Interactive Real-Time Cloth Simulation with Directional Wind Drag

Author: Nathaniel Fargo

Abstract

This project implements a real-time simulation of a deformable cloth under the influence of gravity, spring forces, and aerodynamic drag. Users can interactively adjust wind direction and observe the dynamic behavior of a fabric-like mesh. The simulation visualizes both the cloth surface and the energy components (kinetic, gravitational potential, and elastic) in real time, demonstrating physical principles in an intuitive, engaging format. Designed to be computationally stable and visually informative, the simulation incorporates wind-based drag with directional surface response, replacing traditional velocity-based damping as the primary dissipative mechanism.

Methodology and Key Results

- System Modeling:
  The cloth is modeled as a 2D mass-spring grid, with masses connected by structural and shear springs. Spring forces are computed from Hooke's Law, with damping and external forces applied at each timestep.

- Wind Interaction:
  Wind is applied as a relative velocity vector between the wind and the particles. Aerodynamic drag scales with the square of this velocity and includes a directivity factor based on the angle between surface normals and wind direction.

- Stability and Sub-stepping:
  Integration is performed using Euler's method, with adaptive sub-stepping to ensure stability for stiff springs.

- Visualization:
  The simulation uses Matplotlib for both 3D rendering and live plotting of energy over time. Sliders allow real-time adjustment of wind force. Energy graphs display kinetic, gravitational potential, elastic, and total energy with automatic rescaling.

- Damping Philosophy:
  Instead of heavy velocity damping, aerodynamic drag is used to stabilize the system while preserving dynamic motion.

Files Submitted

- cloth_sim_rt.py - The full Python implementation of the interactive cloth simulation, including wind sliders, energy tracking, and real-time animation.

Use of Artificial Intelligence

This project incorporated the use of ChatGPT (OpenAI GPT-4) and Cursor IDE (AI-enhanced programming tool) to guide development and debugging. Prompts included requests for 'implementing wind force with surface normal directivity,' 'debugging spring instability,' and 'enhancing Matplotlib 3D visualizations with sliders.' AI assistance was most useful during code structuring, identifying unstable integration steps, and designing the aerodynamic model. However, significant manual debugging and testing were required-particularly for visual artifacts, numerical instability, and maintaining physical realism under user-controlled wind conditions. The final functionality, particularly the wind-based drag model, was refined iteratively with trial-and-error and physics intuition, rather than directly produced by AI suggestions.