Archives of Computational Methods in Engineering
https://doi.org/10.1007/s11831-024-10106-6

ORIGINAL PAPER

A Computational Framework for Precise Aerial Agricultural Spray Delivery Processes

J. O. Betancourt[1] · I. Li[1] · E. Mengi[1] · L. Corrales[1] · T. I. Zohdi[1]

Received: 12 January 2024 / Accepted: 5 March 2024

Abstract
As the world's population is expected to increase, so is the global demand for food. Sustainable intensification via precision agriculture of existing farms can increase crop production. Agricultural spray drones have recently taken a physical role within precision agriculture, such as aerial application of fluids, solids, and biological control agents but have difficulties spraying in uncontrolled environments caused by wind shifting spray material away from intended target areas. This work proposes an efficient physics-based framework to provide drone operators with trajectory and spray nozzle configuration for optimal target crop-dusting to mitigate spray drifts while providing quantitative approximations of spray particle trajectory and ground concentration. The framework is coupled with a machine-learning algorithm (MLA) to aid users in their search for optimal results and includes two decoupled models that simulate wind and spray particle trajectories. In the model problem, a genetic algorithm (GA) is used to optimize the system where the optimal trajectory and spray nozzle configuration resulted in 64% of crop targets hit while only losing minimal spray material from spray drifts.

Keywords Precision agriculture · Crop dusting · UAV · Spray drifts · Genetic algorithms

1 Introduction

The world's population is expected to be over 9 billion by 2050, causing global food demand to increase by 35–56% percent to maintain current levels of world hunger [1]. Reaching this level of food production in a short period of time can prove challenging, especially with climate change and global conflicts negatively affecting crop yields [2–4]. Several emerging agricultural techniques combat these issues such as indoor vertical farms placed within close proximity of large metropolitan areas [5]. With their effective land use, vertical farms have the benefit of growing produce within a climate-controlled environment, enabling them to grow crops all year round [5]. However, most commercial vertical farms are limited to growing leafy greens such as kale, arugula, and collard greens due to their economic viability [5, 6]. Therefore, traditional outdoor fields are still best for growing other crops like corn, wheat, and potatoes as well as resources used for grazing livestock [6].

Farmland expansion and sustainable intensification of current farms can increase crop production not suitable for vertical farming. Although farmland area expansion can increase food production relatively quickly, it also leads to more greenhouse gases from the loss of land-based carbon by converting natural ecosystems to agricultural lands [7]. In addition, farmland expansion is the leading cause of deforestation which negatively impacts the natural ecosystem's biodiversity [7]. These environmental effects inhibit the expansion of arable land. As such, there is an apparent need to increase the yield of currently existing farmland to improve the sustainability of crop production to meet the world's population growth.

Sustainable intensification includes the use of unmanned aerial vehicles (UAVs), also known as drones. Existing precision agriculture methods utilizing drones include remote aerial mapping through onboard cameras to track soil moisture levels, weed detection, and seedling emergence [8]. Proposed inexpensive nitrite sensors that can be widely distributed within a field have the potential to provide high spatial resolution data [9] which can be gathered by the use of drones scanning the sensor's RFID tag as it flies by [10, 11].

Drones have recently taken a more physical role within precision agriculture, such as aerial application of fluids,

✉ J. O. Betancourt
omarbetancourt@berkeley.edu

[1] Department of Mechanical Engineering, Etcheverry Hall, University of California, Berkeley, CA 94720-1740, USA

Published online: 12 April 2024

solids, and biological control agents [10]. Since 2015, when the Federal Aviation Administration approved the first crop-dusting drone [12], there has been a rapid development in the agricultural drone market with the potential to reduce the amount of irrigation runoff and occupational exposure to harmful chemicals by eliminating the need to manually crop-dust with backpack sprayers [13]. Research findings indicate that integrating UAVs into farming practices resulted in a significant boost of over $400 per hectare in farmers' revenue, concurrently streamlining the process of pesticide application by reducing time spent [14]. Among the more popular companies manufacturing these drones is DJI with their AGRAS series, capable of spraying 6 liters per second from each nozzle (Fig. 1). These drones also come equipped with onboard radar, lidar, and GPS for manual and autonomous operation [15] and have been used to create 3D digital models of agriculture fields [16].

The use of agricultural drone sprayers has been shown to require less than half the distance for the spray material to settle than the conventional mist blowers used in orchards [17]. However, crop-dusting drones may still have difficulties spraying in uncontrolled environments. 'Spray drift' is a natural phenomenon in which wind shifts spray material away from intended target areas [18]. Generally, higher wind velocities cause more spray drift, hindering the precision of the drone's aerial spray application. Even though an operatoris required to have a clear line of sight of their drone, they are typically a significant distance away. This makes it difficult to accurately judge where the spray material lands. Moreover, first-person observation of spray droplets provided by onboard cameras is only possible immediately after dispensing, and is nearly impossible to observe the distribution of spray material once it moves more than a foot from the onboard camera. Limited camera angles when the drone is far away and the hazard of contacting spray material when the drone is close, paved the way for autonomous drone spray technologies that utilize GPS and camera data to stay on course for pesticide and fertilizer spraying [19, 20].

Due to the relative novelty of employing spray drones, there are limited studies of their performance comparing their efficacy to ground sprayers and conventional crop-dusting aircraft [21–25]. The available literature on spray drone performance may even be contradictory due to significant variations in design parameters among the drones under investigation. Thus, this work proposes an efficient physics-based framework to provide drone operators with custom trajectory and spray nozzle configuration for optimal target crop-dusting while providing quantitative approximations of spray particle trajectory and ground concentration. While the drone is not autonomous, the generated trajectory will allow the operator to maneuver the drone to achieve optimal spray results. The framework is coupled with a machine-learning algorithm (MLA) to aid users in their search for optimal results.

2 Model Problem

The following section explains the framework's physics calculations for generating drone trajectories, particle trajectories, and wind gusts. The simulation includes two models:

- **PARTICLE MODEL**: Simulates the drone displacement and the spray droplet dynamics,
- **WIND MODEL**: Simulates the background transient wind flow field.

The wind model solves for the transient wind velocity field, which affects the particle model through the drag

Fig. 1 A crop-dusting drone spraying a field for precision agriculture. Image by DJI-Agras from pixabay.com

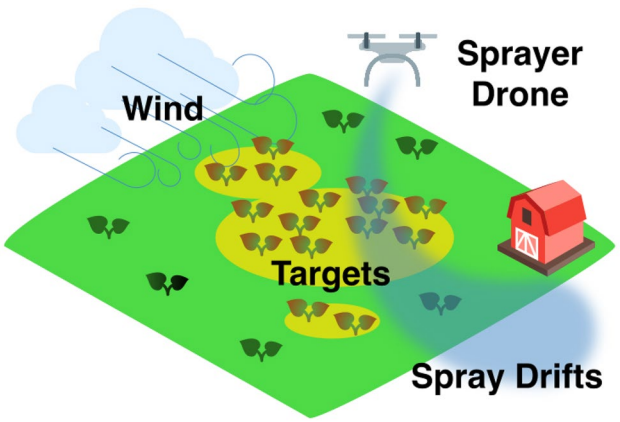


Fig. 2 A schematic of model problem (not to scale)

force induced on the spray droplets. However, the particle model is assumed to not affect the wind model, meaning the air density within the wind model does not include the particulate matter ejected by the drone. The model problem assumes the following (Fig. 2):

- Wind velocity fields are known spatially and temporally throughout the simulated domain.
- Drone propellers and ejected particles do not affect the surrounding wind velocity.
- All sprayed droplets are modeled as particle spheres of equal size. Particle size stays constant throughout time.
- The terrain is entirely flat.

The crop-dusting simulation is executed as follows:

1. **INITIALIZE WIND**: Set wind velocity at the boundaries and evolve the flow field over a user-specified time period or until steady-state is reached. This is done once at the beginning of the simulation before the drone starts to move and spray particles.
2. **SET DRONE AND TARGET LOCATIONS**: Load drone positions and target areas (crop locations) in the domain. Assign the drone design parameters for crop dusting.
3. **CALCULATE PARTICLE FORCES**: Spray particles eject from the nozzle (i.e., the drone's center position). Gravitational and drag forces from localized wind velocities drive the velocity of the spray particles for the next time step.
4. **STEP TIME**: Iterate the drone positions along a trajectory calculated by translational speed and angular velocity. Each spray particle will move with respect to the predetermined forces with an explicit time-stepping scheme.
5. **PERTURB WIND**: Boundary wind velocities are perturbed to simulate random 'gusts', causing the wind to be transient as the drone is moving and spraying material.
6. **TEST**: If a particle is within a voxel distance to a target, the target is considered hit and removed from the domain.
7. **ITERATE**: If unsprayed material remains, or if the maximum simulation time is not reached, loop to Step 3.

This model optimizes a drone's flight path and nozzle design using a Genetic Algorithm (GA) by minimizing a cost function through a non-derivative search.

$$\Pi = W_1 \frac{P_{outside}}{P_{total}} + W_2 \frac{T_{remaining}}{T_{total}} + W_3 \frac{t_{actual}}{t_{max}} \tag{1}$$

where P_{total} and T_{total} are the total number of spray particles and targets at the start of the simulation. t_{max} is the maximum amount of seconds each simulated run is allowed. After each completed simulated run of a drone spray, Eq. 1 will determine how well the drone's design parameters have performed by using the total number of particles that have drifted outside the domain $P_{outside}$, the number of targets remaining that were not hit by spray particles $T_{remaining}$ and the amount of time it took in simulated seconds, t_{actual}. The weights, W_1, W_2, and W_3, can be chosen by the drone operator to prioritize certain objectives.

2.1 Particle Model

2.1.1 Drone Flight Path

The drone's initial direction is defined by the normal vector, $\boldsymbol{n}_d(t = 0)$ at the start of the simulation. At each subsequent step, its movements are governed by rigid body kinematics using Forward-Euler integration to determine the spray nozzle's position, $\boldsymbol{r}_d$, at timestep $(t + \Delta t)$ by

$$\boldsymbol{r}_d(t + \Delta t) \approx \boldsymbol{r}_d(t) + \Delta t \boldsymbol{v}_d(t). \tag{2}$$

A constant angular velocity, $\boldsymbol{\omega}$, and a constant translational speed, V_d, are set in the initializing phase to control the drone's velocity, $\boldsymbol{v_d}$, by

$$\boldsymbol{v}_{d,translational}(t) = V_d \boldsymbol{n}_d, \tag{3}$$

$$\boldsymbol{v}_{d,rotational}(t) = \boldsymbol{r}_d \times \boldsymbol{\omega}, \tag{4}$$

$$\boldsymbol{v}_d(t) = \boldsymbol{v}_{d,translational}(t) + \boldsymbol{v}_{d,rotational}(t). \tag{5}$$

2.1.2 Particle Dynamics

The spray droplets are defined as spherical particles. The nozzle ejects each particle in a randomized direction within the bounds of a right circular cone-shaped region whose apex emanates from the drone's center position. This cone-shaped spray region will have a specified amplitude, A, with its normal vector, $\boldsymbol{n}_g = (0, 0, -1)$, being perpendicular to the ground. Specifically, the initial unit normal direction, $\boldsymbol{n}_i$, of each particle i in the system is given by

$$\boldsymbol{n}_i = \frac{A\boldsymbol{n}_r + \boldsymbol{n}_g}{||A\boldsymbol{n}_r + \boldsymbol{n}_g||}, \tag{6}$$

where $\boldsymbol{n}_r$ is a random unit vector, and A is the spray region amplitude as shown in Fig. 3.

The ejection speed, V_{eject}, is multiplied with $\boldsymbol{n}_i$ to obtain the particle's initial velocity from the drone.

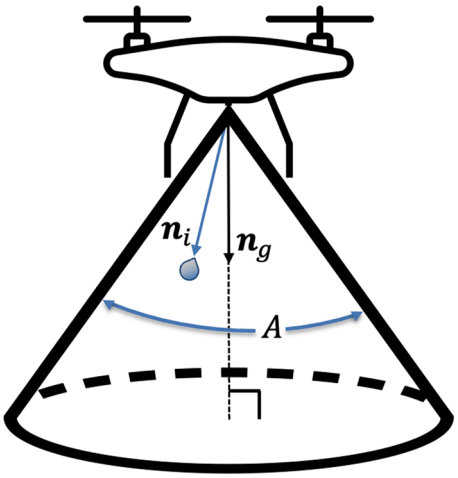


Fig. 3 Geometric representation of the cone-shaped spray region

$$\boldsymbol{v}_{i,0} = V_{eject}\boldsymbol{n}_i. \tag{7}$$

The total velocity of the particle once it is ejected from the drone is $\boldsymbol{v}_{i,o} + \boldsymbol{v}_d$. The drone sprays a total amount of material mass, M_{tot}, for each simulated run. For each particle with radius, r_p, the mass is calculated by

$$m = \frac{4}{3}\pi r_p^3 \rho_p, \tag{8}$$

where ρ_p is the density of the spray material. Therefore the total number of particles released by the drone is found by

$$N_p = \left\lfloor \frac{M_{tot}}{m} \right\rfloor. \tag{9}$$

The number of particles released per time step is calculated by

$$\frac{N_{p,ejected}}{\text{time step}} = \frac{3\eta\Delta t}{4\pi r_p^3}, \tag{10}$$

where the parameter, η, is the drop rate of particles in m^3/s, which will be optimized with a GA.

The airborne particles moving through space are only affected by drag, $\boldsymbol{F}_d$, and gravitational forces, $\boldsymbol{F}_g$. Therefore, the spray particles' equation of motion is given by

$$m\dot{\boldsymbol{v}}_i = \boldsymbol{F}_{d,i} + \boldsymbol{F}_{g,i} = \boldsymbol{F}_i^{tot}, \tag{11}$$

where $\dot{\boldsymbol{v}}_i$ is the time rate of change of particle i's velocity. Forward Euler integration of (11) respectively yields a particle's velocity and position at time $(t + \Delta t))$ to be

$$\boldsymbol{v}_i(t + \Delta t) \approx \boldsymbol{v}_i(t) + \frac{\Delta t}{m}\boldsymbol{F}_i^{tot}(t), \tag{12}$$

$$\boldsymbol{r}_i(t + \Delta t) \approx \boldsymbol{r}_i(t) + \Delta t\boldsymbol{v}_i(t). \tag{13}$$

Note that if the particle's height is less than 0 m (i.e., $r_{i,z}(t + \Delta t) < 0$), then its height is set to 0 m and is no longer tracked in the simulation.

Gravitational forces are constant on each airborne particle and are given as $\boldsymbol{F}_{g,i} = m\boldsymbol{g}$, where $\boldsymbol{g}$ is the gravitational constant at Earth's surface $(0, 0, -9.81)$ m/s^2. The drag force contribution to the motion of the particles is given as

$$\boldsymbol{F}_{d,i} = \tfrac{1}{2}\rho_a C_{d,i} A_x ||\boldsymbol{v}_a - \boldsymbol{v}_i||(\boldsymbol{v}_a - \boldsymbol{v}_i), \tag{14}$$

where ρ_a and $\boldsymbol{v}_a$ are the density and wind velocity of the surrounding air, respectively. A_x is the particle cross-sectional area given by $A_x = \pi r_p^2$, and $C_{d,i}$ is the drag force coefficient which is a function of the Reynolds number, Re. The drag force coefficient is calculated using the following piece-wise continuous function from [26]:

$$C_{d,i} = \begin{cases} \frac{24}{Re}, & 0 < Re \leq 1 \\ \frac{24}{Re^{0.646}}, & 1 < Re \leq 400 \\ 0.5, & 400 < Re \leq 3 \times 10^5 \\ 0.000366Re^{0.4275}, & 3 \times 10^5 < Re \leq 2 \times 10^6 \\ 0.18, & 2 \times 10^6 < Re \end{cases}. \tag{15}$$

2.2 Wind Model

The background transient wind flow is modeled as a *viscous incompressible fluid* and is numerically approximated using a finite difference scheme to solve the mass continuity and conservation of momentum equations. In this section, $\boldsymbol{v}$ only corresponds to the velocity field of the wind flow and ρ only to the air density, which should not be confused with the particle dynamics from the previous section. It is important to note that the wind is modeled separately from the particle dynamics. In this simulation, the wind velocity along the boundary is determined randomly, however, it could also be determined stochastically based on a historical range at the target farmland location. After solving for the initial velocity field prior to the drones' movement, the background wind velocity field is now considered to be transient and would only affect the trajectory of the droplets for the next few user-determined timesteps (e.g. the next 5 timesteps). Afterward, a new set of boundary wind velocities would be provided to solve for a new background transient wind velocity field; thus, it would introduce uncertainty within the simulations. In the context of this work, only the *direction* of boundary wind will be pertabated while its' magnitude will remain constant in order to capture the effects of random gusts. The details of the numerical implementation will be explained in Section 4.

2.2.1 Wind Flow Model

The background wind is modeled as a Newtonian viscous fluid. The Cauchy stress tensor for the fluid is defined as

$$\boldsymbol{T} = -p\boldsymbol{I} + 2\mu\boldsymbol{D}, \tag{16}$$

where p is pressure (an unknown field variable instead of a function of density due to the assumption of the incompressibility), μ is the kinematic viscosity coefficient, and $\boldsymbol{D}$ is the symmetric component of the spatial velocity gradient $\boldsymbol{L} = \nabla\boldsymbol{v}$. Therefore, the governing equation for the wind model is the Navier–Stokes equation for incompressible Newtonian viscous fluid (see Appendix for further discussion on the incompressibility assumption and its restrictions on the simulation parameters), which states

$$\nabla\cdot\boldsymbol{v} = 0, -\nabla p + \nabla\cdot(2\mu\boldsymbol{D}) + \rho\boldsymbol{b} = \rho\dot{\boldsymbol{v}}, \tag{17}$$

where $\boldsymbol{b}$ is the body force and $\nabla\cdot$ is the divergence operator. It is important to note that the kinematic viscosity coefficient is a function of space. Therefore, the $\nabla\cdot(2\mu\boldsymbol{D})$ term cannot be reduced into the more common expression, $\mu\nabla^2\boldsymbol{v}$, where ∇^2 is the Laplacian operator. For convenience, the viscous component of the Cauchy stress tensor will be referred to as $\boldsymbol{\tau} = 2\mu\boldsymbol{D}$.

2.2.2 Numerical Discretization of Wind Field

Further assumptions are made when discretizing the governing equation in (17). Due to the limited height in the simulation, the effect of pressure gradient is neglected. Moreover, the wind flow is modeled without body forces ($\boldsymbol{b} = 0$). With these two assumptions, the linear momentum balance equation from (17) can be simplified as

$$\rho\dot{\boldsymbol{v}} = \nabla\cdot\boldsymbol{\tau}. \tag{18}$$

After expanding the material time derivative of the velocity field, the linear momentum balance becomes

$$\rho\frac{\partial\boldsymbol{v}}{\partial t} = -\rho(\nabla\boldsymbol{v})\boldsymbol{v} + \nabla\cdot\boldsymbol{\tau}. \tag{19}$$

After temporal discretization of the linear momentum balance using Forward Euler ($\frac{\partial\boldsymbol{v}}{\partial t} \approx \frac{\boldsymbol{v}^{n+1}-\boldsymbol{v}^n}{\Delta t}$), the velocity at the $n+1$ timestep can be obtained by

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \Delta t\left(-(\nabla\boldsymbol{v}^n)\boldsymbol{v}^n + \frac{1}{\rho}(\nabla\cdot\boldsymbol{\tau}^n)\right). \tag{20}$$

The only terms remaining to be discretized are the velocity gradient and divergence of the viscous component of the stress tensor. Using finite difference methods for spatial discretization, the domain is discretized into uniform voxels with uniform lengths, h. Each voxel has its own time dependent velocity vector $\boldsymbol{v}(i,j,k) = (v_x(i,j,k), v_y(i,j,k), v_z(i,j,k))$ and kinematic viscosity $\mu(i,j,k)$, where the indices i, j, and k indicate the voxel position. The kinematic viscosity and velocity vector are assumed to be constant within each voxel. Recall the components of the velocity gradient are

$$\nabla\boldsymbol{v}^n = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{bmatrix}. \tag{21}$$

Using $\frac{\partial v_x}{\partial x}$ at the voxel location (i,j,k) as an example, it would be approximated as

$$\frac{\partial v_x}{\partial x} \approx \frac{v_x(i+h,j,k) - v_x(i-h,j,k)}{2h} \tag{22}$$

using the velocities from its neighbor voxels along the x-axis. Recall the components of the viscous stress tensor as

$$\boldsymbol{\tau} = \begin{bmatrix} 2\mu\frac{\partial v_x}{\partial x} & \mu\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) & \mu\left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}\right) \\ \mu\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) & 2\mu\frac{\partial v_y}{\partial y} & \mu\left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right) \\ \mu\left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}\right) & \mu\left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right) & 2\mu\frac{\partial v_z}{\partial z} \end{bmatrix}. \tag{23}$$

The components of the divergence of the viscous component of the stress tensor become

$$\nabla\cdot\boldsymbol{\tau} = \begin{bmatrix} 2\frac{\partial}{\partial x}\left(\mu\frac{\partial v_x}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)\right) + \frac{\partial}{\partial z}\left(\mu\left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}\right)\right) \\ \frac{\partial}{\partial x}\left(\mu\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)\right) + 2\frac{\partial}{\partial y}\left(\mu\frac{\partial v_y}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu\left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right)\right) \\ \frac{\partial}{\partial x}\left(\mu\left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}\right)\right) + \frac{\partial}{\partial y}\left(\mu\left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right)\right) + 2\frac{\partial}{\partial z}\left(\mu\frac{\partial v_z}{\partial z}\right) \end{bmatrix} \tag{24}$$

As seen in (24), the derivative of the flux and the cross-derivative of the flux will need to be discretized accordingly. Using $\frac{\partial}{\partial x}\left(\mu\frac{\partial v_x}{\partial x}\right)$ at the voxel location (i,j,k) as the example for the derivative of the flux, it would be approximated as

$$\begin{aligned} \frac{\partial}{\partial x}\left(\mu\frac{\partial v_x}{\partial x}\right) &\approx \frac{1}{h}\left(\left(\mu\frac{\partial v_x}{\partial x}\right)\Big|_{(i+\frac{h}{2},j,k)} - \left(\mu\frac{\partial v_x}{\partial x}\right)\Big|_{(i-\frac{h}{2},j,k)}\right) \\ &= \mu(i+\frac{h}{2},j,k)\frac{v_x(i+h,j,k) - v_x(i,j,k)}{h^2} \\ &\quad - \mu(i-\frac{h}{2},j,k)\frac{v_x(i,j,k) - v_x(i-h,j,k)}{h^2}, \end{aligned} \tag{25}$$

where

$$\mu(i+\frac{h}{2},j,k) \approx \frac{1}{2}(\mu(i+h,j,k) + \mu(i,j,k)) \tag{26}$$

and

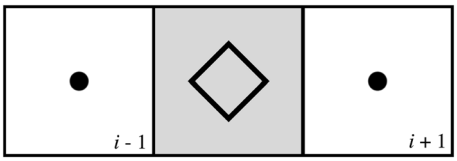


Fig. 4 Discrete gradient $\frac{\partial v_x}{\partial x}$

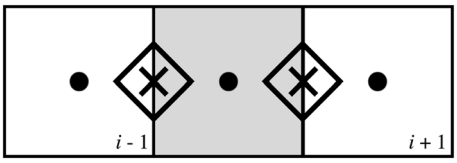


Fig. 5 Discrete derivative of the flux $\frac{\partial}{\partial x}\left(\mu\frac{\partial v_x}{dx}\right)$

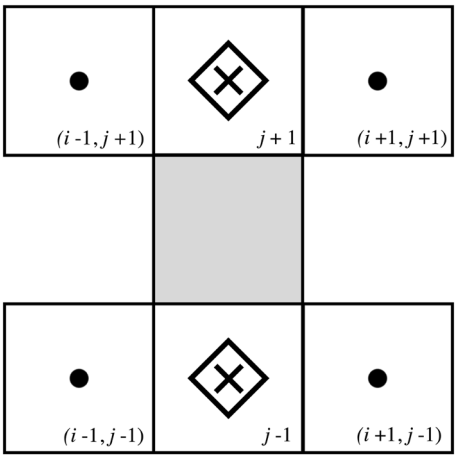


Fig. 6 Discrete cross derivative flux $\frac{\partial}{\partial y}\left(\mu\frac{\partial v_x}{dx}\right)$

$$\mu(i-\frac{h}{2},j,k)\approx\frac{1}{2}(\mu(i,j,k)+\mu(i-h,j,k)). \tag{27}$$

Lastly, using $\frac{\partial}{\partial y}\left(\mu\frac{\partial v_x}{\partial x}\right)$ at the voxel location (i,j,k) as the example for the cross derivative of the flux, it would be approximated as

$$\begin{aligned}\frac{\partial}{\partial y}\left(\mu\frac{\partial v_x}{\partial x}\right)&\approx\frac{\partial}{\partial y}\left(\mu(i,j,k)\frac{v_x(i+h,j,k)-v_x(i-h,j,k)}{2h}\right)\\&=\frac{1}{4h^2}\Big[\mu(i,j+h,k)\big(v_x(i+h,j+h,k)\\&\quad-v_x(i-h,j+h,k)\big)\\&\quad-\mu(i,j-h,k)\big(v_x(i+h,j-h,k)\\&\quad-v_x(i-h,j-h,k)\big)\Big].\end{aligned} \tag{28}$$

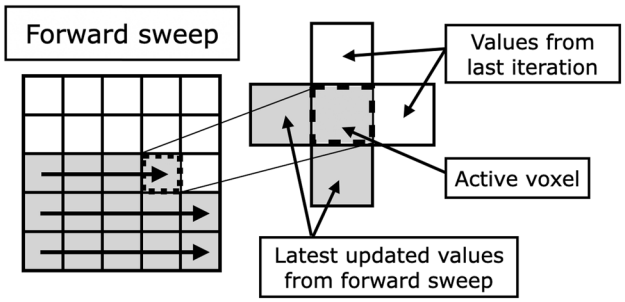


Fig. 7 Successively updating each voxel's properties by Gauss–Seidel method which depends on the most updated values of the surrounding voxels

Other derivatives follow this format, consistent with finite difference practices [27].

Figures 4, 5, and 6 pictorially demonstrate the formulas outlined above. A grey square demonstrates the active voxel where the formula is calculated. The active voxel is positioned at the spacial indices (i, j, k) and each voxel has a length of h. A solid circle represents an evaluated velocity term, likewise, an 'X' represents an evaluated μ term. A hollow diamond represents a computed $\frac{\partial v_x}{\partial x}$ term with respect to the vertically adjacent voxels.

Following Zohdi [28], the equations for wind flow are solved using a matrix-free voxel approach. A voxel, derived from 'volumetric pixel', is used to describe a piece of discretized space that carries the material properties at its nodal location. Storing all voxel properties in arrays and advancing through each voxel successively to solve an equation of interest allows scaling the number of voxels within the domain without occupying too much computer memory as it requires no matrices to be formed, such as in finite element methods. Specifically, voxel velocity values $\boldsymbol{v}$ were successively solved using the Gauss-Seidel method where the spatial derivatives within the velocity gradient $\nabla\boldsymbol{v}$ in Eq. 19 are calculated using its latest updated values from adjacent voxels (Fig. 7).

2.3 Model Performance Optimization

Optimizing simulation run-time performance is imperative for real-time digital twin and GA capabilities. Poor simulation performance is caused by having to calculate drag forces on each spray particle at every timestep. Instead, drag force calculations for these simulations were updated periodically every couple of timesteps. A spatial hashing algorithm helps to rapidly determine each airborne particle's current voxel it is in and the drag force imposed on each particle depends on their voxel's wind velocity. Although periodic drag force updates cut down on computational time,

model accuracy diminishes if the algorithm takes too long to update because particles may traverse through several voxels without having their respective drag forces acting upon those particles. Thus, one must be mindful of determining the frequency of updating the physics which depends on the spray particle speed and voxel size.

3 Machine Learning for Optimized Parameter Search

The model problem in this work involves many design parameters and a nonconvex cost function, making it difficult to use gradient-based optimization schemes. Thus, we elect to use a genetic algorithm for optimization.

For the model in this work, the design parameters the GA optimizes are

1. $\Lambda_1, \Lambda_2, \Lambda_3 =$ the initial drone position, $\boldsymbol{r}_d(0) = (r_{d,x}(0), r_{d,y}(0), r_{d,z}(0))$,
2. $\Lambda_4 =$ the particle eject speed, V_{eject},
3. $\Lambda_5 =$ the total mass of spray material, M_{tot},
4. $\Lambda_6 =$ the nozzle spray amplitude, A,
5. $\Lambda_7, \Lambda_8, \Lambda_9 =$ the initial drone direction, $\boldsymbol{n_d}(0) = (n_{d,x}(0), n_{d,y}(0), n_{d,z}(0))$,
6. $\Lambda_{10} =$ the translational speed of the drone, V_d,
7. $\Lambda_{11} =$ the drop rate of spray particles, η,
8. $\Lambda_{12}, \Lambda_{13}, \Lambda_{14} =$ the angular velocity of the drone, $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$.

Explicitly, the design string is:

$$\boldsymbol{\Lambda} = \{\Lambda_1, \ldots, \Lambda_{14}\} \tag{29}$$

$$= \{r_{d,x}(0), r_{d,y}(0), r_{d,z}(0), V_{eject}, M_{tot}, A, n_{d,x}(0), n_{d,y}(0), n_{d,z}(0), V_d, \eta, \omega_x, \omega_y, \omega_z\}. \tag{30}$$

Genetic algorithms can determine multiple local minima for the cost function, $\Pi(\boldsymbol{\Lambda})$. The user can then choose from the GA results which optimal drone trajectory to use to achieve desired spray results (Fig. 8).

The algorithm was adapted from Zohdi [28, 29] to minimize the cost function presented in Eq. 1. The algorithm for optimizing the system parameters for the drone dynamics and sprayer nozzle follows:

1. **POPULATION GENERATION**: Randomly generate a population of S genetic strings, $\boldsymbol{\Lambda}^i$, $(i = 1, 2, 3, \ldots, S)$: $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda_1, \Lambda_2, \ldots, \Lambda_{14}\}^i$.
2. **PERFORMANCE EVALUATION**: Compute fitness of each string, $\Pi(\boldsymbol{\Lambda}^i)$, $(i = 1, \ldots, S)$:

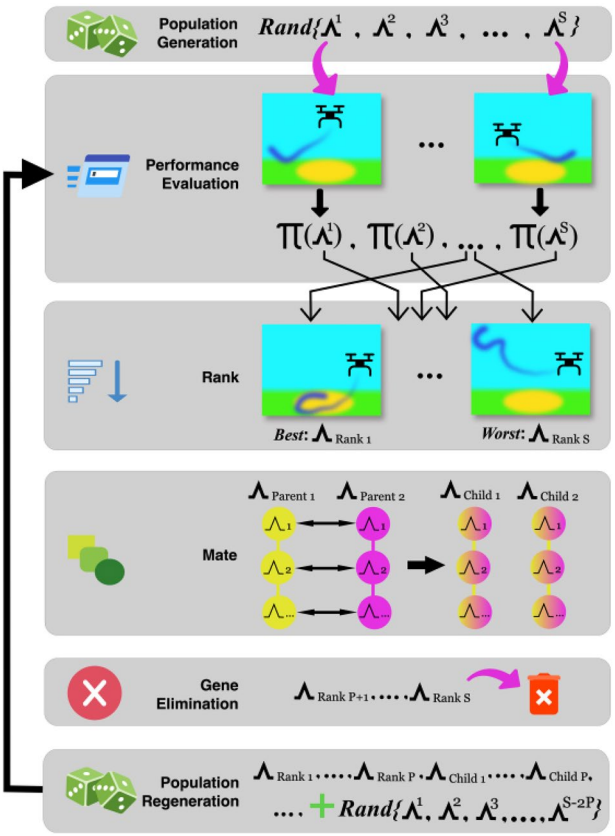


Fig. 8 The GA optimization process for the model problem. S design strings ($\boldsymbol{\Lambda}^i$ for $i = 1, \ldots, S$) are randomly generated, evaluated, ranked, mated, eliminated, and regenerated throughout a set number of generations

$$\Pi(\boldsymbol{\Lambda}^i) \stackrel{\text{def}}{=} W_1 \frac{P_{outside}}{P_{total}} + W_2 \frac{T_{remaining}}{T_{total}} + W_3 \frac{t_{actual}}{t_{max}}.$$

3. **RANK**: Rank each string based on their cost output Π, where Rank 1 is the best-performing design string that produced the lowest cost and Rank S is the worst performing string: $\Pi(\boldsymbol{\Lambda}^1) \leq \Pi(\boldsymbol{\Lambda}^2) \leq \ldots \leq \Pi(\boldsymbol{\Lambda}^S)$.
4. **MATE**: Mate nearest pairs of design strings and produce two offspring: $\Lambda^i \stackrel{\text{def}}{=} \Phi^{(1)}\Lambda^i + (1 - \Phi^{(1)})\Lambda^{i+1}$ and $\Lambda^{i+1} = \Phi^{(2)}\Lambda^i + (1 - \Phi^{(2)})\Lambda^{i+1}$ where $0 \leq \Phi \leq 1$ are distinct for each design parameter within the design string.
5. **GENE ELIMINATION**: Eliminate poorly performing genetic strings, keep top parents, and generate offspring.
6. **POPULATION REGENERATION**: Repeat the process with the parent, offspring, and new *random* genetic strings.
7. **RESTART**: Restart the search around the best-performing parameter if the same parents are carried over every 10 generations.

4 Numerical Example

A 30 m × 30 m × 30 m domain discretized into 21 voxels per axis was initialized. 300 targets were randomly placed within a circular region on the ground, and wind speeds were initialized along the boundaries. The initial conditions, $(v_x(t = 0\ s), v_y(t = 0\ s), v_z(t = 0\ s))$, for each of the six boundaries of the domain were:

- (3.0, 0.0, 0.0) *m/s* for the east-facing boundary,
- (2.0, 0.0, 0.0) *m/s* for the west-facing boundary,
- (0.0, −1.0, 0.0) *m/s* for the north-facing boundary,
- (0.0, −1.5, 0.0) *m/s* for the south-facing boundary,
- (0.0, 0.0, 1.0) *m/s* for the top-facing boundary and
- (0.0, 0.0, 0.0) *m/s* for the bottom-facing (ground) boundary.

The wind velocity field within the domain evolves until it reaches a steady state, then the crop-dusting drone with the design parameters given by the GA was initialized. When the simulation starts, the drone will move based on its angular velocity and translational speed while spraying material droplets whose trajectory is affected by the wind velocity. For every few time steps, the inflow wind velocity direction from the west-facing boundary was perturbed by changing its direction within a cone region with a specified amplitude, similar to randomizing the direction of ejected particles as shown in Eq. 6. This is to allow for randomness in wind and to not allow a steady flow form in the domain. This is to better simulate what may occur in an uncontrolled environment such as an open field. As the drone traverses, if it attempts to go below a height of 2 m, its z position, $r_{d,z}$, will be reset to 2 m. If the drone travels outside the bounds of the domain, the simulation will stop and return a large cost value, penalizing such designs. The simulation parameters are shown in Table 1.

The GA ran for 400 generations, each containing a population of 50 design strings. For each generation, the top 10 parent design strings are kept and mated to make ten additional offspring design strings while the lower ranked strings are eliminated. If the same parent strings are carried over for ten successive generations, the parameter search will restart around the best-performing design. This process involves only keeping the best design and eliminating all the other 49 design strings and replacing them with randomly generated strings. After the GA run, the best-performing design's particle and flight trajectories and the resulting ground concentration of spray material are saved.

5 Results

Figure 9 shows the reduction of the best-performing design string's and population's average cost over 400 generations. Compared to the initial population average cost, the best design cost was reduced by 65% by the end of the GA run. While the overall curve for the best design cost is monotonic, the jaggedness between generations pertains to the random gusts introduced to the simulation at each run. The average

Table 1 Crop-dusting simulation and GA parameters

Symbol	Type	Units	Value	Description
L_{domain}	Scalar	m	30	Domain length, width and height
D_T	Scalar	m	22	Diameter of circular region where targets are placed
T_{total}	Scalar	None	300	Number of targets in the simulation
t_{max}	Scalar	s	60	Maximum simulation time
r_p	Scalar	mm	0.5	Radius of each spray particle
ρ_p	Scalar	kg/m^3	1000	Density of spray material
V_d	Scalar	m/s	[0,3]	Search bounds for drone's translational speed
$r^-_{d,xy}$, $r^+_{d,xy}$	Scalar	m	[− 15,15]	Search bounds for drone's x and y starting position
$r^-_{d,z}$, $r^+_{d,z}$	Scalar	m	[2,30]	Search bounds for drone's z starting position
ω^-_{xyz}, ω^+_{xyz}	Vector	rad/s	[− 5,5]	Search bounds for angular velocity
$n^-_{d,xyz}$, $n^+_{d,xyz}$	Vector	None	[− 1,1]	Search bounds for drone's direction
A^-, A^+	Scalar	None	[0.1, 1.0]	Search bounds for spray nozzle's amplitude
V^-_{eject}, V^+_{eject}	Scalar	m/s	[1,7]	Search bounds for material ejection speed
η^-, η^+	Scalar	m^3/s	[1e−4, 2e−3]	Search bounds for material drop rate
M^-_{tot}, M^+_{tot}	Scalar	kg	2	Search bounds for total amount of material sprayed
W_1	Scalar	None	15	Cost function weight associated with material leaving the domain
W_2	Scalar	None	80	Cost function weight associated with remaining targets
W_3	Scalar	None	5	Cost function weight associated with simulation time

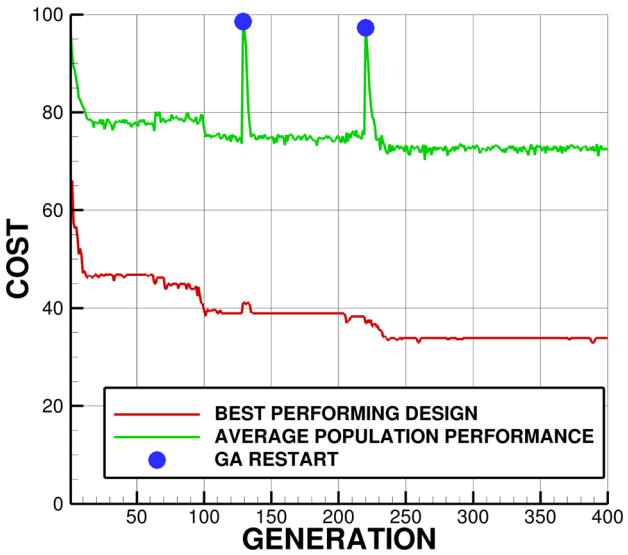


Fig. 9 Costs of best design string (*red*) and the average cost of the population (*green*) throughout 400 generations. The GA restarts at generations 129 and 229 because the same parent strings $\{\mathbf{\Lambda}^1, \ldots, \mathbf{\Lambda}^P\}$ survived for ten consecutive generations. These events caused the average population performance to rise significantly but helped find a more optimal result by generation 237, reducing the cost function by an additional 4.4%. The cost function stayed consistent for the remaining generations

population costs spiked at generations 129 and 229, where the GA had to restart around the best-performing design due to the parent strings being the same for the prior ten generations. For the first restarting at generation 129, there was little difference in the best design's performance, still maintaining a cost of 38.9. However, for the second restarting event at generation 229, it helped reduce the cost significantly from 38.2 to 33.8, which it maintained at that cost until the 400th generation.

After 400 generations, the GA produced the design parameters shown in Table 2, resulting in a crop-dusting drone performing a circular trajectory above the targeted areas while releasing 4.7 kg of material. Snapshots of the drone spraying over time are shown in Fig. 10. The design string resulted in 192 out of 300 targets hit by the material resulting in a 64% hit rate. The optimal trajectory, shown in Fig. 11a, maintains a low altitude averaging 2.6 m with a maximum height of 3.4 m and a minimum height set by the flight bounds of 2 m. This trajectory reduces the chance of spray drifts occurring by having the ejected particle speed of 3.52 m/s be the dominant force over the surrounding wind causing only 0.03 kg of material blown out of domain bounds.

It is also beneficial for a crop-dusting drone operator to know the concentration distribution of material on the ground to ensure proper application. The framework provides post-process results of a simulated run such as Fig. 11b, which show the amount of material in kg/m^2 on the ground. From the optimal design string shown in Table 2, the spray pattern on the ground follows the circular trajectory. Most of the covered area consists of an average amount of 0.035 kg/m^2 except for 10 m^2 worth of space which contained more than 0.055 kg/m^2. Figure 11a shows the starting and ending positions are relatively close to one another which caused the area in between to be sprayed twice, resulting in higher levels of material concentration.

6 Discussion

Precision agriculture involves collecting spatial and temporal data throughout a growing season and using this data to match the inputs to site-specific conditions. As farms are beginning to use a variety of embedded sensors and mapping technologies to monitor the status of their soil and crops, managing site-specific issues requires techniques for efficient use of resources. The framework described can provide guidance and training to a crop-dusting drone operator tasked with spraying fertilizer, water, or other soil amendments in areas of interest. This framework provides an optimal trajectory and nozzle configuration of a crop-dusting drone to minimize the amount of material lost due to spray drifts from given wind drag synced with weather data. Extensions to this framework can be made by using 3D models of agriculture fields for the domain, furthering its use a digital twin, enabling the user of optimal drone configuration in real-time in the physical environment [30–34]. Even though the ground concentration of sprayed material does not play a factor in the GA's search for an optimal design, one may want to extend the cost function to include a desired range of concentrations to avoid high variation. In addition, this work is limited by small droplet particles not having interaction with one another. For distributing solid granular particles that can collide with one another, one can implement the work found in [35]. This modification can further help reduce environmental pollution, such as groundwater contamination and greenhouse gases [36] as plants recover only 30–50% of nitrogen in fertilizers [37], making the rest a potential source of agriculture runoff.

Table 2 Optimized parameters $\{\Lambda_1, \ldots, \Lambda_{14}\}$ generated by the GA after 400 generations with a population of 50 design strings per generation

Λ_1	Λ_2	Λ_3	Λ_4	Λ_5	Λ_6	Λ_7	Λ_8	Λ_9	Λ_{10}	Λ_{11}	Λ_{12}	Λ_{13}	Λ_{14}	$\mathbf{\Pi}$
− 4.24	− 4.33	2.99	3.52	4.70	0.71	0.70	− 0.35	− 0.62	1.89	1.54e−3	0.28	− 0.24	− 2.32	**33.8**

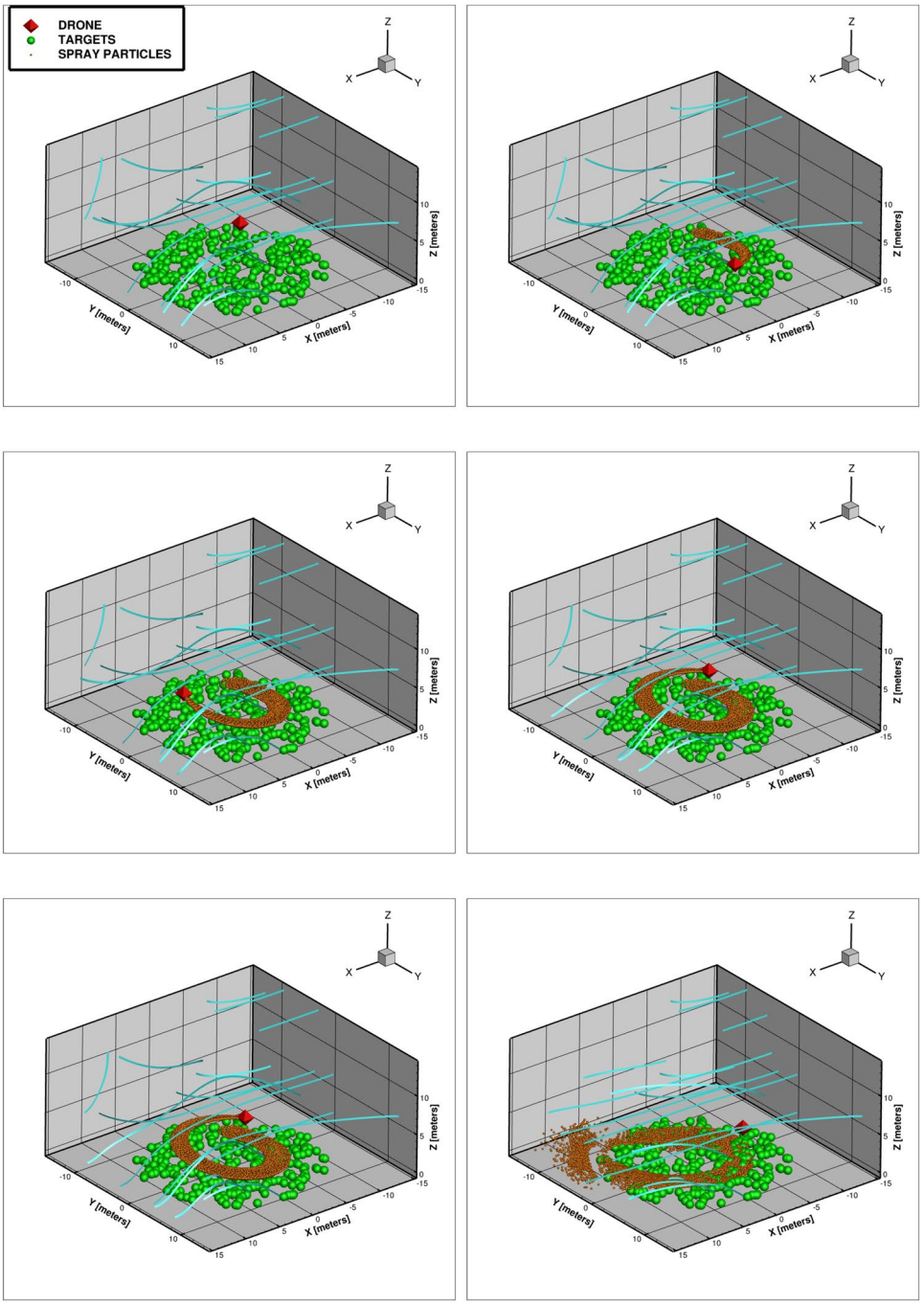
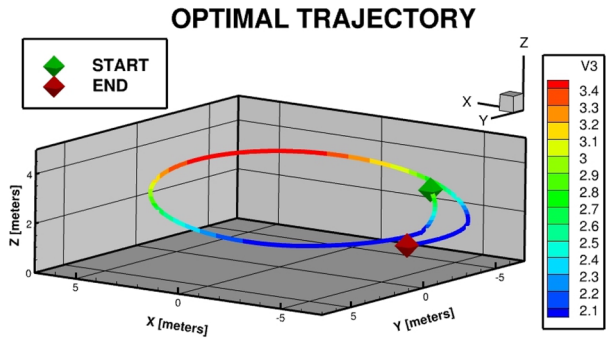


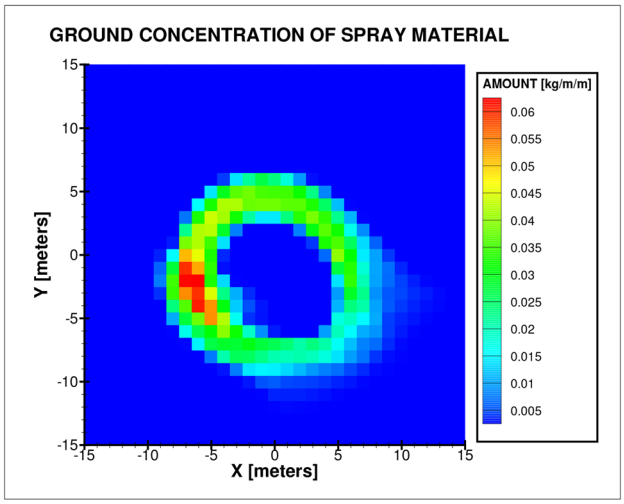
Fig. 10 From left to right and top to bottom are simulation frames of the crop-dusting drone with optimized parameters provided by the GA at $t =$ 1, 2, 3, 4, and 5 s, respectively. The figure on the right-most bottom is the frame at final $t =$ 60 s, showing where particles have landed within the domain. 64% of the targets were hit with only a loss of 0.6% of the total spray mass leaving the domain due to spray drifts. The blue streaks across the domain represent the streamlines of air in that frame

7 Conclusion

The objective of this work was to develop a framework capable of training crop-dusting drone operators on how to spray specific sites within a farm field for the purposes of precision agriculture. The framework uses multiphysics particle dynamics of rigid body motion and fluid dynamics to calculate spray droplet trajectory to mitigate spray drifts from occurring in various wind conditions. A genetic algorithm was used to ascertain the optimal trajectory of the crop-dusting drone as well as the spray rate and amplitude for the nozzle. A model problem was shown of a drone spraying a patch of targets in which the optimal design string generated a drone trajectory and nozzle configuration resulting in 64% of the targets hit while only losing 0.6% of the total spray mass lost from spray drifts. This framework also



(a) Optimal trajectory generated by GA



(b) Ground concentration of spray material

Fig. 11 Post-processed results from the proposed framework with optimized parameters given by the GA shown in Table 2. *Top* The framework provided a circular trajectory for optimal spraying of targets while minimizing the amount of spray material carried away by spray drifts. The trajectory follows a counterclockwise circular pattern 3 m above the ground. *Bottom* Resulting ground concentration of spray material on the ground. The pattern resembles the drone's circular trajectory. Most sprayed areas have an average amount of 0.035 kg/m^2 (*green*) with 10 m^2 of space with more than 0.055 kg/m^2 (*red*) due to that area being sprayed twice

provides further analysis of spray material ground concentration to give more detailed insight into where the spray material landed within the area. The current simulation of a single design runs in the order of seconds on a standard laptop. As the sprayed area increases, it is expected that the simulation time will scale linearly, as the model will run the crop-dusting simulation in cubic partitions of the domain.

One can expand this work by automating drone control input through Model Predictive Control (MPC) algorithms resulting in little to no input from a manned operator [38]. MPC can automate the specific inputs needed for a drone's yaw, pitch, roll, and thrust to keep it within the optimized trajectory provided by this framework all while handling physical and safety constraints. Furthermore, one can include more crop-dusting drones spraying simultaneously, thus creating an agent-based model of a drone swarm. Such collaboration of multiple drones can further reduce the time and energy required to spray the areas of interest. Such frameworks for multi-agent collaboration can be seen in the works of Goodrich et al. [11] and Zohdi [39].

Appendix: Wind Model Incompressibility Assumption

A fluid can only be incompressible if it can only undergo volume-preserving (or isochoric) motions, which implies the determinant of the deformation gradient should always be unity ($J = \det(\boldsymbol{F}) = 1$). Thus, a fluid can only undergo isochoric motions if and only if its velocity field is divergence-free ($\nabla \cdot \boldsymbol{v} = 0$). Observe that

$$\begin{aligned} \dot{J} &= \frac{d}{dt}\det \mathbf{F} = \left(\frac{d}{d\mathbf{F}}\det \mathbf{F}\right) \cdot \frac{d\mathbf{F}}{dt} = Jtr\left(\dot{\boldsymbol{F}}\boldsymbol{F}^{-1}\right) \\ &= Jtr\boldsymbol{L} = J\nabla \cdot \boldsymbol{v}, \end{aligned} \tag{31}$$

where $\mathbf{L}$ is the (spatial) velocity gradient. Assuming that there exists an initial undeformed reference state such that $J = 1$, the sufficient condition can be verified as if $\nabla \cdot \boldsymbol{v} = 0$, then $\dot{J} = 0$ implies that $J = 1$ for all time. Similarly, the necessary condition can be verified as $\dot{J} = 0$ implies that $\nabla \cdot \mathbf{v} = 0$ for every admissible motion ($J \neq 0$).

In general, incompressible flow is a strong assumption to make as it would suggest that there would be no density fluctuation with the fluid. The absence of density perturbation can be shown using the mass continuity equation along with the Reynolds Transport Theorem as any arbitrary subregion or control volume is also a function of time. Under the assumption that there exists a referential/Lagrangian configuration and a current/Eulerian configuration such that the infinitesimal volume element from both configurations is related by $dv = JdV$, the Reynolds Transport Theorem can be derived by first pulling back to the Langrangian configuration to apply the time differentiation and afterward pushing forward back to the Eulerian configuration. Utilizing this approach and for any scalar-field variable ϕ, the Reynolds Transport Theorem states

$$\begin{aligned}\frac{d}{dt}\int_{\mathcal{R}}\phi dv &= \frac{d}{dt}\int_{\mathcal{R}_0}\phi J dV \\ &= \int_{\mathcal{R}_0}\left(\dot{\phi}J + \phi\dot{J}\right)dV \\ &= \int_{\mathcal{R}_0}\left(\dot{\phi}J + \phi J\nabla\cdot\boldsymbol{v}\right)dV \\ &= \int_{\mathcal{R}_0}\left(\dot{\phi} + \phi\nabla\cdot\boldsymbol{v}\right)JdV \\ &= \int_{\mathcal{R}}\left(\dot{\phi} + \phi\nabla\cdot\boldsymbol{v}\right)dv.\end{aligned} \tag{32}$$

Applying the Reynolds Transport Theorem (32) to the mass continuity equation yields

$$\begin{aligned}\frac{d}{dt}m &= \frac{d}{dt}\int_{\mathcal{R}}\rho dv \\ &= \int_{\mathcal{R}}\dot{\rho} + \rho\nabla\cdot\boldsymbol{v}dv \\ &= 0.\end{aligned} \tag{33}$$

After arriving at the local form of (33)

$$\dot{\rho} + \rho\nabla\cdot\boldsymbol{v} = 0, \tag{34}$$

it can be easily observed that if the velocity field is divergence-free ($\nabla\cdot\boldsymbol{v} = 0$), then the material time derivative of the density field is zero ($\dot{\rho} = 0$), suggesting that the density of the air would remain constant. This is a very critical assumption to make as the physics of atmospheric flow is a complex subject. For example, the effective air density can be easily affected by the moisture of the air or other dusting particles. In addition, there could also be a Boussinesq-like Equation of State relationship that connects the fluid's density to its pressure, which would be unphysical under the incompressibility assumption where the density field is constant. For the incompressibility assumption to hold in this framework, certain restrictions will need to be enforced. First, the physical height of the simulation domain is restricted to only 30 ms from ground level, which should be far below the planetary boundary layers and thus be able to neglect any effects of stratified flow. Second, the velocity field of the wind will be significantly restricted in the simulation such that the maximum speed will be under 3 m/s for all time. This second restriction will ensure that the max Mach number, which is generally defined as the ratio of the flow velocity over the speed of sound in the medium, within the simulation domain is small (For a max speed of 3 m/s and using the standard speed of sounds of 343 m/s at 20 degrees Celsius, the maximum Mach number is the simulation is approximately 0.01). These two restrictions would ensure the wind in the simulation behaves similarly to other wind engineering applications, such as single-turbine simulations [40], that assume the wind is incompressible. In fact, the height of the simulation domain is selected to roughly match the hub height of a Vestas V27 (31.5 m) [41]. As a general rule of thumb, most engineering applications assume a fluid flow to be incompressible when the Mach number is less than 0.3, however, it is also important to note that low Mach number does not imply incompressible flow. The general argument for the low Mach number approximation is presented for a steady elastic fluid, where the Cauchy stress tensor is defined as $\boldsymbol{T} = -p\boldsymbol{I}$, where the pressure $p = \hat{p}(\rho)$ is defined as a function of the fluid density ρ. The speed of sound κ is defined as a scalar function $\kappa^2(\rho) = \frac{d\hat{p}}{d\rho}$. For this steady elastic fluid, the local mass continuity (34) equation can be rewritten as

$$\begin{aligned}\dot{\rho} + \rho\nabla\cdot\boldsymbol{v} &= \frac{\partial\rho}{\partial t} + (\nabla\rho)\cdot\boldsymbol{v} + \rho\nabla\cdot\boldsymbol{v} \\ &= \frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{v}) \\ &= \nabla\cdot(\rho\boldsymbol{v}) = 0,\end{aligned} \tag{35}$$

where the $\frac{\partial\rho}{\partial t}$ vanishes due to the assumption of steady flow. This allows the material time derivative of the density field to be rewritten as

$$\dot{\rho} = -\rho\nabla\cdot\boldsymbol{v} = -\nabla\cdot(\rho\boldsymbol{v}) + (\nabla\rho)\cdot\boldsymbol{v} = (\nabla\rho)\cdot\boldsymbol{v}. \tag{36}$$

Under the assumption that the body force is neglectable, the local balance of linear momentum states

$$\rho\dot{\boldsymbol{v}} = \nabla\cdot\boldsymbol{T} = -\nabla p. \tag{37}$$

Equation (37) can be rewritten using chain rule as

$$\rho\dot{\boldsymbol{v}} = -\frac{\partial p}{\partial\rho}(\nabla\rho) = -\kappa^2(\nabla\rho). \tag{38}$$

Now taking the dot product with respect to the velocity on both for (38) and using (36) yields

$$\rho\dot{\boldsymbol{v}}\cdot\boldsymbol{v} = -\kappa^2(\nabla\rho)\cdot\boldsymbol{v} = -\kappa^2\dot{\rho}. \tag{39}$$

Now, the left-hand side of (39) can be rewritten as

$$\rho\dot{\boldsymbol{v}}\cdot\boldsymbol{v} = \frac{1}{2}\rho\dot{\overline{\boldsymbol{v}\cdot\boldsymbol{v}}} = \frac{1}{2}\rho\dot{\overline{||\boldsymbol{v}||^2}} = \rho(||\boldsymbol{v}||)\dot{\overline{||\boldsymbol{v}||}}, \tag{40}$$

where $||\boldsymbol{v}|| = \sqrt{\boldsymbol{v}\cdot\boldsymbol{v}}$ is the standard Euclidean norm of the velocity or the wind speed. Recalling the definition of the Mach number, which is defined as $M = \frac{||\boldsymbol{v}||}{\kappa}$, (39) can now be rewritten as

$$-M^2\frac{\dot{\overline{||\boldsymbol{v}||}}}{||\boldsymbol{v}||} = \frac{\dot{\rho}}{\rho}. \tag{41}$$

When Mach number M tends to zero, the left-hand side of (41) approaches zero, which suggests that the density

fluctuation would be very small. Therefore, $\dot{\rho} = 0$ would be a safe assumption to make.

Acknowledgements This work has been partially supported by the UC Berkeley College of Engineering, UC Berkeley McNair Scholars Program, and the USDA AI Institute for Next Generation Food Systems (AIFS), USDA Award Number 2020-67021-32855.

Declarations

Conflict of interest The authors have no Conflict of interest to declare.

References

1. van Dijk M, Morley T, Rau M-L, Saghai Y (2021) A meta-analysis of projected global food demand and population at risk of hunger for the period 2010–2050. Nat Food 2:494–501. https://doi.org/10.1038/s43016-021-00322-9
2. Brown, M, Antle J, Backlund P, Carr E, Easterling W, Walsh M, Ammann C, Attavanich W, Barrett C, Bellemare M, Dancheck V, Funk C, Grace K, Ingram J, Jiang H, Maletta H, Mata T, Murray A, Ngugi M, Tebaldi C (2015) Climate change, global food security and the U.S. food system. https://doi.org/10.7930/J0862DC7
3. Hanson C, Ranganathan J, Davey E, Searchinger T, Holzer J (2022) The Ukraine crisis threatens a sustainable food future. https://www.wri.org/insights/ukraine-food-security-climate-change
4. Dasgupta S, Robinson E (2022) Impact of covid-19 on food insecurity using multiple waves of high frequency household surveys. Sci Rep 12:1865. https://doi.org/10.1038/s41598-022-05664-3
5. Kalantari F, Mohdtahir O, Akbari Joni R, Fatemi E (2018) Opportunities and challenges in sustainability of vertical farming: a review. J Landsc Ecol. https://doi.org/10.1515/jlecol-2017-0016
6. Adenaeuer L (2014) Up, up and away! the economics of vertical farming. J Agric Stud 2:40–60. https://doi.org/10.5296/jas.v2i1.4526
7. Ranganathan J, Waite R, Searchinger T, Hanson C (2018) How to sustainably feed 10 billion people by 2050, in 21 charts
8. Shafi U, Mumtaz R, García-Nieto J, Hassan SA, Zaidi SAR, Iqbal N (2019) Precision agriculture techniques and practices: from considerations to applications. Sensors 19(17):3796. https://doi.org/10.3390/s19173796
9. Baumbauer CL, Goodrich PJ, Payne ME, Anthony T, Beckstoffer C, Toor A, Silver W, Arias AC (2022) Printed potentiometric nitrate sensors for use in soil. Sensors 22(11):4095. https://doi.org/10.3390/s22114095
10. van der Merwe D, Burchfield D, Witt T, Price K, Sharda A (2020) Drones in agriculture. Adv Agron. https://doi.org/10.1016/bs.agron.2020.03.001
11. Goodrich P, Betancourt O, Arias AC, Zohdi T (2023) Placement and drone flight path mapping of agricultural soil sensors using machine learning. Comput Electron Agric 205:107591. https://doi.org/10.1016/j.compag.2022.107591
12. Dispensing chemicals and agricultural products (part 137) with USA. https://www.faa.gov/uas/advanced_operations/dispensing_chemicals
13. Tudi M, Li H, Li H, Wang L, Lyu J, Yang L, Shuangmei T, Yu Q, Ruan H, Atabila A, Phung D, Sadler R, Connell D (2022) Exposure routes and health risks associated with pesticide application. Toxics 10:335. https://doi.org/10.3390/toxics10060335
14. Quan X, Guo Q, Ma J, Doluschitz R (2023) The economic effects of unmanned aerial vehicles in pesticide application: evidence from Chinese grain farmers. Precision Agric 24:1965–1981. https://doi.org/10.1007/s11119-023-10025-9
15. Enterprise D Precision agriculture with drone technology. https://enterprise-insights.dji.com/blog/precision-agriculture-drones
16. Hovhannisyan T, Efendyan P, Vardanyan M (2018) Creation of a digital model of fields with application of dji phantom 3 drone and the opportunities of its utilization in agriculture. Ann Agrarian Sci 16(2):177–180. https://doi.org/10.1016/j.aasci.2018.03.006
17. Sánchez-Fernández L, Barrera M, Martínez-Guanter J, Pérez-Ruiz M (2023) Drift reduction in orchards through the use of an autonomous UAV system. Comput Electron Agric 211:107981. https://doi.org/10.1016/j.compag.2023.107981
18. Damalas C, Eleftherohorinos I (2011) Pesticide exposure, safety issues, and risk assessment indicators. Int J Environ Res Public Health 8:1402–19. https://doi.org/10.3390/ijerph8051402
19. Spoorthi S, Shadaksharappa B, Suraj S, Manasa V (2017) Freyr drone: pesticide/fertilizers spraying drone-an agricultural approach. In: 2017 2nd international conference on computing and communications technologies (ICCCT), IEEE, pp 252–255
20. Önler E, Özyurt HB, Şener M, Sezen A, Eker B, Çelen İH et al (2023) Spray characterization of an unmanned aerial vehicle for agricultural spraying. Philipp Agric Sci 106(1):39–46
21. Huang Z, Wang C, Li Y, Zhang H, Zeng A, He X (2023) Field evaluation of spray drift and nontargeted soybean injury from unmanned aerial spraying system herbicide application under acceptable operation conditions. Pest Manag Sci 79(3):1140–1153. https://doi.org/10.1002/ps.7285
22. Li X, Giles DK, Andaloro JT, Long R, Lang EB, Watson LJ, Qandah I (2021) Comparison of UAV and fixed-wing aerial application for alfalfa insect pest control: evaluating efficacy, residues, and spray quality. Pest Manag Sci 77(11):4980–4992. https://doi.org/10.1002/ps.6540
23. Biglia A, Grella M, Bloise N, Comba L, Mozzanini E, Sopegno A, Pittarello M, Dicembrini E, Alcatrão LE, Guglieri G, Balsari P, Aimonino DR, Gay P (2022) UAV-spray application in vineyards: flight modes and spray system adjustment effects on canopy deposit, coverage, and off-target losses. Sci Total Environ 845:157292. https://doi.org/10.1016/j.scitotenv.2022.157292
24. Dubuis P-H, Droz M, Melgar A, Zürcher UA, Zarn JA, Gindro K, König SL (2023) Environmental, bystander and resident exposure from orchard applications using an agricultural unmanned aerial spraying system. Sci Total Environ 881:163371. https://doi.org/10.1016/j.scitotenv.2023.163371
25. Herbst A, Glaser M, Bartsch K-U (2023) Spray drift from application of plant protection products with drones in vineyards. Themenheft 75:151–157. https://doi.org/10.5073/JfK.2023.05-06.04
26. Chow C-Y (1979) An introduction to computational fluid mechanics. Colorado, University, Boulder, CO, Tech. rep
27. Kelley W, Peterson A (2001) Difference equations: an introduction with applications. Elsevier, Amsterdam
28. Zohdi T (2022) A digital-twin and machine-learning framework for precise heat and energy management of data-centers. Comput Mech. https://doi.org/10.1007/s00466-022-02152-3
29. Zohdi T (2020) A digital twin framework for machine learning optimization of aerial fire fighting and pilot safety. Comput Methods Appl Mech Eng 373:113446. https://doi.org/10.1016/j.cma.2020.113446
30. Dyck G, Hawley E, Hildebrand K, Paliwal J (2023) Digital twins: a novel traceability concept for post-harvest handling. Smart Agric Technol 3:100079. https://doi.org/10.1016/j.atech.2022.100079
31. Zohdi T (2021) A digital-twin and machine-learning framework for the design of multiobjective agrophotovoltaic solar farms. Comput Mech. https://doi.org/10.1007/s00466-021-02035-z

32. Isied RS, Mengi E, Zohdi TI (2022) A digital-twin framework for genomic-based optimization of an agrophotovoltaic greenhouse system. Proc R Soc A. https://doi.org/10.1098/rspa.2022.0414
33. Mengi E, Samara OA, Zohdi TI (2023) Crop-driven optimization of agrivoltaics using a digital-replica framework. Smart Agric Technol 4:100168. https://doi.org/10.1016/j.atech.2022.100168
34. Mengi E, Becker CJ, Sedky M, Yu S-Y, Zohdi TI (2023) A digital-twin and rapid optimization framework for optical design of indoor farming systems. Comput Mech. https://doi.org/10.1007/s00466-023-02421-9
35. Casas G, Mukherjee D, Celigueta MA, Zohdi TI, Onate E (2015) A modular, partitioned, discrete element framework for industrial grain distribution systems with rotating machinery. Comput Particle Mech 4(2):181–198. https://doi.org/10.1007/s40571-015-0089-9
36. Ladha JK, Pathak H, Krupnik TJ, Six J, van Kessel C (2005) Efficiency of fertilizer nitrogen in cereal production: retrospects and prospects. Adv Agron 87:85–156
37. Cassman KG, Dobermann A, Walters DT (2002) Agroecosystems, nitrogen-use efficiency, and nitrogen management. AMBIO 31(2):132–140
38. Zhu E, Stürz Y, Rosolia U, Borrelli F (2020) Trajectory optimization for nonlinear multi-agent systems using decentralized learning model predictive control
39. Zohdi T (2020) The game of drones: rapid agent-based machine-learning models for multi-UAV path planning. Comput Mech. https://doi.org/10.1007/s00466-019-01761-9
40. Thomas S, Ananthan S, Yellapantula S, Hu J, Lawson M, Sprague M (2019) A comparison of classical and aggregation-based algebraic multigrid preconditioners for high-fidelity simulation of wind turbine incompressible flows. SIAM J Sci Comput 41:S196–S219. https://doi.org/10.1137/18M1179018
41. Markkilde Petersen S (1990) Wind turbine test Vestas V27-225 kW. Risø-M No. 2861