Coursework Report: Winnowing Process

Simulation and Analysis of Particle Separation in the Winnowing Process using Numerical Methods

Siddharthan Somasundaram[a,1]

[a] *Otto von Guericke University, Magdeburg, Germany*

Matriculation number : 254304
Submitted as part of the coursework for Simulations of Mechanical Processes (WiSe 24/25).

Abstract—This report investigates the dynamics of a simplified winnowing process, where spherical grains and chaffs are separated using a horizontal air jet. A numerical simulation is conducted to model the particle trajectories, employing Euler and Runge-Kutta methods to solve the governing equations of motion. The accuracy and efficiency of these numerical methods are assessed through error analysis. The optimal jet velocity required for effective particle separation is determined. Additionally, the impact of initial particle velocities and bin placement on separation efficiency is explored. By considering various combinations of initial particle velocities, bin positions, and jet velocities, the feasibility of achieving perfect separation is investigated. The results provide valuable insights into the design and optimization of winnowing systems, highlighting the potential for achieving efficient separation through careful control of these parameters.

Keywords—*LaTeX class, lab report, academic article, tau class*

Contents

1. Introduction

Winnowing is a traditional agricultural technique used to separate grains from lighter chaff, leveraging differences in mass and aerodynamic properties. Historically, this process relied on natural wind to carry away the lighter chaff while the heavier grains fell back to the ground. Modern adaptations of winnowing utilize controlled airflow to achieve more efficient and selective separation. Optimizing *winnowing processes* is particularly beneficial for industries like rice milling and grain processing, where enhanced efficiency can lead to higher product purity and reduced waste. By precisely controlling airflow parameters, such as jet velocity and bin placement, the separation process can be tailored to the specific properties of the particles.

1.1. Objective

The objective of this study is to investigate the dynamics of particle motion in a simplified winnowing system using numerical simulations. By modeling particle trajectories under varying conditions, this study aims to determine the optimal jet velocity for effective separation and explore the feasibility of achieving perfect separation of grains and chaffs.

1.2. Methods

A numerical model is developed to simulate particle motion in a two-dimensional domain. The model accounts for forces such as gravity, drag, and buoyancy. The governing equations of motion are solved using Euler's method and the fourth-order Runge-Kutta (RK4) method.

1.3. Report Structure

The report is structured as follows:
Section 2: Problem Statement.
Section 3: Methodology.
Section 4: Visualization.
Section 5: Analysis Report: Winnowing System Optimization.
Section 6: Conclusions

2. Problem Statement

Particles are released at a point (x_0, y_0) with no initial velocity. As they fall under the influence of gravity, they pass through a horizontal planar jet, which will push the particles towards the positive x-direction. The heavy particles (grains), should be recovered in $Bin1$, while the lighter particles (chaffs) should be recovered in $Bin2$. Refer Fig. 1

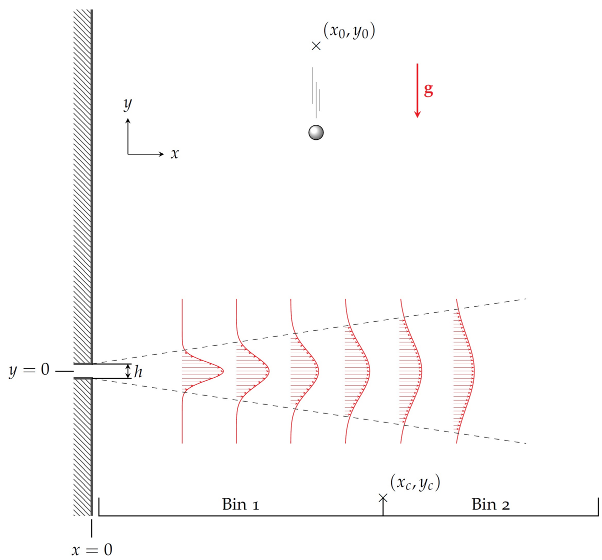

Figure 1. Schematic diagram of the winnowing system

2.1. Governing Equations

The motion of particles is governed by coupled ordinary differential equations (ODEs):

Equation: 1, shows velocity-displacement equation,

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p \tag{1}$$

Equation: 2, shows equation of motion,

$$\rho_p V_p \frac{d\mathbf{u}_p}{dt} = \mathbf{F}_{gravity} + \mathbf{F}_{buoyancy} + \mathbf{F}_{drag} \tag{2}$$

where,

$$\mathbf{F}_{gravity} = \rho_p V_p \mathbf{g} \tag{3}$$

$$\mathbf{F}_{buoyancy} = -\rho_f V_p \mathbf{g} \tag{4}$$

$$\mathbf{F}_{drag} = \frac{1}{2}\pi d_p^2 \rho_f C_D ||\mathbf{u}_f - \mathbf{u}_p||(\mathbf{u}_f - \mathbf{u}_p) \tag{5}$$

The drag coefficient C_D depends on the Reynolds number Re_p:

$$C_D = \begin{cases} \frac{24}{Re_p}\left(1 + 0.15Re_p^{0.687}\right), & \text{if } Re_p < 800 \\ 0.44, & \text{otherwise} \end{cases} \tag{6}$$

$$Re_p = \frac{\rho_f |\mathbf{u}_f - \mathbf{u}_p| d_p}{\mu_f}. \tag{7}$$

Air is blown horizontally through a rectangular slot of height h. As a consequence, the velocity of the fluid in the domain $\{x > 5h\}$ is given by

$$\mathbf{u}_f(x, y) = \begin{bmatrix} 6.2u_0\sqrt{\frac{h}{x}}\exp\left(-50\frac{y^2}{x^2}\right) \\ 0 \end{bmatrix} \tag{8}$$

2.2. Particle Properties and Boundary Conditions:

Particle Properties:

- Refer Table. 1 for particle and fluid properties

Boundary Conditions:

- The jet originates at the origin. (i.e.,) $x_{jet} = 0$, $y_{jet} = 0$.
- Particles fall into Bin 1 if they land to the left of $x_c = 55\,\text{cm}$, $y_c = -50\,\text{cm}$.
- Particles fall into Bin 2 if they land to the right of $x_c = 55\,\text{cm}$, $y_c = -50\,\text{cm}$.
- The ground is assumed at $y_c = -60\,\text{cm}$, where the simulation stops once a particle reaches this boundary.

3. Methodology

This section outlines the numerical methods and implementation used to model particle motion, in the simplified winnowing process. The motion is governed by a set of coupled ordinary differential equations (ODEs), which are solved using the Euler method and the fourth-order Runge-Kutta (RK4) method. Additionally, an iterative bisection method is employed to determine the minimum jet velocity required for optimal particle separation.

3.1. Euler Method

The Euler method is a first-order numerical technique used to approximate solutions of ODEs. It employs a forward difference scheme to estimate the position and velocity of particles at each time step.

Steps in the Euler Method:

Property	Value
Dimensions and flow properties	
Initial particle x-position (x_0)	50 cm
Initial particle y-position (y_0)	50 cm
Initial particle x-velocity (v_x0)	0
Initial particle y-velocity (v_y0)	0
Bin x-coordinate (x_c)	55 cm
Bin y-coordinate (y_c)	-50 cm
Jet height (h)	10 cm
Initial jet velocity (u_0)	20 cm·s^{-1}
Fluid properties (air)	
Fluid density (ρ_f)	1.2 kg·m^{-3}
Fluid viscosity (μ_f)	1.8×10^{-5} Pa·s
Grain particle properties	
Particle density (ρ_p)	750 kg·m^{-3}
Particle diameter (d_p)	2.5 mm
Chaff particle properties	
Particle density (ρ_p)	50 kg·m^{-3}
Particle diameter (d_p)	3.25 mm

Table 1. Flow properties and particle properties

1. Discretization:

$$Position\ :\ \mathbf{r}_p(t + \Delta t) = \mathbf{r}_p(t) + \mathbf{v}_p(t)\Delta t; \tag{9}$$

where,

$$\mathbf{r}_p(t) = \left[x_p(t), y_p(t)\right] \quad and \quad \mathbf{v}_p(t) = \left[v_x(t), v_y(t)\right].$$

$$Velocity\ :\ \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{F}}{m_p}\Delta t; \tag{10}$$

where,

$$\mathbf{v}_p(t) = \left[v_x(t), v_y(t)\right] \quad and \quad \mathbf{F}(t) = \mathbf{F}_{\text{gravity}} + \mathbf{F}_{\text{buoyancy}} + \mathbf{F}_{\text{drag}}.$$

where $\mathbf{F}$ is the net force on the particle and m_p is the particle's mass.

2. Iterative Updates:

- Update the position and velocity based on the forces computed at the current time step.
- Stop the simulation when the particle hits the ground ($y_c <=$ Bin ground) or reaches a bin boundary.

Implementation:

Euler Matlab method:

The Euler method is implemented in the function euler, which: Initializes the particle's position and velocity. Iteratively updates the above equations and Handles the particle's dynamics by iteratively solving the equations of motion until the particle reaches the ground or exits the domain.
Refer fig 2 for the Euler method simulation result

3.1.1. Euler Matlab function:

```
%% Euler method
```

```
function [t, y] = euler(odefun, y0, t_span, dt)
        % Input:
    %   odefun: Function handle for the ODE system
    %   y0: Initial condition vector,
    %   t_span: Time span for the simulation
    %   dt: Time step
    %
    % Output:
    %   t: Time vector
    %   y: Solution matrix, where each row corresponds to a time step

    % Initialize time vector
    t = 0:dt:t_span;

    % Initialize solution matrix
    y = zeros(length(t), length(y0));

    % Set initial condition
    y(1, :) = y0;

    % Time-stepping loop
    for i = 1:length(t) - 1
        % Evaluate the derivative at the current time step
        dydt = odefun(t(i), y(i, :)');

        % Update the solution using Euler's method
        y(i + 1, :) = y(i, :) + dydt' * dt;

        % Check if the particle has hit the ground
        if y(i+1, 2) <= -0.6
            break; % Terminate the simulation
        end
    end

    % Truncate the time vector and solution matrix to the final time step
    t = t(1:i);
    y = y(1:i, :);
end
```

Code 1. Euler Matlab function.

3.2. Runge-Kutta 4th Order (RK4) Method

The Runge-Kutta 4th Order (RK4) method is a higher-order numerical technique used to approximate solutions of ODEs. It provides improved accuracy over the Euler method by evaluating intermediate slopes within each time step.

Steps in the RK4 Method:

1. Discretization: The RK4 method updates the position and velocity by computing weighted averages of slopes (increments) at different stages within a single time step. The equations for position and velocity updates are as follows:

$$\text{Position:} \quad \mathbf{r}_p(t+\Delta t) = \mathbf{r}_p(t) + \Delta t \cdot \mathbf{k}_\mathbf{r}, \tag{11}$$

$$\text{Velocity:} \quad \mathbf{v}_p(t+\Delta t) = \mathbf{v}_p(t) + \Delta t \cdot \mathbf{k}_\mathbf{v}, \tag{12}$$

where the increments $\mathbf{k}_\mathbf{r}$ and $\mathbf{k}_\mathbf{v}$ are weighted averages of intermediate values:

$$\mathbf{k}_{\mathbf{r},\mathbf{v}} = \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

with the stages defined as:

$$\mathbf{k}_1 = f(t, \mathbf{y}(t)),$$
$$\mathbf{k}_2 = f\left(t + \frac{\Delta t}{2}, \mathbf{y}(t) + \frac{\Delta t}{2}\mathbf{k}_1\right),$$
$$\mathbf{k}_3 = f\left(t + \frac{\Delta t}{2}, \mathbf{y}(t) + \frac{\Delta t}{2}\mathbf{k}_2\right),$$
$$\mathbf{k}_4 = f\left(t + \Delta t, \mathbf{y}(t) + \Delta t\mathbf{k}_3\right).$$

2. Iterative Updates:

- Compute the combined $\mathbf{k}$ values for each stage $(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4)$.
- Update the position and velocity using the weighted averages.
- Stop the simulation when the particle hits the ground ($y_c \leq$ Bin ground) or reaches a bin boundary.

Implementation:

RK4 Matlab method:

The RK4 method is implemented in the function rk4, which is similar to Euler, except Compute four intermediate estimates (k-values) of the particle's position , velocity and Handles the particle's dynamics by iteratively solving the equations of motion until the particle reaches the ground or exits the domain.
Refer fig 3 for the RK4 method simulation result

3.2.1. Rk4 Matlab function:

```
%% RK4 method
function [t, y] = rk4(odefun, y0, t_span, dt, Bin_ground)
    % same as Euler

    % Time-stepping loop
    for i = 1:length(t) - 1
        % Calculate RK4 coefficients
        k1 = odefun(t(i), y(i, :)');
        k2 = odefun(t(i) + dt/2, y(i, :)' + dt/2 * k1);
        k3 = odefun(t(i) + dt/2, y(i, :)' + dt/2 * k2);
        k4 = odefun(t(i) + dt, y(i, :)' + dt * k3);

        % Update the solution using the RK4 formula
        dydt = (k1 + 2*k2 + 2*k3 + k4) / 6;
        y(i+1, :) = y(i, :) + dt * dydt';

    %same as Euler
end
```

Code 2. Rk4 Matlab function.

3.3. Determination of Minimum Jet Velocity

A bisection method is used to determine the minimum jet velocity u_0 required to achieve optimal particle separation:

Setup:

- The initial jet velocity (u_0) is set to 0.2 m/s.
- A tolerance value is defined to control convergence.

Bisection Algorithm:

- Simulate particle trajectories at the midpoint of the velocity range.
- Check if the chaff particle lands in Bin 2 using the function check_bin_landing
- Adjust the velocity range based on the result until convergence.

Implementation:

Bisection algorithm Matlab method:

The bisection algorithm is implemented in the function mim_jet_velocity, which:

- Calls rk4 to simulate particle motion.
- Evaluates whether the particle separation criterion is met.
- Iteratively refines u_0 to find the minimum (optimal) velocity.
- Refer fig 5 for the Optimal velocity simulation result for separation.

3.4. Error Analysis:

Error Analysis is performed to assess the accuracy of the Euler and RK4 methods.

Reference Solution for analysis:

- The Trajectory of the particles are governed by a set of coupled ordinary differential equations (ODEs) that generally do not have analytical solutions, especially when considering the complex drag force term.
- The primary challenge in finding an analytical solution for the given system of ODEs lies in the complexity of the drag force term. The drag force depends on the Reynolds number, which is a function of the flow velocity, particle diameter, fluid density, and viscosity. This dependence introduces nonlinearity into the equations, making it difficult to solve analytically.
- Hence, a highly accurate solution is computed using the RK4 method with a very small time step ($\Delta t = 10^{-6}$).

Relative Errors:

- Euclidean Distance (L2 Norm) between the numerical calculated value in actual time step with numerical calculated value in very small time step is used for Error Calculation.

$$Error = \sqrt{(x_{\text{numerical}} - x_{\text{actual}})^2 + (y_{\text{numerical}} - y_{\text{actual}})^2} \tag{13}$$

Implementation:

Order of Error Matlab method:

- The errors are computed for both grain and chaff particles over varying time steps.
- Results are plotted to visualize the error trends.
- Refer fig 4 for the global Error of Euler and Rk4 methods.

4. Visualization

4.1. Particle Trajectories:

4.1.1. Euler:

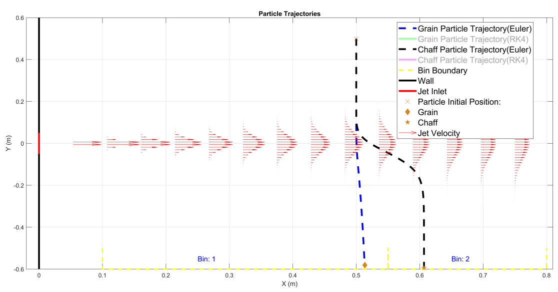

Figure 2. Euler Particle Trajectory.

Interpretation of Figure: Fig 2 shows the trajectory of grain and chaff simulated by Euler method

4.1.2. Rk4

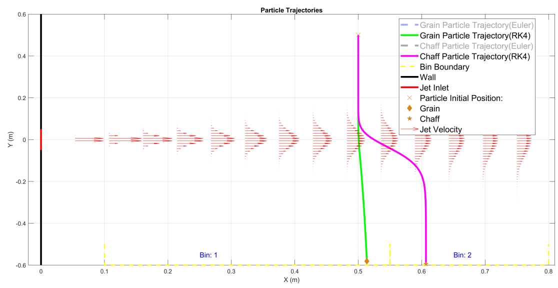

Figure 3. Rk4 Particle Trajectory.

Interpretation of Figure: Fig 3 shows the trajectory of grain and chaff simulated by RK4 method

4.2. Error vs. Time Step:

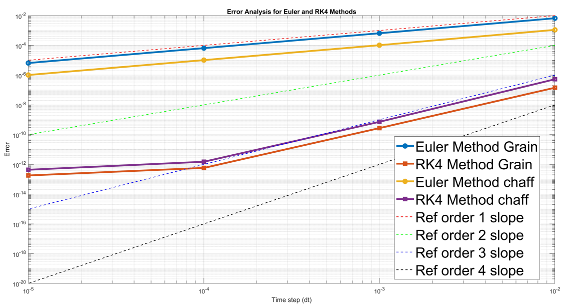

Figure 4. Order of Accuracy of the Euler and Runge-Kutta (RK4) schemes.

Interpretation of Figure:

- From Fig 4, it is evident that the RK4 method has a much steeper slope compared to the Euler method. This indicates that the RK4 method is significantly more accurate than the Euler method.
- The slope for the Euler method is approximately 1 for both grain and chaff. This implies that the error in the Euler method scales linearly with the time step (dt). In other words, doubling the time step roughly doubles the error.
- The slope for the RK4 method is approximately 4 (between 3 and 4)for both grain and chaff. This suggests that the error in the RK4 method scales with the fourth power of the time step. This means that doubling the time step increases the error by a factor of $2^4 = 16$.
- The error slope of the RK4 method at 10^{-5} to 10^{-4} is less steep than the slope observed between 10^{-4} to 10^{-2}. This indicates that while reducing the time step further can still improve accuracy, but the rate of improvement diminishes. This behavior is consistent with the theoretical order of accuracy of the RK4 method, which suggests that halving the time step should reduce the error by a factor of 16. However, as the time step becomes very small, other factors, such as numerical round-off errors, may start to dominate.

4.3. Optimal Jet Velocity:

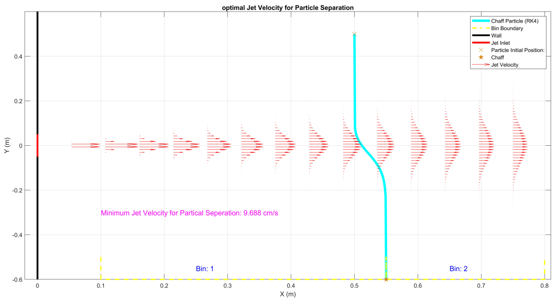

Figure 5. Minimum Jet velocity for particle seperation.

5. Analysis Report: Winnowing System Optimization

Introduction:

The objective of this analysis is to achieve perfect separation of heavy (grain) and light (chaff) particles into their respective bins using a jet flow mechanism. The system parameters include jet velocity (u_0), bin boundary positions (x_c, y_c), and particle properties such as terminal velocity and initial angle. The analysis uses equations of motion, drag force dynamics, and trajectory conditions to determine the feasibility of parameter optimization. Using the equations 1, 2 , 3, 4, 5 , 6 , 7 , 8, the following governing equations are derived.

Governing Equations:

At terminal velocity (v_t), net acceleration is zero:

$$v_t = \sqrt{\frac{4gd_p(\rho_p - \rho_f)}{3C_D\rho_f}}.$$

The particle trajectory released at (x_0, y_0) with velocity v_t at angle θ is given by:

$$x(t) = x_0 + \int_0^t u_p(\tau)d\tau, \quad y(t) = y_0 + \int_0^t v_p(\tau)d\tau,$$

where:

$$u_p(0) = v_t\cos(\theta), \quad v_p(0) = v_t\sin(\theta).$$

5.1. Separation Analysis:

For successful separation:

- **Grain particles (heavy):** Must land in Bin 1:

$$x_g(t) < x_c \text{ when } y_g(t) = y_c.$$

- **Chaff particles (light):** Must land in Bin 2:

$$x_c < x_c(t) \text{ when } y_c(t) = y_c.$$

5.2. Case-by-Case Parameter Analysis:

5.2.1. Case (a): Changing x_c Only:

Mathematical Conditions: Adjusting x_c changes the horizontal separation boundary:

$$\max(x_g(t)) < x_c < \min(x_c(t)) \quad \forall\theta \in [-108°, -72°].$$

Outcome:

- **Heavy particles ($-72°$):** Large $u_{p,x}$ may cause them to cross into Bin 2. Adjusting x_c resolves this.
- **Heavy particles ($-108°$):** Land correctly in Bin 1 regardless of x_c.
- **Light particles ($-72°$):** Naturally cross into Bin 2.
- **Light particles ($-108°$):** May fail to cross into Bin 2 if x_c is too large.

Findings: Adjusting x_c resolves some cases but fails for wide-angle trajectories.

5.2.2. Case (b): Changing y_c Only:

Mathematical Conditions: Adjusting y_c affects vertical separation. The separation condition is:

$$t_{\text{intersection}} = \text{solution of } x_p(t) = x_c,$$

$$y_{\text{intersection}} = y_0 + \int_0^{t_{\text{intersection}}} v_p(\tau)d\tau.$$

Outcome:

- **Heavy particles ($-72°$):** High horizontal velocity may prevent them from descending to Bin 1 if y_c is too high.
- **Heavy particles ($-108°$):** Land in Bin 1 regardless of y_c.
- **Light particles ($-72°$):** May not descend far enough to reach Bin 2.
- **Light particles ($-108°$):** May descend incorrectly into Bin 1.

Findings: Adjusting y_c improves vertical separation but does not resolve horizontal trajectory overlap.

5.2.3. Case (c): Changing u_0 Only:

Mathematical Conditions: Drag force ratio affects particle deflection:

$$\frac{F_{\text{drag, c}}}{F_{\text{drag, g}}} = \frac{d_{p,c}}{d_{p,g}}\frac{C_{D,c}}{C_{D,g}}\frac{|u_0 - u_{p,c}|}{|u_0 - u_{p,g}|}.$$

Outcome:

- **Low $u_0 = 5$ cm/s:** Insufficient drag force fails to separate light particles.
- **High $u_0 = 40$ cm/s:** Excessive drag force may cause heavy particles to enter bin 2.

Findings: Adjusting u_0 may resolve trajectory overlap.

5.2.4. Case (d): Changing Both x_c and y_c:

Outcome:

- Simultaneous adjustment improves both horizontal and vertical separation for most particles. (i.e., it combines the benefits of cases (a) and (b)).
- Fails for extreme shallow angles ($-72°$) in some cases.

Findings: Improves performance but does not guarantee perfect separation.

5.2.5. Case (e): Changing x_c, y_c, and u_0

Findings: Based on the previous four cases, perfect separation is achievable by controlling all three parameters x_c, y_c, and u_0.

6. Final Conclusion

- The simplified model effectively segregates heavy and light particles into bins 1 and 2, respectively, for a jet velocity of $u_0 = 20$ cm/s
- The error analysis demonstrates the superior accuracy and efficiency of the RK4 method compared to the Euler method for this specific problem.
- The minimum jet velocity for optimal separation is $u_0 = 9.688$ cm/s.
- Based on the mathematical analysis, it is evident that:
  1) Cases (a)-(d) lack sufficient degrees of freedom to control both particle trajectories and the Bin separation boundary.
  2) Only case (e) provides adequate control parameters to achieve perfect separation.