V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

FLUID-STRUCTURE COUPLING SIMULATIONS USING A VIRTUAL FLUX METHOD

Koji Morinishi* **and Tomohiro Fukui†**

*Kyoto Institute of Technology
Matsugasaki, Sakyo-ku, Kyoto 606-8585, Japan
e-mail: morinisi@kit.ac.jp

†Kyoto Institute of Technology
Matsugasaki, Sakyo-ku, Kyoto 606-8585, Japan
e-mail: fukui@kit.ac.jp

Key words: Computational Fluid Dynamics, Fluid-Structure Interaction, Coupling Problems, Virtual Flux Method

Abstract. *This paper describes the resent development of a virtual flux method for simulating fluid-structure interaction problems. The virtual flux method is one of the sharp interface Cartesian grid methods. The numerical flux across the interface is replaced with the virtual flux so that proper interface conditions must be satisfied there. In this study, the virtual flux method is applied to numerical flow simulations about reciprocating engines and compressors. In the reciprocating engine simulation, the compressible Navier-Stokes equations are coupled with the equation of motion of the piston, connecting rod, and crank system. Intake and exhaust valves are lifted up and down according with the crank angle in the intake and exhaust strokes. Instead of modeling the complex fuel combustion process, a proper amount of energy is added to the Navier-Stokes equation at the beginning of each expansion stroke, to retain the four stroke engine cycle at a constant revolution rate. The engine comes to work at the revolution rate intended after some initial transition cycles. On the other hand, the intake and exhaust valves of the reciprocating compressors are driven with fluttering by the pressure force acting on them, while the piston is driven at a constant rate. Numerical simulation indicates the compressor comes to work at almost constant flow ate, after the first transition cycle. It is confirmed that the virtual flux method is easily applicable to the simulation of fluid-structure interaction problems*

1 INTRODUCTION

Numerical simulation of fluid-structure interaction problems is a current topics in the computational fluid dynamics. Using the arbitrary Lagrangian Eulerian (ALE) method[1,2] is a straightforward strategy for simulating the fluid-structure interaction problems. The ALE method may be accurate, since the fluid-structure interface is conformed to a moving boundary of computational grid. Conforming the grid boundary to the moving interface is generally difficult and time-consuming especially for the interface under large deformation.

Cartesian grid methods[3,4] are another strategy fit for simulating the fluid-structure interaction problems. The methods need no grid reconstruction even if large deformation of fluid-structure interface takes place. One drawback of the Cartesian grid methods, however, is that most of the methods may smear the interface over several mesh spacing. Thus a sharp interface Cartesian grid method[5,6] is preferable for simulation of fluid-structure interaction problems.

The virtual flux method[7] is one of the sharp interface Cartesian grid methods. The numerical flux of the Navier-Stokes equations across the interface is replaced with the virtual flux so that proper interface conditions must be satisfied there. In this study, the applicability of the virtual flux method to the fluid-structure interaction problems is demonstrated in numerical flow simulations about reciprocating engines and compressors.

In the following, the governing equations are described in Section 2, and Section 3 provides the numerical procedure. The virtual flux method is applied to the reciprocating engines in Section 4 and to the reciprocating compressors in Section 5. The results of this study are summarized in Section 6.

2 GOVERNING EQUATIONS

Fluid flows in the reciprocating compressors and engines may be described by the compressible Navier-Stokes equations, which can be written in the following form.

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F} = \nabla \cdot \mathbf{S} \tag{1}$$

where $\mathbf{q}$ is the conservative vector, $\mathbf{F}$ the convective flux vector, and $\mathbf{S}$ the viscous flux vector. The conservative vector and the flux vectors are given with:

$$\mathbf{q} = \begin{pmatrix} \rho \\ \mathbf{u} \\ e \end{pmatrix} \quad , \quad \mathbf{F} = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u} + p\mathbf{I} \\ \mathbf{u}(e+p) \end{pmatrix} \quad , \quad \mathbf{S} = \begin{pmatrix} 0 \\ \tau \\ \mathbf{u}\cdot\tau - \mathbf{h} \end{pmatrix} \tag{2}$$

where ρ is the density, $\mathbf{u}$ the velocity vector, p the pressure, e the total energy per unit volume, τ the viscous stress tensor, and $\mathbf{h}$ the heat flux vector. The total energy e is defined for a perfect gas as:

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\mathbf{u}^2 \tag{3}$$

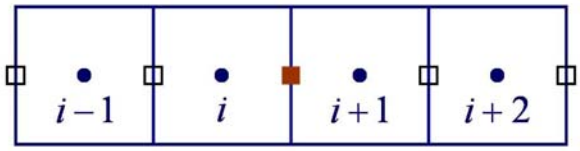

Figure 1: Stencil about a regular point.

where γ is the ratio of specific heats.

The components of the viscous stress tensor and the heat flux vector, for example, τ_{xx}, τ_{xy}, and h_x, may be written as:

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \tag{4}$$

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{5}$$

$$h_x = -\kappa\frac{\partial T}{\partial x} \tag{6}$$

where u, v, and w are the velocity components, T is the temperature, μ the viscosity and κ the thermal conductivity. In this study Smagorinsky's eddy viscosity model is used for practical turbulent flow simulations,

3 NUMERICAL PROCEDURE

Obtaining numerical solution of the compressible Navier-Stokes equations for flows about complex geometries on a Cartesian grid, the virtual flux method is introduced in conventional spatial discretizing methods.

3.1 Spatial discretizing method

The convective terms of the compressible Navier-Stokes equations can be evaluated simply at a regular point on the Cartesian grid, for example, as:

$$\left(\frac{\partial \mathbf{F}_x}{\partial x}\right)_i = \frac{\mathbf{F}_x(\tilde{\mathbf{q}}^+_{i+1/2}, \tilde{\mathbf{q}}^-_{i+1/2}) - \mathbf{F}_x(\tilde{\mathbf{q}}^+_{i-1/2}, \tilde{\mathbf{q}}^-_{i-1/2})}{\Delta x} \tag{7}$$

where $\tilde{\mathbf{q}}$ is the primitive variable, Δx denotes the mesh spacing, and i the cell index as shown in Fig. 1. The numerical flux at the cell surface is obtained as:

$$\mathbf{F}_x(\tilde{\mathbf{q}}^+_{i+1/2}, \tilde{\mathbf{q}}^-_{i+1/2}) = \frac{1}{2}\left\{\mathbf{F}_x(\tilde{\mathbf{q}}^+_{i+1/2}) + \mathbf{F}_x(\tilde{\mathbf{q}}^-_{i+1/2}) - |\mathbf{A}_x|(\tilde{\mathbf{q}}^+_{i+1/2} - \tilde{\mathbf{q}}^-_{i+1/2})\right\} \tag{8}$$

where $|\mathbf{A}_x|$ is defined as:

$$|\mathbf{A}_x| = \mathbf{R}_x|\mathbf{\Lambda}_x|\mathbf{R}_x^{-1} \tag{9}$$

Here $\mathbf{\Lambda}_x$ and $\mathbf{R}_x$ are, respectively, the eigenvalue matrix and the right eigenvector matrix of the flux Jacobian matrix $\mathbf{A}_x$, which is defined as:

$$\mathbf{A}_x = \frac{\partial \mathbf{F}_x}{\partial \tilde{\mathbf{q}}} \tag{10}$$

The third order method may be obtained, for example, if $\tilde{\mathbf{q}}^-_{i+1/2}$ are evaluated with the following reconstruction.[8]

$$\tilde{\mathbf{q}}^-_{i+1/2} = \tilde{\mathbf{q}}_i + \frac{1}{2}\left(\omega_0 \Delta\tilde{\mathbf{q}}_{i+1/2} + \omega_1 \Delta\tilde{\mathbf{q}}_{i-1/2}\right) \tag{11}$$

Here $\Delta\tilde{\mathbf{q}}_{i+1/2}$ are obtained with:

$$\Delta\tilde{\mathbf{q}}_{i+1/2} = \tilde{\mathbf{q}}_{i+1} - \tilde{\mathbf{q}}_i \tag{12}$$

The weights ω_0 and ω_1 for the third order linear method are:

$$\omega_0 = \frac{2}{3}\,, \quad \omega_1 = \frac{1}{3}\,. \tag{13}$$

The weights ω_0 and ω_1 are defined for a nonlinear method as:

$$\omega_0 = \frac{2\alpha_0}{2\alpha_0 + \alpha_1}\,, \quad \omega_1 = \frac{\alpha_1}{2\alpha_0 + \alpha_1} \tag{14}$$

where α_0 and α_1 are obtained as:

$$\alpha_0 = \frac{1}{\Delta\tilde{q}_{i+1/2} + \epsilon}\,, \quad \omega_1 = \frac{1}{\Delta\tilde{q}_{i-1/2} + \epsilon} \tag{15}$$

Here ϵ is a small number which prevents null division in smooth flow regions.

The viscous terms of the Navier-Stokes equations are evaluated simply with the second order central difference approximation. For example,

$$\left(\frac{\partial \mathbf{S}_x}{\partial x}\right)_i = \frac{\mathbf{S}_x(\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}_{i+1}) - \mathbf{S}_x(\tilde{\mathbf{q}}_{i-1}, \tilde{\mathbf{q}}_i)}{\Delta x} \tag{16}$$

3.2 Virtual flux method

If an immersed solid boundary is located between the points i and $i+1$ as shown in Fig. 2, the numerical fluxes of Eq.(7) must be modified so that no-slip and no-penetration velocity boundary conditions are satisfied on the solid boundary.

$$\left(\frac{\partial \mathbf{F}_x}{\partial x}\right)_i = \frac{\mathbf{F}_x(\tilde{\mathbf{q}^*}^+_{i+1/2}, \tilde{\mathbf{q}^*}^-_{i+1/2}) - \mathbf{F}_x(\tilde{\mathbf{q}^*}^+_{i-1/2}, \tilde{\mathbf{q}}^-_{i-1/2})}{\Delta x} \tag{17}$$

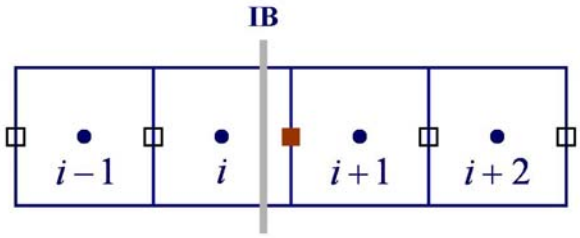

Figure 2: Stencil about a interfacial point.

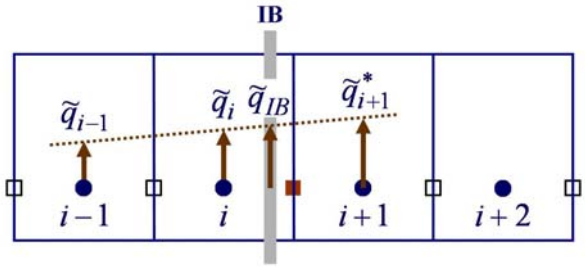

Figure 3: Stencil for a one-sided exterpolation.

where $\tilde{\mathbf{q}^*}^+_{i+1/2}$, $\tilde{\mathbf{q}^*}^-_{i+1/2}$, and $\tilde{\mathbf{q}^*}^+_{i-1/2}$ are reconstructed with considering the immersed solid boundary conditions. For example,

$$\tilde{\mathbf{q}^*}^-_{i+1/2} = \tilde{\mathbf{q}}_i + \frac{1}{2}\left(\omega_0 \Delta\tilde{\mathbf{q}^*}_{i+1/2} + \omega_1 \Delta\tilde{\mathbf{q}}_{i-1/2}\right) \tag{18}$$

where $\Delta\tilde{\mathbf{q}^*}_{i+1/2}$ are calculated with:

$$\Delta\tilde{\mathbf{q}^*}_{i+1/2} = \tilde{\mathbf{q}^*}_{i+1} - \tilde{\mathbf{q}}_i \tag{19}$$

Here $\tilde{\mathbf{q}^*}_{i+1}$ are obtained with one-sided first or second order exterpolating operators L as shown in Fig. 3:

$$\tilde{\mathbf{q}^*}_{i+1} = L\left[\tilde{\mathbf{q}}_{i-1}, \tilde{\mathbf{q}}_i, \tilde{\mathbf{q}}_{IB}\right] \tag{20}$$

or

$$\tilde{\mathbf{q}^*}_{i+1} = L\left[\tilde{\mathbf{q}}_{i-1}, \tilde{\mathbf{q}}_i, \left(\frac{\partial\tilde{\mathbf{q}}}{\partial x}\right)_{IB}\right] \tag{21}$$

where $\tilde{\mathbf{q}}_{IB}$ and $\left(\frac{\partial\tilde{\mathbf{q}}}{\partial x}\right)_{IB}$ are Dirichlet and Neumann boundary conditions at the immersed boundary, respectively.

The numerical viscous fluxes of Eq.(16) must be also modified so that no-slip and no-penetration velocity boundary condistions are satisfied on the solid boundary as:

$$\left(\frac{\partial\mathbf{S}_x}{\partial x}\right)_i = \frac{\mathbf{S}_x(\tilde{\mathbf{q}}_i, \tilde{\mathbf{q}^*}_{i+1}) - \mathbf{S}_x(\tilde{\mathbf{q}}_{i-1}, \tilde{\mathbf{q}}_i)}{\Delta x} \tag{22}$$

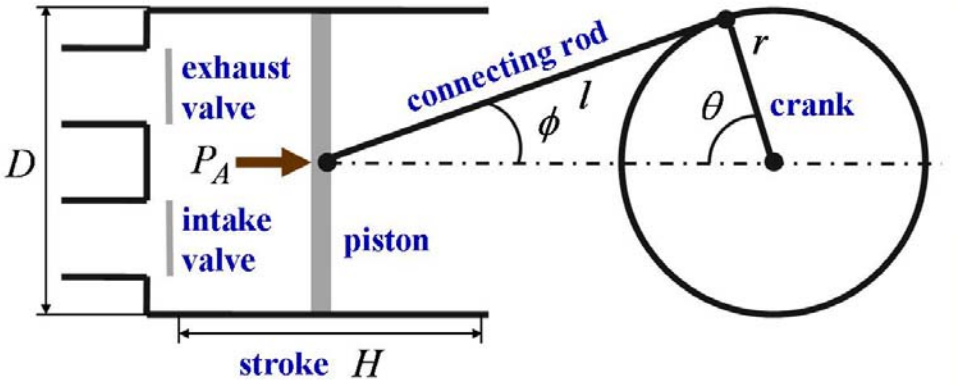

Figure 4: Schematic model of a reciprocating engine.

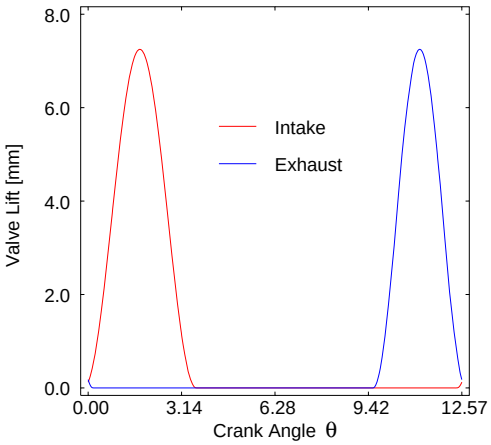

Figure 5: Valve lift history as a function of crank angle.

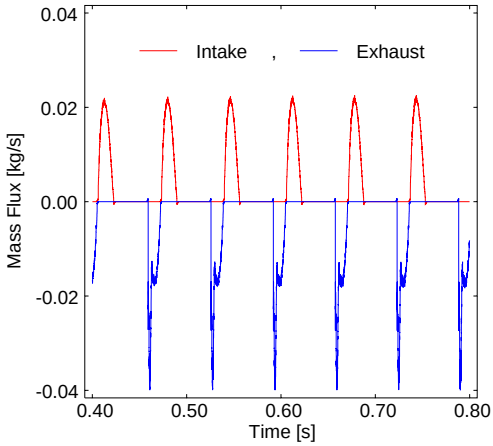

Figure 6: Intake and exhaust mass flux.

3.3 Time stepping method

After discretizing the spatial derivatives, the Navier-Stokes equations can be written in the form:

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{Q}(\mathbf{q}) \tag{23}$$

The second order Runge-Kutta method is used for solving the equation as:

$$\mathbf{q}^{n+1/2} = \mathbf{q}^n + \frac{1}{2}\Delta t \mathbf{Q}(\mathbf{q}^n) \tag{24}$$

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \mathbf{Q}(\mathbf{q}^{n+1/2}) \tag{25}$$

where the superscript n denotes the time index and Δt the time step size.

4 APPLICATION TO RECIPROCATING ENGINE

In this section, a fluid-structure coupling simulation is carried out to reproduce the four strokes of a reciprocating engine. At the beginning of the expansion stroke, a proper amount of energy is added to the Navier-Stokes equation, instead of modeling the complex fuel combustion process, to retain the engine cycle.

4.1 Reciprocating engine model

The computational reciprocating engine model is schematically drawn in Fig. 4. The revolution of crank shaft is described with the following equation of motion.

$$I(\theta)\frac{d^2\theta}{dt^2} + \frac{1}{2}\frac{dI(\theta)}{d\theta}\left(\frac{d\theta}{dt}\right)^2 = M \tag{26}$$

where θ is the crank angel and M the moment of force acting on the crank shaft. The total inertia moment of piston, connecting rod, and crank $I(\theta)$ is defined as

$$I(\theta) = I_c + (m_p + m_{cr})r^2 sin^2\theta \left(1 + \frac{r}{l}\frac{cos\theta}{cos\phi}\right)^2 + I_{cr}\left(\frac{r}{l}\right)^2 \frac{cos^2\theta}{cos^2\phi} \tag{27}$$

Here m_p and m_{cr} are the mass of the piston and connecting rod, and I_c and I_{cr} are the inertia moment of the crank and connecting rod, respectively. Between the radius of the crank r, the length of the connecting rod l , and its slope angle ϕ, the following equations are obtained from Fig. 4.

$$sin\phi = \frac{r}{l}sin\theta \tag{28}$$

$$cos\phi = \sqrt{1 - \left(\frac{r}{l}\right)^2 sin^2\theta} \tag{29}$$

For the moment of force acting on the crank shaft, the moment of pressure force M_p acting on the piston head and the moment of load M_l acting on the crank shaft are considered and assumed as:

$$M_p = P_A cos\phi\, sin(\theta + \phi) \tag{30}$$

$$M_l = -\zeta \left|\frac{d\theta}{dt}\right| \frac{d\theta}{dt} \tag{31}$$

where P_A is the pressure force acting on the piston head and ζ a proper constant.

Figure 5 shows the valve lift designed as a function of the crank angle. The intake valve is lifted up and down according with the crank and connecting rod pulling the piston down from the top dead center (TDC) to the bottom dead center (BDC) in the intake stroke. The exhaust valve is also lifted up and down according with connecting rod driving the piston from BDC to TDC in the exhaust stroke.

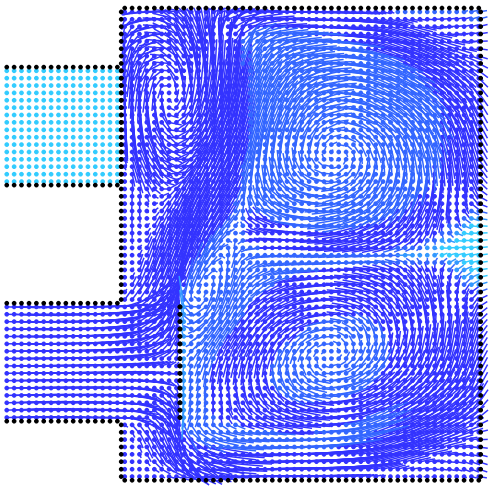

Figure 7: Velocity vectors at an instant in intake stroke.

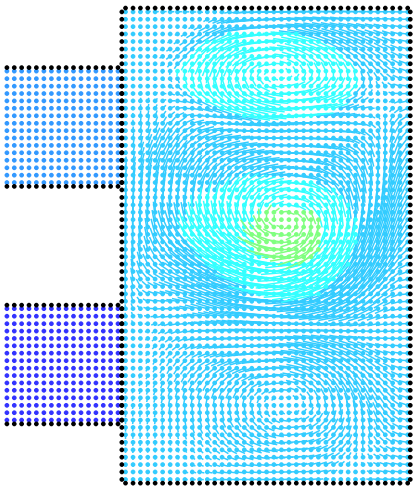

Figure 8: Velocity vectors at an instant in compression stroke.

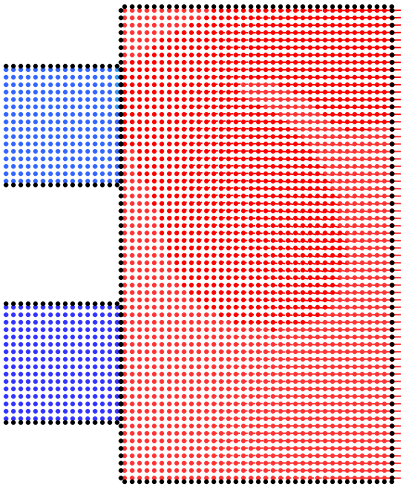

Figure 9: Velocity vectors at an instant in expansion stroke.

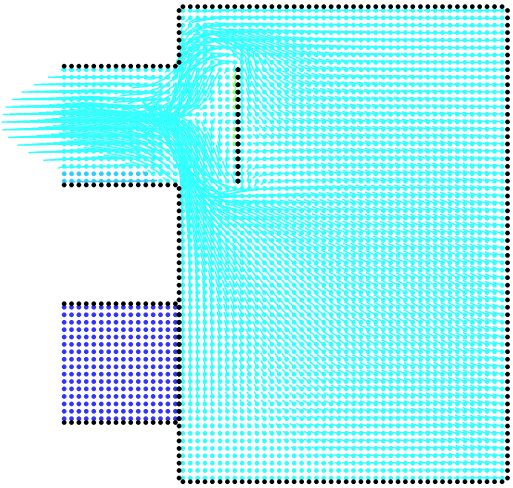

Figure 10: Velocity vectors at an instant in exhaust stroke.

4.2 Numerical results

The computation is carried out for the cylinder width D of 5.8 cm and stroke H ($= 2r$) of 5.8 cm. Certain amount of energy is added to the Navier-Stokes equation at the exapntion stroke to retain the angular velocity of crank at about 1800 rpm.

The intake mass flux and exhaust mass flux are plotted in Fig. 6. After some initial transition cycles, the engine comes to work at the almost constant rate of 1818 rpm. Typical velocity vectors, colored by the temperature obtained for the four strokes, are plotted in Figs. 7 - 8. Fresh gases flow into the cylinder through the intake port (Fig. 7).

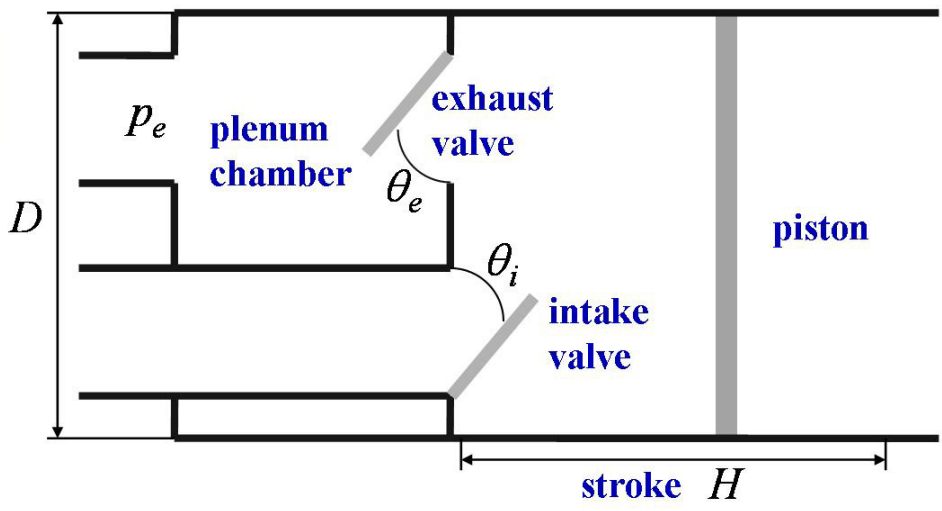

Figure 11: Schematic model of a reciprocating compressor.

As connecting rod driving the piston from BDC to TDC, the gases are being compressed (Fig. 8). The hot gases, heated by energy added, are driving the piston to BDC (Fig. 9), and the gases flow out through exhaust port (Fig. 10). The four stroke engine cycle is clearly reproduced in the simulation.

5 APPLICATION TO RECIPROCATING COMPRESSOR

In a reciprocating compressor, the intake and exhaust valves are driven by the pressure force acting on them, while the piston is driven at a constant rate. Here the fluid-structure coupling simulation is carried out to reproduce the valve response and gas flows in the reciprocating compressor.

5.1 Reciprocating compressor model

The computational model of reciprocating compressor considered is schematically drawn in Fig. 11. The motion of intake valve and exhaust valve is assumed to be described with the following equation of motion.

$$I\frac{d^2\theta}{dt^2} + k\theta = M \tag{32}$$

where θ is the open angle of intake valve θ_i or exhaust valve θ_e, k the spring stiffness, and M the moment of force acting on each valve.

5.2 Numerical results

The computation is carried out for the cylinder diameter D of 2.5 cm and stroke H of 2.5 cm. The piston is operated at the constant revolution rate of 3000 rpm. The pressure ratio of exhaust pressure p_e to intake pressure p_∞ is 10.

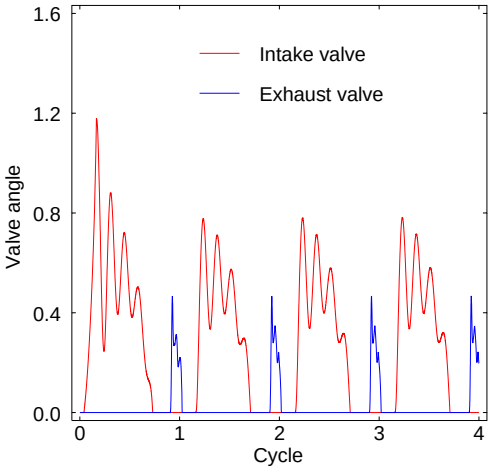

Figure 12: Valve angle history as a function of cycle.

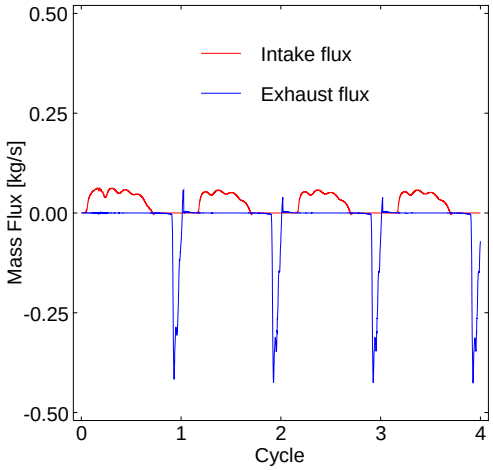

Figure 13: Intake and exhaust mass flux as a function of cycle.

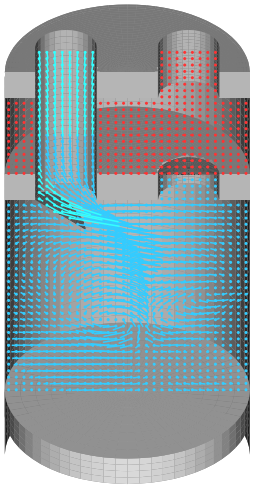

Figure 14: Velocity vectors at an instant in intake stroke.

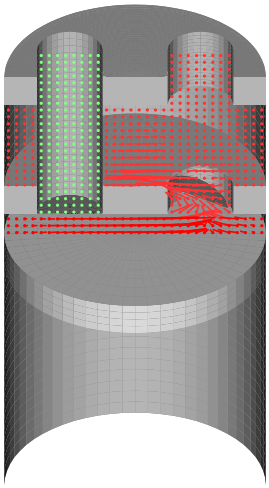

Figure 15: Velocity vectors at an instant in exhaust stroke.

Figure 12 shows the history of valve angle as a function of the cycle. Valve fluttering is observed. Corresponding intake mass flux and exhaust mass flux are plotted in Fig. 13. After the first transition cycle, the compressor comes to work at an almost constant flow rate. Typical velocity vectors, colored by the pressure obtained for intake and exhaust strokes, are plotted in Figs. 14 and 15, respectively. Fresh gases flow into the cylinder through the intake port (Fig. 14), and the gases are compressed and driven out through exhaust port (Fig. 15). The work of reciprocating compressor is clearly reproduced in the simulation.

6 CONCLUSIONS

The virtual flux method has been developed for simulating fluid-structure interaction problems. The numerical flux across the fluid-structure interface is successfully replaced with the virtual flux, so that proper interface conditions are satisfied there. In this study, the virtual flux method is applied to numerical flow simulations about reciprocating engines and compressors. In the reciprocating engine simulation, in order to retain the four stroke engine cycle, a proper amount of energy is added to the Navier-Stokes equation at the beginning of the expasion stroke, instead of modeling the complex fuel combustion process. After some initial transition cycles, the engine comes to work at the constant revolution rate intended. On the other hand, the valves of reciprocating compressor are driven with fluttering by the pressure force acting on them. After the first transition cycle, the compressor comes to work at almost constant flow rate.

REFERENCES

[1] C. W. Hirt, A. A. Amsden, J. L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, *Journal of Computational Physics*, **14**, 227–253 (1974).

[2] J. Donea, S. Giuliani and J.P. Halleux, An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions, *Computer Methods in Applied Mechanics and Engineering*, **33**, 689–723 (1982).

[3] C. S. Peskin, Numerical analysis of blood flow in the heart, *Journal of Computational Physics*, **25**, 220–243 (1977).

[4] J. Kim, D. Kim, and H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *Journal of Computational Physics*, **171**, 132–150 (2001).

[5] H. S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, *Journal of Computational Physics*, **174**, 345–380 (2001).

[6] D. V. Le, B. C. Khoo, and J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *Journal of Computational Physics*, **220**, 109–138 (2006).

[7] I. Tanno, K. Morinishi, K. Matsuno, and N. Nishida, Validation of virtual flux method for forced convection flow, *JSME International Journal* **49-4**, 1141–1148 (2006).

[8] G. S. Jiang and C. W. Shu, Efficient implementation of weighted ENO schemes, *Journal of Computational Physics*, **126**, 202–228 (1996).