0* (1998) Exa Corporation.
- 2 Huang, K. (1963) *Statistical Mechanics*, John Wiley & Sons, Inc.
- 3 Cercignani, C. (1975) *The Boltzmann Equation and its Applications*, Springer.
- 4 Chen, C.H., Teixeira, C. and Molvig, K. (1994) Digital Physics Approach to Computational Fluid Dynamics: Some Basic Theoretical Features. *International Journal of Modern Physics C*, **8**, 4, 675-684.
- 5 Frisch, U., Hasslacher, B. and Pomeau, Y. (1986) *Phys. Rev. Lett.*, **56**, 1505.
- 6 Wolfram, S. (1986) *J. Stat. Phys.*, **45**, 471.
- 7 Frisch, U., d'Humieres, D., Hasslacher, B., Lallemand, P., Pomeau, Y. and Rivet, J.P. (1987) *Complex Syst.*, **1**, 649.
- 8 Chen, H. (1993) Discrete Boltzmann Systems and Fluid Flows. *Computers in Physics*, **7**, 6.
- 9 Teixeira, C. (1992) Continuum Limit of Lattice Gas Fluid Dynamics. *PhD Thesis*, MIT.
- 10 Bhatnagar, P., Gross, E. and Krook, M. (1954) *Phys. Rev.*, **94**, 511.
- 11 Teixeira, C. (1999) Incorporating Turbulence Models into the Lattice-Boltzmann Method. *Int. J. of Mod. Phys. C*, in press.
- 12 Smagorinsky, J. (1963) *Monthly Weather Review*, **91**, 99-164.

Final manuscript received in March 1999