Lattice Boltzmann Method – Kármán Vortex Street

Neil Ghugare

Physics 5810, The Ohio State University, Department of Physics, Columbus, OH 43210

April 24, 2026

1 Introduction and Problem Statement

The objective of this project is to simulate the Kármán Vortex Street, a classic fluid dynamics phenomenon characterized by periodic vortex shedding behind a blunt body [1]. This occurs when the flow instabilities lead to unsteady separation around an obstacle, such as a cylinder [1]. Capturing this behavior requires high spatial resolution and extensive temporal iteration, making it an ideal candidate for high-performance computing. We utilized the Lattice Boltzmann Method (LBM) due to its locality and inherent parallelism, which allows for efficient execution on a GPU.

2 Methodology: The D2Q9 Model

The simulation employs the D2Q9 (2 dimensions, 9 velocities) model. Unlike traditional Navier-Stokes solvers, LBM tracks the probability distribution functions f_i of particles moving in specific discrete directions [2].

The lattice constants, including weights W and unit velocity vectors CX and CY, define the discrete velocity space [2]. The evolution of the fluid is governed by the streaming and collision steps [2]. The collision is modeled using the Bhatnagar-Gross-Krook (BGK) relaxation toward equilibrium [2]:

$$f_i(\mathbf{x}+\mathbf{e}_i, t+1) = f_i(\mathbf{x},t) - \frac{1}{\tau}\left[f_i(\mathbf{x},t) - f_i^{\text{eq}}(\mathbf{x},t)\right] \tag{1}$$

where τ is the relaxation time, determining the kinematic viscosity $\nu = \frac{1}{3}(\tau - 0.5)$ [2]. We must keep $\tau > 0.5$ for our simulation to ensure numerical stability.

3 GPU Implementation

The solver is implemented in CUDA to leverage the massive parallelism of the GPU. Specifically, we utilize powerful NVIDIA H100 Hopper GPUs.

3.1 Memory and Streaming Strategy

To ensure efficient memory access, distribution functions are stored in a Structure of Arrays (SoA) format (`[direction][y][x]`), enabling coalesced memory transactions [2]. We implemented a *Pull Streaming* approach where each thread, mapped to a lattice node, pulls the particle populations from its neighbors [2]. This avoids race conditions inherent in "push" strategies [2].

3.2 Stability and Boundaries

Boundary conditions include a constant inlet velocity u_{inlet} at the left wall. No-slip conditions at the cylinder and channel walls are enforced using a *Bounce-Back* scheme, where distributions hitting a solid boundary (defined by a mask) are reflected in the opposite direction. To trigger vortex shedding in a numerically symmetric environment, the cylinder was placed slightly off-center ($y = ny/2 + 1$).

4 Analysis and Verification

The physical validity of the simulation is verified through the Reynolds number (Re) and the Strouhal number (St).

The Reynolds number is calculated as:

$$Re = \frac{u_{\text{inlet}} \cdot (2r)}{\nu} \tag{2}$$

where r is the cylinder radius (set to 4% of the domain height) and ν is the lattice viscosity [1]. This maintains a blockage velocity similar to the literature [3].

Instead of controlling the lattice viscosity directly as a simulation parameter, we utilize the relaxation time τ in Eq. 1 through the relationship [2]

$$\nu = \frac{1}{3}(\tau - 0.5). \tag{3}$$

The Strouhal number, representing the dimensionless shedding frequency f, is derived from probe data extracted behind the cylinder:

$$St = \frac{f \cdot (2r)}{u_{\max}} \tag{4}$$

where $u_{\max}$ is the free-stream velocity, given by

$$u_{\max} = u_{\text{inlet}} \frac{n_y}{n_y - 2r}, \tag{5}$$

where n_y is the number of y cells in the lattice (the y-domain or y-resolution) [1].

A Fast Fourier Transform (FFT) is applied to the time-series of the vertical velocity u_y to identify the dominant shedding frequency. This is done by placing a "probe" in the area of the vortex (we choose the middle of the entire domain), and storing the u_y values to be able to perform this FFT. Once we have found the frequency, we can find the experimental St value. The experimental Re value can be calculated analytically without running the simulation itself, based on the initial parameters.

The Kármán Vortex Street doesn't technically have an exact analytical solution that we can use to validate results. What we can do is use a study by C. H. K. Williamson, which analyzes this style of vortex shredding, specifically plotting St vs. Re [3]. We note that we need to keep $Re \lesssim 200$, since we are running a 2D simulation, and larger values of Re start opening up 3D terms [3]. We run 5 simulations with differing values of both τ and u_{inlet}, yielding the results in Fig. 1.

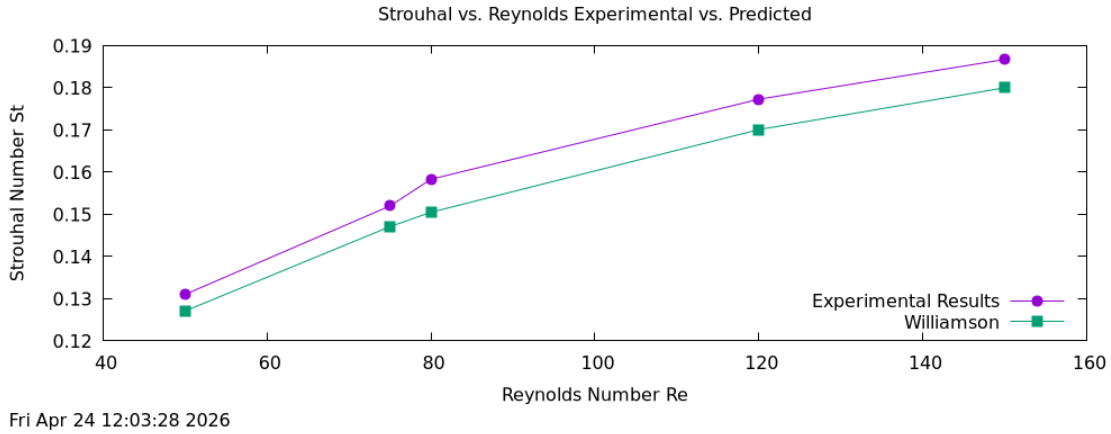

Figure 1: Comparison of simulated Strouhal numbers (St) against experimental data from Williamson (1989) for Reynolds numbers (Re) for the 2D regime of $50 \leq Re \leq 150$ [3]. We keep the resolution $n_x = 4000$ and $n_y = 1000$ fixed, and change τ, u_{inlet}, or both to show consistency across shifting parameter values.

The simulation results demonstrate excellent agreement with the experimental trend, capturing the characteristic increase in shedding frequency as the Reynolds number rises. The slight, consistent over-prediction of the Strouhal number (approximately 3–5%) is a hallmark of high-quality 2D numerical simulations. This discrepancy is primarily attributed to the "2D vs. 3D" effect: while physical experiments encounter three-dimensional spanwise instabilities that slightly dissipate energy and slow the shedding rate, 2D models constrain the flow to a single plane, resulting in a slightly higher, idealized frequency [3]. Numerical diffusion could also be a factor, as the simulation resolution may not be high enough to properly resolve the frequencies from the FFT for the experimental St number. The consistency of this offset across the entire range validates the solver's ability to resolve the underlying transport phenomena and confirms that the chosen lattice resolution and domain boundaries are sufficient for accurate Kármán Vortex Street modeling.

5 Lessons Learned

- **Memory Bottlenecks:** The transition from text-based (.dat) to binary (.bin) output significantly reduced I/O overhead during data snapshots. I also had to program efficiently to manage and output visualizations from the files without crashing the system due to RAM overuse.
- **Data Transfer:** Extracting single-point probe data from the GPU to the host at every step is a bottleneck that could be optimized by buffering on the device.
- **SoA vs. AoS:** Organizing data for coalesced access is critical for achieving high throughput on NVIDIA architectures, and learned how that applies to the D2Q9 methodology.
- **CFD Analysis:** Learned validation through Strouhal and Reynolds numbers, and generally how to validate CFD simulations with no true analytical solution.
- **GPU Programming:** Learned how to use NVIDIA's code suite (CUDA, NVCC, etc.) and how to program on a GPU itself using `.cu` files. Learned how to get a CPU and GPU to work together for simulating and outputting data.

6 In the Submission

In the submission zip file, there are the following files:

- `config.json`: The configuration JSON file that allows for easy updating of the simulation parameters.
- `helper.py`: Helper Python files to load JSON data for Python analysis.
- `strouhal.py`: Calculates experimental Strouhal and Reynolds numbers.
- `visualize.py`: Creates the MP4 visualizations from simulation output.
- `json.hpp`: JSON C++ file from Niehls Lohmann, included to get the simulation to run properly.
- `lbm_kernels.cu`: The CUDA file that implements the LBM method to run on the GPU.
- `main.cu`: The main CUDA file that runs the simulation and saves the output.
- `lbm_solver`: The latest LBM executable.
- `Makefile`: The makefile to compile the simulation.
- `plot_vortex.plt`: Plots the vortex from the last timestep dat file.

- `plot_strouhal.plt`: Plots the experimental Strouhal results to the predicted results.
- `load_osc.sh`: Loads the required modules and environments on the Cardinal desktop on the OSC.
- `.dat` and `.bin` files: In the zip file, I have included a small set of the dat files for a single simulation. All data files are not included to keep the zip file small in size. The bin file is for the main simulation output to keep the output small in size and easy to load (instead of the intensive dat file). The details of the data files are in Appendix A.
- `.png` and `.mp4` files: Likewise, I have included output PNG and MP4 visualizations. All visualizations are not given to keep the zip file small in size. Some PNG files are shown in Appendix B, but in the PDF the MP4 files cannot be viewed directly.

The submission can also be found on Github. No data files are on the Github.

7 Possible Further Work

There is a lot of possible further work with varying levels of difficulty and abstraction.

- **Simple Additions:**
  - Different Barriers: Allow for different kinds of barriers, like squares, walls, and more.
  - Barrier Placement/Sizing Control: Allow for unique placement and sizing of the barrier to control blockage percentage, instead of a static 4%.
  - More Strouhal Analysis: Run more simulations to do better analysis, and create a more rigorous version of Fig. 1. We could also resolve competing results of changing both τ and u_{inlet} better.
- **Complex Additions:**
  - 3D Mode Simulations: Transition from 2D to 3D to capture the "Mode A" and "Mode B" instabilities observed in the literature, especially at high *Re* [3].
  - Multi-GPU or Local Grid Refinement: Utilize multiple GPUs to overcome VRAM limitations, to run much larger domains to resolve the dynamics and Strouhal number better. Implement local grid refinement to increase accuracy near the barrier without the massive memory cost of increasing resolution for the entire domain.
  - Moving/Immersed Boundaries: Implement the Immersed Boundary Method (IBM) or a modified bounce-back scheme to allow the cylinder to vibrate or oscillate in response to the vortex shedding (Vortex-Induced Vibration).

References

[1] Blevins, R. D. (1990). Flow-induced vibration (2nd ed.). Van Nostrand Reinhold.

[2] Li, W., Fan, Z., Wei, X., & Kaufman, A. (2005). Flow simulation with complex boundaries. In M. Pharr (Ed.), GPU Gems 2: Programming techniques for high-performance graphics and general-purpose computation (pp. 747–764). Addison-Wesley.

[3] Williamson, C. H. K. (1989). Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers. Journal of Fluid Mechanics, 206, 579–627. https://doi.org/10.1017/S0022112089002429.

Supplement

A Data Files

The larger data files are not included in the zip file to save space.

A.1 Main Data File (`.bin` File) and Last Step File

Below is an example of the last time step data file. It stores the *xy*-coordinates in the grid and the magnitude of the velocity. This is used for Gnuplot or `visualize.py` in order to make outputs that we can see from the simulation.

```
# Last step data file for Karman Vortex Street Simulation for GNUPlot.
# x  y  velocity_magnitude
0 0 0.129717
1 0 0.0776971
2 0 0.0446465
3 0 0.0364366
4 0 0.0363942
5 0 0.0308166
6 0 0.0243736
7 0 0.0220436
8 0 0.020819
9 0 0.018771
10 0 0.0169558
...
```

The main data file (the `.bin` file) takes one of these and stores it over multiple time steps. We use the binary format to save storage space and keep I/O efficient.

A.2 Probe Data File

The following is the head of the probe data file (used for calculating Strouhal numbers experimentally). It tracks the time step and the normalized y-velocity u_y. It starts at $t = 5000$ to avoid some of the initial transients.

```
# Probe data file for Karman Vortex Street Simulation for Strouhal Number.
# t     uy/rho
```

```
5001 3.60998e-07
5002 2.82845e-07
5003 3.42393e-07
5004 2.82848e-07
5005 3.57283e-07
5006 2.90294e-07
...
```

A.3 Strouhal Data File

This is the manually-created `.dat` file that shows the experimental and predicted values for *St* vs. *Re*, for use with `plot_strouhal.plt`.

```
# Manually created .dat file to store experimental Strouhal values.
# Re St St_true
150 0.186724 0.18
75 0.152000 0.1471
120 0.177300 0.1701
50 0.131034 0.1271
80 0.15833 0.1505
...
```

B Visualizations and Snapshots

We ran 5 simulations with a fixed $n_x = 4000$ and $n_y = 1000$ resolution. We plot the following vortex outputs, keeping the colorbar constant to showcase the differences. In practice, you would scale the colorbar accordingly to see the vortex pattern better for each specific simulation result. For these simulations, we change the relaxation time τ, the inlet velocity u_{inlet}, or both. Our fiducial simulation (the one included in the zip file) is for $\tau = 0.6$ and $u_{\text{inlet}} = 0.1$.

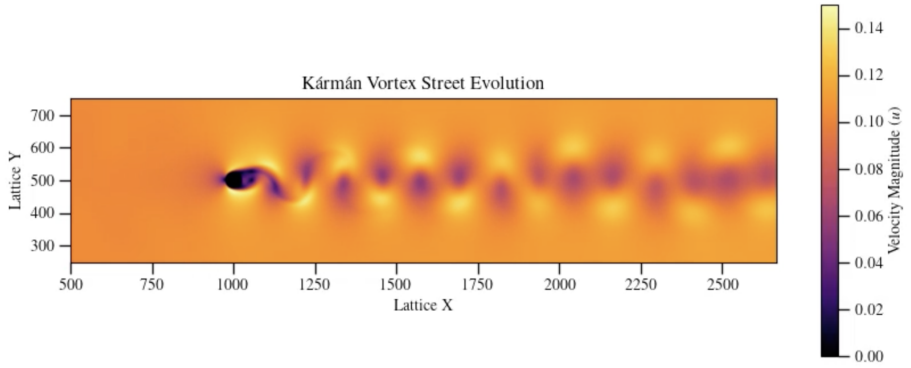

Figure B.1: Simulation with $\tau = 0.6$ and $u_{\text{inlet}} = 0.1$. This is our fiducial simulation for comparison with the other ones to check if we are getting the expected behavior. Analysis on this simulation yields $Re = 150$ and $St \approx 0.187$.

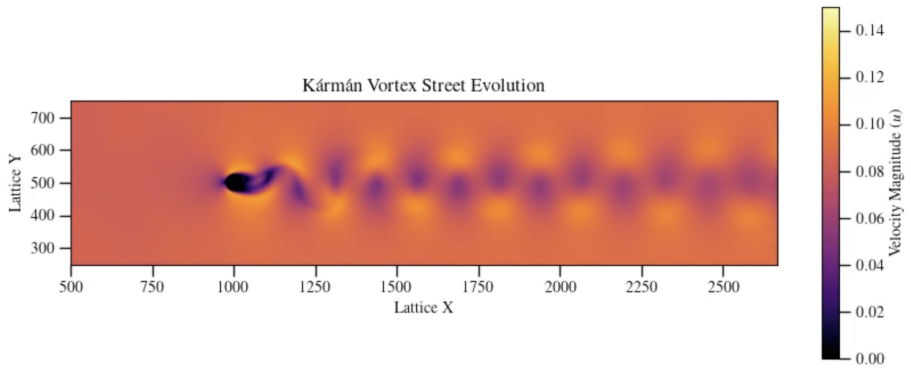

Figure B.2: Simulation with $\tau = 0.6$ and $u_{\text{inlet}} = 0.08$. As u_{inlet} decreases, we expect a decrease in Re and decrease in the aggression of the vortex shredding (more visible in the animation). As expected, analysis yields $Re = 120$ and $St \approx 0.177$.

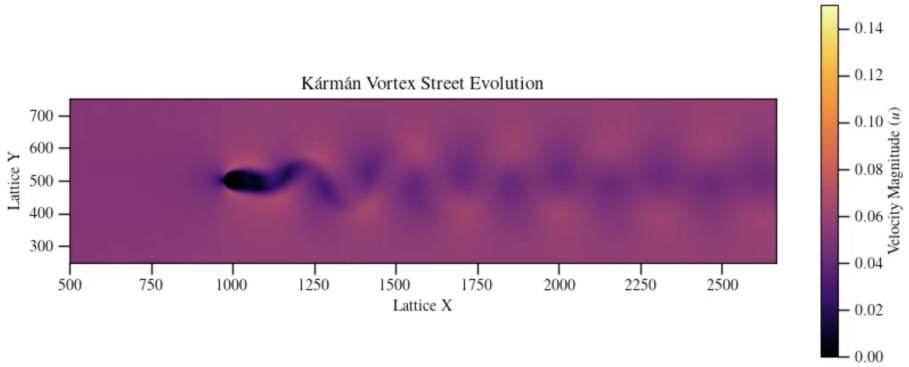

Figure B.3: Simulation with $\tau = 0.6$ and $u_{\text{inlet}} = 0.05$. As u_{inlet} decreases, we expect a decrease in Re and decrease in the aggression of the vortex shredding (more visible in the animation), and even more so than the prior simulation. As expected, analysis yields $Re = 75$ and $St \approx 0.152$.

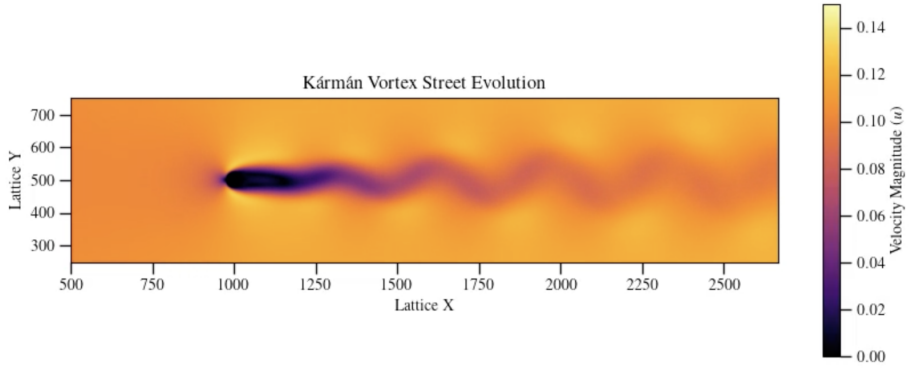

Figure B.4: Simulation with $\tau = 0.8$ and $u_{\text{inlet}} = 0.1$. Keeping u_{inlet} fixed but increasing τ makes the fluid "thicker". As such, we anticipate the Re to drop and that the vortex "dies off" a bit. By comparing to Fig. B.1, we can see how the intensity of the vortex has decreased (more visible in animation). As such, we yield the anticipated results with $Re = 50$ and $St \approx 0.131$.

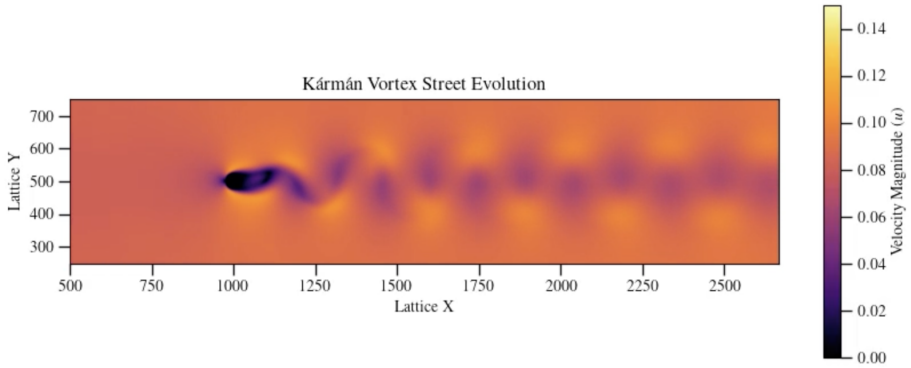

Figure B.5: Simulation with $\tau = 0.65$ and $u_{\text{inlet}} = 0.08$. With both values changed from our fiducial simulation, we can observe competing effects. However, for this instance, we "pull the lever" in the same direction for both. The *Re* value should drop due to increased τ and due to less u_{inlet}. As such, we expect an *Re* value less then Fig. B.2. We get a result of $Re = 80$ and $St \approx 0.158$.

Animations of these simulations allow us to see the initial transient period and the evolution and dynamics of the vortex over simulation time. One such MP4 of the fiducial simulation is provided in the zip file.