Simulation of airflow through a window with an anti-bug grid

Martin Opat (s4704126), Robin Sachsenweger Ballantyne (s4617096)

November 2024

1 Introduction

Bug grids can be placed over open windows to allow air to flow through without letting bugs into someone's house. Presumably, when people open their windows, they want air to flow through their house. However, our hypothesis is that a bug grid will decrease the flux and that different grids will affect the flux differently. Using a fluid flow simulation, we want to study how the grid's thickness and hole density affect the flux. Finding optimal thickness and hole density can help design anti-bug grids.

Figure 1.1: Anti-bug grid (Flickr.com)

1.1 Research question

Main:

- How much does the air flux through a window with a bug grid change with the grid's thickness and density?

Subquestions:

- What are the optimal parameters of the grid that maximise airflow while keeping the bugs out?
- In which section of the window has the airflow decreased the most due to the presence of the bug grid?

2 Model and Methodology

We aim to implement a numerical 3D simulation of airflow based on Navier-Stokes equations in Python. The numerical simulation will include discretising space and time. The cells that make up the bug grid will be modelled using a no-slip boundary condition.

2.1 Assumptions:

1. Mass conservation
2. Isotropy (i.e., no gravity)
3. Compressible fluid
4. Dynamic (μ) and bulk (ζ) viscosities are uniform in space
5. Stoke's hypothesis ($\zeta = 0$)
6. Newtonian fluid

2.2 Math

The assumptions above yield the Navier-Stokes equation(s) **batchelor2000introduction** in the following form:

$$\left(\partial_t + \mathbf{u} \cdot \nabla - \nu \nabla^2 - \left(\tfrac{1}{3}\nu + \xi\right) \nabla(\nabla \cdot)\right) \mathbf{u} = -\frac{1}{\rho} \nabla p + \mathbf{f}, \tag{2.1}$$

where ρ is the mass density, $\mathbf{u}$ is velocity, p is the pressure, ν is a shear kinematic viscosity parameter, ξ is the bulk kinematic viscosity, and $\mathbf{f}$ is the body force (which is 0 due to isotropy). Further "massaging" the equation:

$$\partial_t \mathbf{u} + \underbrace{(\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \nabla^2 \mathbf{u} - \left(\tfrac{1}{3}\nu + \xi\right) \nabla(\nabla \cdot \mathbf{u})}_{f(\mathbf{u})} = -\frac{1}{\rho} \nabla p, \tag{2.2}$$

where $f(\mathbf{u})$ is a function of $\mathbf{u}$ and its spatial (partial) derivatives.

$$\partial_t \mathbf{u} = -f(\mathbf{u}) - \frac{1}{\rho} \nabla p \tag{2.3}$$

Applying the finite difference method to the time derivative in eq. 2.3, we get:

$$\frac{\mathbf{u}_{ijk}^{n+1} - \mathbf{u}_{ijk}^{n}}{\Delta t} = -f(\mathbf{u}^n) - \frac{1}{\rho} \nabla p \tag{2.4}$$

$$\mathbf{u}_{ijk}^{n+1} = \mathbf{u}_{ijk}^{n} - \Delta t \left(f(\mathbf{u}^n) + \frac{1}{\rho} \nabla p \right), \tag{2.5}$$

where Δt is the discrete time-step size, n is the time-step index, and $i, j,$ and k are the indices in the $x, y,$ and z spatial coordinates respectively. We can apply the finite difference method to all the (spatial) derivatives in f in the same way as we did above.

We also impose the no-slip boundary condition at the location of the rectangular bug grid, as well as on the walls around the window:

$$\forall i, j, k \in [\text{grid},\ \text{window}]\ \forall n : \ \mathbf{u}_{ijk}^{n} = \mathbf{0} \tag{2.6}$$

3 Simulation setup

There are two ways of approaching the numerical simulation of the equations above:

1. Giving eq. (2.3) into a PDE solver
2. Implement and perform a numerical integration using eq. (2.5)

Option 1. is the preferred option, assuming a PDE-solve Python package exists that can accommodate eqs. (2.5) and (2.6). Option 2. is a backup option, and will only be used if no existing package fits our criteria well enough.

4 Results

5 Discussion

6 Conclusion