

For office use only
T1 _____
T2 _____
T3 _____
T4 _____

Team Control Number

73052

Problem Chosen

A

For office use only
F1 _____
F2 _____
F3 _____
F4 _____

2018
MCM/ICM
Summary Sheet

Optimization to Tackling with Formaldehyde Hazard

Summary

The history of formaldehyde dates back to 1859 when Aleksandr Butlerov first reported it, yet its long term hazard wasn't evident until 1980s. In recent years, the widely use of formaldehyde in chemical industry has made the health problem caused by formaldehyde even worse. There are lots of circumstances when people have to live in recent decorated apartments. In this case, we are modeling to find out how the factors influence the emission of formaldehyde in a house. Then, we offer some strategies to optimize the process of lowering down the density of formaldehyde in a room.

First of all, we discuss the problem in a microscopic way and create two models to simulate the behavior of gas particles. In our Lattice Gas Automata Model, we treat every formaldehyde molecular as lattice gas and solve the problem using Cellular Automata. While in the Molecular Dynamics Model, we use classical Newton mechanics to simulate the diffusion of formaldehyde, with a simple sphere to represent the coarse-grained model of formaldehyde molecular.

Secondly, we introduced a macroscopic model - Continuous Fluid Model, which is based on Fluid Dynamics. By simplify the problem into a partial differential equation, we tackle it with Finite Element Method as well as visualize results using COMSOL.

Thirdly, we compare the three models evaluating their accuracy and conciseness to find out the optimal model to this problem.

In the end, we analyze the best strategy to reduce the concentration of formaldehyde to a healthy threshold. Two steps are suggested: exhausting fans can help to reduce it quickly in the first few weeks and then open up windows as often as possible. Temperature and humidity should be controlled under a certain level.

Throughout our article, the common small family house has been examined. Using COMSOL, we do build up a room with a window, inner wall, furniture and a fan. By changing its temperature and volume and other parameters we find the general strategy.

Keywords: Lattice gas automaton; Molecular dynamics; Finite element method

Optimization to Tackling with Formaldehyde Hazard

February 2, 2018

Contents

1	Introduction	3
1.1	Problem Statement	3
1.2	Our Work	3
2	Assumptions and Notations	4
2.1	Assumptions	4
2.2	Notations	5
3	Model Construction and Implementation	5
3.1	Theory of Diffusion	5
3.2	Lattice Gas Automata(LGA) Model	6
3.3	Molecular Dynamics(MD) Model	8
3.4	Continuous Fluid(CF) Model	9
3.5	Model Analysis and Selection	10
4	Strategies	11
4.1	Opening Windows	11
4.2	Exhausting Fans	13
4.3	Temperature and Humidity	13
4.4	Other Strategies	13
4.5	Final Evaluation	14
5	Model Analysis	14
5.1	Sensitivity Analysis	14
5.1.1	Impact of Volume	14
5.1.2	Impact of Temperature	15
5.1.3	Impact of Additional Emission Sources	16

5.1.4	Impact of Inner Structure	17
5.2	Strengths and Weakness	18
5.2.1	Strengths	18
5.2.2	Weakness	18
6	Conclusions	19
Letter		21
Appendices		22
Appendix A	Implemented LGA Model	22

1 Introduction

1.1 Problem Statement

The formaldehyde gas released by newly painted walls and recently-bought furniture has caused a lot damage to people. Scientist have discovered that formaldehyde has great damage to DNA and it's repair[1]. On December 12, 2016, EPA published in the Federal Register a final rule[2] to reduce exposure to formaldehyde emissions from certain wood products produced domestically or imported into the United States. However, the emission of formaldehyde in the decorating process is inevitable, what we can do is to wait them volatile and decrease.

We need to develop a model that describes the change of the concentration of formaldehyde over time in the space of the room, then determine the best strategy to reduce the concentration.



Figure 1: Houses that are decorated

1.2 Our Work

We discuss this problem from two perspectives.

From the microscopic point of view, all the properties that we are interested can be derived from the behavior of the mechanics of gas particles, so we discuss two models. On the one hand, we can treat every formaldehyde molecular as lattice gas, this is a trivial and simple model, which we call Lattice Gas Automata(LGA) Model. On the other hand, we can directly use classical Newton mechanics to simulate the behavior of formaldehyde gas diffusion, with a simple sphere to represent the coarse-grained model of formaldehyde molecular. Actually, lattice gas perfectly resembles classical gas and should be identical under the thermodynamic limit, which we will find out in later calculations.

Meanwhile, we can construct a Continuous Flow(CF) model based on Fluid Dynamics, without any single particle taken into consideration. Hence the problem transforms into solving a partial differential equation. We tackle this with Finite Element Method and state-of-the-art simulation software COMSOL.

After discussion from two perspective, we evaluate them carefully and pick out the best one to carry on our work. We then apply different strategies including opening

windows, exhausting fans, temperature and humidity control and so on, among them the best compound strategy will be given. We will make full assessment of our model and our final composite strategy. In the end, we'll provide model analysis, giving the strengths and weakness of our model and conclusions.

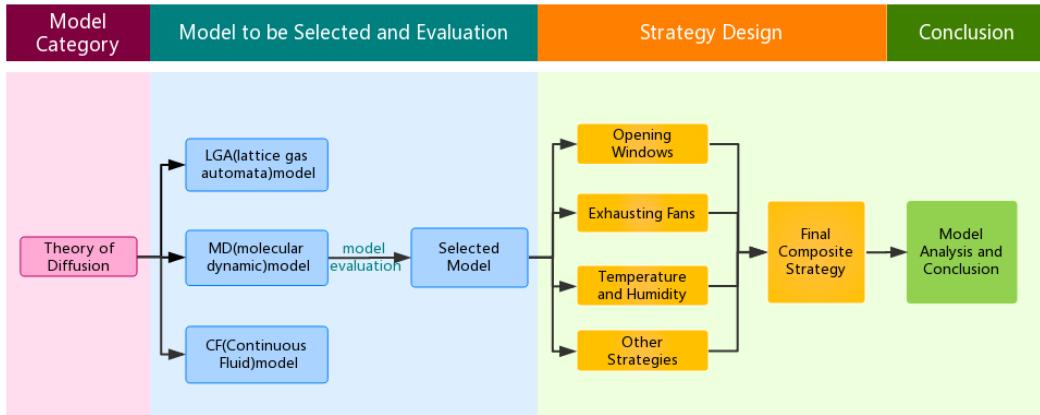


Figure 2: Framework of Our Work

2 Assumptions and Notations

2.1 Assumptions

We derived three models and they make the following assumptions in common.

- Without loss of generality, we consider our room a cube because almost every apartment in reality consists of a few cubes or cuboids. Meanwhile, the discussion on cube room can be easily generalized to cuboid room. The effect of interior walls will be discussed in an individual section.
- The thickness of the wall can be ignored so that the volume of the room equals the capacity of the room.
- In our basic models, rooms are closed all around with its interior containing only formaldehyde gas.
- The formaldehyde gas is uniformly distributed in every wall except the floor and can only be released inward.
- There's no gas exchange between the house and the outside unless the window is open or the exhausting fans are applied, which will be discussed as additional strategies later.

2.2 Notations

Table 1: Notations.

Symbol	Definition	Unit
W	Width of a house	m
L	Length of a house	m
H	Height of a house	m
σ	Formaldehyde gas particles contained in the wall per unit area	m^3
$\Delta\tau$	the time per frame for the simulation of LGA model	s
$\bar{\rho}$	Average density of formaldehyde gas in house	m^{-3}
$\rho_{threshold}$	Threshold density of formaldehyde gas	m^{-3}
J	Diffusion flux	$\text{mol}\cdot\text{m}^{-2}\text{s}^{-1}$
D	Diffusion coefficient	m^2/s
$\varphi(\vec{x}, t)$	Molar concentration in position \vec{x} and time t	$\text{mol}\cdot\text{m}^{-3}$
$t_{threshold}$	Time needed to decline to health threshold	d

3 Model Construction and Implementation

3.1 Theory of Diffusion

Diffusion is the net movement of molecules or atoms from a region of high concentration with high chemical potential to a region of low concentration with low chemical potential. The rate of diffusion, which is quantified by diffusion flux, the rate of flow of amount of substance per unit area is proportional to the concentration gradient. This is described by Fick's First Law as following.

$$J = -D\nabla\varphi \quad (1)$$

This law has told us that the concentration gradient is the driving force of transport phenomena, but what we care most is the evolution of concentration with time. This is given by Fick's Second Law.

First, use mass conservation law

$$\frac{\partial\varphi}{\partial t} + \nabla J = 0 \quad (2)$$

Then Fick's first law is applied,

$$\frac{\partial\varphi}{\partial t} = \nabla(D\nabla\varphi) = D\Delta\varphi \quad (3)$$

What we need to do next is set all the boundary conditions and solve the partial equation above to get distribution function $\varphi(\vec{x}, t)$, which will offer us all the preferred physical quantity such as the minimum time which concentration falls below threshold, the concentration of formaldehyde gas in a place of interest at a given time.

3.2 Lattice Gas Automata(LGA) Model

We start with Lattice Gas Automata model, a rough model which is enough to help us understand the behavior of diffusion of formaldehyde gas. This model treat real gas as lattice gas and makes the following additional assumptions:

1. We consider the room as a lattice, and the formaldehyde gas as a gas unit point regardless of its shape or intrinsic properties.
2. Evolution of the simulation is done in discrete time steps. After each time step, the state at a given site can be determined by the state of the site itself and neighboring sites, before the time step.
3. The state at each site is purely boolean. At a given site, there either is or is not a unit gas moving in each direction.
4. The effect of air is ignored, since compared with formaldehyde, the concentration of air is too large to matter on the behavior of flow of formaldehyde gas. Although the average velocity, escape time can be totally different in vacuum or atmosphere, the parameters of properties of gas unit can be adjusted.
5. The formaldehyde gas generating rate is even over time and space along the wall.

The LGA model was introduced by Hardy, de Passis, and Pomeau in 1976 [3] and is also called the HPP model, a name derived from its inventors. Initially used for the description of the molecular dynamics of a classical lattice gas, the LGA was later used to model large numbers of uniformly interacting particles (cells). In our model, LGA are used to simulate the diffusion of formaldehyde.

LGA employs a regular, finite lattice and includes a finite set of particle states, an interaction neighborhood and local rules which determine the movement of particles (cells) and their transitions between states.[4] LGA differs from traditional CA by incorporating the movement of particles and an exclusion principle. The particles in the model select from a finite number of permissible discrete velocity channels. The velocity specifies the direction and magnitude of movement, which may include zero velocity (rest). In a simple exclusion rule, only one particle may have each allowed velocity at each lattice site.

According to the dimensions, LGA model can be divided into 2 categories:2D and 3D[5]. Every node in the 2D LGA model is associated with five velocity channels, namely stay, left, right, up, and down. Similarly, each node in a 3D LGA model has seven velocity channels associated with it-stay, left, right, up, down, front, and back. For the case of a 2D LGA model, the number of velocities and their possible directions are shown in Fig1.

a	Number of Velocities	Velocities Available
	1	● → ← ↑ ↓
	2	← ← ↑ ↓ ↑ ← → ↓ ↑ ← ↓ ↓
	3	↑ ← → ↓ ↓ ↑ ↗ ↘ ↑ ↑ ← ↓ ↗
	4	↔ ↔ ↗ ↘ ↗ ↘
	5	◆◆◆◆◆

b	Velocity	s	l	r	u	d	Binary format
	●	1	0	0	0	0	10000
	→	0	0	1	0	0	00100
	←	0	1	0	0	0	01000
	↑	0	0	0	1	0	00010
	↓	0	0	0	0	1	00001
	↔	0	1	1	0	0	01100
	↗	0	1	0	1	1	01011
	↖	0	1	1	1	1	01111

Figure 3: (a) The number of velocities available in a node and their possible directions for a 2D LGA. (b) Examples of binary representation of some possible velocities at a node in a 2D LGA

In our model, the density of formaldehyde follows the formula:

$$WLH \frac{d\rho}{dt} = \sigma[2H(W+L) + WL] \quad (4)$$

where σ is not a constant because of the emission of formaldehyde. To simplify the process of simulation of our modeling, we ignore the emission from the ceiling so the density of formaldehyde follows:

$$WLH \frac{d\bar{\rho}}{dt} = 2\sigma H(W + L) \quad (5)$$

as there is no emission from ceiling and floor and in order to give a better intuition, we first run our LGA in 2-dimension scene using C++.

In the simulation, the house are set to be $5(L) \times 3(W) \times 3.5(H)$, so we draw 1500×900 lattices to represent the cross section of the house. In the beginning there is no formaldehyde particles in the air, they are only stored in the walls. The yellow dot in 4 represents the formaldehyde particles, it's clear that formaldehyde cluster near the wall and it's density takes the minimum in the center of the house's cross section.

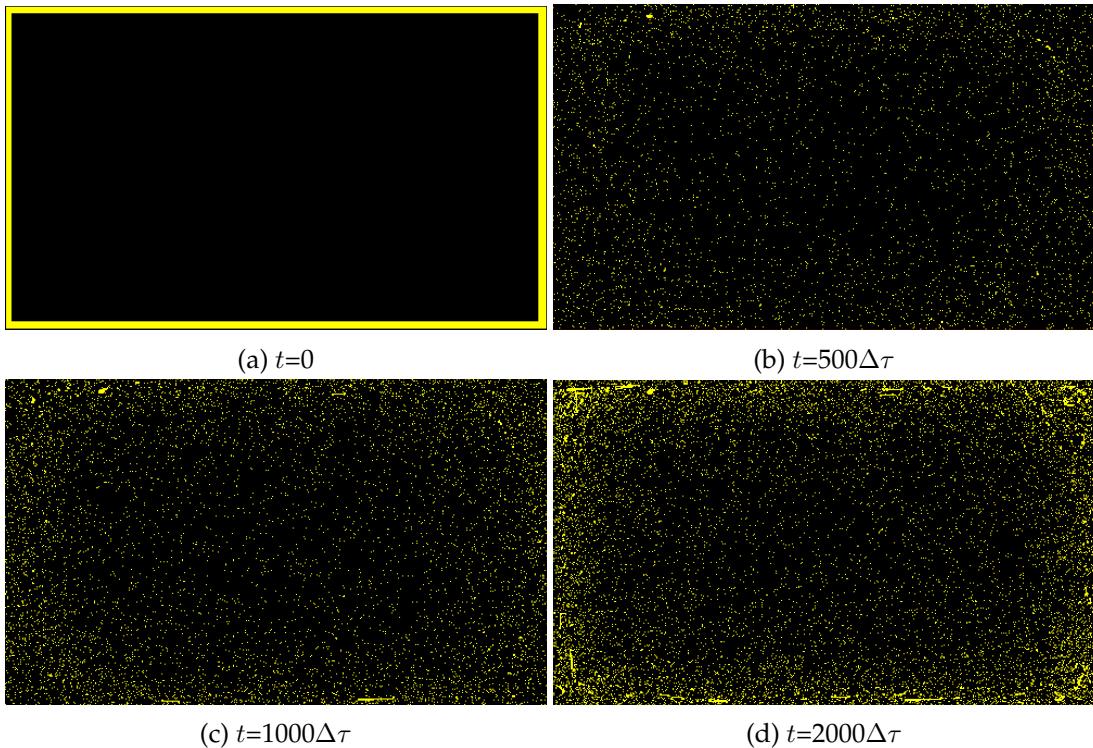


Figure 4: Simulation of the LGA model using C++

3.3 Molecular Dynamics(MD) Model

Despite the rough LGA model mentioned above, we now try to be more specific by simulating the diffusion of particles following the rules of classical mechanics. Therefore, we introduce the additional assumptions as follows:

1. We use a simple sphere to represent a molecule of formaldehyde and take the strategy of coarse-grained modelling. By decreasing the degrees of freedom much longer simulation times can be studied than using classical atomistic models.
2. The interactions between the particles strictly abide by the Newton's second law of mechanics and Momentum conservation law.
3. Since the concentration of the air is at a much higher scale than formaldehyde, the behavior of diffusion of formaldehyde has no much difference to the particles in vacuum except velocity which can be modified by adjusting parameters.

To improve computational accuracy, we tackle this molecular dynamic problem using Verlet integration method [6], which is a numerical method used to integrate Newton's equations of motion. At first, we define: $v(t + \Delta t/2) \equiv (x(t + \Delta t) - x(t))/\Delta t$

$$\text{It is obvious that: } x(t + \Delta t) \approx 2x(t) - x(t - \Delta t) + \frac{F(x_i)}{m} \Delta t$$

$$\text{so we can get } x(t + \Delta t/2) = x(t) + \Delta t v(t + \Delta t/2)$$

Because of this, the equations for updating position and velocity are:

$$x_i = x_{i-1} + v_{i-1/2} \Delta t \quad (6)$$

$$a_i = F(x_i) \quad (7)$$

$$v_{i+1/2} = v_{i-1/2} + a_i \Delta t \quad (8)$$

where x_i is position at step i , $v_{i+1/2}$ is the velocity, or first derivative of x , at step $i + 1/2$, a_i is the acceleration, or second derivative of x , at step i , and Δt is the size of each time step.

Note that the potential for particles should be set individually, we choose use Lennard-Jones potential[7] to do our simulation, which is widely used in molecular dynamics and really resembles the real condition.

3.4 Continuous Fluid(CF) Model

In physics and engineering, fluid dynamics is a sub-disciplines of fluid mechanics that describes the flow of fluids (liquids and gases). Fluid dynamics has a wide range of applications, including calculating forces and moments on aircraft, determining the mass flow rate of petroleum through pipelines, predicting weather patterns and so on. In our model, the flow of formaldehyde can be viewed as continuous fluid rather than discrete particles. We derive Continuous Fluid(CF) model from it, the fluid of formaldehyde follows the conservation laws' equation:

$$\frac{\partial}{\partial t} \iiint_V \rho dV = - \oint_S \rho \mathbf{u} \cdot d\mathbf{S} \quad (9)$$

$$\frac{\partial}{\partial t} \iiint_V \rho \mathbf{u} dV = - \oint_S (\rho \mathbf{u} \cdot d\mathbf{S}) \mathbf{u} - \oint_S p d\mathbf{S} + \iiint_V \rho \mathbf{f}_{\text{body}} dV + \mathbf{F}_{\text{surf}} \quad (10)$$

$$\rho \frac{Dh}{Dt} = \frac{Dp}{Dt} + \nabla \cdot (k \nabla T) + \Phi \quad (11)$$

Based on the conservation laws' equation of fluid and the Fick's first law we have discussed before, we can derive Fick's second law:

$$\frac{\partial}{\partial x} \left(D \frac{\partial}{\partial x} \varphi \right) = D \frac{\partial}{\partial x} \frac{\partial}{\partial x} \varphi = D \frac{\partial^2 \varphi}{\partial x^2} \quad (12)$$

Then we are discussing constants needed and all boundary conditions that are necessary to numerically solve the partial differential equation.

- The diffusion coefficient of formaldehyde $D = 1.8 \times 10^{-5} \text{ m}^2/\text{s}$ in air, according to GSI Database¹.
- The concentration on the surface of the wall are given by the assumption that formaldehyde is uniformly distributed initially, and then evolves according to Fick's law.
- As for windows taken into consideration later, that treating the concentration along the window as 0 is enough to simulate the real-life window.

¹<https://www.gsi-net.com/en/publications/gsi-chemical-database/single/288-CAS-50000.html>

Solving this differential equations, we'll get the solution and the characterization concentration of formaldehyde, namely, φ . As we assume that the air in the house to be the homogeneous isotropic media, finite element method can be applied to the calculation in the house. We divide the house into large quantity of finite element and apply Fick's Law to it. The result of our calculation is visualized through COMSOL software, the results are shown in 5.

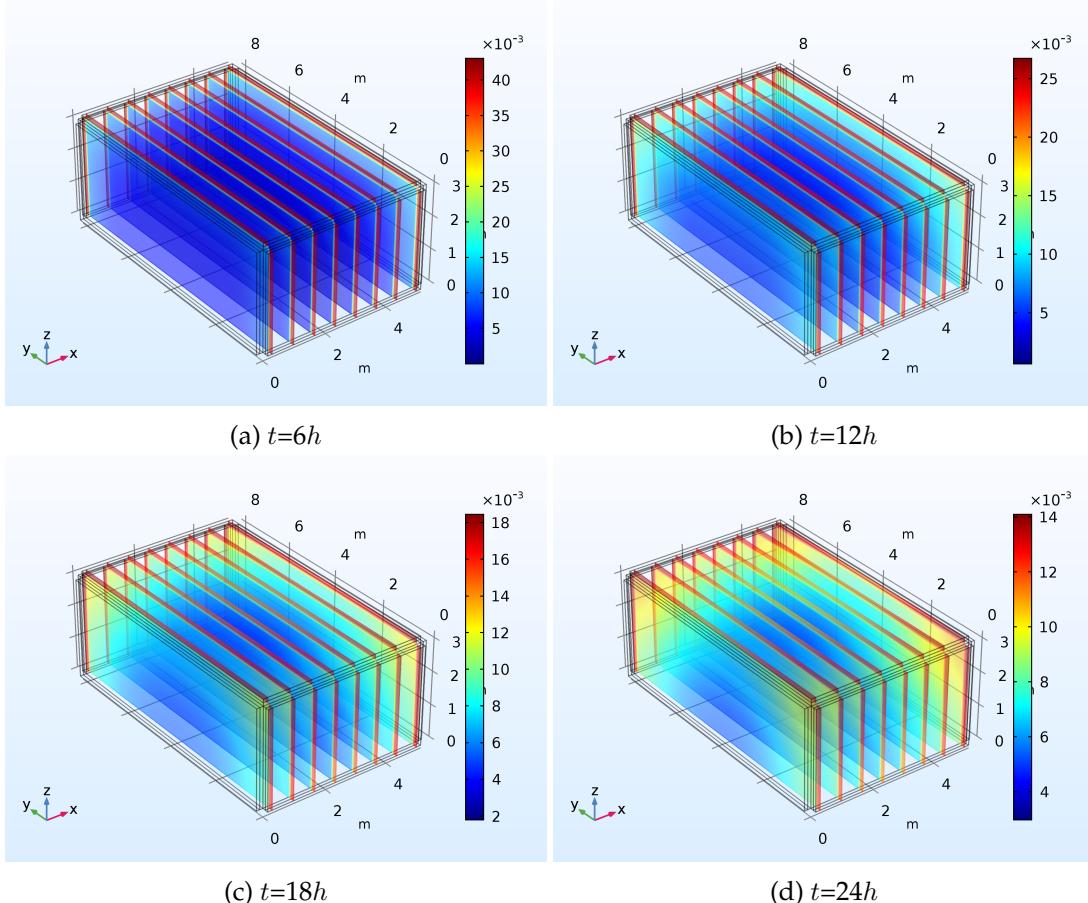


Figure 5: Simulation of the CF model

3.5 Model Analysis and Selection

Among the model we put forward above, the best model must be selected to continue our work to examine the different strategies. We evaluate our model in terms of accuracy and conciseness.

For accuracy, LGA model makes assumption that the direction of particles' motivation and particles' position is limited, thus leading to inaccuracy of simulation. MD model has better accuracy over LGA model, but it is still difficult to simulate continuous presence of the formaldehyde, because the number of particles is limited by the capability of computation. CF model based on the assumption that the distribution of formaldehyde gas is described by the density function and motion described by Fick's law. To tackle with the partial differential equation, we use finite element method and made a state-of-the-art simulation.

For conciseness, LGA model's assumptions leave out too much information of particles motion and oversimplify the rules of collision. MD model and CF model both reflect the actual motion of particles well as MD model behaves better when the density of formaldehyde is low and CF model behaves better when it's high. In this problem, we know the threshold of concentration which do no harm to people is 3.3×10^{-6} mol/m³, which contains still much more number of particles that MD simulation can tackle with.

Taking all these aspects into consideration, we select CF(Continuous Fluid) as our model. Then we'll find and discuss strategies of this model.

4 Strategies

4.1 Opening Windows

In order to simulate the effect of opening windows, it's necessary to assume that there is no wind outside the window and concentration of formaldehyde gas outside window is 0. The formaldehyde gas can only diffuse outside through the window. As for the emission source, all the walls and the ceiling are considered storing a certain amount of formaldehyde inside it, which is obtained by data in real life. As for our house, we choose a 8m(L)×5m(W)×3m(H) cuboid house for convenience, the result are shown in Figure 6

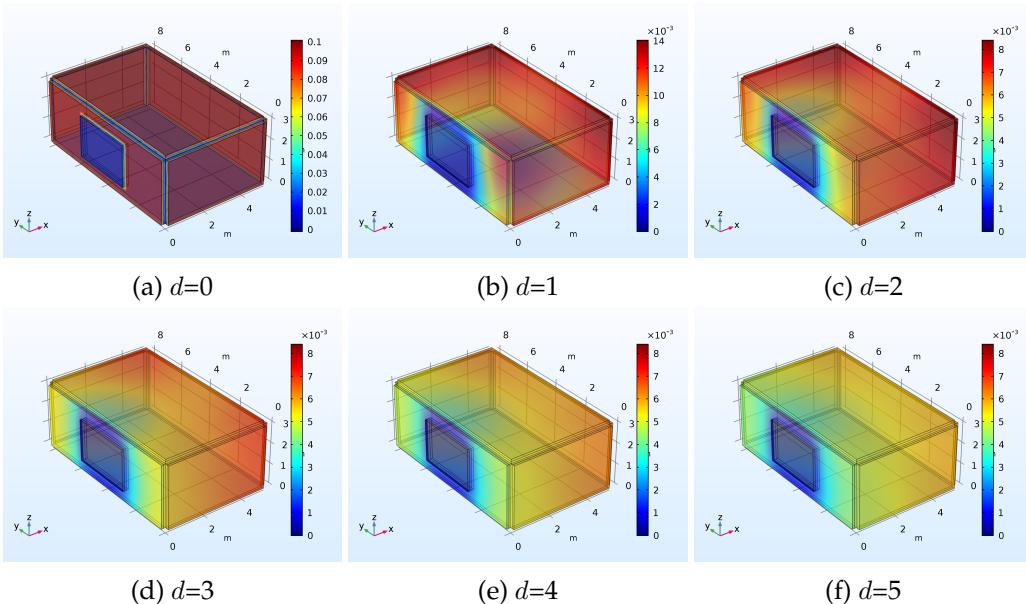


Figure 6: Simulation of the window opening

As can be shown above, windows make a difference to the reducing of the formaldehyde, especially in the first week after decoration. It still works well when the density of the formaldehyde is lower than the health threshold, the change of average density of formaldehyde can be shown in Figure 7 , notice that the second diagram's vertical axis are set to be logarithm.

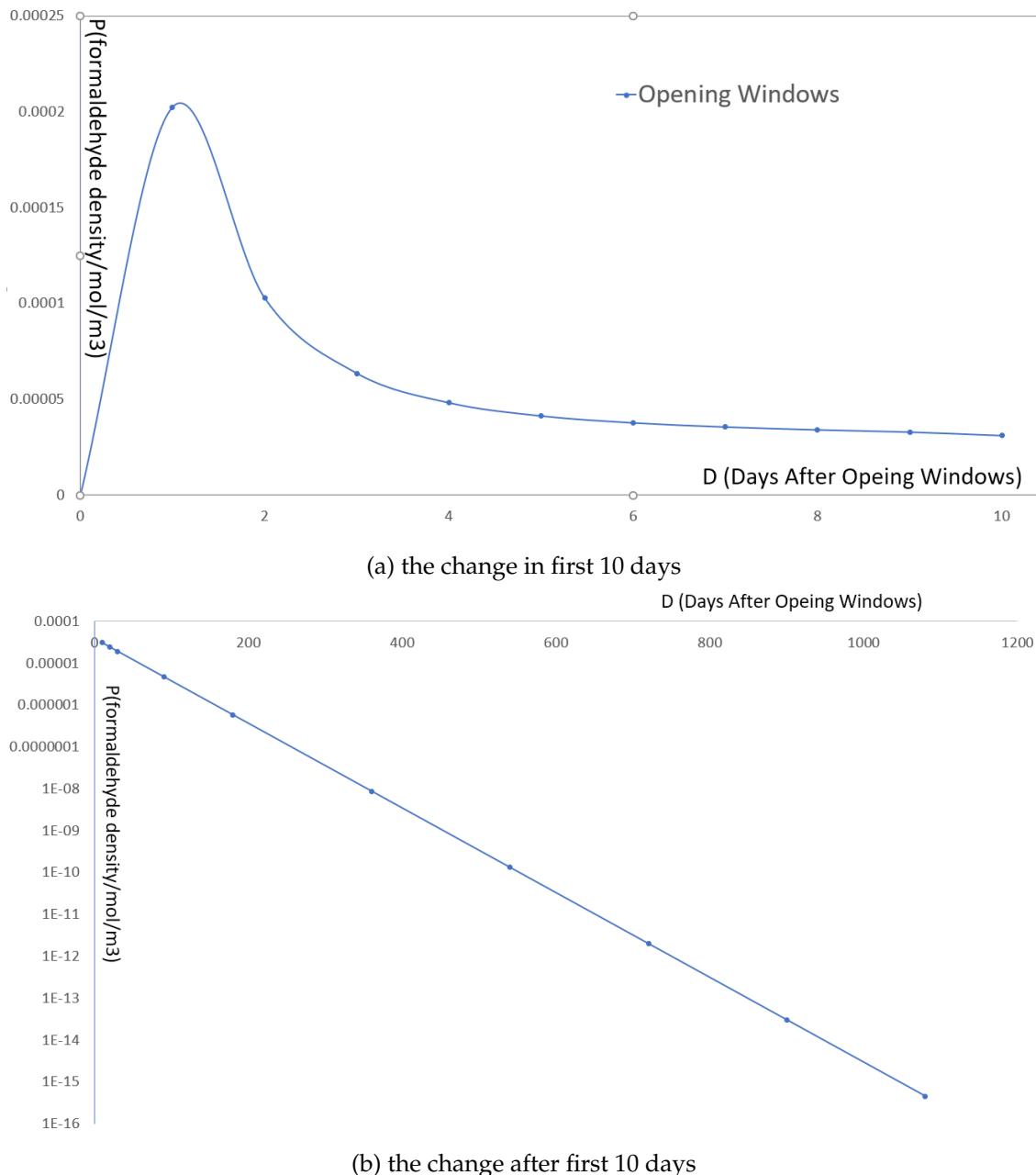


Figure 7: The change of density in 3 years (Opening Windows)

As the result shows, the density of formaldehyde first increase as the walls starting to release the gas. It reaches the top after 1.5 day, from when it starts to fall down and the its derivative's absolute value also falls down. It follows a negative logarithm law and the density reach the health threshold in 106 days. From the data, we can know that opening windows makes a difference but its contribution is limited as it takes a long time to reach the health threshold. However, this strategy is the most energy-saving one. Being combined with other more effective strategies will work better.

4.2 Exhausting Fans

Opening exhausting fans is a quite effective method from our intuition because most place take it as a significant way of ventilation. We simulate an exhausting fans with a constant wind velocity and compare it to opening the windows alone, the results are shown in Figure 8. From the curve we can clearly find out that using exhausting fans is much more effective than opening windows. The maximum value is 0.00005 when exhausting fans are turned on while this value equals 0.0002 when opening windows. From the fitting curve we can know that it only takes 21 days for the density of formaldehyde to decline to threshold compared to the 161 days of window-opening.

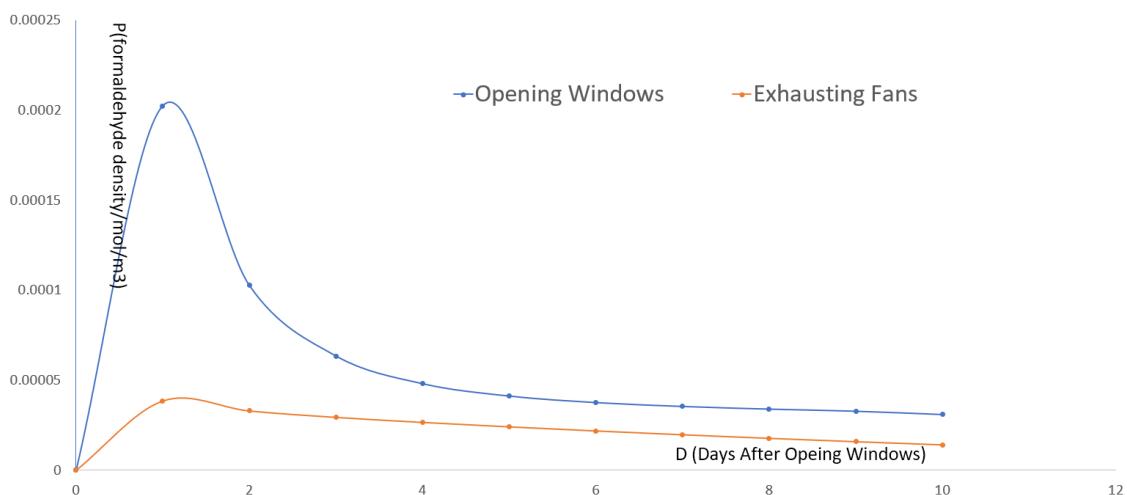


Figure 8: Exhausting Fans Compare to Opening Windows

4.3 Temperature and Humidity

According to our test results, the rise of temperature and increase of humidity have a synergistic effect on the increase of concentration of formaldehyde in the room.

We simulated the diffusion of the formaldehyde under different temperature ranging from 10°C to 35°C. It turns out that under 19°C the emission of formaldehyde can be controlled at an relatively low level. However, when the temperature is higher than 19°C , every increase of 5°C in temperature results in the rise of 1.3 to 2.5 times the concentration of formaldehyde.[8]

On the other hand, when we increase the relative humidity by every 20 percent, the density of formaldehyde increase by 1.1 to 1.3 times.

4.4 Other Strategies

Besides the strategies we put forward above, there are quite a few other ways to cut down the production of formaldehyde and accelerate the emission, Here are other selected ways to make a difference:

- Shop smart for furniture and cabinets, find Wood products that meet regulations

- Choose paints wisely and skip the wallpapers, stop using carpet from re-emitting
- Fill your room with formaldehyde-absorbing plants[9]

4.5 Final Evaluation

Among all the strategies we mentioned above, it's no denying that the exhausting fans works the best although it costs energy the most. Opening windows is the most energy-friendly way and it works well the same. Keeping temperature and humidity also works but the energy costs will be high. Other strategies can be adopted in some specific situation.

To strike a balance between effectiveness and energy cost, we strongly recommend householders to open exhausting fans in the first few days and open the window simultaneously. After the first few days, householders can stop using the exhausting fans and turn to opening windows while adopt some other strategies such as fill the room with formaldehyde-absorbing plants.

5 Model Analysis

5.1 Sensitivity Analysis

Note that our model consists a lot parameters such as room temperature and the volume of the room. Now we are studying the impact of these two parameters on the declining curve of concentration of formaldehyde and $t_{threshold}$, using CF model with a window.

5.1.1 Impact of Volume

In this section we ignore the impact of shape and concentrate on a series of similar cuboid room with linear scaling.

Table 2: Selected volume.

Scale	Size(m)	Volume(m^3)
0.8	$6.4 \times 4.0 \times 2.4$	61.44
0.9	$7.2 \times 4.5 \times 2.7$	87.48
1.0	$8.0 \times 5.0 \times 3.0$	120.00
1.1	$8.0 \times 5.0 \times 3.0$	159.72
1.2	$8.0 \times 5.0 \times 3.0$	207.36

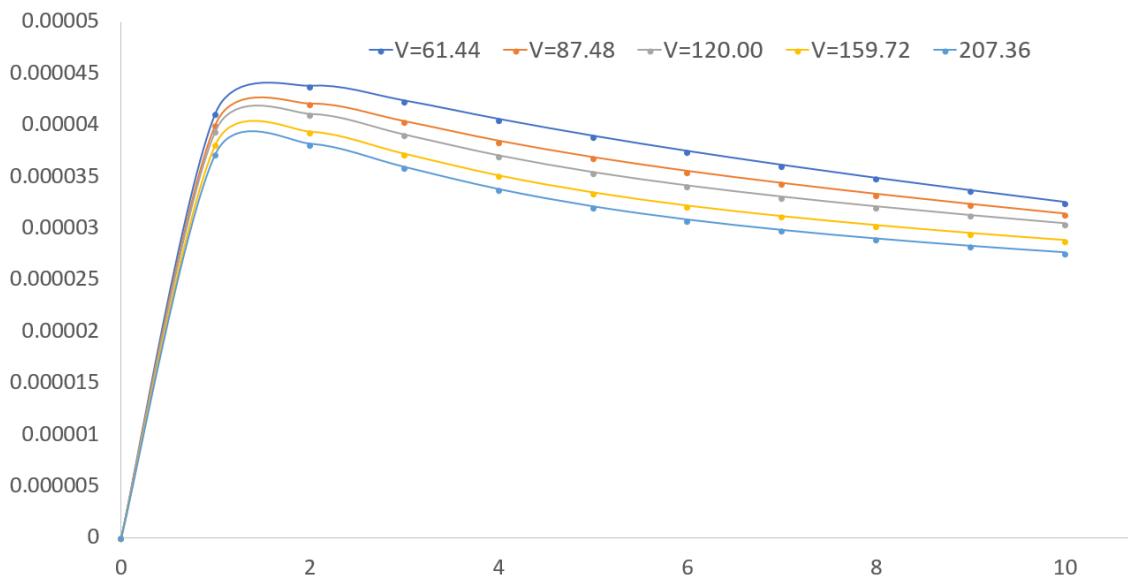


Figure 9: Impact of Volume

As the curve of volume shows, the density's change pattern varies when the volume differs. When the volume is high, the maximum value is lower. However, compared to the volume change itself, the change of density is minor. In the meantime, the relation between volume and $t_{threshold}$ is shown in Table 3, in which we can see that larger house needs much longer time to be healthy for people to live in.

Table 3: Relation between volume and $t_{threshold}$.

Volume(m^3)	61.44	87.48	120.00	159.72	207.36
$t_{threshold}(\text{s})$	88	98	106	114	122

5.1.2 Impact of Temperature

Considered that the range that an air conditioner can adjust, we think considering step of temperature as 5°C is reasonable. Meanwhile, using Gilliland's semiempirical[10] equation:

$$D_{12} = 435.7 \frac{T^{3/2}}{p_0(V_1^{1/3} + V_2^{1/3})^2} \sqrt{\frac{1}{M_1} + \frac{1}{M_2}} \text{ cm}^2/\text{s} \quad (13)$$

We know that the diffusion coefficient is proportional to $T^{3/2}$ in a narrow range, so we can know the diffusion coefficient of temperature varying from 283.15K to 313.15K.

Table 4: Diffusion Coefficient.

T(K)	283.15	288.15	293.15	298.15	303.15	308.15	313.15
$D(10^{-5}\text{cm}^2/\text{s})$	1.67	1.71	1.75	1.80	1.85	1.89	1.94

From the results shown in Figure 10, we can conclude that temperature has minor impact on the diffusion of formaldehyde, but we should still be aware that the higher the temperature is, the faster formaldehyde diffuses. Then we have a relation between temperature and $t_{threshold}$ in Table 5.

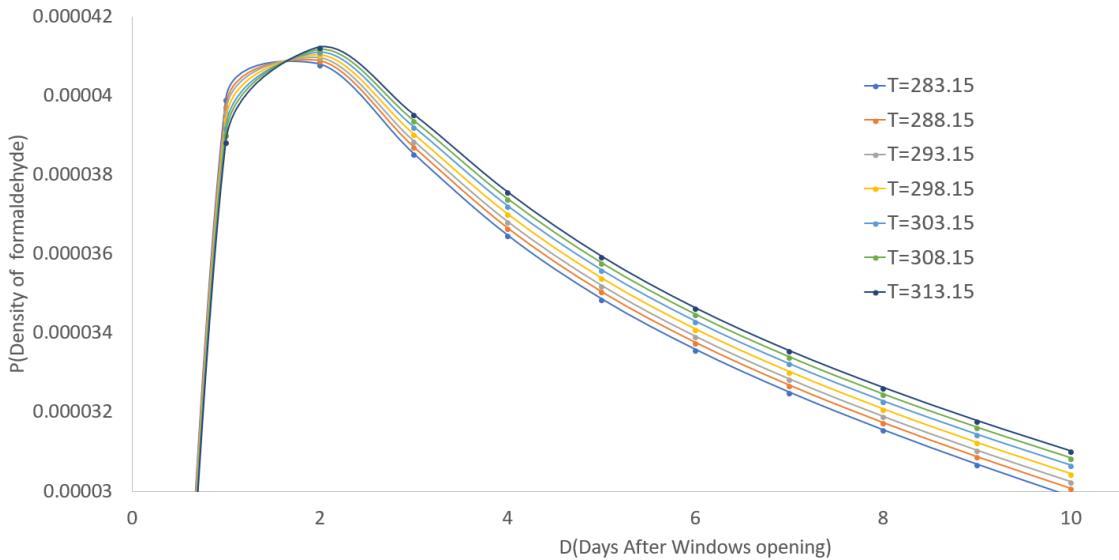


Figure 10: Impact of Temperature

As the figure of temperature shows, the concentration's curve also varies when the temperature differs. When the temperature is high, the maximum value appears quicker and starting to fall down faster. In the meantime, the relation between temperature and $t_{threshold}$ is shown in Table 5, in which we can see that the time needed to reduce to the health threshold are shorter when the temperature are higher.

Table 5: Relation between temperature and $t_{threshold}$.

Temperature(T)	313.15	308.15	303.15	298.15	293.15	288.15	283.15
$t_{threshold}(\text{d})$	99	102	104	106	109	111	114

5.1.3 Impact of Additional Emission Sources

Besides the wall and ceiling, additional emission sources like wooden furniture and wallpaper also influence a lot. According to a research [11] conduct by SK Brown, some kinds of wooden material produce formaldehyde even more than latex paint and oil paint we use now. We simulate a wooden furniture in the house and simulate the distribution of the formaldehyde, the results are shown in Figure 11:

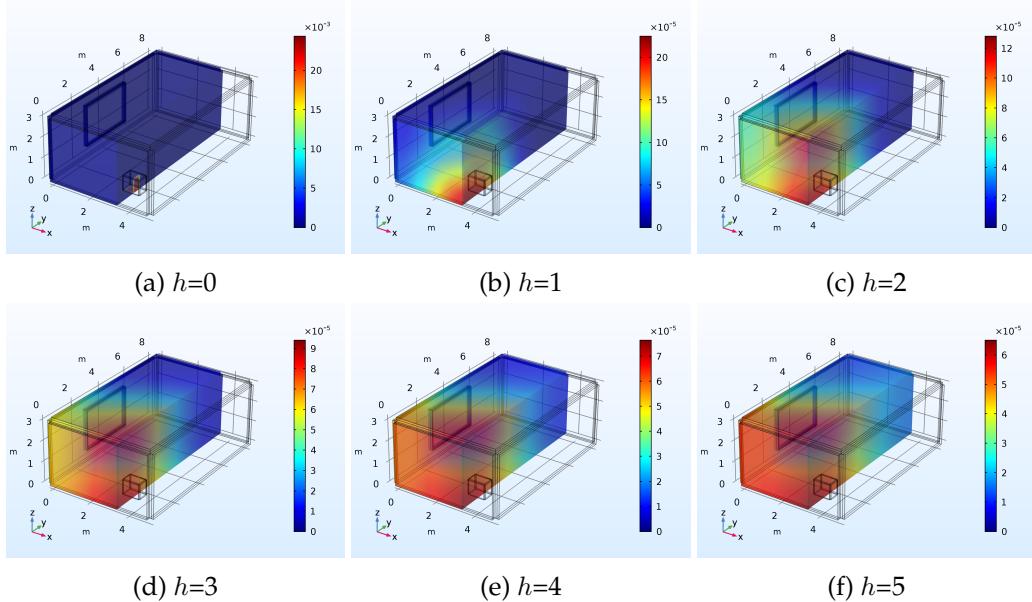


Figure 11: Simulation of the emission sources

As shown in the figure above, additional emission sources will bring nonuniform distribution of formaldehyde in the house. Those space closed to the sources will have higher density of formaldehyde. Also, the degree of the influence are determined by the concentration of formaldehyde in the sources.

5.1.4 Impact of Inner Structure

Most houses are not simply combination of cuboid as inner structure like screens and interior walls may influence the distribution of formaldehyde. We simulate the whole process with a wall and an interior door in the house, the simulation result are shown in Figure 12:

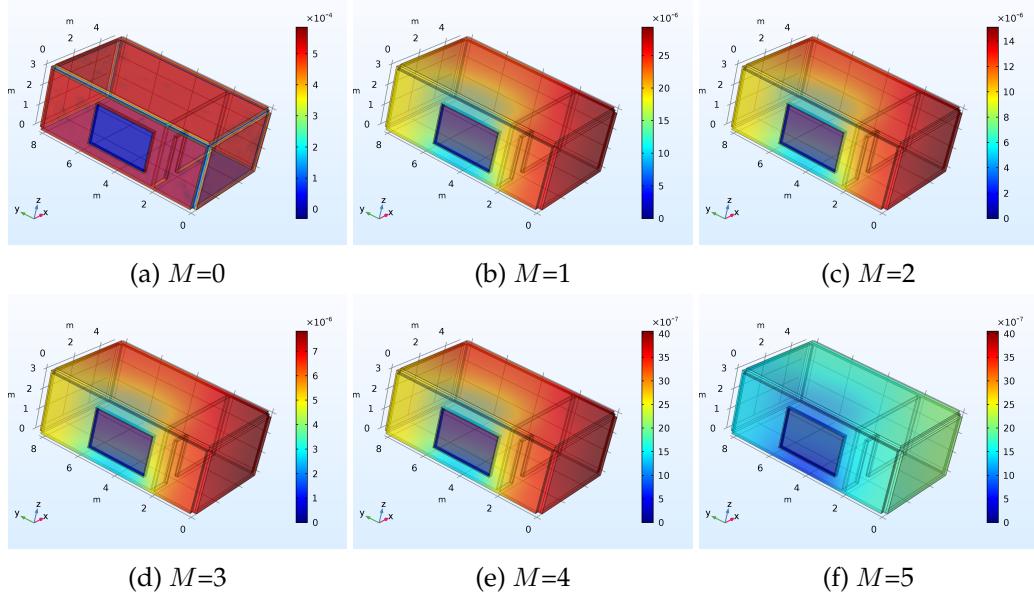


Figure 12: Simulation of the inner structure: interior walls

From the figure above we can see that interior wall also bring inhomogeneous distribution just like the additional interior emission sources. What's more, the data derived from the simulation shows that it will take more time for the density of formaldehyde to decline, mainly because the wall will block the spread of formaldehyde.

5.2 Strengths and Weakness

5.2.1 Strengths

- We introduce three distinct models covering both discrete and continuous : LGA, MD, CF. After carefully analysis and evaluation we choose the most appropriate model to continue our work.
- Our sensitivity analysis shows that our models are fairly robust to changes in parameter value so it can apply to many different situation in reality.
- The assumptions in our models are reasonable, which can not only simplify the problems, but lay the foundation for the establishment of the latter model as well.
- In each model we discuss, we provide clear and understandable graph to visualize the result of our model analysis.

5.2.2 Weakness

- In LGA model our simulation focus on 2D as it's hard to visualize the results in 3D.
- In the strategy selection part we lack the detailed discussion over the trade-off between effect and energy cost.

6 Conclusions

Our paper provides a detailed analysis of how the formaldehyde was emit and diffused, what strategies can we adopt to tackle with the formaldehyde hazard. Before introducing the model, we discussed the theory of diffusion. Then our models are proposed, made up of three distinct model: Lattice Gas Automata(LGA) model, Molecular Dynamic(MD) model and Continuous Fluid(CF) model. We use real data to simulate the emission and spread of formaldehyde.

We have discovered that the CF model works well and have the best accuracy. Based on CF model, Different selective Strategies are fully discussed. Balance between effectiveness and energy-cost have been taken into consideration. Finally we put forward the composite strategy of combining window-opening with other effective strategy such as exhausting fans to adopt in the first few days, filling room with formaldehyde-absorbing plants. When different strategies are adopted, the time needed to decline to the health threshold are given: it takes over 106 days to decline to threshold by opening windows while it takes only 21 days when using exhausting fans.

References

- [1] R. C. Grafstrom, A. J. Fornace, H. Autrup, J. F. Lechner, and C. C. Harris, "Formaldehyde damage to dna and inhibition of dna repair in human bronchial cells," *Science*, vol. 220, no. 4593, pp. 216–218, 1983.
- [2] E. Marshall, "Epa indicts formaldehyde, 7 years later," *Science*, vol. 236, pp. 381–382, 1987.
- [3] U. Frisch, B. Hasslacher, and Y. Pomeau, "Lattice-gas automata for the navier-stokes equation," *Phys. Rev. Lett.*, vol. 56, pp. 1505–1508, Apr 1986. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.56.1505>
- [4] J. R. Weimar, *Simulation with cellular automata*. Logos-Verlag, 1997.
- [5] D. d'HumiÃšres, P. Lallemand, and U. Frisch, "Lattice gas models for 3d hydrodynamics," *EPL (Europhysics Letters)*, vol. 2, no. 4, p. 291, 1986. [Online]. Available: <http://stacks.iop.org/0295-5075/2/i=4/a=006>
- [6] M. G. Paterlini and D. M. Ferguson, "Constant temperature simulations using the langevin equation with velocity verlet integration," *Chemical physics*, vol. 236, no. 1-3, pp. 243–252, 1998.
- [7] J. E. Lennard-Jones, "Cohesion," *Proceedings of the Physical Society*, vol. 43, no. 5, p. 461, 1931.
- [8] X. W. K. Q. R. Z. Yinping Zhang, Xiaoxi Luo, "Influence of temperature on formaldehyde emission parameters of dry building materials," *Atmospheric Environment*, vol. 38, no. 7, p. 14, 2006.
- [9] B. Wolverton and J. D. Wolverton, "Plants and soil microorganisms: removal of formaldehyde, xylene, and ammonia from the indoor environment," *Journal of the Mississippi Academy of Sciences*, vol. 38, no. 2, pp. 11–15, 1993.

- [10] W. J. Christian and S. P. Kezios, "Sublimation from sharp-edged cylinders in ax-symmetric flow, including influence of surface curvature," *AIChE Journal*, vol. 5, no. 1, pp. 61–68, 1959.
- [11] S. Brown, "Chamber assessment of formaldehyde and voc emissions from wood-based panels," *Indoor air*, vol. 9, no. 3, pp. 209–215, 1999.

Letter

Dear local residents,

In the last few years, many of you have redecorated your houses by painting walls and replenishing rooms with some new furniture. It is universally acknowledged that paint and pressed-wood furniture emit formaldehyde. However, not all people attach much significance to its hazards: it can cause watery eyes in short terms while trigger cancer in long terms. Owing to these facts, we are writing this letter to offer you some simple advice to minimize the hazards.

We have been recently worked on the problem dealing with speeding up the emission of formaldehyde. We set up three models and proved that they are reliable by comparing their results to the actual circumstances. By modelling the exhaling of formaldehyde, we figured out that the best method to present the diffusion of the model is Continuous Fluid model, and our work on taking different strategies is based on that model.

First of all, opening your windows as often as possible is definitely an easy approach you can take. By letting in sweet freeze, it can take away the contamination without consuming any energy. Our model has proved that by opening a window in a relatively small apartment for three months can ensure the concentration of formaldehyde meet the healthy threshold. If your room is larger, additional time is needed for the density to come to the safety threshold.

Secondly, you can install some exhausting fans. These would cost you some money, but the effect is prominent especially in the first three days. It can shorten the process of formaldehyde diffusion to only ten days. So, if you are in bad need to move into a newly painted room, you may consider this advice.

Thirdly, keeping the temperature and humidity in your house at a certain level also helps to speed up the diffusion. In our simulating experiments, temperature and humidity are in positive correlation with the diffusion of the formaldehyde. Therefore, keep in mind that the temperature in the room shouldn't be too high, and is suggested to be approximately 21 Celsius degrees. The best humidity is 50%, which can be controlled by dehumidifiers and air-conditioners.

In a nutshell, if you have just renovated your house, you can try our advice. In this case, you are bound to get rid of the formaldehyde much faster. We sincerely hope that those simple approaches can make your life healthier and merrier.

Best regards,

Team 73052

Appendices

Appendix A Implemented LGA Model

```

#pragma once
#include <cmath>
class cell
{
public:
    cell();
    ~cell();
    struct node {
        double x, y; //indicate the position of the particle
        bool i, j, k, l, E, N, W, S;
        double phi;
        double get_phi() {
            //the center point of the circle is(x0, y0), and radius is r;
            double x0 = 0.5; double y0 = 0.5;
            double r = 0.2;
            phi = r - sqrt((x - x0)*(x - x0) + (y - y0)*(y - y0));
            return phi;
        }
        node()
        {
            x = 0.0; y = 0.0;
            i = false; j = false; k = false; l = false;
            E = false; N = false; W = false; S = false;
        }

        void operator=(const node temp) {
            x = temp.x; y = temp.y;
            i = temp.i; j = temp.j; k = temp.k; l = temp.l;
            E = temp.E; N = temp.N; W = temp.W; S = temp.S;
        }

        node(double fx, double fy)
        {
            x = fx; y = fy;
            i = false; j = false; k = false; l = false;
            E = false; N = false; W = false; S = false;
        }
        ~node() {}
    }elem[1500][900];

    void initial();
    void output(int time);
    void impact();
    void move();
    double get_theta(const node & norigin, const node & nodeb);
    void cmain();

private:
    int m;
    int n;
    double dx;
    double dy;
    int maxTime;
    double stheta; //the standard line of open boundary
}

```

```
    int boubdaryTheta;//the open boundary of the circle;
};



---



```
#include "stdafx.h"
#include "cell.h"
#include <ctime>
#include <iostream>
#include <fstream>
#include <cstring>
#include <vector>
#include <sstream>
#include <cmath>
using namespace std;

int my_rand(int);

cell::cell()
{
 m = 200;
 n = m;//200
 dx = 1.0 / m;
 dy = 1.0 / n;
 maxTime = 1500;
 stheta = -0.0; //the standary line of open boundary
 boubdaryTheta = 5; //the open boundary of the circle;

 int imax = m + 1;
 int jmax = n + 1;
 for (int i = 0; i < imax; ++i) {
 for (int j = 0; j < jmax; ++j) {
 //node temp(i*dx, j*dy);
 elem[i][j].x = i * dx;
 elem[i][j].y = j * dy;
 elem[i][j].get_phi();
 }
 }
}

void cell::initial() {

 int imax = 1500;
 int jmax = 900;
 //time_t nowtime = time(nullptr);
 int nowtime = 2;
 auto seed = my_rand(nowtime);
 int random = seed;
 for (int i = 0; i < imax; ++i) {
 for (int j = 0; j < jmax; ++j) {
 elem[i][j].i = false; elem[i][j].j = false;
 elem[i][j].k = false; elem[i][j].l = false;
 if (elem[i][j].phi>0) {
 seed = my_rand(seed);
 random = 1.0 * (seed% m) / m;
 if (random < 0.25) {
 elem[i][j].i = true; elem[i][j].k = true;
 }
 else if (random < 0.5) {
 elem[i][j].j = true; elem[i][j].l = true;
 }
 }
 }
 }
}
```


```

```
        else if (random < 0.75) {
            elem[i][j].k = true;      elem[i][j].i = true;
        }
        else if (random<1) {
            elem[i][j].l = true;      elem[i][j].j = true;
        }
        else {
            cout << "random error in initial!" << endl; cin.get();
        }
        //cout << "come to here " << endl;
    }
}
}

void cell::output(int time) {
    string tempa = "axissym_";

    int temp = 10000 + time;
    stringstream ss;
    string str;
    ss << temp;
    ss >> str;

    string title = tempa + str + ".plt";
    ofstream fcout;
    fcout.open(title);

    int imax = m + 1;
    int jmax = n + 1;
    int maxParticleNum = 0;
    for (int i = 0; i < imax; ++i) {
        for (int j = 0; j < jmax; ++j) {
            if (elem[i][j].i || elem[i][j].j || elem[i][j].k || elem[i][j].l) {
                maxParticleNum++;
            }
        }
    }

    fcout << " title=random " << endl;
    fcout << " VARIABLES = \"X\", \"Y\" " << endl;
    fcout << "zone I= " << maxParticleNum << ",datapacking=POINT" << endl;

    for (int i = 0; i < imax; ++i) {
        for (int j = 0; j < jmax; ++j) {
            if (elem[i][j].i || elem[i][j].j || elem[i][j].k || elem[i][j].l) {
                fcout << i * dx << " " << j * dy << endl;
            }
        }
    }

    fcout.close();
}

cell::~cell() {}

void cell::impact()
{
    //cout << "in impact !" << endl;
    int imax = 1500;
    int jmax = 900;
```

```
for (int i = 1; i < imax - 1; ++i) {
    for (int j = 1; j < jmax - 1; ++j) {
        elem[i][j].i = false; elem[i][j].j = false;
        elem[i][j].k = false; elem[i][j].l = false;
        if (elem[i][j].E || elem[i][j].N || !elem[i][j].W || !elem[i][j].S) {
            if (elem[i][j].E && !elem[i][j].N && !elem[i][j].W && !elem[i][j].S)
                { //one particle comes from East
                    elem[i][j].k = true;
                }
            else if (!elem[i][j].E && !elem[i][j].N && elem[i][j].W && !elem[i][j].S)
                { //one particle comes from West
                    elem[i][j].i = true;
                }
            else if (!elem[i][j].E && elem[i][j].N && !elem[i][j].W && !elem[i][j].S)
                { //one particle comes from North
                    elem[i][j].l = true;
                }
            else if (!elem[i][j].E && !elem[i][j].N && !elem[i][j].W && elem[i][j].S)
                { //one particle comes from South
                    elem[i][j].j = true;
                }
            else if (!elem[i][j].E && elem[i][j].N && !elem[i][j].W && elem[i][j].S)
                { //two particles come from South and North
                    elem[i][j].k = true; elem[i][j].i = true;
                }
            else if (elem[i][j].E && !elem[i][j].N && !elem[i][j].W && elem[i][j].S)
                { //two particles come from South and East
                    elem[i][j].j = true; elem[i][j].k = true;
                }
            else if (!elem[i][j].E && !elem[i][j].N && elem[i][j].W && elem[i][j].S)
                { //two particles come from South and West
                    elem[i][j].j = true; elem[i][j].i = true;
                }
            else if (!elem[i][j].E && elem[i][j].N && elem[i][j].W && !elem[i][j].S)
                { //two particles come from North and West
                    elem[i][j].i = true; elem[i][j].l = true;
                }
            else if (elem[i][j].E && elem[i][j].N && !elem[i][j].W && !elem[i][j].S)
                { //two particles come from North and East
                    elem[i][j].l = true; elem[i][j].k = true;
                }
            else if (elem[i][j].E && !elem[i][j].N && elem[i][j].W && !elem[i][j].S)
                { //two particles come from East and West
                    elem[i][j].j = true; elem[i][j].l = true;
                }
            else if (!elem[i][j].E && elem[i][j].N && elem[i][j].W && elem[i][j].S)
                { //three particles except coming from East
                    elem[i][j].j = true; elem[i][j].i = true; elem[i][j].l = true;
                }
            else if (elem[i][j].E && elem[i][j].N && !elem[i][j].W && elem[i][j].S)
                { //three particles except coming from West
                    elem[i][j].j = true; elem[i][j].k = true; elem[i][j].l = true;
                }
            else if (elem[i][j].E && elem[i][j].N && elem[i][j].W && !elem[i][j].S)
                { //three particles except coming from South
                    elem[i][j].i = true; elem[i][j].k = true; elem[i][j].l = true;
                }
            else if (elem[i][j].E && !elem[i][j].N && elem[i][j].W && elem[i][j].S)
                { //three particles except coming from North
                    elem[i][j].i = true; elem[i][j].k = true; elem[i][j].j = true;
                }
        }
    }
}
```

```
        else if (elem[i][j].E && elem[i][j].N && elem[i][j].W && elem[i][j].S)
        { //four particles
            elem[i][j].i = true; elem[i][j].k = true; elem[i][j].j = true; elem[i][j].l = true;
        }
    }

for (int i = 0; i < imax; ++i) { //the uppest and the lowest boundary
    elem[i][0].i = false; elem[i][0].j = false;
    elem[i][0].k = false; elem[i][0].l = false;
    elem[i][jmax - 1].i = false; elem[i][jmax - 1].j = false;
    elem[i][jmax - 1].k = false; elem[i][jmax - 1].l = false;
    if (elem[i][0].N) {
        int temp = 0;
        elem[i][0].j = true;
    }
    if (elem[i][jmax - 1].S) {
        elem[i][jmax - 1].l = true;
    }
}
int temp = 0;
for (int j = 0; j < jmax; ++j) { //the left and the right boundary
    elem[0][j].i = false; elem[0][j].j = false;
    elem[0][j].k = false; elem[0][j].l = false;
    elem[imax - 1][j].i = false; elem[imax - 1][j].j = false;
    elem[imax - 1][j].k = false; elem[imax - 1][j].l = false;
    if (elem[0][j].E) {
        elem[0][j].i = true;
    }
    if (elem[imax - 1][j].W) {
        elem[imax - 1][j].k = true;
    }
}
temp = 0;
}

void cell::move()
{
//cout << "in move !" << endl;
node norigin(0.5, 0.5);

double theta = 0.0;

int imax = 1500;
int jmax = 900;
for (int i = 0; i < imax; ++i) {
    for (int j = 0; j < jmax; ++j) {
        elem[i][j].E = false; elem[i][j].N = false;
        elem[i][j].W = false; elem[i][j].S = false;
    }
}
for (int i = 0; i < imax; ++i) {
    for (int j = 0; j < jmax; ++j) {
        if (elem[i][j].i) {
            if (elem[i + 1][j].phi * elem[i][j].phi <= 0) {
                theta = get_theta(norigin, elem[i][j]);
                if (abs(theta - stheta) > boubdaryTheta) {
                    elem[i][j].E = true;
                }
            }
        }
    }
}
```

```

        else {
            elem[i + 1][j].W = true;
        }
    }
    else {
        elem[i + 1][j].W = true;
    }
}
if (elem[i][j].j) {
    if (elem[i][j].phi * elem[i][j + 1].phi <= 0) {
        theta = get_theta(norigin, elem[i][j]);
        if (abs(theta - stheta) > boubdaryTheta) {
            elem[i][j].N = true;
        }
        else {
            elem[i][j + 1].S = true;
        }
    }
    else {
        elem[i][j + 1].S = true;
    }
}
if (elem[i][j].k) {
    if (elem[i][j].phi * elem[i - 1][j].phi <= 0) {
        theta = get_theta(norigin, elem[i][j]);
        if (abs(theta - stheta) > boubdaryTheta) {
            elem[i][j].W = true;
        }
        else {
            elem[i - 1][j].E = true;
        }
    }
    else {
        elem[i - 1][j].E = true;
    }
}
if (elem[i][j].l) {
    if (elem[i][j].phi * elem[i][j - 1].phi <= 0) {
        theta = get_theta(norigin, elem[i][j]);
        if (abs(theta - stheta) > boubdaryTheta) {
            elem[i][j].S = true;
        }
        else {
            elem[i][j - 1].N = true;
        }
    }
    else {
        elem[i][j - 1].N = true;
    }
}

}

}

double cell::get_theta(const node & norigin, const node & nodeb)
//get the theta by giving origin point and another point
{
    const double PI = 3.14159;
    double x = nodeb.x - norigin.x;
}

```

```
double y = nodeb.y - norig.y;
double thet = 0.0;

if (x == 0) {
    if (y > 0) {
        thet = 90;
    }
    else if (y < 0) {
        thet = -90;
    }
}
else {
    thet = atan2(y, x);
}
thet = thet * 180 / PI;

return thet;
}

void cell::cmain() {

cell::node anode(0, 0);
cell::node bnode(-1, -2);
double th = get_theta(anode, bnode);
cout << th << endl;
initial();
output(0);
for (int i = 0; i < maxTime; ++i) {
    output(i);
    cout << "time = " << i << endl;
    for (int j = 0; j != 20; ++j, ++i) {
        /*cout << "time = " << i << endl;*/
        move();
        impact();
    }
}
}

int my_rand(int z)
// 16807 way to create random numbers
// z is the seed number, num is the total random number to create
{
//z(n+1)=(a*z(n)+b) mod m
//describe m=a*q+r to avoid that
//the number is large than the computer can bear
const int m = pow(2, 31) - 1;
const int a = 16807;
const int q = 127773;
const int r = 2836;

int random = a * (z%q) - r * (z / q);

if (random<0)
{
    random = m + random;
}
//z is the seed number
//z = temp;
//double t = z*1.0 / m;

//cRandom cr;
```

```
//cr.random = t;
//cr.seed = z;

return random;
}

// LGA.cpp
// MCM/ICM modeling
#include "stdafnx.h"
#include "cell.h"
#include <iostream>
#include <ctime>
#include <cmath>

using namespace std;

int my_rand(int);

int main() {

    cell solver;

    solver.cmain();
    cout << sizeof(solver) << endl;

    int temp = 2;
    int timeseed = my_rand(2);
    time_t nowtime = time(nullptr);
    //int nowtime = time(nullptr);
    cout << "press any button to exit "<< timeseed << " " << nowtime << endl;
    system("pause");
    return 0;
}
```
