

## Simulation of hydrogen dispersion by the domain decomposition method

Kanayama, Hiroshi  
Graduate School of Engineering, Kyushu University

Tsukikawa, Hisayoshi  
Graduate School of Engineering, Kyushu University

Ismail, Idris  
Graduate School of Engineering, Kyushu University

<https://hdl.handle.net/2324/25386>

---

出版情報 : Japan Journal of Industrial and Applied Mathematics. 28 (1), pp.43-53, 2011-04.  
Springer Japan  
バージョン :  
権利関係 : (C) The JJIAM Publishing Committee and Springer 2011

# Simulation of Hydrogen Dispersion by the Domain Decomposition Method

Hiroshi KANAYAMA, Hisayoshi TSUKIKAWA, and Idris ISMAIL

*Graduate School of Engineering, Kyushu University,*

*744, Motoooka, Nishi-ku, Fukuoka 819-0395, Japan*

E-mail: [kanayama@mech.kyushu-u.ac.jp](mailto:kanayama@mech.kyushu-u.ac.jp)

We demonstrate the feasibility of the domain decomposition method in simulating large scale finite element models through the ADVENTURE code, an open source freeware partly developed by the Computational Mechanics Laboratory at Kyushu University. Our model is that of hydrogen dispersion in a partially open space, chosen because of its relevance to the safe use of hydrogen as a potential replacement for fossil fuels. An analogy of the Boussinesq approximation is applied in our simulation. We describe the formulations and the model, followed by some results.

*Key words: hierarchical domain decomposition method, finite element method, hydrogen dispersion*

# 1. Introduction.

The benefits of using computational simulations in science and engineering cannot be understated. This is evidenced by the savings that may be achieved in terms of cost and time, by minimizing physical experiments through extensive testing via numerical analysis. Much progress in the development of such tools has been made by various groups and organizations, as can be seen from the continuous expansion of facilities in today's commercial software. However, there is still a need for robust, flexible codes that can handle large scale analyses in acceptable time. Recognizing this, a group of researchers from several universities and industrial bodies have developed the ADVENTURE system [1], an open source freeware that utilizes the Hierarchical Domain Decomposition Method (HDDM) in its finite element code. HDDM enables parallel processing to be performed; thus, very large scale analyses may be computed, with the upper limit to size theoretically bound by the number of processors available.

This paper demonstrates the suitability of the ADVENTURE code ('the Code', unless otherwise stated) in simulating large scale, real-world phenomena. In this work, the dispersion of hydrogen in a partially open space is simulated using ADVENTURE\_sFlow ('sFlow') [2]-[2d], one of several modules in the ADVENTURE system. Such a simulation was chosen in view of both the current and predicted world energy situation. Hydrogen is seen as a possible replacement to fossil fuels, which, besides being highly polluting, are fast being depleted. However, several obstacles must be overcome if hydrogen is to be used as a mainstream source of energy, of which safe storage is one of them. The minimum ignition energy of hydrogen is very low; as a result of this, the leakage of hydrogen in a confined space introduces the possibility of accidental ignition, which in the worst case, may result in an explosion. Such safety issues necessitate special facilities if extensive physical experiments are to be carried

out safely and frequently, further justifying the use of computational simulations. In addition to these, experimental data for the simulation exist [3].

The sFlow module was originally written to solve incompressible viscous flow problems. Later versions have been written to handle thermal convection problems, and by applying an analogy of the Boussinesq approximation, may be adapted to handle gas dispersion problems. A similar model and method was employed in a previous paper [4]; however, the used program was written specifically for computation with a single processor. In our work, the same model is used, but on a much larger scale through the use of parallel processing via the Code and HDDM. The aim of this paper is to demonstrate the feasibility of the Code and HDDM in simulating large scale engineering problems. To the best of our knowledge, no work has been published thus far on the use of HDDM to facilitate parallel processing for non-stationary gas dispersion problems – the current work is the first in which the Code, through the sFlow module, is used to compute such a problem. By using an analogy to simulate the dispersion of hydrogen in a partially open space, we also show the flexibility and robustness of the Code.

## 2. Formulations

Let  $\Omega$  be a three-dimensional polyhedral domain with the boundary  $\partial\Omega$ . We consider the non-stationary Navier-Stokes equations and the advection diffusion equation as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} - 2\nu \nabla \cdot \mathbf{D}(\mathbf{u}) + \nabla p = -\beta C \mathbf{g} \quad \text{in } \Omega \times (0, T), \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T), \quad (2)$$

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C - a \Delta C = S \quad \text{in } \Omega \times (0, T), \quad (3)$$

where  $\mathbf{u} = (u_1, u_2, u_3)^T$  is the velocity [m/s],  $t$  is time [s],  $\nu$  is the kinematic viscosity coefficient [ $\text{m}^2/\text{s}$ ],  $p$  is the gas mixture gauge pressure ('pressure') normalized by the density [ $\text{m}^2/\text{s}^2$ ],  $\mathbf{g} = (g_1, g_2, g_3)^T$  is gravity [ $\text{m}/\text{s}^2$ ],  $\beta$  is the coefficient [-],  $C$  is the mass

concentration of hydrogen [mass%],  $a$  is the hydrogen diffusion coefficient in air [ $\text{m}^2/\text{s}$ ],  $S$  is the source term [ $1/\text{s}$ ] and  $D_{ij}$  is the rate of strain tensor [ $1/\text{s}$ ] defined by

$$D_{ij}(\mathbf{u}) \equiv \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad i, j = 1, 2, 3. \quad (4)$$

The following boundary conditions are applied, where  $\Gamma_u$  and  $\Gamma_c$  denote the boundary with specified velocity and concentration respectively:

$$\mathbf{u} = \hat{\mathbf{u}} \quad \text{on } \Gamma_u \times (0, T), \quad (5)$$

$$C = \hat{C} \quad \text{on } \Gamma_c \times (0, T), \quad (6)$$

$$\sum_{j=1}^3 \sigma_{ij} n_j = 0 \quad \text{on } (\partial\Omega - \Gamma_u) \times (0, T), \quad (7)$$

$$a \frac{\partial C}{\partial \mathbf{n}} = 0 \quad \text{on } (\partial\Omega - \Gamma_c) \times (0, T), \quad (8)$$

$$\mathbf{u} = \mathbf{u}^0, C = C^0 \quad \text{in } \Omega \quad \text{at } t = 0, \quad (9)$$

where  $T$  is the total time [ $\text{s}$ ],  $\mathbf{u}^0$  is the initial velocity [ $\text{m/s}$ ],  $C^0$  is the initial concentration [mass%],  $\hat{\mathbf{u}}$  is the boundary velocity [ $\text{m/s}$ ], and  $\hat{C}$  is the boundary concentration [mass%].

As the weak form, the following system is considered:

$$\left( \frac{\partial \mathbf{u}}{\partial t}, \mathbf{v} \right) + ((\mathbf{u} \cdot \nabla) \mathbf{u}, \mathbf{v}) + (2\nu D(\mathbf{u}), D(\mathbf{v})) - (p, \nabla \cdot \mathbf{v}) + (\beta C \mathbf{g}, \mathbf{v}) = 0 \quad \text{for } \mathbf{v} \in \mathbf{V}, \quad (10)$$

$$-(\nabla \cdot \mathbf{u}, q) = 0 \quad \text{for } q \in Q, \quad (11)$$

$$\left( \frac{\partial C}{\partial t}, \theta \right) + (\mathbf{u} \cdot \nabla C, \theta) + (a \nabla C, \nabla \theta) = (S, \theta) \quad \text{for } \theta \in \Theta. \quad (12)$$

Here,  $\mathbf{L}^2(\Omega)$  denotes the space of square summable functions in  $\Omega$ , and  $\mathbf{H}^1(\Omega)$  is the space of functions in  $\mathbf{L}^2(\Omega)$  with derivatives up to the first order.

$$\mathbf{V}(\mathbf{g}_1) \equiv \left\{ \mathbf{v} \in (\mathbf{H}^1(\Omega))^3; \mathbf{v} = \mathbf{g}_1 \text{ on } \Gamma_u \right\}, \mathbf{V} \equiv \mathbf{V}(0), \quad (13)$$

$$Q \equiv \{q \in \mathbf{L}^2(\Omega)\}, \quad (14)$$

$$\Theta(\mathbf{g}_2) \equiv \left\{ \theta \in \mathbf{H}^1(\Omega); \theta = \mathbf{g}_2 \text{ on } \Gamma_c \right\}, \Theta \equiv \Theta(0), \quad (15)$$

and  $(\cdot, \cdot)$  denotes the  $\mathbf{L}^2$ -inner product over  $\Omega$ .

## 2.1 Finite Element Approximation

Let us consider approximations of above formulations. The finite element method [2d] is used for discretization of space, and the backward Euler method is used for discretization of time. Linear tetrahedral elements are used for approximations of velocity, pressure and concentration. The approximation scheme without stabilization terms is first explained as follows:

$$\begin{aligned}
& \left( \frac{\mathbf{u}_h^{(n+1)} - \mathbf{u}_h^{(n)}}{\Delta t}, \mathbf{v}_h \right) + \left( (\mathbf{u}_h^{(n)} \cdot \nabla) \mathbf{u}_h^{(n+1)}, \mathbf{v}_h \right) + (2\nu D(\mathbf{u}_h^{(n+1)}), D(\mathbf{v}_h)) - (p_h^{(n+1)}, \nabla \cdot \mathbf{v}_h) \\
& + (\beta C_h^{(n+1)} \mathbf{g}, \mathbf{v}_h) - (\nabla \cdot \mathbf{u}_h^{(n+1)}, q_h) + \left( \frac{C_h^{(n+1)} - C_h^{(n)}}{\Delta t}, \theta_h \right) + (\mathbf{u}_h^{(n)} \cdot \nabla C_h^{(n+1)}, \theta_h) \\
& + (a \nabla C_h^{(n+1)}, \nabla \theta_h) = (S, \theta_h)
\end{aligned} \tag{16}$$

$$\text{for } \mathbf{v}_h \in \mathbf{V}_h, q_h \in Q_h, \theta_h \in \Theta_h,$$

where  $\Delta t$  denotes a time increment.  $\mathbf{u}_h^{(n)}$ ,  $p_h^{(n)}$  and  $C_h^{(n)}$  denote the finite element approximations of  $\mathbf{u}$ ,  $p$  and  $C$  at time  $n\Delta t$ , respectively.  $\mathbf{V}_h$ ,  $Q_h$  and  $\Theta_h$  are finite element approximation spaces of  $\mathbf{V}$ ,  $Q$  and  $\Theta$ , respectively.

## 2.2 A Stabilization Method

Now, our computational scheme with stabilization terms [2c] is introduced as follows:

$$\begin{aligned}
& \left( \frac{\mathbf{u}_h^{(n+1)} - \mathbf{u}_h^{(n)}}{\Delta t}, \mathbf{v}_h \right) + \left( (\mathbf{u}_h^{(n)} \cdot \nabla) \mathbf{u}_h^{(n+1)}, \mathbf{v}_h \right) + (2\nu D(\mathbf{u}_h^{(n+1)}), D(\mathbf{v}_h)) \\
& - (p_h^{(n+1)}, \nabla \cdot \mathbf{v}_h) + (\beta C_h^{(n+1)} \mathbf{g}, \mathbf{v}_h) + \sum_{K \in \mathfrak{S}_h} \tau_K^{\text{NS}} \left( \begin{array}{c} \frac{\mathbf{u}_h^{(n+1)} - \mathbf{u}_h^{(n)}}{\Delta t} + (\mathbf{u}_h^{(n)} \cdot \nabla) \mathbf{u}_h^{(n+1)} \\ + \nabla p_h^{(n+1)} + \beta C_h^{(n+1)} \mathbf{g}, \\ (\mathbf{u}_h^{(n)} \cdot \nabla) \mathbf{v}_h - \nabla q_h \end{array} \right)_K \\
& - (\nabla \cdot \mathbf{u}_h^{(n+1)}, q_h) + \sum_{K \in \mathfrak{S}_h} \tau_K^{\text{CO}} (\nabla \cdot \mathbf{u}_h^{(n+1)}, \nabla \cdot \mathbf{v}_h)_K
\end{aligned}$$

$$\begin{aligned}
& + \left( \frac{C_h^{(n+1)} - C_h^{(n)}}{\Delta t}, \theta_h \right) + (\mathbf{u}_h^{(n)} \cdot \nabla C_h^{(n+1)}, \theta_h) + (a \nabla C_h^{(n+1)}, \nabla \theta_h) \\
& + \sum_{K \in \mathfrak{T}_h} \tau_K^{\text{AD}} \left( \frac{C_h^{(n+1)} - C_h^{(n)}}{\Delta t} + \mathbf{u}_h^{(n)} \cdot \nabla C_h^{(n+1)}, \mathbf{u}_h^{(n)} \cdot \nabla \theta_h \right)_K \\
& = (\mathbf{S}, \theta_h) + \sum_{K \in \mathfrak{T}_h} \tau_K^{\text{AD}} (\mathbf{S}, \mathbf{u}_h^{(n)} \cdot \nabla \theta_h)_K
\end{aligned} \tag{17}$$

$$\text{for } \mathbf{v}_h \in \mathbf{V}_h, \quad q_h \in Q_h, \quad \theta_h \in \Theta_h,$$

where  $(\cdot, \cdot)_K$  denotes the  $L^2$ -inner product over each tetrahedral element  $K$ . The stabilization parameters  $\tau_K^{\text{NS}}$ ,  $\tau_K^{\text{CO}}$  and  $\tau_K^{\text{AD}}$  are defined by

$$\tau_K^{\text{NS}} \equiv \min \left\{ \frac{\Delta t}{2}, \frac{h_K}{2|\mathbf{u}_h^{(n)}|_\infty}, \frac{h_K^2}{24\nu} \right\}, \tag{18}$$

$$\tau_K^{\text{CO}} \equiv \min \left\{ \frac{\lambda |\mathbf{u}_h^{(n)}|_\infty^2 h_K^2}{12\nu}, \lambda |\mathbf{u}_h^{(n)}|_\infty h_K \right\}, \tag{19}$$

$$\tau_K^{\text{AD}} \equiv \min \left\{ \frac{\Delta t}{2}, \frac{h_K}{2|\mathbf{u}_h^{(n)}|_\infty}, \frac{h_K^2}{12a} \right\}, \tag{20}$$

where the constant  $\lambda$  is set to be 1,  $|\mathbf{u}|_\infty$  denotes the maximum norm of  $\mathbf{u}$  in  $K$ , and  $h_K$  denotes the diameter of  $K$ . The symbol  $\mathfrak{T}_h$  denotes the finite element division of  $\Omega$ .

### 2.3 Domain Decomposition Method

Let  $\mathbf{K}\mathbf{x} = \mathbf{f}$  be the finite element system derived from (17), where  $\mathbf{K}$  denotes the regular, asymmetric coefficient matrix corresponding to (17),  $\mathbf{x}$  the vector corresponding to the velocity, pressure, and the concentration at the  $(n+1)$ th step, and  $\mathbf{f}$  the right hand side vector. Let  $\Omega$  be divided into subdomains. Let  $\mathbf{x}_i$  be the vector corresponding to the degrees of freedom in the interior subdomains,  $\mathbf{x}_b$  the vector on the interface between subdomains, and  $\mathbf{x}_t$  a given vector. Then, the system  $\mathbf{K}\mathbf{x} = \mathbf{f}$  can be rewritten as follows:

$$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ib} & \mathbf{K}_{it} \\ \mathbf{K}_{bi} & \mathbf{K}_{bb} & \mathbf{K}_{bt} \\ 0 & 0 & \mathbf{E} \end{bmatrix} \begin{Bmatrix} \mathbf{x}_i \\ \mathbf{x}_b \\ \mathbf{x}_t \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_i \\ \mathbf{f}_b \\ \mathbf{f}_t \end{Bmatrix}, \quad (21)$$

where  $\mathbf{E}$  is an identity matrix. Eliminating  $\mathbf{x}_i$  from (21), we get the linear system on the interface:

$$\mathbf{S}\mathbf{x}_b = \boldsymbol{\chi}, \quad (22)$$

where

$$\mathbf{S} \equiv \mathbf{K}_{bb} - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib},$$

$$\boldsymbol{\chi} \equiv \mathbf{f}_b - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{f}_i - (\mathbf{K}_{bt} - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{K}_{it})\mathbf{x}_t .$$

GPBiCG [5][5a] is applied to (22), and  $\mathbf{x}_b$  is obtained. In the implementation, the matrix  $\mathbf{S}$  is not constructed explicitly. The products of matrices and vectors appearing in GPBiCG can be replaced by solving the basic equations in each subdomain, which implies that the method is fit for parallel computing; see, for example, [2][6][6a]. The application of the skyline method [7] to a problem in each subdomain yields  $\mathbf{x}_i$  from  $\mathbf{x}_b$ . The solution in the whole domain at the (n+1)th step of the scheme is then obtained. In the actual parallel computing, we adopt HDDM [8] for data and processor management to have the workload balanced among processors. It has already been shown that HDDM is effective for a structural problem where the number of degrees of freedom (DOF) is 100 million [9].

### 3. Modeling Aspects

The model considered in this work is shown schematically in Figure 1, used by several researchers [4][10][11][12] to assess the risk of an accident due to a hydrogen leak into a partially open space. Called the hallway model, it may be taken, for example, to represent a garage with a hydrogen powered vehicle parked within. Hydrogen enters from an inlet at the base near one end of the model; ventilation is through a roof vent and a door vent near the opposite end. Four sensors are placed within the model, their locations defined as per the table embedded in Figure 1. It should be noted that the sensor positions of both the experimental model and the numerical model do not exactly match these coordinates. In the case of the experimental model, this was due to the sensors being of finite size. For the numerical model, nodal values at the nodes closest to the defined sensor positions are used.



**Fig.1** The ventilation model.

### 3.1 Boundary and Initial Conditions

The boundary conditions are as follows [4].  $\Gamma_{\text{inlet}}$ ,  $\Gamma_{\text{roof}}$  and  $\Gamma_{\text{door}}$  denote the boundary of the hydrogen inlet, the boundary of the roof vent, and the boundary of the door vent, respectively.

At the inlet, hydrogen leaks in the vertical direction. The velocity and the concentration are as follows:

$$\begin{cases} u_1 = u_2 = 0 & [\text{m/s}] \\ u_3 = 0.02 & [\text{m/s}] \\ C = 6.94 & [\text{mass \%}] \end{cases} \quad \text{on } \Gamma_{\text{inlet}}. \quad (23)$$

The mass concentration is set to be 6.94 [mass%] by considering the density difference between air and hydrogen. The Reynolds number at the inlet is below 60.

Boundary conditions at the vents are as follows. At the roof vent, hydrogen is discharged outside freely. However, at the door vent, air enters from the boundary because conditions for inflow appear in the simulation.

$$\begin{cases} \sum_{j=1}^3 \sigma_{ij} n_j = 0 & [\text{m}^2/\text{s}^2] \\ a \frac{\partial C}{\partial \mathbf{n}} = 0 & [\text{m/s}] \end{cases} \quad \text{on } \Gamma_{\text{roof}}, \quad (24)$$

$$\begin{cases} \sum_{j=1}^3 \sigma_{ij} n_j = 0 & [\text{m}^2/\text{s}^2] \\ C = 0 & [\text{mass \%}] \end{cases} \quad \text{on } \Gamma_{\text{door}}, \quad (25)$$

where  $\sigma(\mathbf{u}, p)$  is the stress tensor normalized by the density [ $\text{m}^2/\text{s}^2$ ] defined by

$$\sigma_{ij} = -p\delta_{ij} + 2\nu D_{ij}(\mathbf{u}) \quad i, j = 1, 2, 3, \quad (26)$$

with  $\delta_{ij}$  being the Kronecker delta and  $\mathbf{n}$  being the unit normal vector.

At the other boundary, there is no inflow of hydrogen:

$$\begin{cases} \mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}_3 = 0 & [\text{m/s}] \\ a \frac{\partial C}{\partial \mathbf{n}} = 0 & [\text{m/s}] \end{cases} \quad \text{on } \partial\Omega - (\Gamma_{\text{inlet}} + \Gamma_{\text{roof}} + \Gamma_{\text{door}}), \quad (27)$$

The initial conditions are as follows:

$$\begin{cases} \mathbf{u}_1 = \mathbf{u}_2 = \mathbf{u}_3 = 0 & [\text{m/s}] \\ C = 0 & [\text{mass \%}] \end{cases} \quad \text{in } \Omega. \quad (28)$$

### 3.2 Numerical Parameters

The parameters involved and their respective values are given in **Table 1** [4].

**Table 1** Parameters.

### 3.3 Computational Aspects

For completeness, we first compute the hallway model with 90,536 (84,355), 1,155,450 (991,950) and 11,540,027 (9,473,785) elements (DOF), all with the timestep set at 1.0 [s] for a total of 200 [s]. Then, to further demonstrate large scale computing with sFlow, we run the largest model (11,540,027) at timesteps 0.1 [s] and 0.5 [s] for 200 [s]. Computation time for the largest model was 1,324, 245 and 125 hours for timesteps 0.1, 0.5, and 1.0 [s], respectively. The largest model was decomposed into 20 domains, and each domain was further broken down into 578 subdomains. Each subdomain had roughly 1,000 elements. The number of interface DOF was 4,435,925. The relative residual was set at  $1.0 \times 10^{-6}$  for the GPBiCG convergence criterion. The computation was performed on 20 CPUs in parallel, each with a Pentium 4 (3.2 GHz) processor.

Previous work involving the Code for large scale modeling has been published before e.g. Kanayama et al. [2] analyzed a stationary flow problem with 10 million DOF. However, we believe this to be the first very large scale computation of a problem that represents both non-stationary flow as well as gas dispersion using the Code.

## 4. Results

Figures 2a-d compare the hydrogen volumetric concentration values at the nodes closest to the four sensor positions, for the three computational models of varying mesh densities against experimental values [3]. Figures 3a-d compare, by sensor position, the concentration values of the largest model run at different timesteps against experimental values. In all cases, the hydrogen concentration is measured in terms of volumetric concentration [%], where 0 indicates that the entire volume is occupied by air and 1 [100%] represents the opposite. We note that the experimental results are to be taken merely as an indicator of what to expect in terms of the magnitude of concentration and general dispersion behaviour of hydrogen.

Comparing the effects of mesh refinement, we see that a progressively finer mesh results in less oscillatory concentration values. However, significant variations in the values can be seen, particularly that for sensors 2 and 3. Sensor 2, for example, has the following concentration values at  $t = 170$  [s]: 5% (experimental), 3% (90,536 elements), 4% (1,155,450), and 4% (11,540,027). Due to the random, transient nature of real gas dispersion, we do not draw any conclusions as to whether the computational results have increased in accuracy as the mesh is refined, regardless of whether or not the results approach the experimental values. Nonetheless, we note that in general, the magnitude of concentration and therefore the general dispersion behaviour of hydrogen is similar to that suggested by the experimental model.

Comparing the effects of changing the timestep of the largest model, we note again that significant variations can be seen for sensors 2 and 3. A possible explanation for this may be given by observing that sensor 2 is located in the region above the hydrogen inlet, where steep concentration gradients may be expected for small changes in sensor position. The same explanation may be used for sensor 3, which is located near the roof vent. This point was also noted by Matsuura [12] and may be seen in Figure 9 of [4]. In addition, and as noted previously, the sensor positions for the experimental model and the nodal positions of the computational model do not necessarily overlap, nor do they match exactly with the predefined sensor coordinates. The 0.5 [s] timestep gives results closest to the experimental results; this suggests that the ideal timestep for this mesh size might lie in this range.

**Fig.2a** Volumetric Concentration vs Time – Experimental Data [3].

**Fig.2b** Volumetric Concentration vs Time – 90,536 Elements, Timestep 1.0 s.

- Fig.2c** Volumetric Concentration vs Time - 1,155,450 Elements, Timestep 1.0 s.
- Fig.2d** Volumetric Concentration vs Time - 11,540,027 Elements, Timestep 1.0 s.
- Fig.3a** Volumetric Concentration vs Time – Sensor 1, 11,540,027 Elements.
- Fig.3b** Volumetric Concentration vs Time – Sensor 2, 11,540,027 Elements.
- Fig.3c** Volumetric Concentration vs Time – Sensor 3, 11,540,027 Elements.
- Fig.3d** Volumetric Concentration vs Time – Sensor 4, 11,540,027 Elements.

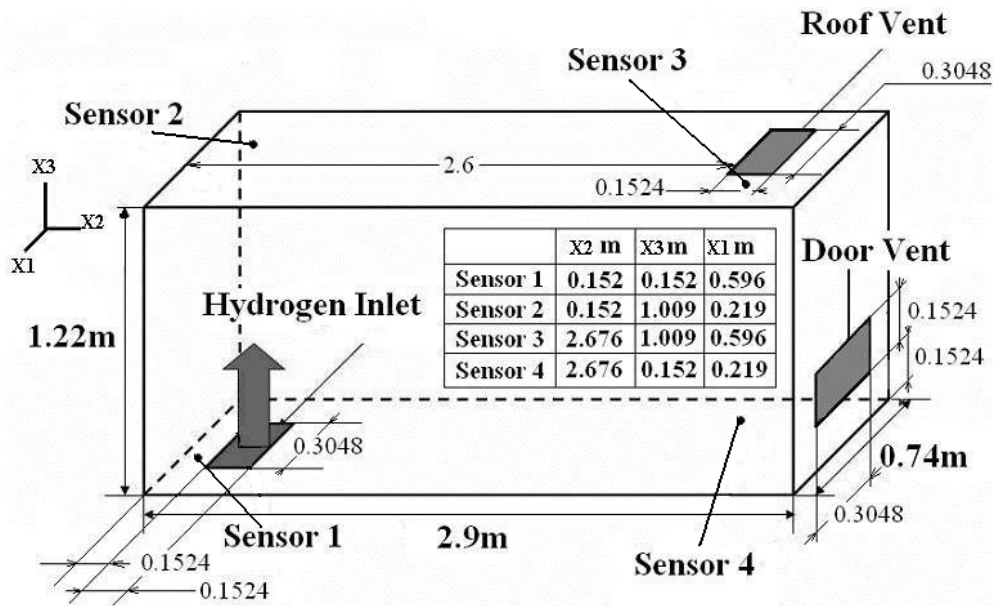
## 5. Conclusion

We have reported the use of the hierarchical domain decomposition method in simulating a large scale model of hydrogen dispersion in a partially open space. We have used an open source freeware, called ADVENTURE\_sFlow, and applied an analogy of the Boussinesq approximation in our analysis. After taking possible sources of error into account, we conclude that our results follow the general trend shown in the experimental data. This indicates that our code, for which the domain decomposition method is a standard feature, can give a realistic representation of real-world problems modeled in large scale.

## REFERENCES

1. ADVENTURE Project, <http://adventure.sys.t.u-tokyo.ac.jp/>
2. Kanayama, H., Tagami, D., Chiba, M.: Stationary incompressible viscous flow analysis by a domain decomposition method. *Decompos. Methods Sci. Eng.* XVI, 611–618 (2006)
- 2a. Kanayama, H., Ogino, M., Takesue, N., Mukaddes, A.M.M: Finite element analysis for stationary incompressible viscous flow analysis by a domain decomposition method. *Theor. Appl. Mech.*, Vol 54, 211-219 (2005)
- 2b. Kanayama, H., Kume, H., Tagami, D.: Incompressible viscous flow analysis by a domain decomposition method. *4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS2004)*, Volume II, 1-12 (2004)
- 2c. Kanayama, H., Tagami, D., Araki, T., Kume, H.: A stabilization technique for stationary flow problems. *Int. J. Comput. Fluid Dyn.*, Vol.18, No.4, 297-301 (2004)
- 2d. Girault, V., Raviart, P.A.: *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*. Springer 1986.
3. Inoue, M., Tsukikawa, H., Kanayama H., Matsuura, K.: Experimental study on leaking hydrogen dispersion in a partially open space. *J. Hydrog. Energy Syst. Soc. Jpn*, Vol.33, No.4, 32-43 (2008) in Japanese.
4. Kanayama, H., Tsukikawa, H., Ndong-Mefane, S. B., Sakuragi, O.: Finite element simulation of hydrogen dispersion by the analogy of the Boussinesq approximation. *J. Comput. Sci. Tech.*, Vol.2, No.4, 643-654 (2008)
5. Zhang, S.L.: GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems. *SIAM J.Sci.Comput.* 18, 537-551 (1997)

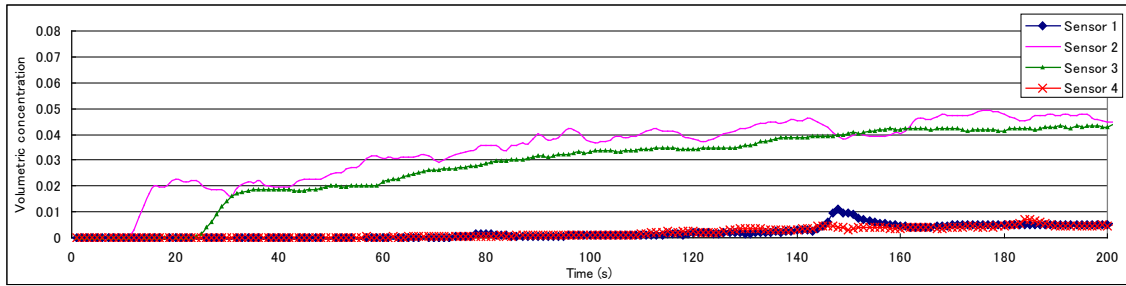
- 5a. van der Vorst, H.A.: Iterative Krylov Methods for Large Linear Systems. Cambridge University Press, United Kingdom 2003.
6. Glowinski, R., Dinh, Q.V., Periaux, J.: Domain decomposition methods for nonlinear problems in fluid dynamics. *Comput. Methods Appl. Mech. Eng.*, 40, 27–109 (1983)
- 6a. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Oxford University Press, USA 1999.
7. Hasbani, Y., Engelman, M.: Out-of-core solution of linear equations with non-symmetric coefficient matrix. *Comput. Fluid* 7, 13-31 (1979).
8. Yagawa, G., Shioya, R.: Parallel finite elements on a massively parallel computer with domain decomposition. *Comput. Syst. Eng.*, 4, 495–503 (1993)
9. Shioya, R., Yagawa, G.: Iterative domain decomposition FEM with preconditioning technique for large scale problem. ECM'99, Progress in Experimental and Computational Mechanics in Engineering and Material Behaviour, 255–260 (1999)
10. Agarant, V., Cheng, Z., Tchouvelev, A.: CFD modeling of hydrogen releases and dispersion in hydrogen energy station. Proceedings of The 15<sup>th</sup> World Hydrogen Energy Conference, (2004)
11. Swain, M.R., Grilliot, E.S., Swain, M.N.: Risks incurred by hydrogen escaping from containers and conduits. Proceedings of the 1998 US DOE Hydrogen Program Review, NREL/CP-570-25315, 1998.
12. Matsuura, K., Kanayama, H., Tsukikawa, H., Inoue, M.: Numerical simulation of leaking hydrogen dispersion behavior in a partially open space. *Int. J. Hydrog. Energy* 33, 240-247 (2008)



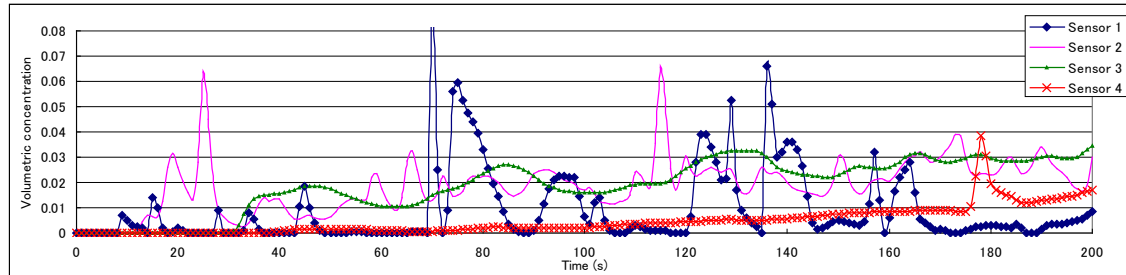
**Fig.1** The ventilation model.

Parameters	Values
Kinematic viscosity, $\nu$	$1.05 \times 10^{-4}$ [m <sup>2</sup> /s]
Diffusion coefficient in air, $a$	$6.1 \times 10^{-5}$ [m <sup>2</sup> /s]
Coefficient, $\beta$	13.4 [-]
Gravity, $g$	(0,0, -9.8) [m/s <sup>2</sup> ]
Source term, $S$	0 [1/s]

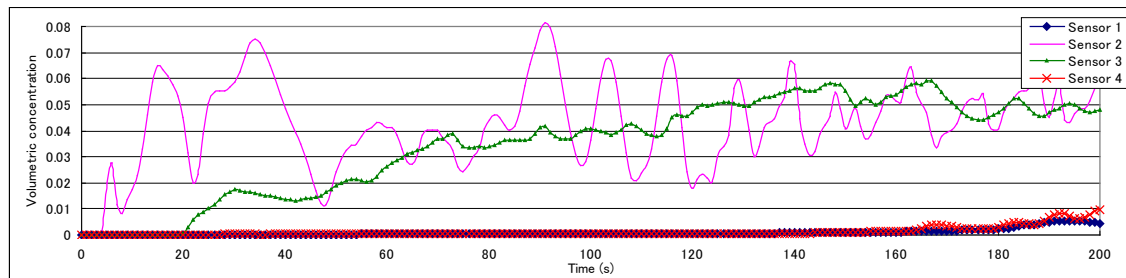
**Table 1** Parameters.



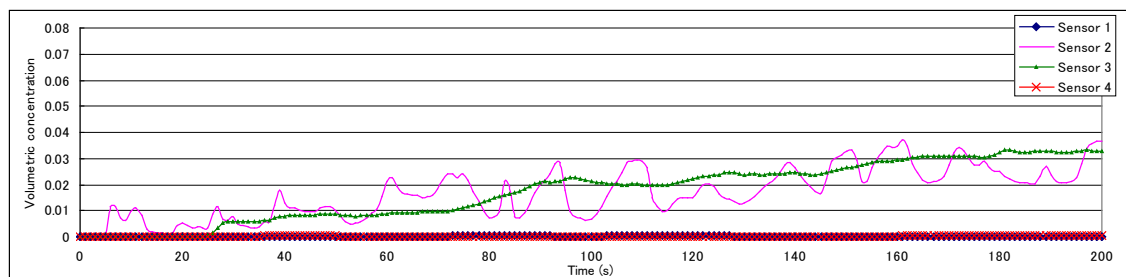
**Fig.2a** Volumetric Concentration vs Time – Experimental Data [3].



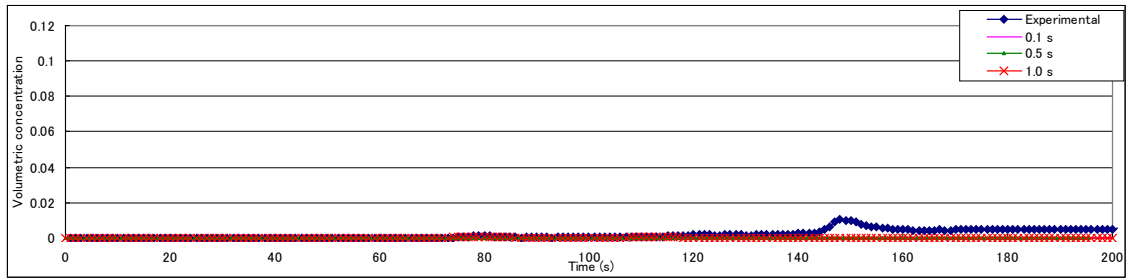
**Fig.2b** Volumetric Concentration vs Time – 90,536 Elements, Timestep 1.0 s.



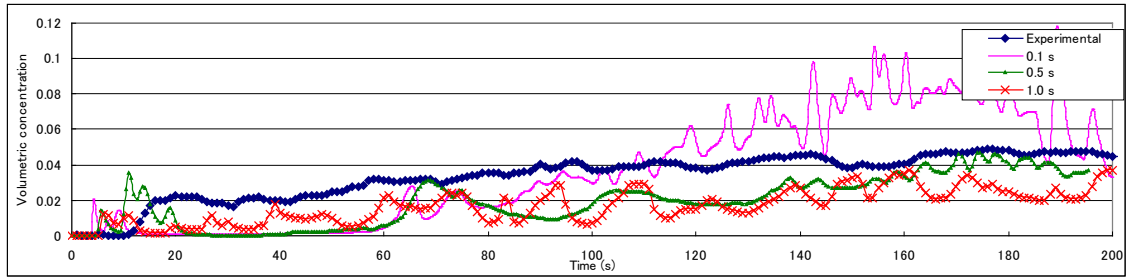
**Fig.2c** Volumetric Concentration vs Time - 1,155,450 Elements, Timestep 1.0 s.



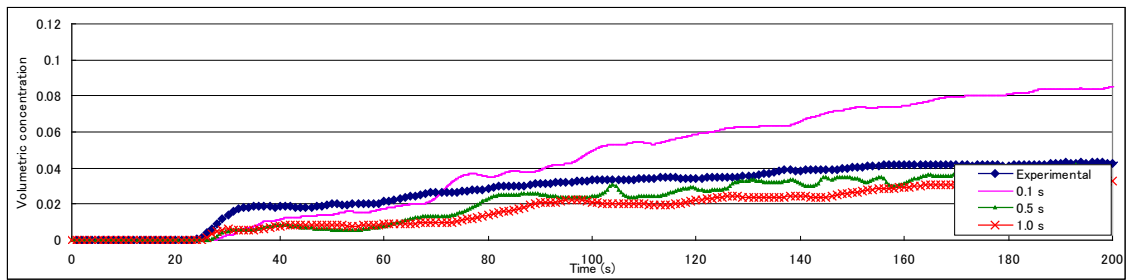
**Fig.2d** Volumetric Concentration vs Time - 11,540,027 Elements, Timestep 1.0 s.



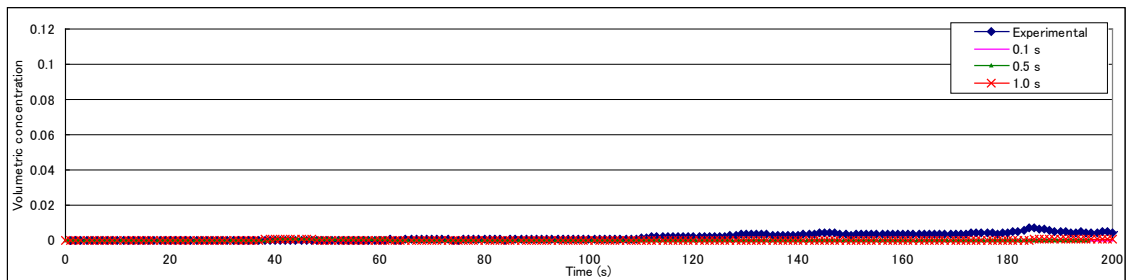
**Fig.3a** Volumetric Concentration vs Time – Sensor 1, 11,540,027 Elements.



**Fig.3b** Volumetric Concentration vs Time – Sensor 2, 11,540,027 Elements.



**Fig.3c** Volumetric Concentration vs Time – Sensor 3, 11,540,027 Elements.



**Fig.3d** Volumetric Concentration vs Time – Sensor 4, 11,540,027 Elements.