Research presentation

By John McCance

Introduction and background

Understanding transport phenomena in heterogeneous porous media is critical in fields such as hydrogeology and environmental engineering. One way to model such transport is by simulating the behavior of passive solutes in two-dimensional incompressible Gaussian random velocity fields. These random fields, derived from a linearized Darcy flow with log-Gaussian hydraulic conductivity, capture the complex structure of real-world subsurface flow while remaining analytically tractable.

In this project, I aimed to replicate and extend aspects of this modeling approach. I began by implementing the velocity field using a spectral method that represents the random field as a superposition of harmonic modes, as outlined in Dentz et al. (2003). This was first done using a basic numerical approach and later improved by incorporating an extended Runge-Kutta method for greater accuracy when computing particle trajectories governed by the Langevin equation.

To analyze and visualize the results, I exported the simulation data and created detailed streamline plots and transport pathlines in MATLAB. By experimenting with different node counts, I was able to observe how spatial resolution affects the structure of streamlines, including the formation of closed paths. This is a notable feature of 2D Gaussian random velocity fields that can trap particles and induce anomalous dispersion.

This work provides computational and visual insights into how solutes spread in complex flow environments.

Hypothesis

Using a higher-order numerical integration method, such as the extended Runge-Kutta scheme, will yield more accurate representations of particle trajectories in a 2D Gaussian random velocity field compared to simpler methods. Specifically, I hypothesize that:

1. The Runge-Kutta method will better capture the formation of closed streamlines, a key feature of Gaussian random fields.
2. Visualizations with higher node counts will reveal more detailed and accurate flow structures, allowing for better insight into the spatial variability and trapping behavior within the field.
3. As a result, simulations using the Runge-Kutta method will produce dispersion coefficients that more closely align with theoretical predictions for long-time asymptotic behavior than those from simpler integration schemes.

Some constraints:

Mass Conservation: The simulation must preserve the total mass of the passive scalar over time, ensuring that numerical errors do not introduce artificial sources or sinks.
Incompressibility: Since the velocity field is derived from an incompressible flow ($\nabla \cdot u = 0$), the model must strictly satisfy divergence-free conditions. The harmonic-mode construction of the velocity field should preserve this, and accurate interpolation and integration are required to avoid numerical violations.
Energy Consistency: The fluctuations in the flow field, governed by the specified autocorrelation function, must remain consistent with the theoretical variance and correlation length.
Dispersion Scaling: The computed effective and ensemble dispersion coefficients should exhibit correct long-time scaling behavior, particularly the linear growth of longitudinal dispersion in advection-dominated regimes and the saturation of transverse dispersion.

Methodology

To simulate solute transport in a two-dimensional incompressible Gaussian random velocity field, I followed a multi-step numerical approach grounded in the framework described by Dentz et al. (2003). The workflow can be broken down into three core components:

1. I generated a spatially varying, divergence-free (∇·u = 0) Gaussian random velocity field using a spectral method.his involved summing N = 64 harmonic modes, with randomly sampled wave vectors and phases, as in:

$$u_i(x) = \bar{u}\delta_{i1} + 2\sigma\bar{u}\sqrt{\frac{2}{N}}\sum_{j=1}^{N} p_i(k^{(j)})\cos(k^{(j)} \cdot x + \omega^{(j)})$$

2 te/particle trajectories, I used the equation below.

$$x_{t+\Delta t} = x_t + u(x_t)\Delta t + \xi\sqrt{2D\Delta t}$$

I will now go into the Results/Analysis/Discussion

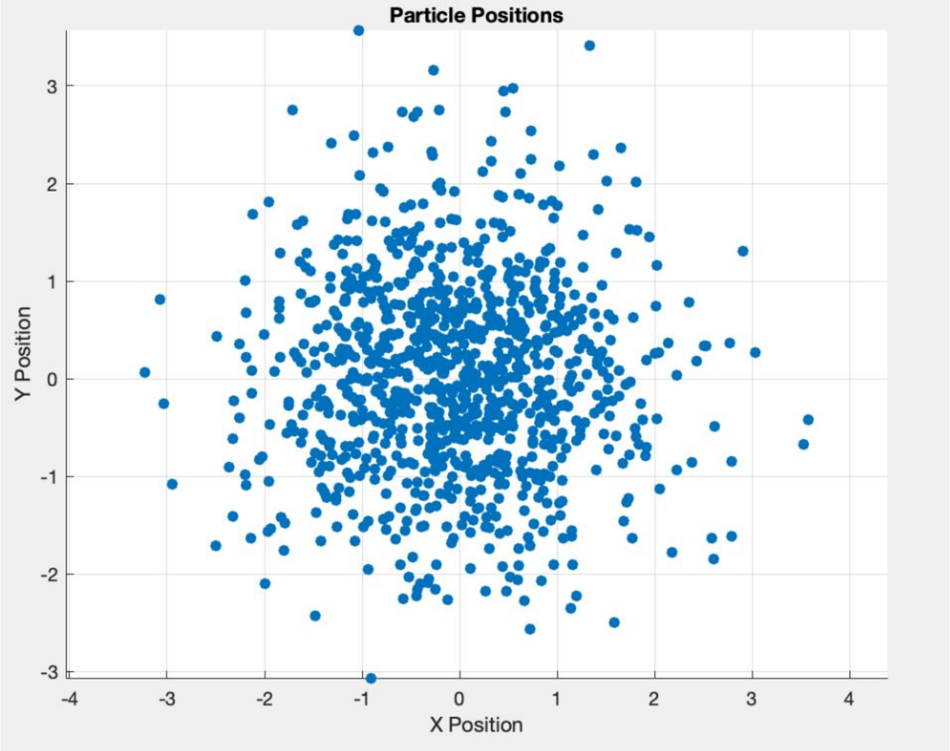
Now that I’ve outlined the numerical methods and theoretical framework guiding this project, I’ll move into the results, analysis, and discussion. In this section, I present the outcomes of simulating particle transport in 2D Gaussian random velocity fields using both the basic Euler method and the more advanced extended Runge-Kutta method.

The core goal here is to evaluate how these numerical schemes differ in capturing key flow behaviors such as the formation of closed streamlines and the long-time dispersion properties of particles. I compare streamline structures, particle trajectories, and dispersion metrics across simulations with varying spatial resolution and integration schemes.

By doing so, I aim to determine whether the Runge-Kutta method indeed offers the greater accuracy predicted by theory: especially in advection-dominated, low-diffusion regimes where numerical precision is critical. These results directly address the hypothesis and connect back to the findings of Dentz et al. (2003), which stress the importance of accurate trajectory integration in modeling realistic transport phenomena.

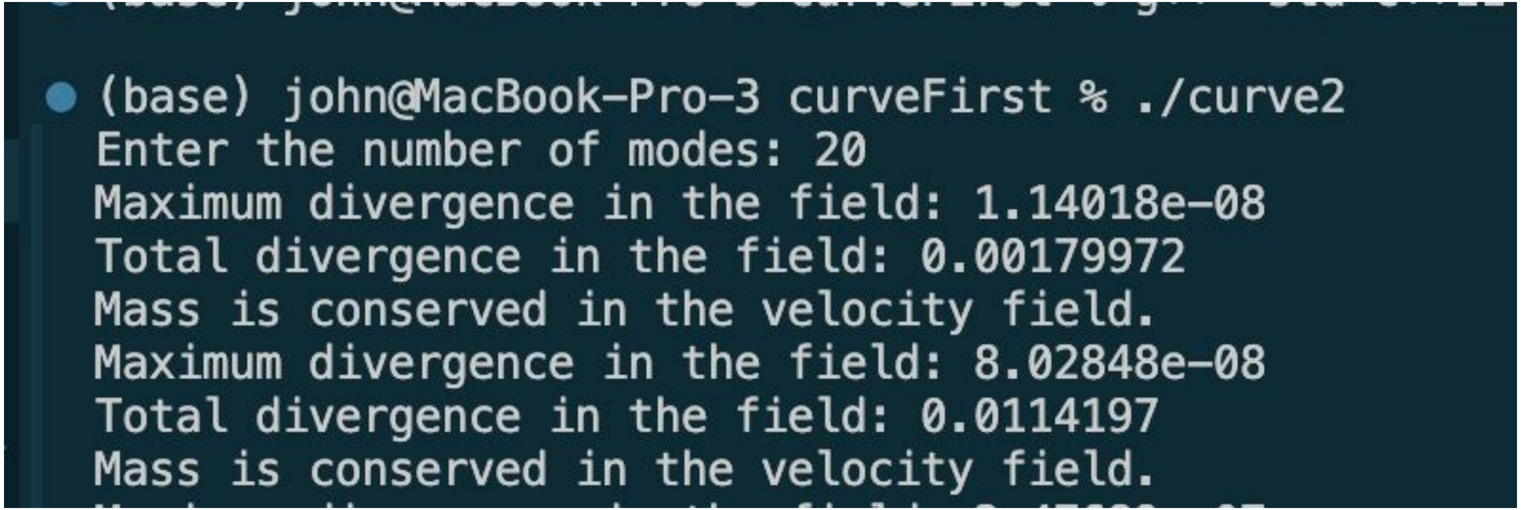
First Part: Basic Particle Tracking

I was tasked with doing a random particle tracking by my Professor as a first step



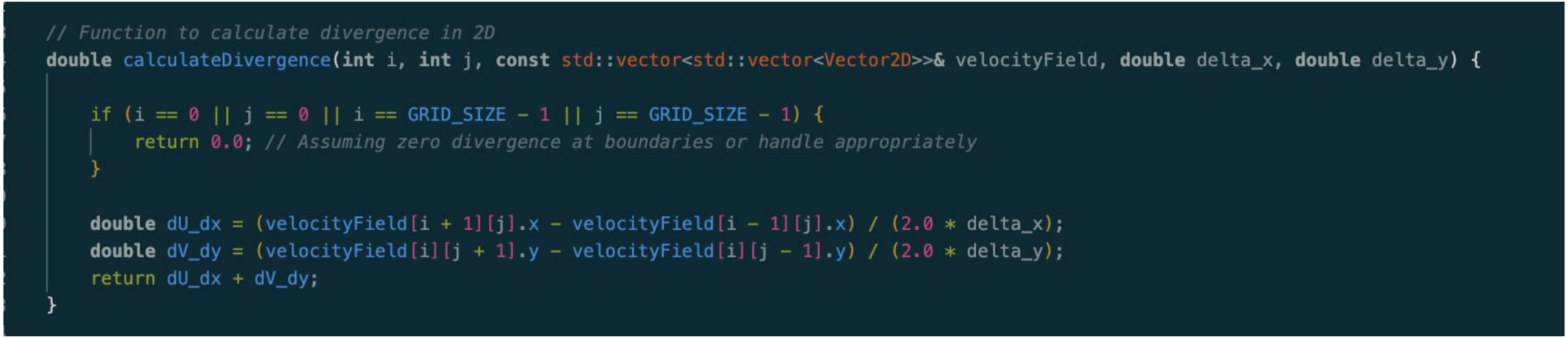
Second part of project: Create velocity random velocity field with mass conservation checks.

For each of 20 nodes, I ran and checked for mass conservation/divergence close to 0. Both constraints held. .



```
● (base) john@MacBook-Pro-3 curveFirst % ./curve2
  Enter the number of modes: 20
  Maximum divergence in the field: 1.14018e-08
  Total divergence in the field: 0.00179972
  Mass is conserved in the velocity field.
  Maximum divergence in the field: 8.02848e-08
  Total divergence in the field: 0.0114197
  Mass is conserved in the velocity field.
```

THE PAPER Assumes 0 divergence, so in my code, I made sure to check for this in the below code

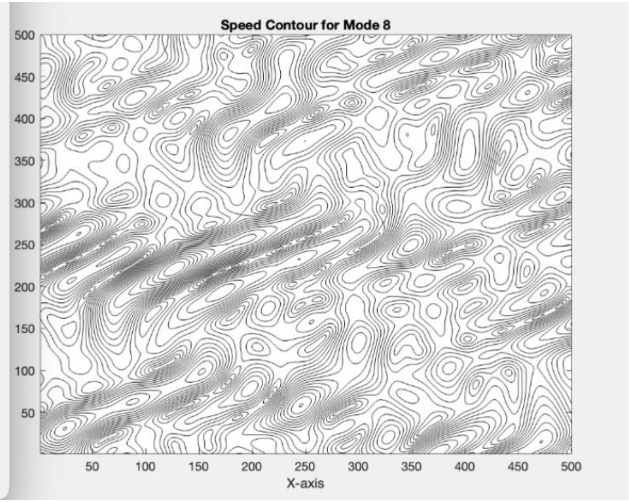


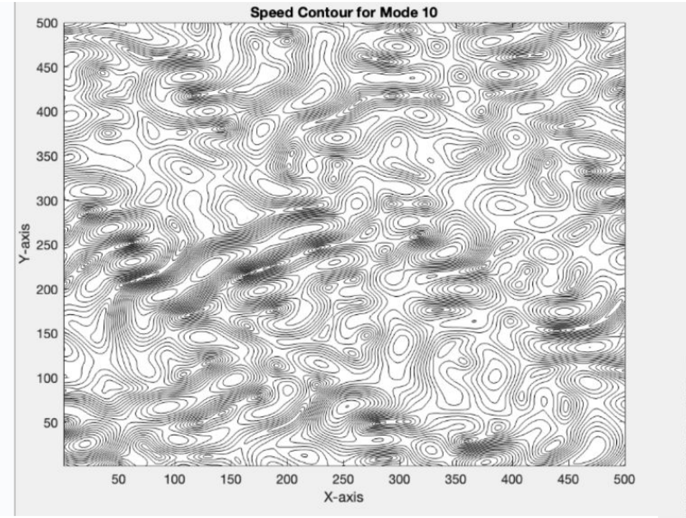
```cpp
// Function to calculate divergence in 2D
double calculateDivergence(int i, int j, const std::vector<std::vector<Vector2D>>& velocityField, double delta_x, double delta_y) {

    if (i == 0 || j == 0 || i == GRID_SIZE - 1 || j == GRID_SIZE - 1) {
        return 0.0; // Assuming zero divergence at boundaries or handle appropriately
    }

    double dU_dx = (velocityField[i + 1][j].x - velocityField[i - 1][j].x) / (2.0 * delta_x);
    double dV_dy = (velocityField[i][j + 1].y - velocityField[i][j - 1].y) / (2.0 * delta_y);
    return dU_dx + dV_dy;
}
```

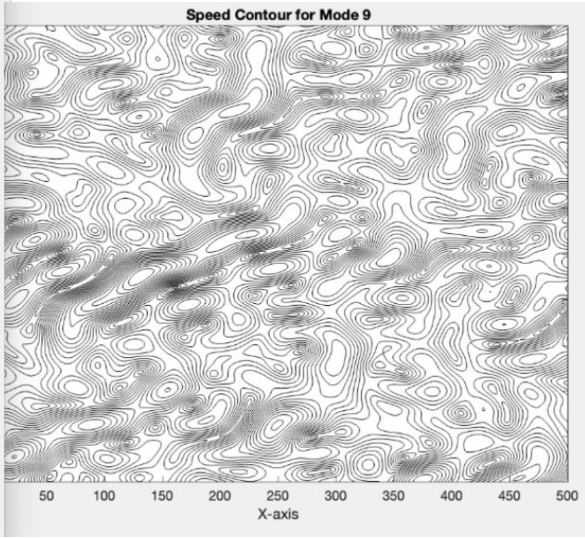
Incompressible flow → $\nabla \cdot \vec{u} = 0$

$$\nabla \cdot \vec{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y}$$

Streamlines:



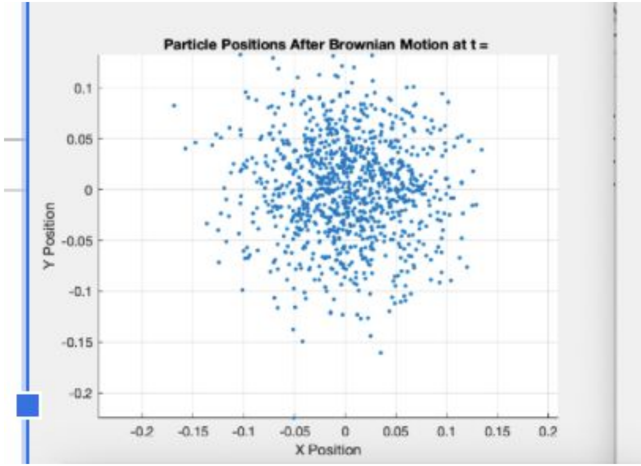




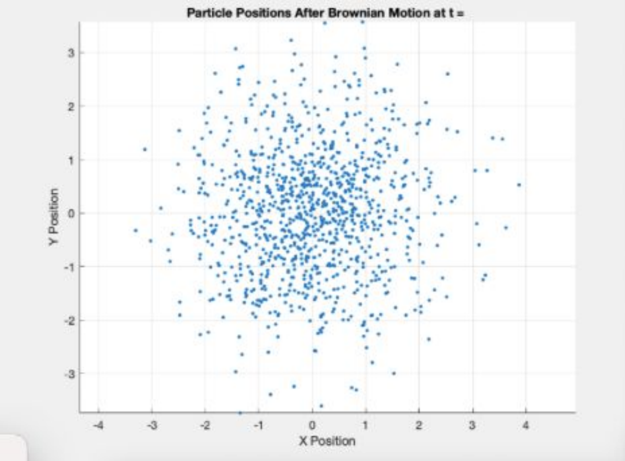
The contour lines become more intricate and tightly packed as the mode number increases. This indicates higher spatial frequency and more rapid variations in speed across space.

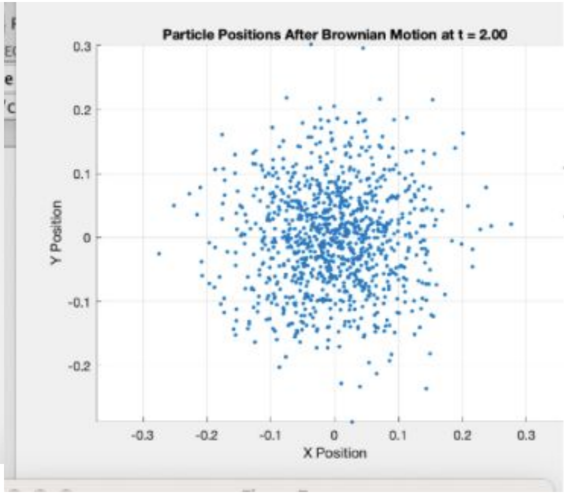
Now for the second part of our project, I plotted particles with Brownian motion. On the next slide I show the particles movement with different Brownian steps.

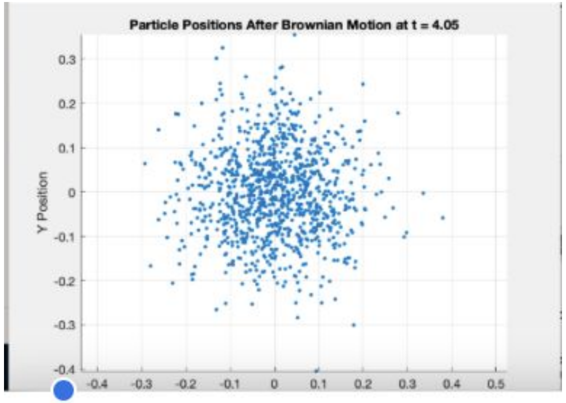
T = 0

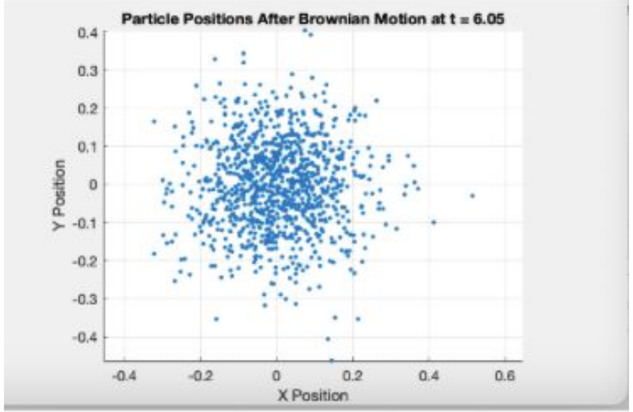

Particle Positions After Brownian Motion at t =
Y Position
X Position
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2

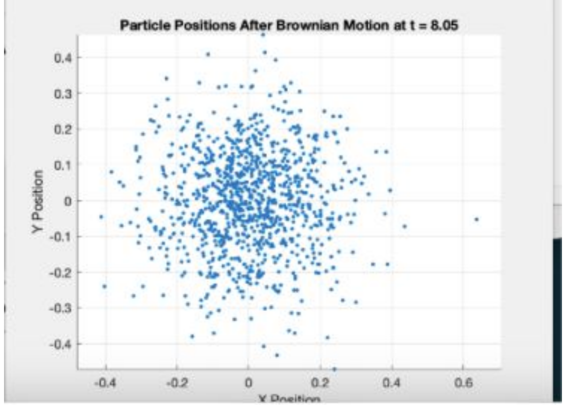
T = 1


Particle Positions After Brownian Motion at t =
Y Position
X Position
3
2
1
0
-1
-2
-3
-4
-3
-2
-1
0
1
2
3
4


Particle Positions After Brownian Motion at t = 2.00
Y Position
X Position
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.2
-0.1
0
0.1
0.2
0.3


Particle Positions After Brownian Motion at t = 4.05
Y Position
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


Particle Positions After Brownian Motion at t = 6.05
Y Position
X Position
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.4
-0.2
0
0.2
0.4
0.6


Particle Positions After Brownian Motion at t = 8.05
Y Position
X Position
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.4
-0.2
0
0.2
0.4
0.6

I then went into the Runge Kutta method.

The Runge-Kutta method in *Dentz et al. (2003)* is used to numerically simulate particle trajectories in a random velocity field — that is, to solve the stochastic differential equation (SDE) governing the motion of a solute particle in a spatially varying and random flow.

The extended Runge-Kutta scheme integrates this stochastic equation by:

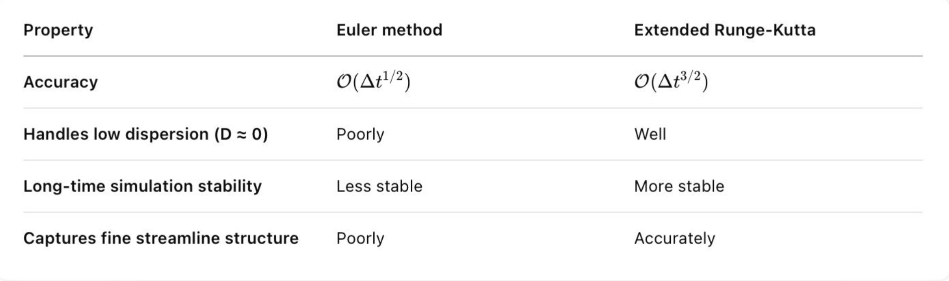
- Evaluating intermediate velocities and noise terms at multiple substeps
- Accurately tracking the trajectory of each solute particle over time
- Maintaining high-order accuracy, even when local dispersion (diffusion) is small or zero

The Euler-Maruyama method, the simplest integration scheme for SDEs, updates position as:

$\vec{x}(t+\Delta t)=\vec{x}(t)+\vec{u}(\vec{x}(t))\Delta t+2D\Delta t \cdot$ Gaussian noise

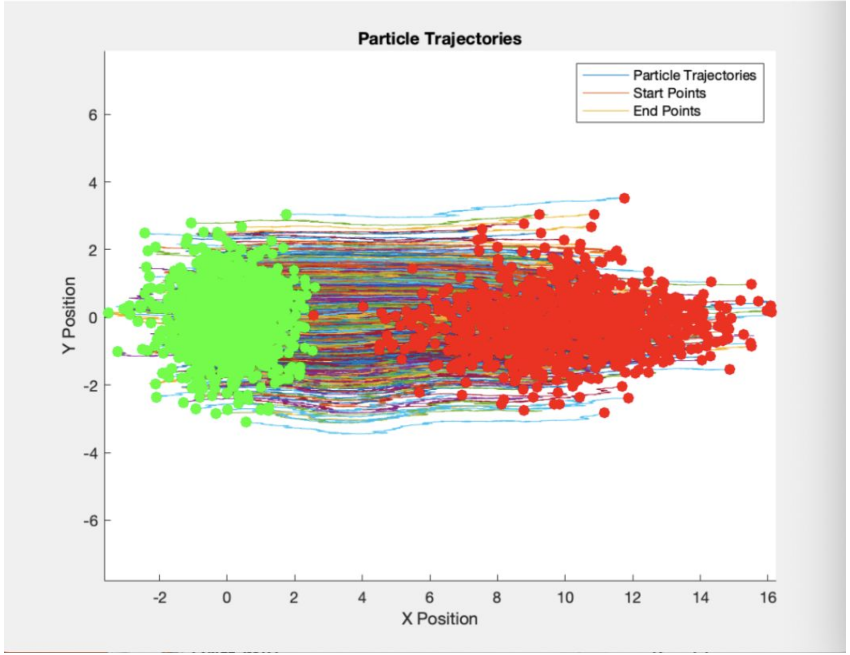
While simple, it's only first-order accurate and becomes very inaccurate in:

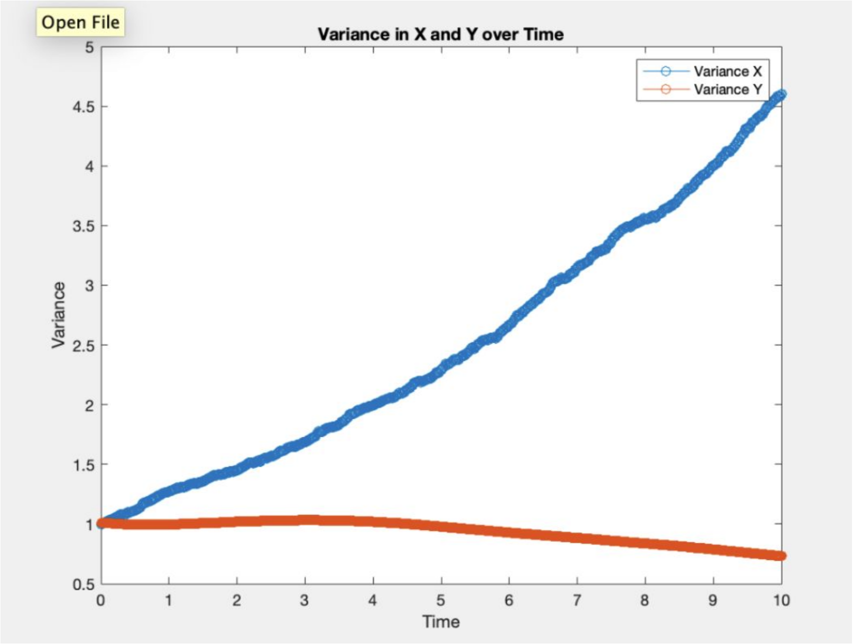
- High Péclet number flows (advection dominates over diffusion)
- Highly heterogeneous fields (large velocity gradients)
- Long simulation times



Property	Euler method	Extended Runge-Kutta
Accuracy	$\mathcal{O}(\Delta t^{1/2})$	$\mathcal{O}(\Delta t^{3/2})$
Handles low dispersion (D ≈ 0)	Poorly	Well
Long-time simulation stability	Less stable	More stable
Captures fine streamline structure	Poorly	Accurately

Runge Kutta results





Key Observations from Simulation Results

Brownian Motion Only (Euler-Maruyama – Slide 12)

- Particle spread is noisy and isotropic, consistent with random diffusion.
- No directional structure is observed as particle cloud expands outward symmetrically.
- Lacks streamline-following behavior → cannot capture real flow effects.
- As a result, it overestimates transverse dispersion and misses the trapping dynamics of real velocity fields.

Advection + Diffusion with Velocity Field (Runge-Kutta – Slide 14)

- Streamline-following trajectories emerge clearly.
- Particle movement is directional: stretching longitudinally (x) while remaining tight transversely (y).
- Variance vs. time plot shows:
  - Linear x-dispersion growth, matching theory
  - Suppressed y-dispersion, showing anisotropy
- Captures complex transport behavior like trapping and anisotropic spreading that simpler Brownian models miss.

These results confirm that the extended Runge-Kutta method accurately captures key transport phenomena in Gaussian random velocity fields. The suppression of transverse dispersion and emergence of streamline-following trajectories validate the importance of higher-order integration in simulating realistic flow behavior in heterogeneous porous media.

Conclusion/Limitations/Recommendations

- The extended Runge-Kutta method proved significantly more accurate than the Euler method for simulating solute transport in 2D Gaussian random velocity fields.
- Runge-Kutta better captured streamline-following behavior, preserved anisotropy, and produced dispersion trends that aligned with theory.
- The Euler method, by contrast, introduced numerical noise that obscured key physical phenomena like particle trapping and directional advection.

Limitations

- Simulations were limited to 2D systems, real-world porous media often require 3D modeling for full accuracy.
- High computational cost of Runge-Kutta makes large-scale simulations resource intensive.

Recommendations for Future Work

- Implement ensemble averaging over multiple realizations of the velocity field to obtain statistically robust dispersion metrics.
- Extend simulations to 3D fields and explore interactions between closed streamline regions and local dispersion.
- Incorporate adaptive time stepping to improve performance of Runge-Kutta methods in complex regions.
- Develop a hybrid scheme to blend efficiency of Euler with accuracy of RK in non-critical regions.

Project Scope

- This was an individual project.
- I was responsible for:
    - Coding the spectral method for velocity field generation
    - Implementing both Euler and Runge-Kutta solvers
    - Designing and running all simulations
    - Producing MATLAB visualizations and analyzing dispersion trends