

The first scientific evidence for the hail cannon

Krzysztof Misan¹, Maciej Koziejka¹,
Marcin Łoś¹[0000-0002-8426-6345], Dominik Gryboś¹[0000-0002-4332-0771],
Jacek Leszczyński¹[0000-0001-7512-994X], Paweł Maczuga¹[0000-0002-5111-6981],
Maciej Woźniak¹[0000-0002-5576-5671], Albert Oliver Serra²[0000-0002-3783-8670],
and Maciej Paszyński¹[0000-0001-7766-6052]

¹AGH University of Science and Technology, Kraków, Poland
maciej.paszynski@agh.edu.pl

²Universidad de Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain

Abstract. The hail cannon has been used to prevent hail storms since the 19th century. The idea of the hail cannon is to create a sequence of shock waves to prevent the formation of clouds before the hail storm. Modern hail cannons employ a mixture of acetylene and oxygen to ignite a sequence of explosions in the lower chamber traveling through the neck and into the cone of the cannon, creating shock waves. The shock waves propagate upwards to the cloud, and they are supposed to prevent the formation of the cloud. According to Wikipedia, there is no scientific evidence for the hail cannon, even though it is commonly used in several countries. In this paper, we propose a numerical simulation to verify the idea of the hail cannon. We employ isogeometric analysis and variational splitting methods. We compare our numerical results with the experimental data. We show that our numerical simulation is indeed the scientific evidence for the hail cannon. We also compare our numerical simulations with the experimental measurements performed with a drone before and after a sequence of generated shock waves.

Keywords: Hail cannon, Cloud formation, Advection-Diffusion model, Variational Splitting, Isogeometric Analysis

1 Introduction

A hail cannon is a shock wave generator that is supposed to prevent the formation of hail storms. Modern hail cannons employ a sequence of explosions with an acetylene-oxygen mixture in the combustion chamber. The shock wave created by the explosions travels upward through reversed cone shape pipes. It is supposed to create a shock wave that travels upwards and causes the cloud vapor particles to travel up and to the side. Albert Stiger created the first modern kind of hail cannon in 1895. He was a farmer, and he had a large wine plantation in Austria [6], damaged by local hail storms. The first international congress on hail shooting happened in 1902 in [27]. Despite claims that there is no scientific evidence in favor of hail cannons [30], they are still successfully manufactured [12]. One claim that there is no physical evidence for the hail cannon to work

is that a thunderstorm is much stronger than the hail cannon, and it does not seem to disturb the hailstorms [8].

In this paper, we perform the first three-dimensional numerical simulations of the hail cannon. We employ the advection-diffusion partial differential equation to model the process. The unknown scalar field is the concentration of the cloud vapor. The shock waves generated by the hail cannon are modeled as the advection vector field. We show that a sequence of generated shock waves can move the cloud vapor up and to the sides (see Figure 1). After finishing a sequence of shock wave generation, the "hole" in the cloud remains intact for a long time. After that time, if the neighboring clouds are still there, the hole is filled with the cloud vapor particles by the diffusion mechanism. In this sense, we show that generating a sequence of shock waves can produce a hole in the cloud vapor. The argument that the thunderstorm is not removing the hail storm and, thus, the hail cannons are not working is not valid since we need to produce a long sequence of shock waves to obtain the desired effect.

In this paper, we apply the isogeometric finite element method (IGA-FEM) [7] for three-dimensional simulations of the hail cannon. The IGA-FEM employs higher-order and continuity basis functions to approximate different physical phenomena described by Partial Differential Equations. Several researchers applied the IGA-FEM to model different engineering applications. To name a few, IGA-FEM was applied to deformable shell theory [3], phase field modeling [10, 9], phase separation simulations with either Cahn-Hilliard [13] or Navier-Stokes-Korteweg higher order models [14], wind turbine aerodynamics [16], incompressible hyper-elasticity [11], turbulent flow simulations [5], transportation of drugs in arterials [15] or the blood flow simulations [4]. In our simulations, we use an explicit dynamics solver, and we employ linear computational cost alternating directions solver [23]. We will use our C++ IGA-ADS code [24] linking LAPACK [2], parallelized into shared-memory multi-core servers using GALOIS library [26]. We developed an interface into the VTK visualization toolkit [28]. Due to the IGA-ADS solver's ability to run fast and accurate three-dimensional simulations on a laptop, it was employed to simulate several phenomena. They include tumor growth [21, 19, 25], non-linear flow in heterogeneous media with possible applications to CO2 sequestration process [24, 20], as well as pathogen propagation problem [22].

The structure of the paper is the following. We start with detailed description of the hail cannon and our experimental verification of the cloud vapor reduction by using a drone measurements in Section 2. Next, Section 3 introduces the Partial Differential Equations modeling the phenomena, together with IGA-FEM discretizations. Section 4 describes details of our simulations of the hail cannon generating a sequence of the shock-waves into the hail cloud. Finally, we conclude the paper in Section 5.

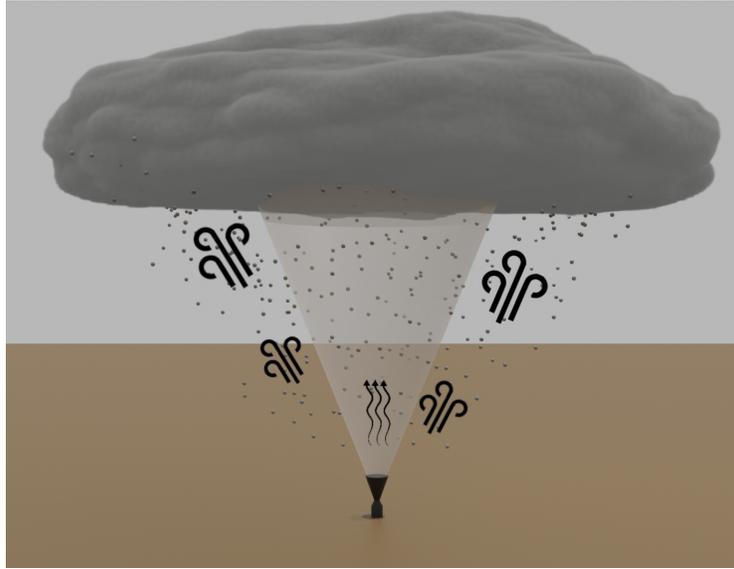


Fig. 1: The idea of using shock waves to mix and lift cloud vapor particles.

2 Experimental verification

In the technological experiments we use the the Inopower anti-hail cannon [17], shown in Fig. 2. The device consists of a container with dimensions of 6.00m x 2.45m x 2.60m with a combustion chamber of 150 dm³, three fuel inlets and a control panel. The shock wave created by the ignition of the acetylene-air mixture is directed upward through the conical outlet pipe. On the other side of the container there are acetylene cylinders and a gas pressure reduction system.

During our experiments, the gauge of the acetylene fed into the explosion chamber was 2.9 bar. During the explosion of the mixture of acetylene and air, a pressure of about 1 MPa was reached. During the experiment, about 300 shock waves were generated in half an hour.

For the experimental verification we measured the temperature, the humidity, and the particular matter concentration in the vertical profile using the equipment placed on the DJI Matrice 200 V2 drone. For each test, a flight was performed immediately before and after the generator was started and 20 minutes after its completion. During the test, measurements with the drone were made before the experiment and several times after the experiment, in particular, the flight was made 15 and 5 minutes before starting the generator, 5, 15 and 30 and 40 minutes after its completion. Fig. 3 shows the measurement data of the altitude profile from 0 m to 130 m.

From the experimental data, we can read the two times reduction of the cloud vapor (possibly with pollution particles) around 15 minutes after we start



Fig. 2: Hail cannon: Container with acetylene cylinders and control panel. Combustion chamber with the conical outlet pipe.

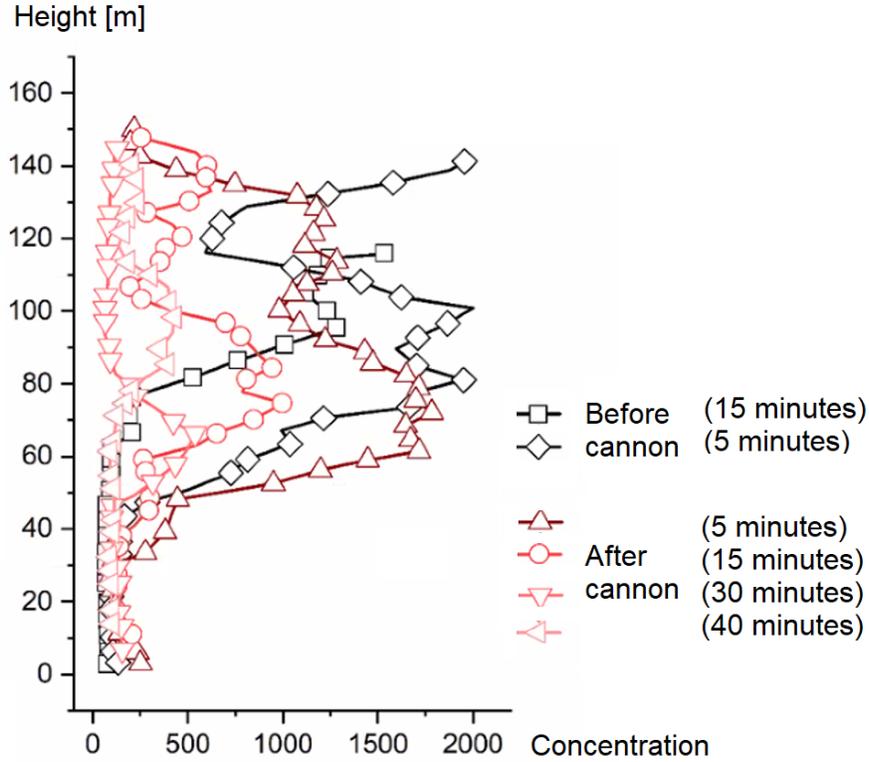


Fig. 3: Reduction of cloud vapor particles after a sequence of shots.

the sequence of shock wave generation, and the significant reduction around 30 minutes of the shock wave generation. Finally, 40 minutes from the beginning of the sequence, 10 minutes after we finish the sequence, we can still see around five times reduction of the cloud vaport. We can conclude that the long sequence of shots can significantly reduce the cloud vapor concentration.

3 Numerical simulations

We employ advection-diffusion-reaction equations to model the concentration of the water vapor forming a cloud. The equations in the strong form are

$$\begin{aligned}
 \frac{\partial u}{\partial t} + (\mathbf{b} \cdot \nabla)u - \nabla \cdot (\mathbf{K}\nabla u) &= 0 \text{ in } \Omega \times (0, T], \\
 \nabla u \cdot n &= 0 \text{ in } \Omega \times (0, T], \\
 u &= u_0 \text{ in } \Omega \times 0,
 \end{aligned}
 \tag{1}$$

where u is the concentration scalar field, $\mathbf{b} = (b_x, b_y, b_z)$ is the assumed "wind" velocity vector field,

$$\mathbf{K} = \begin{pmatrix} K_{11} & 0 & 0 \\ 0 & K_{22} & 0 \\ 0 & 0 & K_{33} \end{pmatrix}, \quad (2)$$

is the isotropic diffusion matrix. The scalar concentration field u represents cloud vapor particles. The initial conditions on the vertical cross-section of the domain are presented in Figure 4.

In the atmospheric modeling [1, 29, 18] there are two types of diffusion: molecular diffusion and turbulent diffusion (also known as eddy diffusion). In the air, turbulent diffusion is quicker than molecular diffusion. The turbulent diffusion is not isotropic. For example, in a stable atmosphere (all air masses are stratified where the lower ones are denser and less warm than the upper ones that are lighter and warmer), the horizontal turbulence diffusion is greater than the vertical diffusion (there is no transfer between layers). However, when the atmosphere is unstable (some lighter masses are going up through denser ones), the vertical turbulent diffusion can be quicker than the horizontal.

In our model, the diffusion is assumed to be ten times smaller in the vertical direction $K_x = K_y = 1.0$, and $K_z = 0.1$. The advection term models the air movement. We do not investigate the influence of the wind in this simulation. Instead, the advection term will be employed to model the air movement as enforced by the shock-wave generator, according to equation (15). We will generate a sequence of shock waves to remove the cloud. Of course, it is possible to add the wind to the model and check how the shock wave generator works if we have a strong wind moving the cloud vapor.

We employ the explicit time integration scheme, where we approximate the time derivative with $\frac{\partial u}{\partial t} = \frac{u^{t+1} - u^t}{dt}$, where dt is the time step size. We also assume that we evaluate the remaining terms in the previous time step. As the result we get the explicit Euler time integration scheme

$$\frac{u^{t+1} - u^t}{dt} = \nabla \cdot (\mathbf{K} \nabla u^t) - (\mathbf{b} \cdot \nabla) u^t = 0. \quad (3)$$

The weak formulation is obtained by testing with B-spline basis functions

$$(u^{t+1}, v) = (u^t, v) - dt (\mathbf{K} \nabla u^t, \nabla v) - dt (\mathbf{b} \cdot \nabla u^t, v) + (cu^t, v) \forall v \in V. \quad (4)$$

We discretize with B-spline basis functions defined over the cube shape domain $\Omega = [0, 1]^3$

$$\begin{aligned} u^{t+1} &= \sum_{i=1, \dots, N_x; j=1, \dots, N_y; k=1, \dots, N_z} u_{ij}^{t+1} B_i^x B_j^y B_k^z, \\ u^t &= \sum_{i=1, \dots, N_x; j=1, \dots, N_y; k=1, \dots, N_z} u_{ij}^t B_i^x B_j^y B_k^z, \end{aligned} \quad (5)$$

and we test with B-spline basis functions

$$\begin{aligned}
 \sum_{ijk} u_{ijk}^{t+1} (B_i^x B_j^y B_k^z, B_l^x B_m^y B_n^z) &= \sum_{ij} u_{ijk}^t (B_i^x B_j^y B_k^z, B_l^x B_m^y B_n^z) - \\
 &dt \sum_{ijk} u_{ijk}^t \left(K_x \frac{\partial B_i^x}{\partial x} B_j^y B_k^z, \frac{\partial B_l^x}{\partial x} B_m^y B_n^z \right) - \\
 &dt \sum_{ijk} u_{ijk}^t \left(K_y B_i^x \frac{\partial B_j^y}{\partial y} B_k^z, B_l^x \frac{\partial B_m^y}{\partial y} B_n^z \right) - \\
 &dt \sum_{ijk} u_{ijk}^t \left(K_z B_i^x B_j^y \frac{\partial B_k^z}{\partial z}, B_l^x B_m^y \frac{\partial B_n^z}{\partial z} \right) - \quad (6) \\
 &dt \sum_{ij} u_{ijk}^t \left(b_x(x, y, z) \frac{\partial B_i^x}{\partial x} B_j^y B_k^z, B_l^x B_m^y B_n^z \right) + \\
 &dt \sum_{ijk} u_{ijk}^t \left(b_y(x, y, z) B_i^x \frac{\partial B_j^y}{\partial y} B_k^z, B_l^x B_m^y B_n^z \right) + \\
 &dt \sum_{ijk} u_{ijk}^t \left(b_z(x, y, z) B_i^x B_j^y \frac{\partial B_k^z}{\partial z}, B_l^x B_m^y B_n^z \right) \\
 &l = 1, \dots, N_x; m = 1, \dots, N_y; n = 1, \dots, N_z,
 \end{aligned}$$

where $(u, v) = \int_{\Omega} u(x, y, z)v(x, y, z)dx dy dz$.

We separate directions

$$\begin{aligned}
 \sum_{ijk} u_{ijl}^{t+1} (B_i^x, B_l^x)_x (B_j^y, B_m^y)_y (B_k^z, B_n^z)_z &= \sum_{ij} u_{ijk}^t (B_i^x, B_l^x)_x (B_j^y, B_m^y)_y (B_k^z, B_n^z)_z - \\
 &dt \sum_{ijk} u_{ijk}^t \left(K_x \frac{\partial B_i^x}{\partial x}, \frac{\partial B_l^x}{\partial x} \right)_x (K_y B_j^y, B_m^y)_y (K_z B_k^z, B_n^z)_z - \\
 &dt \sum_{ijk} u_{ijk}^t (K_x B_i^x, B_l^x)_x \left(K_y \frac{\partial B_j^y}{\partial y}, \frac{\partial B_m^y}{\partial y} \right)_y (K_z B_k^z, B_n^z)_z - \\
 &dt \sum_{ijk} u_{ijk}^t (K_x B_i^x, B_l^x)_x (K_y B_j^y, B_m^y)_y \left(K_z \frac{\partial B_k^z}{\partial z}, \frac{\partial B_n^z}{\partial z} \right)_z + \\
 &dt \sum_{ijk} u_{ijk}^t \left(b_x \frac{\partial B_i^x}{\partial x}, B_l^x \right)_x (b_y B_j^y, B_m^y)_y (b_z B_k^z, B_n^z)_z + \\
 &dt \sum_{ijk} u_{ijk}^t (b_x B_i^x, B_l^x)_x \left(b_y \frac{\partial B_j^y}{\partial y}, B_m^y \right)_y (b_z B_k^z, B_n^z)_z + \\
 &dt \sum_{ijk} u_{ijk}^t (b_x B_i^x, B_l^x)_x (b_y B_j^y, B_m^y)_y \left(b_z \frac{\partial B_k^z}{\partial z}, B_n^z \right)_z \\
 &l = 1, \dots, N_x; m = 1, \dots, N_y; n = 1, \dots, N_z.
 \end{aligned}$$

We introduce

$$\mathbf{M}_x = \{(B_i^x, B_l^x)_x\}_{il} = \left\{ \int B_i^x B_l^x dx \right\}_{il}, \quad (7)$$

$$\mathbf{M}_y = \{(B_j^y, B_m^y)_y\}_{jm} = \left\{ \int B_j^y B_m^y dy \right\}_{jm}, \quad (8)$$

$$\mathbf{M}_z = \{(B_k^z, B_n^z)_z\}_{kn} = \left\{ \int B_k^z B_n^z dz \right\}_{kn}. \quad (9)$$

In general, Kronecker product matrix $\mathcal{M} = \mathcal{A}^x \otimes \mathcal{B}^y \otimes \mathcal{C}^z$ over 3D domain $\Omega = \Omega_x \times \Omega_y \times \Omega_z$ is defined as

$$\mathcal{M}_{ijklmn} = \mathcal{A}_{il}^x \mathcal{B}_{jm}^y \mathcal{C}_{kn}^z. \quad (10)$$

Due to the fact, that one-dimensional matrices discretized with B-spline functions are banded and they have $2p + 1$ diagonals (where p stands for the order of B-splines), since

$$(\mathcal{M})^{-1} = (\mathcal{A}^x \otimes \mathcal{B}^y \otimes \mathcal{C}^z)^{-1} = (\mathcal{A}^x)^{-1} \otimes (\mathcal{B}^y)^{-1} \otimes (\mathcal{C}^z)^{-1}. \quad (11)$$

we can solve our system in a linear computational cost. The Kronecker product decomposition on the right-hand-side can also help in developing fast integration algorithms.

4 IGA-ADS simulation of the hail cannon

We perform three-dimensional computer simulations of the hail cannon using our IGA-ADS code [24], employing isogeometric finite element method and linear computational cost solver. In our simulation, the scalar field u represents the water vapor forming a cloud (possibly mixed with the pollution particles). Our initial configuration is the cloud "fixed" at the height of $3/4$ of the domain. We formulate the problem in the domain, with $\Omega = [0, 100m] \times [0, 100m] \times [0, 100m]$. The initial state is presented in Figure 4. The vertical axis on the left-hand side of the picture represents the vertical dimension of the domain. The vertical axis on the right-hand side represents the cloud vapor concentration field. The display scale is fixed from 0 to 10,000, on all the plots.

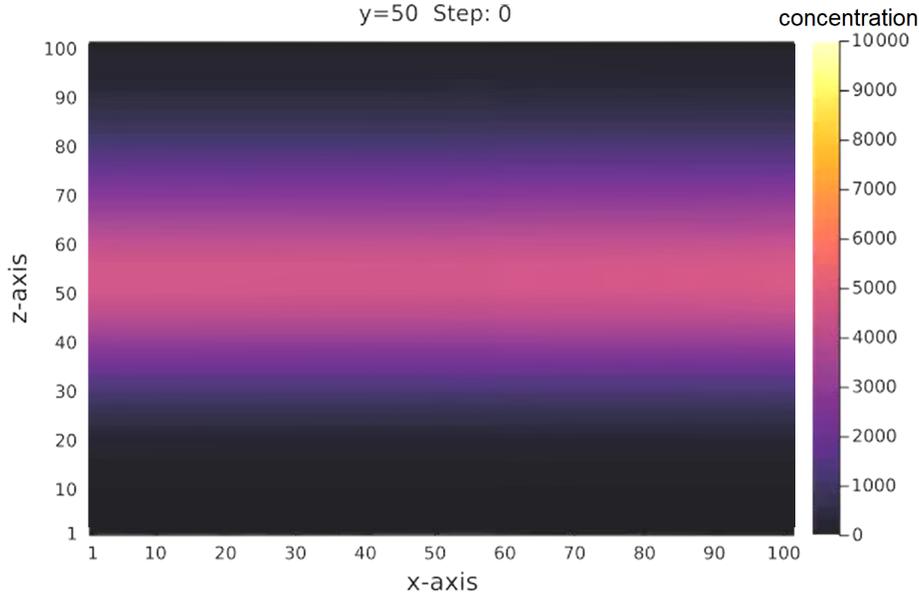


Fig. 4: Initial configuration for the cloud.

We employ the advection-diffusion equations,

$$\begin{aligned} & \frac{\partial u(x, y, z; t)}{\partial t} + b_x(x, y, z; t) \frac{\partial u(x, y, z; t)}{\partial x} \\ & \quad + b_y(x, y, z; t) \frac{\partial u(x, y, z; t)}{\partial y} \\ & \quad + b_z(x, y, z; t) \frac{\partial u(x, y, z; t)}{\partial z} \\ & - K_x \frac{\partial^2 u(x, y, z; t)}{\partial x^2} - K_y \frac{\partial^2 u(x, y, z; t)}{\partial y^2} - K_z \frac{\partial^2 u(x, y, z; t)}{\partial z^2} = 0, \\ & \quad (x, y, z; t) \text{ in } \Omega \times (0, T], \end{aligned} \quad (12)$$

$$\nabla u(x, y, z; t) \cdot n(x, y, z) = 0, \quad (x, y, z; t) \text{ in } \Omega \times (0, T], \quad (13)$$

$$u(x, y, z; 0) = u_0 \text{ in } \Omega \times 0, \quad (14)$$

where u is the concentration scalar field, where the shock-waves are modeled by the advection field, namely $(b_x(x, y, z; t), b_y(x, y, z; t), b_z(x, y, z; t)) = (0, 0, cannon(x, y, z; t))$ (given by equation (15)), $K_x = K_y = 1.0$ are the horizontal diffusion coefficients, and $K_z = 0.1$ is the vertical diffusion.

We employ the implementation of the linear computational cost Kronecker product structure solver as described in [24]. The explicit method formulation is implemented in the `compute_rhs` routine

```
void compute_rhs(int iter)
```

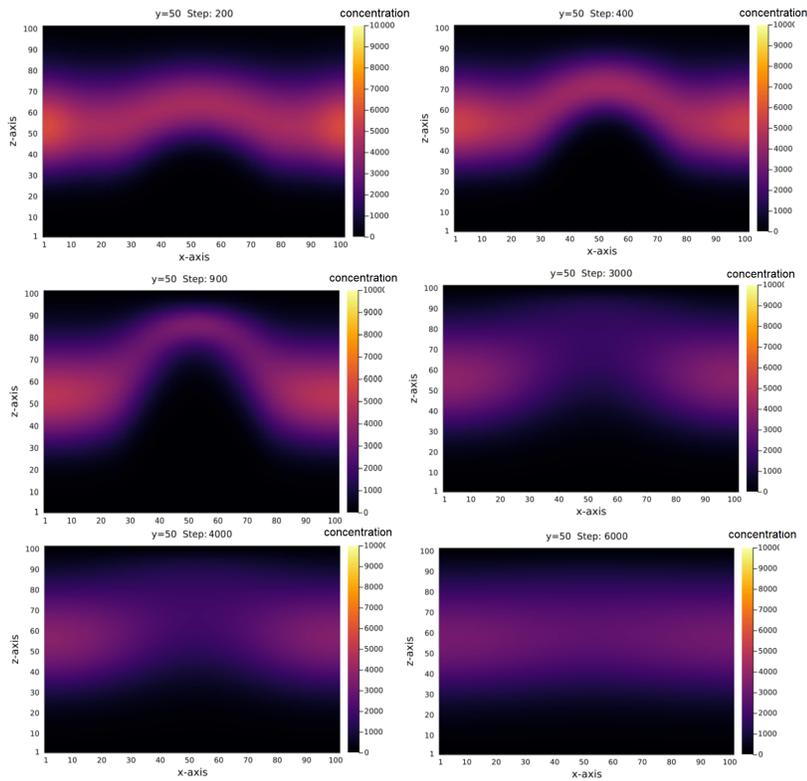


Fig. 5: Reduction of the cloud vapor by generated shock waves. Side view.

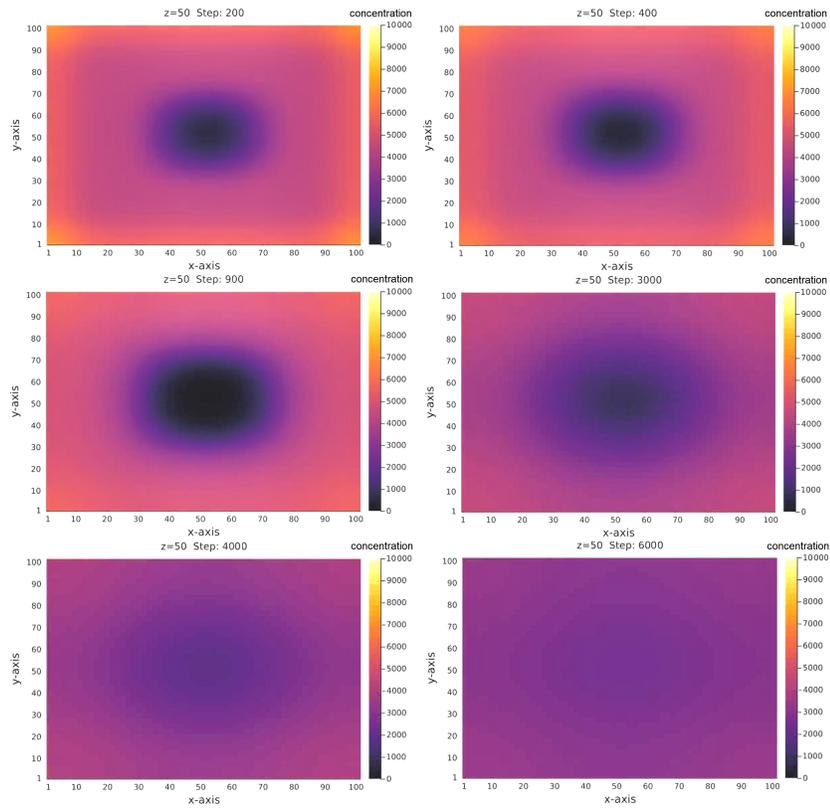


Fig. 6: Reduction of the cloud vapor by generated shock waves. Top view.

```

1  auto& rhs = u;
2  zero(rhs);
3  for(auto e:elements()) {
4      auto U = element_rhs();
5      double J = jacobian(e);
6      for (auto q : quad_points()) {
7          double w = weight(q);
8          for (auto a : dofs_on_element(e)) {
9              auto aa = dof_global_to_local(e, a);
10             value_type v = eval_basis(e, q, a);
11             value_type u = eval_fun(u_prev, e, q);
12             double grad = 1.*u.dx * v.dx +
13                 1.*u.dy * v.dy +
14                 0.1*u.dz * v.dz;
15             double val = u.val * v.val - steps.dt * grad+
16                 steps.dt*cannon(e[1])*u.dz*v.val;
17             U(aa[0], aa[1]) += val * w * J;
18         }
19     }
20     update_global_rhs(rhs, U, e);
21 }

```

In order to simulate the atmospheric cannon, we introduce the shock wave as the the advection function in a separable way as

$$\begin{aligned}
 \text{cannon}(x, y, z; t) = & \text{const} * (1 - z) * \\
 & \sin(10 * \pi * x) * \sin(10 * \pi * y) * \\
 & \max(0, \sin(\pi * t/10)),
 \end{aligned} \tag{15}$$

for $t = s - 100$, where s is the time step size. In other words we run the cannon from time step 100, and we shoot for 10 time steps with a function $(1 - z) * \sin(10 * \pi * x) * \sin(10 * \pi * y)$ that runs in time like $\max(0, \sin(\pi * t/10))$.

```

1  double cannon(double x, double y, double z, int iter) {
2      x=x/40.; y=y/40.; z=z/40.
3      double t=iter;
4      if(x>0.3 && x<0.6 && y>0.3 && y<0.6 &&t>0 && t<1000)
5          return 200.*(1.-z)*
6              max(sin(10*PI*x),0) *max(sin(10*PI*y),0)*
7              max(0,sin(PI*t/10));
8      else
9          return 0.;
10 }

```

We add this cannon function to the right-hand-side computing routine

```

void compute_rhs(int iter)
...
13     double val = u.val * v.val - steps.dt * grad+
        steps.dt*(delta_T(e[1])-
        cannon(e[0],e[1],e[2], iter))*u.dz*v.val;
...

```

The simulations are executed on a laptop. The whole simulation takes around 1 hour on Corsair Vengeance LPX, DDR4, 64 GB (2x32GB), 3200MHz, CL16 with processor AMD Ryzen 9 - 3900X with 12 physical cores, and a total of 24 virtual cores. We run fully three-dimensional simulations, and we present a horizontal cross-section along OXZ in the middle of the domain in Figure 5, and the vertical cross-section along OXY in the middle height of the domain in Figure 6. We start generating the shock waves at time step $t = 100$. The configuration at time step $t = 200$, 100-time steps after the start of the sequence is presented in the first panel in Figures 5-6. We maintain the sequence of generated shock waves, and we present the next configuration at time step $t = 400$ in the second panel in Figures 5-6. We can read from this simulation that generating a sequence of shock waves results in a local mixing of the layers and a reduction of the cloud vapor (possibly mixed with the pollution particles). We continue this sequence of shock waves until time step $t = 1000$ presented in the third panel in Figures 5-6. At this moment, we can observe the maximum reduction of the cloud vapor. Then, we stop the sequence and observe the behavior of the cloud. The "hole" in clouds remains there for another 2000 time steps (two times longer than the generation of a sequence of shock waves) until the time moment $t = 3000$ illustrated in the fourth panel. At this moment, the neighboring cloud vapor particles return slowly to the center by the diffusion mechanism. This is illustrated in the fifth and sixth panels (time moments $t = 4000$ and $t = 6000$) in Figures 5-6. We conclude that this local water vapor reduction maintains if the cannon creates shocking waves over a repeated period and stays for a significant period. The repetition of the shock waves for a long time is a critical phenomenon for reducing water vapor concentration.

5 Conclusion and future work

We claim that the numerical results presented in this paper are the first three-dimensional simulational verification of the hail cannon. We showed that repeating a sequence of shock waves for a long time significantly reduces cloud vapor. Our model is simple and may require several improvements, like computing the real shape of the cannon force modeled by the advection. This will require a solution of the three-dimensional Navier-Stokes equations. This computed field can be introduced into the advection-diffusion model. We can also include different cloud components and the reaction terms between them. The thermal effects that consider the additional movement of the cloud particles can also be incorporated by considering the Navier-Stokes-Boussinesq model. Nevertheless,

the presented numerical results are very interesting, and they also confirm our experimental findings.

Acknowledgement

The Authors are thankful for support from the funds assigned to AGH University of Science and Technology by the Polish Ministry of Science and Higher Education. The work of Albert Oliver-Serra is supported by the “Ayudas para la recualificación del sistema universitario español” grant funded by the ULPGC, the Ministry of Universities by Order UNI/501/2021 of 26 May, and the European Union-Next Generation EU Funds.

References

1. Height of the mixed layer in the stably stratified planetary boundary layer. In: Frenkiel, F., Munn, R. (eds.) *Turbulent Diffusion in Environmental Pollution, Advances in Geophysics*, vol. 18, pp. 73–92. Elsevier (1975)
2. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: *LAPACK Users’ guide*, vol. 9. Siam (1999)
3. Benson, D.J., Bazilevs, Y., Luycker, E.D., Hsu, M.C., Scott, M., Hughes, T.J.R., Belytschko, T.: How it works. A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to X-FEM **83**, 765–785 (2010)
4. Calo, V., Brasher, N., Bazilevs, Y., Hughes, T.: Multiphysics model for blood flow and drug transport with application to patient-specific coronary artery flow. *Computational Mechanics* **43**(1), 161–177 (2008)
5. Chang, K., Hughes, T., Calo, V.: Isogeometric variational multiscale large-eddy simulation of fully-developed turbulent flow over a wavy wall. *Computers & Fluids* **68**, 94–104 (2012)
6. Changnon, S.A., Ivens, J.L.: History repeated: The forgotten hail cannons of Europe. *Bulletin of the American Meteorological Society* **62**(3), 368–375 (1981)
7. Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y.: *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons (2009)
8. Curran, J.: Vt. Orchard wakes the neighbors with hail cannon (2018)
9. Dedè, L., Borden, M.J., Hughes, T.: A user’s view of solving stiff ordinary differential equations. *SIAM review* **21**(1), 1–17 (1979)
10. Dede, L., Hughes, T., Lipton, S., Calo, V.: Structural topology optimization with isogeometric analysis in a phase field approach. In: *16th US National Conference on Theoretical and Applied Mechanics*. pp. 1–2 (2010)
11. Duddu, R., Lavier, L.L., Hughes, T.J., Calo, V.M.: A finite strain eulerian formulation for compressible and nearly incompressible hyperelasticity using high-order b-spline finite elements. *International Journal for Numerical Methods in Engineering* **89**(6), 762–785 (2012)
12. Egger, M.: How it works. Hail Cannon Manufacturer (2022)
13. Gomez, H., Calo, V.M., Bazilevs, Y., Hughes, T.J.: Isogeometric analysis of the cahn–hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering* **197**(49-50), 4333–4352 (2008)

14. Gomez, H., Hughes, T.J., Nogueira, X., Calo, V.M.: Isogeometric analysis of the isothermal navier–stokes–korteweg equations. *Computer Methods in Applied Mechanics and Engineering* **199**(25-28), 1828–1840 (2010)
15. Hossain, S.S., Hossainy, S.F., Bazilevs, Y., Calo, V.M., Hughes, T.J.: Mathematical modeling of coupled drug and drug-encapsulated nanoparticle transport in patient-specific coronary artery walls. *Computational Mechanics* **49**(2), 213–242 (2012)
16. Hsu, M.C., Akkerman, I., Bazilevs, Y.: High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids* **49**(1), 93–100 (2011)
17. Inopower: Hail cannon user’s manual (2009)
18. Lamb, R., Durran, D.: Eddy diffusivities derived from a numerical model of the convective planetary boundary layer. *Il Nuovo Cimento C* **1**, 1–17 (1978)
19. Łoś, M., Klusek, A., Hassaan, M.A., Pingali, K., Dzwiniel, W., Paszyński, M.: Parallel fast isogeometric l2 projection solver with galois system for 3d tumor growth simulations. *Computer Methods in Applied Mechanics and Engineering* **343**, 1–22 (2019)
20. Łoś, M., Paszyński, M.: Stability of non-linear flow in heterogeneous porous media simulations using higher order and continuity basis functions. *Journal of Computational and Applied Mathematics* p. 115141 (2023)
21. Łoś, M., Paszyński, M., Klusek, A., Dzwiniel, W.: Application of fast isogeometric l2 projection solver for tumor growth simulations. *Computer Methods in Applied Mechanics and Engineering* **316**, 1257–1269 (2017)
22. Łoś, M., Woźniak, M., Muga, I., Paszyński, M.: Three-dimensional simulations of the airborne covid-19 pathogens using the advection-diffusion model and alternating-directions implicit solver. *Bulletin of the Polish Academy of Sciences: Technical Sciences* **69**(4), e137125 (2021)
23. Łoś, M., Woźniak, M., Paszyński, M., Dalcin, L., Calo, V.M.: Dynamics with matrices possessing Kronecker product structure. *Procedia Computer Science* **51**, 286–295 (2015)
24. Łoś, M., Woźniak, M., Paszyński, M., Lenharth, A., Hassaan, M.A., Pingali, K.: Iga-ads: isogeometric analysis finite element method using alternating directions solver. *Computer Physics Communications* **217**, 99–116 (2017)
25. Paszyński, M., Siwik, L., Dzwiniel, W., Pingali, K.: Supermodeling, a convergent data assimilation meta-procedure used in simulation of tumor progression. *Computers and Mathematics with Applications* **113**(C), 214–224 (2022)
26. Pingali, K., Nguyen, D., Kulkarni, M., Burtscher, M., Hassaan, M.A., Kaleem, R., Lee, T.H., Lenharth, A., Manevich, R., Méndez-Lojo, M., Proutzos, D., Sui, X.: The tao of parallelism in algorithms. *SIGPLAN Not.* **46**(6), 12–25 (Jun 2011)
27. Plumandon, J.R.: Troisieme congres international de defense contre la grele (in French). *Ciel et Terre* **22**, 457–471 (1902)
28. Schroeder, W., Martin, K., Lorensen, B.: *The Visualization Toolkit* (4th ed.), vol. 9. ISBN 978-1-930934-19-1 (2006)
29. Shir, C.C.: A preliminary numerical study of atmospheric turbulent flows in the idealized planetary boundary layer. *Journal of the Atmospheric Sciences* **30**(7), 1327–1339 (1973)
30. Wieringa, J., Holleman, I.: If cannons cannot fight hail, what else? *Meteorologische Zeitschrift* **15**(6), 659–669 (2006)