Fast Isogeometric Analysis Simulations of a Process of Air Pollution Removal by Artificially Generated Shock Waves

Krzysztof Misan, Weronika Ormaniec, Adam Kania, Maciej Kozieja,
Marcin Łoś ⓘ, Dominik Gryboś ⓘ, Jacek Leszczyński ⓘ,
and Maciej Paszyński$^{(\boxtimes)}$ ⓘ

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

Abstract. Large concentrations of particulate matter in residential areas are related to the lack of vertical movements of air masses. Their disappearance is associated with the occurrence of the most common ground temperature inversion, which inhibits the natural air convection. As a result, air layers separated by a temperature inversion layer are formed, which practically do not interact with each other. Therefore, to reduce the concentration of particulate matter, mixing of air layers should be forced, or natural processes should be restored. For this purpose, it was proposed to generate shock waves of high pressure in the vertical direction to mix the polluted air and break the inversion layer locally. This paper performs fast isogeometric analysis simulations of the thermal inversion and pollution removal process. We employ a linear computational cost solver using Kronecker product-based factorization. We compare our numerical simulations to an experiment performed with an anti-hail cannon in a highly polluted city of Kraków, Poland.

Keywords: Thermal inversions · Atmoshperic pollution reduction methods · Advection-diffusion-reaction · Variational splitting · Isogeometric analysis

1 Introduction

One of the main challenges of human civilization is air pollution, which is an environmental, social, and economic problem. According to WHO data, almost 90% of people live in poor air quality, which causes over 4 million deaths annually [19]. Air pollution adversely affects human health, and causes such ailments as asthma, respiratory tract infections, and heart attacks [22]. Air pollution can be divided according to the type of harmful substances that cause this phenomenon: smog (particulate matter) and photochemical smog (ozone). Smog is a suspension of aerosol particles smaller than 10 $\frac{\mu g}{m^3}$ or 2.5 $\frac{\mu g}{m^3}$. Usually, it is created in the autumn and winter months and is associated with low emissions,

i.e., heating with solid fuels in households. As a result of unfavorable meteorological conditions and the formation of ground inversion, pollutants get stuck in the living zone, reaching dangerous concentrations. The phenomenon is particularly enhanced in unfavorable terrain features, such as valleys or basins [9,15]. Due to the lack of commercial methods of removing particulate matter in open spaces, only a change in weather conditions, i.e., the disappearance of temperature inversion, rain or snowfall or wind, causes the pollution to disperse in the entire atmosphere and/or fall to the ground. As a result of the formation of inversion layers, the vertical air movement between the layers of the atmosphere above and below this layer disappears. By introducing forced mixing of the layers, it can thin out or even pierce the inversion layer. This will cause at least a temporary and partial natural mixing of the air layers and a decrease in air pollution. The method presented by Leszczynski et al. (2020) [14], uses a shock wave generator which, by producing an impact, mixes the layers of air in the atmosphere. The idea of using shock waves to mix and lift polluted air upwards is schematically shown in Fig. 1.
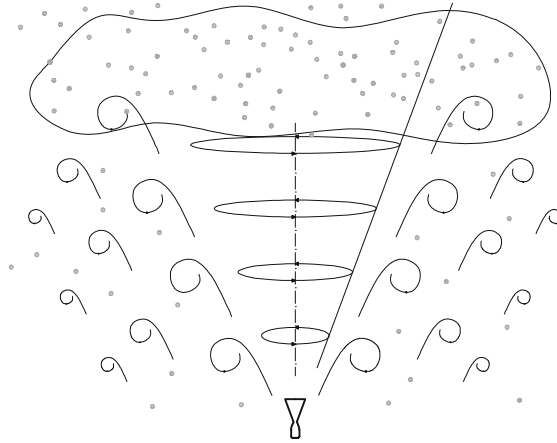
**Fig. 1.** The idea of using shock waves to mix and lift polluted air.

In this paper, we propose a numerical verification of the process of air pollution removal by artificially generated shock waves. We employ fast isogeometric analysis [17,18] simulations. Our solver is based on the Kronecker product decomposition of the system of linear equations.

We employ the advection-diffusion-reaction model in a weak form. We first discretize in space using finite elements and higher-order B-spline basis functions employed by isogeometric analysis (IGA) [5]. Then, we employ the explicit time marching scheme for discretization in time. This method requires factorization of the mass matrix built from higher-order B-spline basis functions spans over tensor product grid. The mass matrix from the explicit scheme can be factorized

in a linear computational cost resulting from the Kronecker product decomposition [7,8,16,17]. We factorize the matrix in the first step and then forward and backward substitute in the following steps. Using our solver, we first simulate the effect of thermal inversion on the formulation of clouds with the pollution particles. Next, we show how we can locally force mixing of the air layers using the artificially generated shock waves, resulting in a local decrease of the concentration of particles. We compare our numerical simulations to an experiment performed with a cannon in a highly polluted city of Kraków, Poland.

2 Experimental Verification

The method proposed by Leszczynski et al. (2020) [14], using a generation of high-pressure vertical shock waves, introduces the propagating disturbance of the air medium, which causes the air with particulate matter to be lifted out. Moreover, it generates air vortices in the adjacent volume, which causes the horizontal mixing of the air layers. The method is based on an intervention reduction of the suspended dust concentration when the standards are exceeded. The Inopower anti-hail cannon [11], shown in Fig. 2 was used in the conducted technological tests. The device consists of a container (1) with dimensions $6.00 \times 2.45 \times 2.60$ in which is the combustion chamber (4) with the volume of $150 \, dm^3$, three fuel inlet ports (5), and a control cabinet (6). The shock wave produced by the ignition of an acetylene-air mixture is directed vertically upward through the widening conical outlet tube (2). There is a bundle of acetylene cylinders on the other side of the container with gas pressure reduction installation (3). After detecting
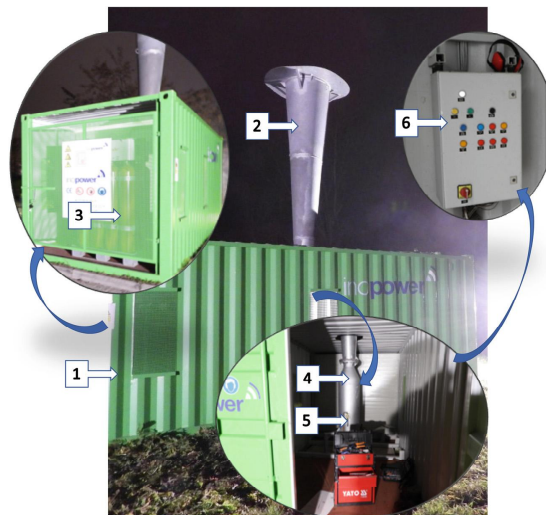
Fig. 2. Shock wave generator. 1 - Container, 2 - Outlet tube, 3 - Bundle of acetylene cylinders, 4 - Combustion chamber, 5 - Fuel inlet ports, 6 - Control cabinet.

unfavorable air pollution forecasts, intervention actions were taken to end the unfavorable stagnation of the atmosphere. During the tests, the acetylene gauge pressure supplied to the combustion chamber was 2.9 bar. During the explosion, the acetylene-air mixture reaches a pressure of about 1 MPa. The frequency of generating the shock wave was 0.17 Hz, and the total operating time was about 30 min. Temperature, pressure, humidity, and particulate matter concentration in the vertical profile were measured using the equipment placed on the drone DJI Matrice 200 V2. For each test, a flight was performed immediately before and after the generator was started and 20 min after its completion. Additionally, stationary Airly PM sensors measured dust concentrations at 0.1 km, 1.5 km, and 2 km from the generator. The preliminary results are shown in two papers: Jedrzejek et al. (2021) [12] and Jedrzejek et al. (2021) [13]. Figure 3 shows the measurement data of the altitude profile from 0 m to 130 m for the temperature and PM2.5 concentration for the situation before and after the generation of shock waves. The altitude profile of the temperature before the generation of the shock waves indicates a temperature inversion from 90 m to 130 m, where the temperature change was $0.38 \frac{^{o}\mathrm{C}}{100\,\mathrm{m}}$. However, after using the cannon, the temperature change in a given height range decreased to about $-0.6 \frac{^{o}\mathrm{C}}{100\,\mathrm{m}}$. Which is a typical humid-adiabatic gradient value for the atmosphere. Moreover, the monitored concentration of particulate matter PM2.5 decreased by up to 50% at specific points, e.g. at 120 m. The average concentration of PM2.5 from 0 to 120 m decreased from $85.1 \frac{\mu\mathrm{g}}{\mathrm{m}^3}$ to $61.9 \frac{\mu\mathrm{g}}{\mathrm{m}^3}$.
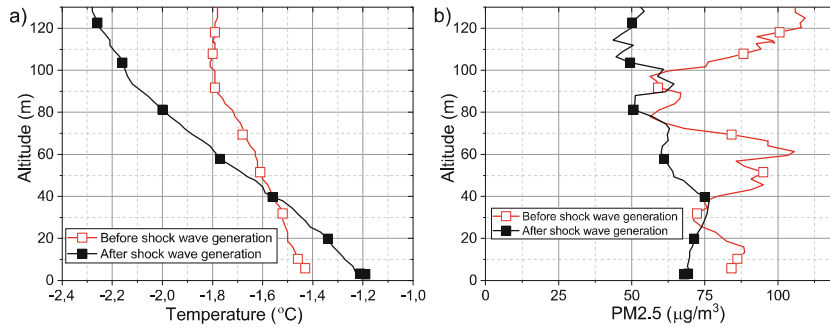
Fig. 3. Measurement data of the altitude profile: a) Temperature; b) PM2.5 concentration.

3   Numerical Simulations

We employ advection-diffusion-reaction equations to model the concentration of the water vapor forming a cloud, mixed with the pollution particles.

The equations in the strong form are

$$\frac{\partial u}{\partial t} + (b \cdot \nabla)u - \nabla \cdot (K\nabla u) + cu = f \text{ in } \Omega \times (0, T] \tag{1}$$

$$\nabla u \cdot n = 0 \text{ in } \Omega \times (0, T] \tag{2}$$

$$u = u_0 \text{ in } \Omega \times 0 \tag{3}$$

where u is the concentration scalar field, b is the assumed wind velocity vector field, $K = \begin{pmatrix} K_{11} 0 \\ 0 K_{22} \end{pmatrix}$ is the diffusion matrix, c is the reaction parameter, and f is the source term.

We employ the explicit method formulation

$$\frac{u^{t+1} - u^t}{dt} = \nabla \cdot (K\nabla u^t) - (b \cdot \nabla)u^t + cu^t = f^t \tag{4}$$

we derive the weak formulation, testing again with B-spline basis functions

$$\left(u^{t+1}, v \right) = \left(u^t, v \right) - dt \left(K\nabla u^t, \nabla v \right) - dt \left(b \cdot \nabla u^t, v \right) + \left(cu^t + f^t, v \right) \forall v \in V \tag{5}$$

We discretize with B-spline basis functions defined over the square domain $\Omega = [0, 1]^2$

$$u^{t+1} = \sum_{i=1,...,N_x; j=1,...,N_y} u_{ij}^{t+1} B_i^x B_j^y; \quad u^t = \sum_{i=1,...,N_x; j=1,...,N_y} u_{ij}^t B_i^x B_j^y \tag{6}$$

and we test with B-spline basis functions

$$
\begin{aligned}
\sum_{ij} u_{ij}^{t+1} \left(B_i^x B_j^y, B_k^x B_l^y \right) =& \sum_{ij} u_{ij}^t \left(B_i^x B_j^y, B_k^x B_l^y \right) \\
&- dt \sum_{ij} u_{ij}^t \left(K\frac{\partial B_i^x}{\partial x} B_j^y, \frac{\partial B_k^x}{\partial x} B_l^y \right) \\
&- dt \sum_{ij} u_{ij}^t \left(K B_i^x \frac{\partial B_j^y}{\partial y}, B_k^x \frac{\partial B_l^y}{\partial y} \right) \\
&- dt \sum_{ij} u_{ij}^t \left(b\frac{\partial B_i^x}{\partial x} B_j^y, B_k^x B_l^y \right) \\
&+ dt \sum_{ij} u_{ij}^t \left(b B_i^x \frac{\partial B_j^y}{\partial y}, B_k^x B_l^y \right) \\
&+ \sum_{ij} u_{ij}^t \left(c B_i^x B_j^y, B_k^x B_l^y \right) + \left(f^t, B_k^x B_l^y \right) \\
& k = 1, ..., N_x; l = 1, ..., N_y
\end{aligned}
\tag{7}
$$

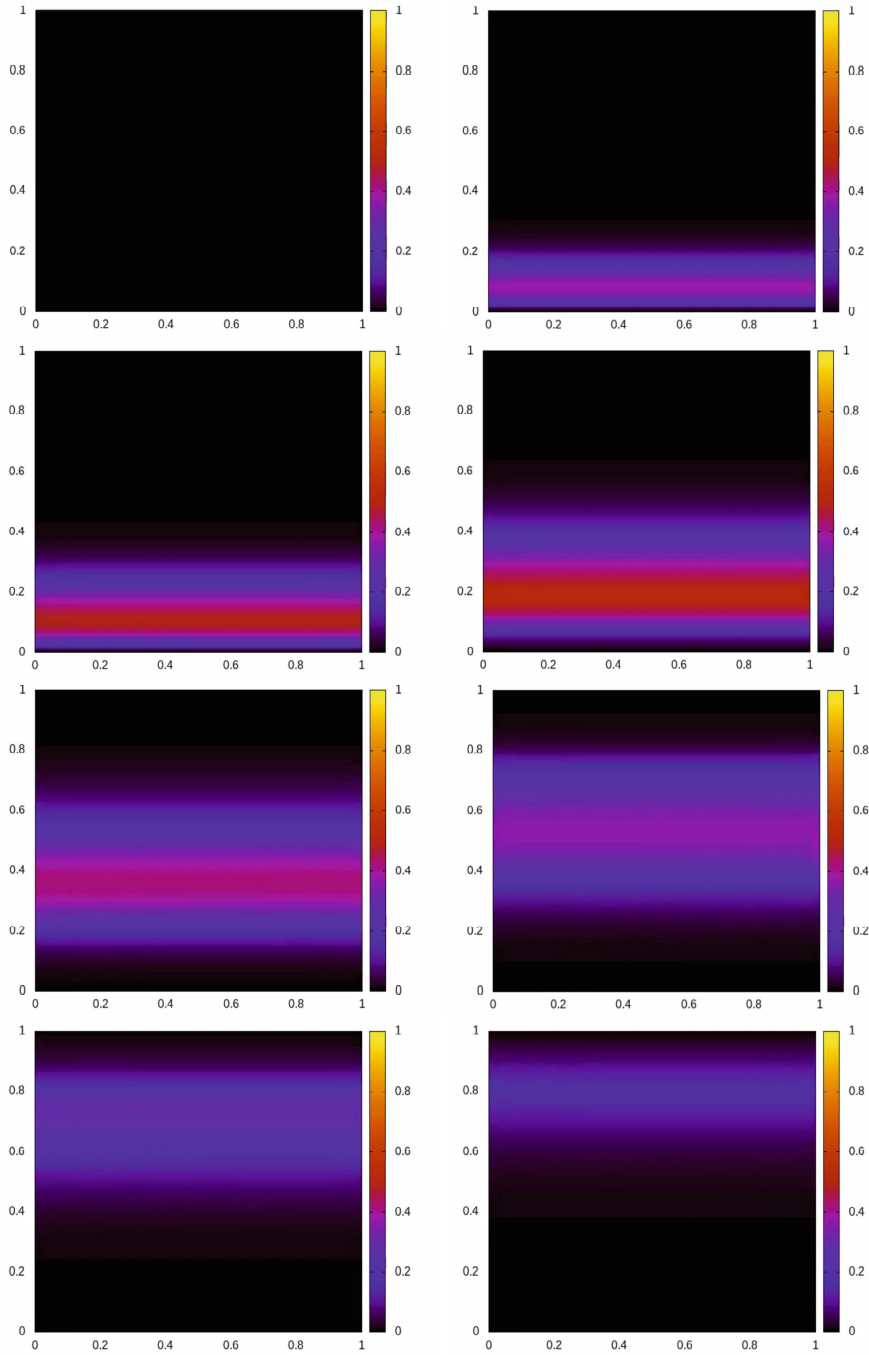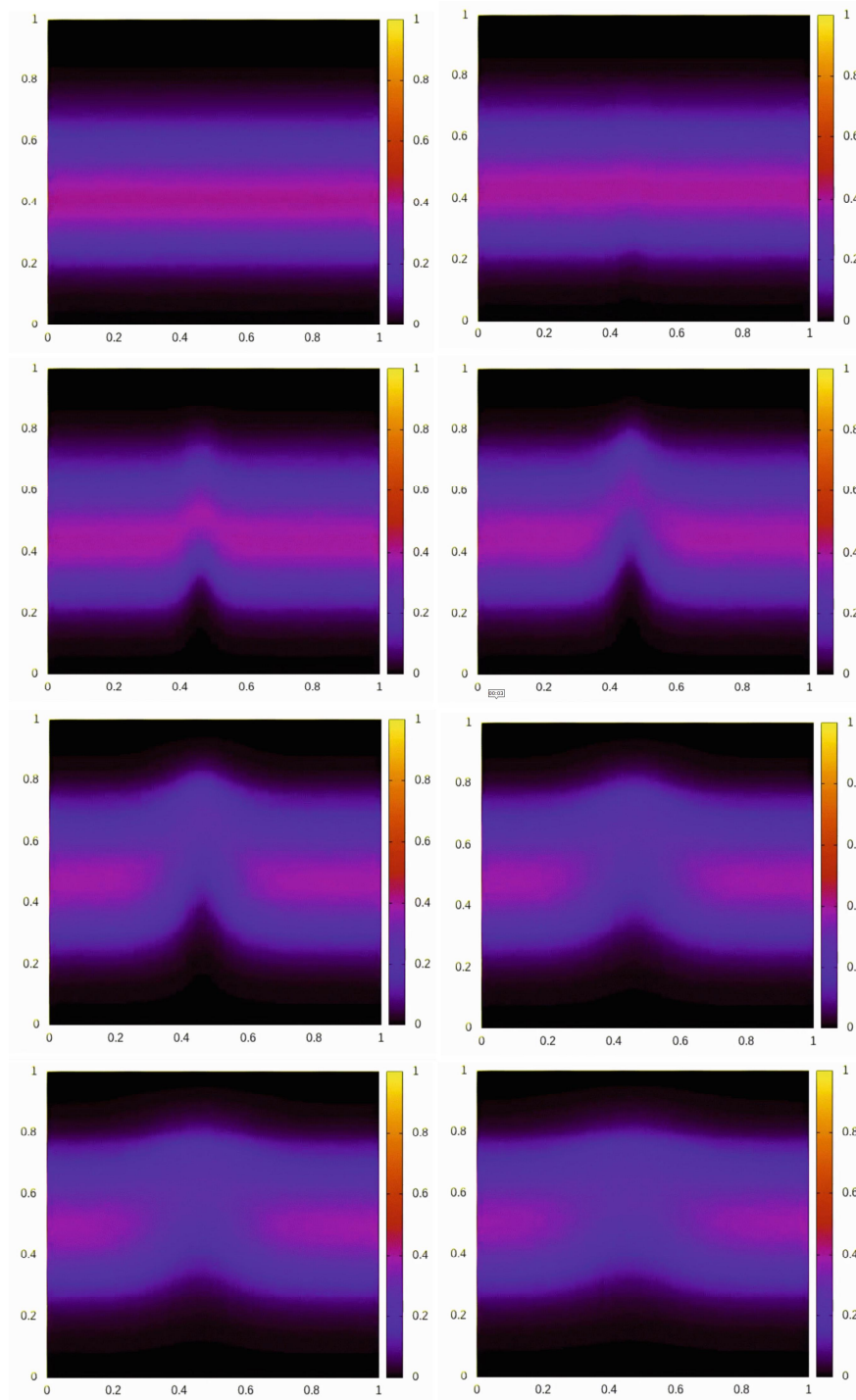where $(u, v) = \int_\Omega uv x dy$.

Fig. 4. Formulation of the cloud through termal inversion.

Fig. 5. Pollution reduction by generated shock waves.

We introduce

$$\mathbf{M}_x = \{(B_i^x, B_k^x)_x\}_{ik} = \{\int B_i^x B_k^x dx\}_{ik} \tag{8}$$

$$\mathbf{M}_y = \{(B_j^y, B_l^y)_y\}_{jl} = \{\int B_j^y B_l^y dy\}_{jl} \tag{9}$$

In general, Kronecker product matrix $\mathcal{M} = \mathcal{A}^x \otimes \mathcal{B}^y$ over 2D domain $\Omega = \Omega_x \times \Omega_y$ is defined as

$$\mathcal{M}_{ijkl} = \mathcal{A}_{ik}^x \mathcal{B}_{jl}^y. \tag{10}$$

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

$$(\mathcal{M})^{-1} = (\mathcal{A}^x \otimes \mathcal{B}^y)^{-1} = (\mathcal{A}^x)^{-1} \otimes (\mathcal{B}^y)^{-1} \tag{11}$$

we can solve our system in a linear computational cost.

3.1   Fast Simulation of Thermal Inversion and Cloud Formation

We concentrate first on the fast simulation of cloud formation and thermal inversion. In our simulation, the scalar field u represents the water vapor forming a cloud, mixed with the pollution particles. Following [1] we introduce the linear temperature decrease from 20 C at ground level to -30 C at the height 3/4 of the domain. The thermal inversion effect is obtained by introducing the advection field as a temperature gradient.

The equations in the strong form are

$$\frac{\partial u(x,y;t)}{\partial t} + \frac{\partial T(y;t)}{\partial y}\frac{\partial u(x,y;t)}{\partial y}$$

$$-K_x\frac{\partial^2 u(x,y;t)}{\partial x^2} - K_y\frac{\partial^2 u(x,y;t)}{\partial y^2} = f(x,y;t)$$

$$(x,y;t) \text{ in } \Omega \times (0,T] \tag{12}$$

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

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

where u is the concentration scalar field, where the advection is driven by the temperature gradient in the vertical direction

$$\frac{\partial T(y;t)}{\partial y} = \begin{cases} 0 \text{ for } y > 30 \ -5 \text{ for } y <= 30 \end{cases} \tag{15}$$

$K_x = 1.0$ is the horizontal diffusion, $K_y = 0.1$ is the vertical diffusion, and the source term

$$f(x,y;t) = \begin{cases} -25y \text{ for } y < 5, t < 5000 * dt \ 0 \text{ otherwise} \end{cases} \tag{16}$$

We employ the implementation of the linear computational cost Kronecker product structure solver as described in [18]. The explicit method formulation is implemented in the **compute_rhs** routine

```
void compute_rhs(int iter)
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 gradient_prod = 1.*u.dx * v.dx + 0.1*u.dy * v.dy;
13         double val = u.val * v.val - steps.dt * gradient_prod+
               steps.dt*Tprim(e[1])*u.dy*v.val+
               steps.dt*f(e[1],iter)*v.val*50.;
14         U(aa[0], aa[1]) += val * w * J;
15      }
16   }
17   update_global_rhs(rhs, U, e);
18}
```

where the temperature gradient

```
void Tprim(int iter, int y)
1 if(y>30)
2   return 0.0;
3 return -5.;
```

The cloud formation is modeled by the right-hand side term, that is active in first 5000 steps

```
void f(int iter, int y)
1 if(y<5 && iter<5000 )
2   return (-y/5.);
3 return 0.;
```

The simulation results are summarized in Fig. 1. The water vapor formulated first at the ground level is driven by the temperature gradient.

3.2   Fast Simulation of the Pollution Removal by Artificially Generated Shock Waves

In order to simulate the atmospheric cannon, we modify the advection function in a sepparable way as

$$cannon(x, z) = const * (1 - z) * sin(10 * \pi * x) * max(0, sin(\pi * t/10)) \quad (17)$$

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 $(1 - z) * sin(10 * \pi * x)$ that runs in time like $max(0, sin(\pi * t/10))$.

```
  double cannon(double x, double y,int iter) {
1 x=x/40.; y=y/40.;
2 double t=iter;
3 if(x>0.3 && x<0.6 &&t>10000)
4   return 200.*(1.-y)*max(sin(10*PI*x),0)*
      max(0,sin(PI*(t-8000)/1000));
5 else
6   return 0.;
```

We add this cannon function to the formulation

```
void compute_rhs(int iter)
  ...
13      double val = u.val * v.val - steps.dt * gradient_prod+
          steps.dt*(delta_T(e[1])-
          cannon(e[0],e[1],iter))*u.dy*v.val+
          steps.dt*f(e[1])*v.val*50.;
  ...
```

The simulation results are summarized in Fig. 2. We can read from this simulation that shooting a cannon results in a local mixing of the layers and reduction of the water vapor mixed with the pollution particles. This local pollution and water vapor reduction maintain if the cannon creates shocking waves in a repeated period of time.

4   Comparison of Higher-Order and Continuity Isogeometric Finite Element Method with Finite Difference Method

Our simulator employs higher-order and continuity B-spline basis functions. This requires computing the integrals with higher-order B-spline basis functions on the right-hand side. We compute them element-wise:

$$\mathcal{I}_1 = \int_E B_i^x(x) B_j^y(y) B_k^x(x) B_l^y(y) dx dy$$

$$= \int_{[0,1]^2} B_i^x(x) B_j^y(y) B_k^x(x) B_l^y(y) J_E(x,y) dx dy$$

$$= \sum_q w_q B_i^x(x_q) B_j^y(y_q) B_k^x(x_q) B_l^y(y_q) J_E(x_q, y_q)$$

$$\mathcal{I}_2 = \int_E K(x,y) \frac{\partial B_i^x(x)}{\partial x} B_j^y(y) \frac{\partial B_k^x(x)}{\partial x} B_l^y(y) dx dy$$

$$= \int_{[0,1]^2} K(x,y) \frac{\partial B_i^x(x)}{\partial x} B_j^y(y) \frac{\partial B_k^x(x)}{\partial x} B_l^y(y) J_E(x,y) dx dy$$

$$= \sum_q w_q K(x_q, y_q) \frac{\partial B_i^x(x_q)}{\partial x} B_j^y(y_q) \frac{\partial B_k^x(x_q)}{\partial x} B_l^y(y_q) J_E(x_q, y_q)$$

$$\mathcal{I}_3 = \int_E K(x,y) B_i^x(x) \frac{\partial B_j^y(y)}{\partial y} B_k^x(x) \frac{\partial B_l^y(y)}{\partial y} dx dy$$

$$= \int_{[0,1]^2} K(x,y) B_i^x(x) \frac{\partial B_j^y(y)}{\partial y} B_k^x(x) \frac{\partial B_l^y(y) J_E(x,y)}{\partial y} dx dy$$

$$= \sum_q w_q K(x_q, y_q) B_i^x(x_q) \frac{\partial B_j^y(y_q)}{\partial y} B_k^x(x_q) \frac{\partial B_l^y(y_q)}{\partial y} J_E(x_q, y_q)$$

$$\text{(18)}$$

$$\mathcal{I}_4 = \int_E b_x(x,y) \frac{\partial B_i^x(x)}{\partial x} B_j^y(y) B_k^x(x) B_l^y(y) dx dy$$

$$= \int_{[0,1]^2} b_x(x,y) \frac{\partial B_i^x(x)}{\partial x} B_j^y(y) B_k^x(x) B_l^y(y) J_E(x,y) dx dy$$

$$= \sum_q w_q b_x(x_q, y_q) \frac{\partial B_i^x(x_q)}{\partial x} B_j^y(y_q) B_k^x(x_q) B_l^y(y_q) J_E(x_q, y_q)$$

$$\mathcal{I}_5 = \int_E b_y(x,y) B_i^x(x) \frac{\partial B_j^y(y)}{\partial y} B_k^x(x) B_l^y(y) dx dy \partial y$$

$$= \int_{[0,1]^2} b_y(x,y) B_i^x(x) \frac{\partial B_j^y(y)}{\partial y} B_k^x(x) B_l^y(y) J_E(x,y) dx dy$$

$$= \sum_q w_q b_y(x_q, y_q) B_i^x(x_q) \frac{\partial B_j^y(y_q)}{\partial y} B_k^x(x_q) B_l^y(y_q) J_E(x_q, y_q)$$

$$\mathcal{I}_6 = \int_E f(x,y) B_k^x(x) B_l^y(y) dx dy$$

$$= \int_{[0,1]^2} f(x,y) B_k^x(x) B_l^y(y) J_E(x,y) dx dy$$

$$= \sum_q w_q f(x_q, y_q) B_k^x(x_q) B_l^y(y_q) J_E(x_q, y_q)$$

We change the variables into the reference $[0,1]^2$ element and introduce the Jacobian of the element transformation $J_E(x,y)$. We employ $q = 1, ..., N_q$ quadrature points (x_q, y_q) and weights w_q to compute the integrals, where $N_q \approx \mathcal{O}(p^2)$. Finally, we collect the integrals into a global right-hand-side vector according to (7).

The mass matrix on the left-hand side is a Kronecker product $\mathcal{M} = \mathcal{M}_x \otimes \mathcal{M}_y$ of matrices with one-dimensional B-spline basis functions.

$$\{\mathcal{M}_x\}_{i,k} = \int_\Omega B_i^x(x) B_k^x(x) dx, \quad \{\mathcal{M}_y\}_{j,l} = \int_\Omega B_j^y(y) B_l^y(x) dy \qquad (19)$$

These matrices are $2p+1$ diagonal, because of the overlap of the one-dimensional B-splines. Thus, the factorization cost is $\mathcal{O}(Np^3)$.

The alternative finite-difference formulation approximates the derivates in the strong equation using the point-wise values

$$u_{i,j}^{t+1} = u_{i,j}^t - dtK_{i,j}\frac{u_{i-1,j}^t - 2u_{i,j}^t + u_{i+1,j}^t}{h} - dtK_{i,j}\frac{u_{i,j-1}^t - 2u_{i,j}^t + u_{i,j+1}^t}{h}$$
$$+ dtb_{i,j}^x\frac{u_{i,j}^t - u_{i-1,j}^t}{h} - +dtb_{i,j}^y\frac{u_{i,j}^t - u_{i,j-1}^t}{h} + dtf_{i,j}^t \qquad (20)$$

where h stands for the horizontal or vertical distance between grid points, $u_{i,j}^t$ stands for the concentrations at point (i,j), and $K_{i,j}, (b_{i,j}^x, b_{i,j}^y), f_{i,j}^t$ stand for the diffusion, advection, and forcing values estimated at the grid points. In the explicit method of the finite difference there are no matrices to factorize, there is only an $\mathcal{O}(N)$ update operation.

The advantage of the isogeometric analysis method is that we obtain a global smooth solution that is globally differentiable. The finite difference method gives us pointwise values.

The disadvantage of the isogeometric analysis is that we have to compute the integrals with higher order polynomials, which is $\mathcal{O}(p^2)$ more expensive than generating and computing pointwise values. We also have to factorize the mass matrix, which is $\mathcal{O}(p^3)$ more expensive than an update operation in the explicit method in finite differences.

5   Conclusion

The article presents a numerical verification of the experimental method of intervention reduction of particulate matter using shock waves generated by the combustion of the acetylene air mixture. The shock wave cycle restores the natural movements of air masses by temporarily acting on the inversion layer. We employed a fast isogeometric solver to model the thermal inversion, cloud formation, and pollution removal by artificially created shock waves. The numerical experiments confirm the experimental effect of mixing of the air layers and local decrease of the pollution and water vapor in the neighborhood of the created shock waves. Future work will include generalizing the developed simulational

software to three dimensions. We also plan parallelization into shared memory Linux cluster nodes [4,10]. Additionally, we plan to employ inverse algorithms [2,3,6,21] to solve several related inverse problems. They are related to the following open research questions: what is the optimal location of the cannon in a given area, what is the optimal frequency and power of the cannon, and what is the best moment during the cloud and pollution formation process to start the process. A detailed comparison of the numerical accuracy, convergence, and execution times of the isogeometric analysis and alternative methods can be a subject of a future study. The future work may also include incorporation of mesh adaptation techniques [20].

References

1. U.s. standard atmosphere vs. altitude. (2003). https://www.engineeringtoolbox.com/standard-atmosphere-d_604.html. Accessed 21 Dec 2021
2. Barabasz, B., Gajda-Zagórska, E., Migórski, S., Paszyński, M., Schaefer, R., Smołka, M.: A hybrid algorithm for solving inverse problems in elasticity. Appl. Math. Comput. Sci. **24**(4), 865–886 (2014)
3. Barabasz, B., Migórski, S., Schaefer, R., Paszyński, M.: Multi-deme, twin adaptive strategy hp-HGS. Inverse Prob. Sci. Eng. **19**(1), 3–16 (2011)
4. Calo, V., Collier, N., Pardo, D., Paszyński, M.: Computational complexity and memory usage for multi-frontal direct solvers used in p finite element analysis. Procedia Comput. Sci. **4**, 1854–1861 (2011)
5. Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y.: Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley & Sons (2009)
6. Gajda-Zagórska, E., Schaefer, R., Smołka, M., Paszyński, M., Pardo, D.: A hybrid method for inversion of 3D dc resistivity logging measurements. Natl. Comput. **14**(3), 355–374 (2015)
7. Gao, L., Calo, V.M.: Fast isogeometric solvers for explicit dynamics. Comput. Meth. Appl. Mech. Eng. **274**, 19–41 (2014)
8. Gao, L., Calo, V.M.: Preconditioners based on the alternating-direction-implicit algorithm for the 2D steady-state diffusion equation with orthotropic heterogeneous coefficients. J. Comput. Appl. Math. **273**, 274–295 (2015)
9. Giemsa, E., Soentgen, J., Kusch, T., Beck, C., Munkel, C., Cyrys, J., Pitz, M.: Influence of local sources and meteorological parameters on the spatial and temporal distribution of ultrafine particles in Augsburg, Germany. Front. Environ. Sci **8** (2021)
10. Goik, D., Jopek, K., Paszyński, M., Lenharth, A., Nguyen, D., Pingali, K.: Graph grammar based multi-thread multi-frontal direct solver with Galois scheduler. Procedia Comput. Sci. **29**(29), 960–969 (2014)
11. Inopower: Hail cannon user's manual (2009)
12. Jedrzejek, F., et al.: The innovative method of purifying polluted air in the region of an inversion layer. Front. Environ. Sci. **600** (2021)
13. Jedrzejek, F., et al.: An innovative method of reducing particulate matter concentration levels in the atmosphere. Laboratorium - Przeglàd Ogólnopolski nr 1 (2021). (in Polish)

14. Leszczyński, J., et al.: The method of reducing dust accumulation in the smog layer, which is the inversion layer. Munich, Germany, European Patent Office EP20217680 (2020)
15. Li, W.W., et al.: Analysis of temporal and spatial dichotomous pm air samples in the el paso-cd. juarez air quality basin. J. Air Waste Manag. Assoc. **51**, 1551–1560 (2001)
16. Łoś, M., Paszyński, M., Kłusek, A., Dzwinel, W.: Application of fast isogeometric l2 projection solver for tumor growth simulations. Comput. Meth. Appl. Mech. Eng. **316**, 1257–1269 (2017)
17. Łoś, M., Woźniak, M., Paszyński, M., Dalcin, L., Calo, V.M.: Dynamics with matrices possessing Kronecker product structure. Procedia Comput. Sci. **51**, 286–295 (2015)
18. Łoś, M., Woźniak, M., Paszyński, M., Lenharth, A., Hassaan, M.A., Pingali, K.: IGA-ads: isogeometric analysis fem using ads solver. Computer. Phys. Commun. **217**, 99–116 (2017)
19. Observatory, W.: Mortality from Household Air Pollution. World Health Organization, Luxembourg (2018)
20. Paszyńska, A., Paszyński, M., Grabska, E.: Graph transformations for modeling *hp*-adaptive finite element method with mixed triangular and rectangular elements. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2009. LNCS, vol. 5545, pp. 875–884. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01973-9_97
21. Paszyński, M., Barabasz, B., Schaefer, R.: Efficient adaptive strategy for solving inverse problems. Lect. Comput. Sci. **4487**, 342–354 (2007)
22. Sinha, J., Kumar, N.: Mortality and air pollution effects of air quality interventions in Delhi and Beijing. Front. Environ. Sci. **7**, 15 (2019)