Wind Tunnel Aerodynamics: A Real-time Navier-Stokes Simulation Approach

Alexandru Petrusca
University of California – Santa Cruz
apetrusc@ucsc.edu

Abstract – **Aerodynamic testing is a primary concern for companies specializing in aircraft and automotive manufacturing. Physical testing of aerodynamic forces on transportation vehicles is commonly achieved with the aid of wind tunnels, large tubes with air moving inside used to copy the actions of an object in flight. This project focuses on digitally simulating the operations of a wind tunnel and developing different techniques to visualize the air flow within the tunnel as it interacts with obstacle objects. The project is composed of two main parts: the simulation and the visualization. For the simulation, the Navier-Stokes equations are used to model the air flow through the tunnel. For the visualization, my application incorporates heat maps to visualize wind velocity and pressure, and uses texture advected material to visualize the movement of wind inside the tunnel in two dimensions.**

INTRODUCTION

Aerodynamics is everywhere: a jet airplane streaking though the sky, a race car speeding through a race course, a skyscraper experiencing a gust of wind, and the dispersal of seeds from a dandelion. Underlying all of them is the flow of fluids around a solid obstacle. All are phenomena that we would like to simulate realistically in interactive visualization applications for the purpose of aerodynamic analytics.

The goal of this project is to scratch the surface of accurate aerodynamic simulations in order to get a rough sense of how the particular design of an object affects its interactions with wind flows around it. Its primary purpose is to communicate a practical perspective on aerodynamic design and provide an intuitive sense of how turbulent flows form around an object. To accomplish this, this application runs air flow simulations in real-time and allows for arbitrary obstacle geometries. Through experimentation with various obstacle geometries and orientations, it is possible to gain insight into what geometries are best suited for real-world aerodynamic purposes.

RELATED WORKS

The simulation of a fluid around an obstacle object is a fundamental concept in computational fluid dynamics. This field is highly concerned with running computer simulations over the Navier-Stokes equations, a set of differentiable equations that describe the motion of viscous fluid substances. These equations have been known since the 18th century and arise arise from applying Isaac Newton's second law to fluid motion. They have historically been used to describe fluids mathematically and make predictions on real world scenarios involving fluids. In practice, the Navier-Stokes equations are too difficult to solve analytically. Over the years, engineers and mathematicians have made many approximations and simplifications to the equation set to allow for fast analytical and numerical solutions to be possible

With the advent of the digital computer and the start of the digital era, scientists have employed powerful computers to solve approximations to the equations using a variety of techniques. Among the use cases for these equations in the modern era

were computer-based wind-tunnel simulations. Some of the earliest papers on Navier-Stokes based wind tunnel simulation come from NASA. A 1988 paper titled "Navier-Stokes Simulation of Wind-Tunnel Flow Using LU-ADl Factorization Algorithm" describes a simulation of a wind-tunnel using walls and objects that comply with the no-slip and adiabatic wall conditions for flow simulation. These simulations were used by NASA to test aerodynamic wind flows around a plane wing model. It ran on an Amdahl 1200 supercomputer having 128 Mbytes main memory.

Another 1987 paper titled "Numerical Simulation of Turbulent Flows Using Navier Stokes Equations" discusses the industrial implications of Navier-Stokes based wind tunnel simulation as well as the limitations of such simulation techniques. The paper also describes fast ways to simulate turbulent air flows by employing back-then recent advances in vector computing.

Nowadays, powerful software suites such as Autodesk Flow Design, MicroCFD, and SOLIDWORKS Flow Simulation can perform industry-grade wind tunnel flow simulations using a variety of different visualization techniques. These simulations are so precise and convenient that they almost replace the need for physical wind tunnels entirely. Although computer simulation of wind flows through air tunnels have advanced significantly in recent years, aerodynamic vehicle manufacturers still often times prefer using physical wind tunnels to test prototype vehicles. They often cite precision of measurement as a key motivator to use physical rather than simulated wind tunnels.

TECHNICAL DETAIL

I. Simulation

This component of my final project comprises of implementing an accurate simulation of 3d fluid dynamics which accounts for turbulent flows and objects as obstacles in the flow. To accomplish this, I use the Navier-Stokes equations to model the air flow inside the tunnel. The direct numerical solution of the Navier-Stokes equations for turbulent flow is extremely difficult however. To get anywhere close to a real-time simulation with turbulence, it is necessary to employ a stripped-down numerical-approximation of the Navier–Stokes equations. To solve the equations and acquire a description of the air flow inside the tunnel, I use GPU calculation done via WebGL in JavaScript.

To achieve an accurate representation of obstacles, objects inside the wind-tunnel walls act as obstacles to the flow and cause turbulent flows to arise. In interactions with the flow, obstacle objects obey the no-slip and adiabatic wall conditions, such that no matter can pass through the obstacle and the velocity of flow on the obstacle's surface is zero.

To solve the Navier-Stokes equation numerically and operate my simulation, I use the following pipeline of operations: first I load my obstacle objects from jpeg image files stored in my "assets" folder. These obstacle images represent the obstacle as black pixels on a white background (black meaning that the pixel is an obstacle and white meaning that the pixel is not an obstacle). Every obstacle image is exactly 1500px by 750px, which is the size of my HTML canvas.

I then go into a render loop where I first advect my velocity field to get an intermediate velocity field, then apply fluid diffusion over the course of 20 Jacobi iterations (consequence of fluid's resistance to motion), then compute a pressure gradient, and subtract this pressure gradient from my intermediate velocity field to get my next velocity field to use in my next render iteration. Finally, after computing this new velocity field each cycle, I also advect a material (dye) in my fluid (that originates from the leftmost edge of the canvas screen) using my current velocity field. This process repeated over and over again gives me all the complex behavior of a fluid moving over time around my obstacle object. To achieve a real-time rendering, I perform all of the necessary Navier-Stokes calculations on a grid that is a sixth of the size of my HTML canvas. All necessary calculations are done on the GPU using a

programming methodology that treats textures as arrays and fragment shaders as loop bodies that iterate over these texture arrays. This allows for the fast computation of the Navier-Stokes equations on the GPU and is another reason why I was able to get my simulation to run in real-time.

Once I have the data from my simulations, I can begin visualizing it with a host of different techniques.

II. Visualization

To begin visualizing the data, I implement a graphical user interface to display objects and air flows inside the tunnel. To do this, I employ a HTML5 canvas element with WebGL. For my application, I implement two primary methods to visualize the air flows inside the wind tunnel.

First, I use texture-advected material to represent air flow. In this method, "dye" material steadily injected into the fluid at a fixed point (leftmost edge of screen) extends along the path that the fluid takes in time. The path this dye takes over time represents the direction and general shape of particular flow bands through time.

The color of this dye is specially coded for in my application. Regions of dye that form dense clusters turn red in color while sparser regions of dye change to a green color. The sparsest of dye regions appear blue-green and blend in with the blue color of the background. Thus, with this color encoding, red indicates regions of high dye density while green represents regions of sparse dye density and blue represents regions of no dye density.

Second, I use heat maps to represent air flow. In this method, "hotter" regions denote high wind velocity whereas "colder" regions denote low wind velocity. Similarly, when visualizing pressure "hotter" regions denote high wind velocity whereas "colder" regions denote low wind velocity. I use a standard rainbow transfer function to color my heat map and give this "hot" vs "cold" appearance.

RESULTS

Results from the final application proved mixed. Most of the problems come from the simulation and not the visualizations. The visualizations look fine and use many of the techniques we talked about early on in this class. They give you a good understanding of which designs are good aerodynamically and which aren't.

For example, I imported three geometries of plane wings into my application to test which one would fare the best in my wind tunnel. As I expected and is in line with physical expectations, longer, thinner teardrop-shaped did better than more bulbous and circular wings. I could tell this by seeing less turbulent flow on the longer, thinner wing compared to other wings. The vortices being shedded were much smaller than those of worse performing wings.

In addition, the visualizations for wind velocity and pressure turned out very. They ended up giving me crucial insight into the aerodynamic properties of the wings. The wind velocity heat map visualization allowed me to spot regions of high velocity flow and pockets of low velocity flow easily. The pressure heat map visualization likewise did the same. Vortices could be visualized in v

On top of this, every single wing outperformed object such as circles and squares, which are quite un-aerodynamic. This leads me to believe that my simulation at least captures a little bit of the physical interactions present in the real world.

The major problem with my simulation is the resolution that it runs at. In order to get my Navier-Stokes simulation to run at real time on the GPU, sacrifices had to be made concerning the resolution of the vector velocity field and texture advection being computed every GPU cycle.

While my application renders a 1500px by 750px scene to the user, it does not solve the Navier-Stokes equations for each pixel in this scene. Instead it only solves the Navier-Stokes equations for a grid that is a sixth of the size of the scene. The solutions for each of these grid points are bilinearly interpolated across my scene. This leads

to many apparent miscalculations and mis-approximations in my simulation. These can be more clearly seen when analyzing the boundary between my obstacle objects and their surrounding fluid. The boundaries of the obstacles look jagged and pixelated because they are being up-scaled from a lower resolution. Unfortunately, finding a solution for this problem proved to be too hard to implement. This restricts me from getting precise surface information on my obstacle objects without increasing the grid size and in effect making my application lose its real-time execution.

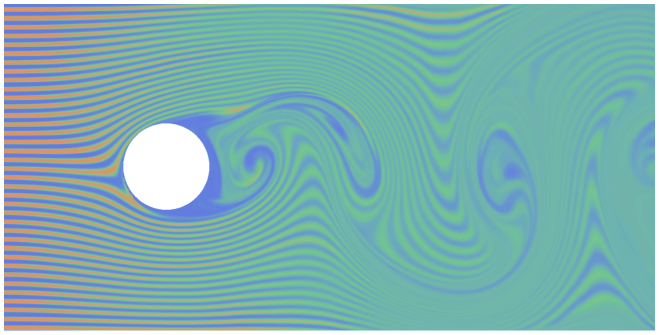

FIGURE 1
MATERIAL BEING ADVECTED IN APPLICATION AND FLOWING AROUND CIRCLE OBSTACLE

FIGURE 2
HEAT MAP OF WIND SPEED AROUND A 2D GEOMETRY OF MINI-SUV VEHICLE

CONCLUSION

Numerical solutions to the Navier-Stokes equations lead to satisfying visualization and simulation results. Interactive flow visualization applications such as this one help users grasp the concept of aerodynamics and aerodynamic design better. Such applications also provide a platform to compare and contrast different object designs based on their aerodynamic properties and interactions with wind flows. Therefore, it is feasible that these sorts of applications could be used for educational purposes to help interested students and even aerodynamic engineers understand the fundamental principals behind aerodynamic designs and turbulence.

REFERENCES

[1] Schmidt, Wolfgang, "Numerical Simulation of Turbulent Flows Using Navier Stokes Equations," *Perspectives in Turbulence Studies,* May 11–12, 1987, 473-497.

[2] Obayashi, Shigeru, "Navier-Stokes Simulation of Wind-Tunnel Flow Using LU-ADI Factorization Algorithm," Nasa Technical Memorandum 100042, Feb 1988, 1-29