Modelling Physics Problems On An Augmented Reality Sandbox

by

Ethan Denny

A thesis submitted to the

Department of Computer Science

to fulfill the requirements for the degree of

Bachelor of Science (Honour's).

Memorial University of Newfoundland

St. John's, NL, Canada

April 9, 2025

St. John's Newfoundland

Abstract

Most research on augmented reality (AR) sandboxes has been done in geoscience and related fields, specifically how sandboxes' use in visualizing complex problems can facilitate education and outreach. However, this research has been narrow in scope and has yielded mixed results on the educational benefits of AR sandboxes. Our work takes a different approach by exploring problems from physics, notably the heat transfer equation and simple models from meteorology for predicting wind trajectories. We constructed and demonstrated several interactive visualizations using an augmented reality sandbox. This involved re-imagining how the sandbox could be used as an input (e.g. having the sand represent conductivity instead of a topographical surface) and creating more complicated simulations than have been used in previous work. By extending this technology to a new field, we aim to demonstrate its potential for enhancing conceptual understanding in physics and to encourage further exploration of augmented reality in science education.

Acknowledgments

Thank you to my supervisors, Dr. David Churchill and Dr. Joseph Fitzgerald. Their constant and continuing support, patience, and encouragement have made this process a valuable and productive experience, as well as a personal joy.

Thank you to Memorial University of Newfoundland, the Faculty of Science, and the Department of Computer Science. Without the help of their faculty and administration's teaching and support, I would not have had the knowledge or resources needed to complete this project.

Thank you to my family, friends, and loved ones. Your love, kindness, and support have allowed me to go far beyond what I could have ever accomplished alone. There is no way I can repay all that you have done for me.

Contents

List of Figures

Chapter 1

Introduction

1.1 Augmented Reality

Augmented reality (AR) presents unique opportunities for human-computer interaction, scientific research, and education. Because AR may be presented in many different ways, people may experience it in multiple forms without recognizing the commonalities between these experiences. Products like Apple's Vision Pro imagine augmented reality as something which is worn, while other companies like Avalon Holographics are creating technology that offers users a way to view three-dimensional scenes with real depth through holographics and lightfields. Both of these technologies are examples of emissive augmented reality — which, because they require expensive, cutting-edge displays, are less accessible to laypeople and researchers. However, there are other AR technologies that are much cheaper and easier to build with or develop

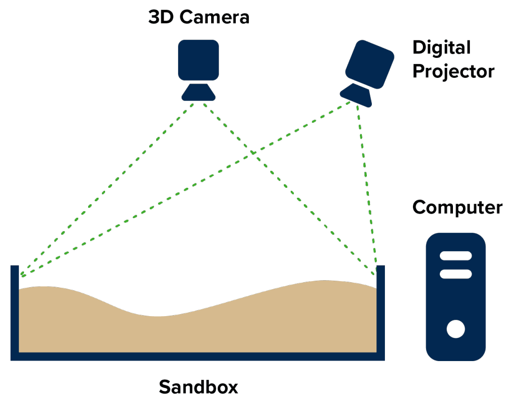

Figure 1.1: A diagram showing how the AR sandbox is constructed, (UC Davis, n.d.)

for. One example of this is an augmented reality "sandbox", which provides a cheap and unique way to create interactive physical simulations.

1.2 Augmented Reality Sandboxes

Fundamentally, an augmented reality sandbox is made of up of three components:

- A box with low sides filled about 2/3 of the way to the top with pale sand and which comes to approximately waist height to facilitate manipulation of the sand.
- An image projector which is mounted above the box.
- A camera capable of creating a depth map of the sand below, also mounted above the box.

The camera creates a depth map of the sand, which is then passed to software (see Figure 3.1). The software in the computer processes the data and outputs an image (typically the result of some kind of shader or simulation) to the projector, which is then projected back on to the sand. Since the sand is very pale, it takes on the colours of the projection, creating the illusion that the sand has been coloured. Users of the sandbox can move the sand with their hands or other tools, creating a pleasing visual effect as the projected image updates to reflect the new depth map.

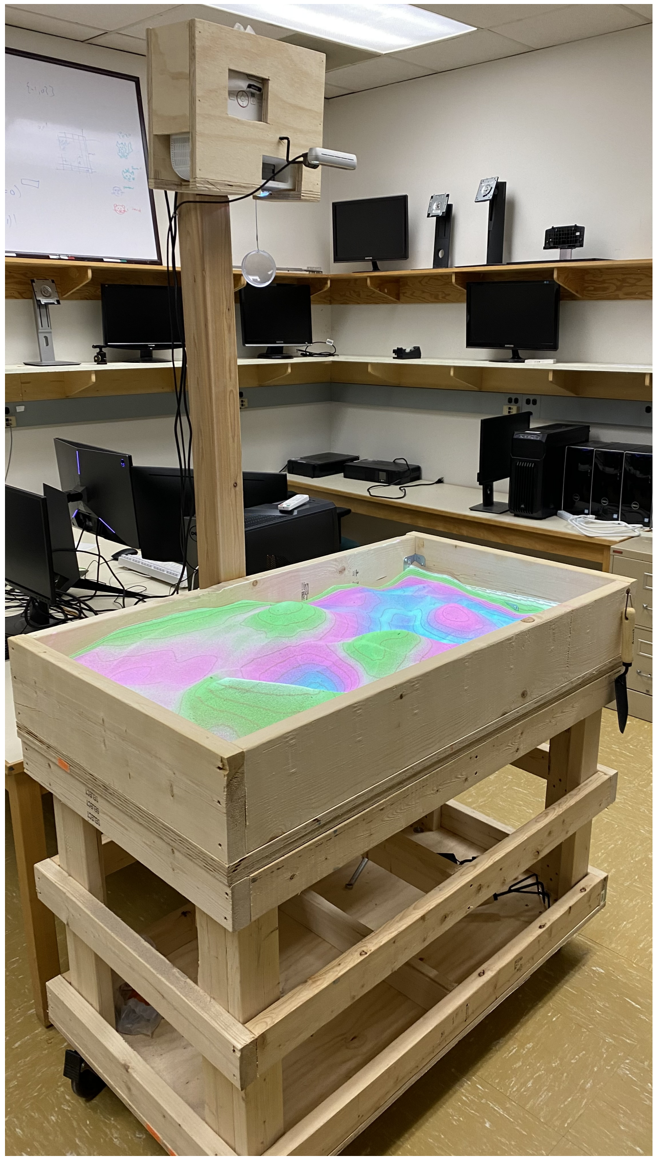

Figure 1.2: The AR sandbox used for this research, housed in MUN's AI & Games Lab

Chapter 2

Background & Related Work

2.1 Augmented Reality Sandboxes

The first augmented reality "sandbox" did not use sand at all. Created by Hiroshi Ishii of MIT's Media Lab and first published about in 2004, the IlluminatingClay and SandScape projects used clay and glass beads, respectively, along with a Minolta Vivid-900 laser scanner to produce impressive visualizations (I. et al., 2004). However, the scanner was very slow (approx. 0.8 frames per second) and cost tens of thousands of dollars (Minolta, 2006).

A major breakthrough - in quality, public awareness of the technology, and scientific impact - came in 2014 with Oliver Kreylos' work at UC Davis (Kreylos, 2023). Krelylos' sandbox used an Xbox Kinect sensor (released in 2010 (Whitworth, 2010), after Ishii's research), which was a major leap forwards in both interactivity and affordability. The development of an easy-to-use software library, SARndbox (Kreylos, 2022), made the technology even more accessible to both the public and researchers. Kreylos' research was centred on the geophysical educational opportunities of the sandbox, a thread that would be continued by future research. Since the release of Kreylos' sandbox blueprints and software, many labs, organizations, and companies have built their own versions of his design. The sandbox at MUN's AI & Games Lab is not based on Kreylos' design, nevertheless, his work remains the most accessible entry point for those looking to construct and use their own sandbox.

2.2 Using Numerical Methods for Physics Modelling

There are many equations in physics for which no known closed-form solution exists - that is, the equation cannot be expressed as a finite combination of symbols, functions, operators, etc. Take for example the sum of inverse squares:

$$1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25}$$

is not a closed-form solution, because it requires an infinite number of terms. However, with some basic analysis, we can easily find that:

$$1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} = \frac{\pi^2}{6}$$

$\frac{\pi^2}{6}$ is a closed-form solution, and is thus much easier to work with.

Unfortunately, many equations are not so simple. The approach taken with the sum of inverse squares above is known as an "analytical" method, but as equations get more complex, the problem of simplifying them in this way becomes mathematically impossible. Instead, physicists and computer scientists turn to "numerical" methods. Suppose we have a complicated integral, where there is no known closed-form solution, or where finding one requires substantial effort. In this case, it is often more useful to proceed with numerical methods, by computing the integral at a range of points and then using this to approximate its area. In this way, the error can be controlled, and a satisfactory result can be obtained without excessive analytical work.

2.3 *A Numerical Method for Predicting the Perturbations of the Middle Latitude Westerlies*

As related by Joseph Smagorinsky in his account of the history of weather prediction through numerical methods, the "formation of the Meteorology Group at the Institute for Advance Study (IAS) in Princeton and its first numerical forecasts on the Electronic Numerical Integrator and Computer (ENIAC) were key events in the early history of numerical weather prediction" (Smagorinsky, 1983). Smagorinsky went on to explain that it was Jules Charney who was responsible for many of the early work in applying numerical methods to weather prediction.

In 1949, Charney (along with Arnt Eliassen) authored the paper "A Numerical Method for Predicting the Perturbations of the Middle Latitude Westerlies", which "gave the results of one-dimensional predictions (along a latitude band)" (Smagorinsky, 1983). This simple model relied on a few key constants and some simple functions to produce a very effective and efficient forecast of the 500 millibar (mbar) pressure surface.

The 500 mbar pressure surface is the area in the atmosphere halfway between sea level and the edge of space. The reasons this is a good measurement for meteorological purposes are outside the scope of this thesis, but its main attraction is that it can easily be approximated as a vector field by sampling arbitrary points and considering the vector between them, which provides an excellent basis for visualizing wind trajectories.

2.4 Applications for Education

Research into augmented reality sandboxes has heavily focused on topographical lab activities. (M. et al., 2020) centered their research on administering different kinds of lessons (unstructured, structured, and semi-structured) to one group of university students, while another group received more traditional instruction. They did not find any significant difference between the labs that used the sandbox and the labs that did not, echoing the results of (G. et al., 2017), who found that while the sandbox incited interest, a short session (approx. 20 minutes) with it did not lead to an appreciable difference in learning outcomes. A recent, much more extensive study, (B. et al., 2023), also found no major benefits to using an augmented reality sandbox to teach geoscience labs. They performed a semester-long study of an entire class, with many controls to account for some suspected deficiencies in the methods of the previous two papers, but still failed to produce a positive result.

Other researchers have used less traditional approaches to measure the sandbox's effectiveness for education. (S. et al., 2020) placed electrodes on students' skin, interpreting the measurements as being representative of the students' engagement. They concluded that the sensors were indeed useful as a measure of engagement, and were able to show at least some value in the use of structured activities using the sandbox. However, we felt that their data and results were unconvincing, as their "curve fitting" seemed to oversimplify their observations. A more thorough review or comparison of studies using the same sensor method may be necessary. (G. et al., 2020) focused on the benefits that an augmented reality sandbox could bring to younger children (4- and 5- year-olds). They found value in the sandbox, but we felt that a sample size of $n = 4$ was far too small to be significant.

(Y. et al., 2020) did a thorough investigation into not just how students learn, but how researchers measure and draw conclusions from students' activities. They were highly critical of past work in the field, while still drawing lessons from it, and presented a new framework (constellations) for measuring students. Their research involved multiple sleek, well-made sandboxes, which let more students interact at the same time and may have increased students' willingness to take the research seriously.

Chapter 3

Methodology

3.1 The AR Sandbox

The sandbox used for this research is not bespoke. It was constructed for a series of research projects being conducted at MUN's AI & Games Lab. There is nothing remarkable about its construction: it was built with regular lumber and tools and typical sand, all of which were acquired from local hardware stores. Nonetheless, it is still as useful as any other sandbox design, because most of the benefits comes from the software and projection, not the sand or other aspects.

3.1.1 Software Implementation

The software used on the sandbox is also custom (C. et al., 2024), mostly written by my supervisor David Churchill with additional contributions from the students

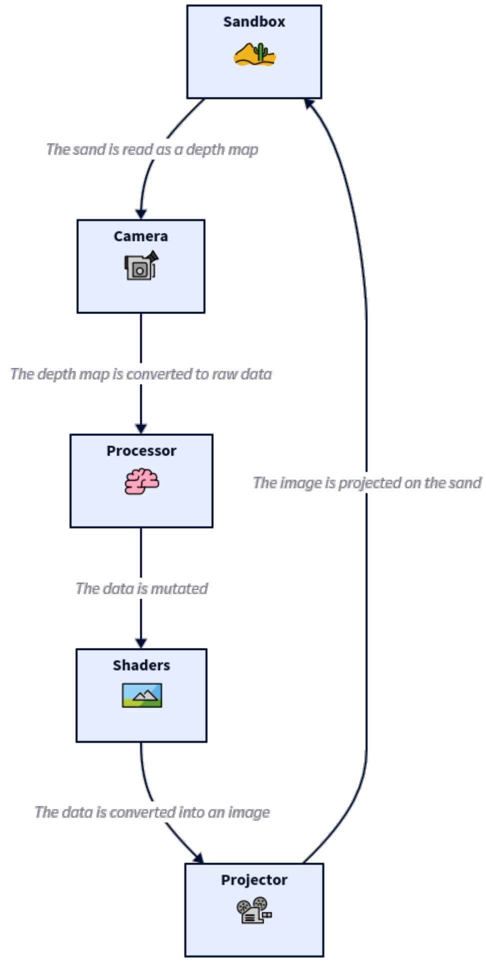

Figure 3.1: A diagram showing how the software powers the sandbox (Emojis courtesy of OpenMoji, CC BY-SA 4.0)

working in his lab, including myself. The software is very efficient both in terms of frames per second and in terms of development time. It is built around the concept of “processors”, which provide simulations for different tasks — some simply map the height of the sand to different colours so that it looks like a landscape, some integrate with other programs like *Minecraft*, while the ones we have worked on are used to simulate solutions to physics problems.

3.1.2 Shaders

One of the keys to ensuring fast processing on the sandbox is the use of shaders. Shaders can do heavy graphics processing in a massively parallel fashion, by taking advantage of both the dedicated hardware of the GPU and by including only a limited set of APIs compared to more ”standard” programming languages. The shaders used for this project are simple fragment shaders that use a single channel to represent the “value” of a point. This “value” is typically representative of the height of the sand — but may be modified (possibly along with additional channels) to represent other values. These value(s) are then processed in the shader, which creates a fully coloured image which is then fed to the projector.

3.1.3 Contour Lines

One of the most useful pre-existing features the sandbox had was its “contour lines”. These are roughly analogous to topographical lines on a map, and helped distinguish different heights on the sandbox. Figure 3.2 demonstrates this effectively — note the black lines on the second image. While contour lines are most often found on topographical maps, visually separating different colour ranges has proven useful in other areas as well such as the heat equation, especially when similar colours make it hard to distinguish how values are changing.

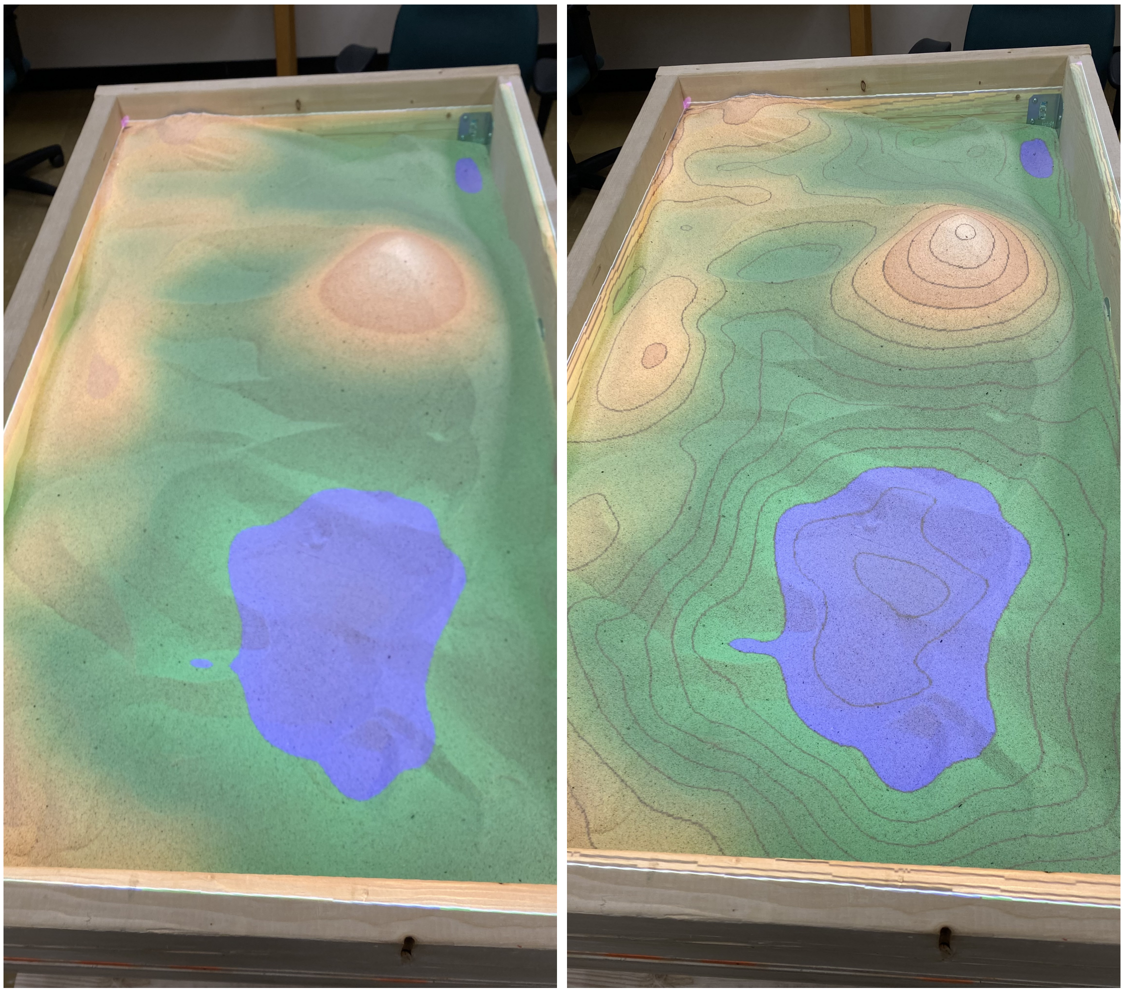

Figure 3.2: On the left, an image of the sandbox without contour lines. On the right, an image of the sandbox with contour lines

3.2 The Heat Transfer Equation

The first physics problem we modelled on the sandbox was the "heat transfer equation", a simple problem that any physics undergrad should be familiar with. We selected it because despite being straightforward to solve, it may be challenging for students to conceptualize how heat will spread in complex situations.

3.2.1 Heat Transfer Implementation

Consider a bowl of hot soup — if I drop an ice cube in it, how will the heat (or lack thereof) spread through the soup? More generally, how fast does heat spread through different materials, and what are the behaviours it exhibits? To explore this, we reimagined the sandbox's height map input as the k-value of the materials at a given point in a system. This value is representative of how fast a point will reach temperature equilibrium with its neighbours. The numerical method for determining the temperature T at some point (x, y) at some time step $t + \Delta t$ given $T_t(x, y)$ (the temperature of the point at the previous time step) is:

$$T_{t+\Delta t}(x, y) = k \cdot (H + V)$$

$$H = \frac{T_t(x - \Delta x, y) - 2T_t(x, y) + T_t(x + \Delta x, y)}{\Delta x^2}$$

$$V = \frac{T_t(x, y - \Delta y) - 2T_t(x, y) + T_t(x, y + \Delta y)}{\Delta y^2}$$

and where Δx, Δy, Δt represent the distance / duration between discrete points in both space and time. This formula easily translated to code and we quickly built a simple simulation.

A few assumptions were made for this simulation. First, k was restricted to a range of $[0, 1]$, which is how the height of the sandbox is represented. There is no particular physical phenomenon this is meant to represent, it was merely an easily accessible range of values that led to useful results. This will be a pattern throughout this work - eschewing extreme accuracy in physical simulations in favour of directing more effort towards improving visualizations and evaluating the sandbox's educational possibilities. Further work could easily improve the simulation, which would almost certainly have a positive effect on both the visualizations and any educational activities, but this effect cannot be realized without building those visualizations and activities in the first place. The second assumption we made was to the edges of the sandbox as boundary conditions with a temperature of $0\,^{\circ}\mathrm{C}$. All heat sources were considered to be rectangles held at $100\,^{\circ}\mathrm{C}$. These values were easy to understand (ice vs. boiling water) and, more importantly, led to interesting visual results. Third, and finally, there was no need to consider out-of-bounds issues as the borders were held at a constant value and thus did not require evaluation.

3.2.2 Challenges

There were two major challenges with implementing the heat transfer equation. The first was how to interpret the data. As mentioned above, the k-value is representative of how fast a point will reach temperature equilibrium with its neighbours. It was necessary to envision the sandbox as an input device, but k was not the first option — we also considered using a uniform k and imagining the sandbox as water, but quickly realized this led to much less interesting results. Effectively communicating that a lower sand height $\Rightarrow$ a higher k is difficult, as it implies the "material" the sandbox represents has a bizarre, heterogeneous conductivity.

Another issue we encountered was performance. Simulating every single point, every single frame, caused our frame rate to drop to 24 FPS or even as low as 12. Responsiveness is a core design goal for the sandbox project as a whole, so this was unacceptable. Thanks to the help of my supervisor, David Churchill, the code for the simulation was converted to run in parallel, which greatly reduced the computational time and returned responsiveness to ideal levels.

3.2.3 Survey

The final part of the research for the heat transfer equation was done as part of a demonstration at MUN's "Whale of a Day!" event. At this event, a sign with a link to a survey was presented, which asked the following questions:

- Overall, how would you rate your experience with the Augmented Reality Sandbox? (Rating from 1-5, with 1 presented as "Poor" and 5 presented as "Good")
- How engaging did you find the Augmented Reality Sandbox? (Rating from 1-5, with 1 presented as "Not engaging" and 5 presented as "Engaging")

- Before today, how familiar were you with augmented reality technology? (Rating from 1-5, with 1 presented as “Not familiar” and 5 presented as “Familiar”)
- Did the Augmented Reality Sandbox help you understand topography or geographical concepts better? (Rating from 1-5, with 1 presented as “No” and 5 presented as “Yes”)
- Did you learn something new from interacting with the Augmented Reality Sandbox? (Multiple choice, with options of “Yes”, “No”, and “Not sure”)
- If you answered “Yes” to the previous question, please describe what you learned (Long answer)
- If you were thinking about going to an event in the future and a demonstration was being given with the Augmented Reality Sandbox, would that make you more or less interested in going? (Rating from 1-5, with 1 presented as “Less interested” and 5 presented as “More interested”)

Participants were prompted with the following: “How did you feel about the sandbox exhibit at “A Whale of a Day”? Please contribute to our research by letting us know!”.

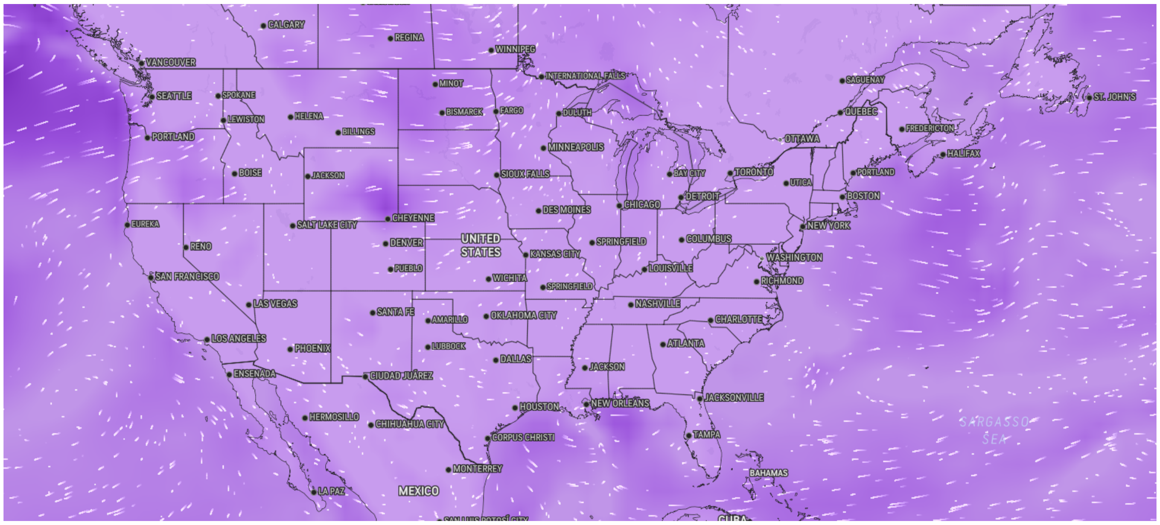

Figure 3.3: Example of a wind visualization (AccuWeather, 2025)

3.3 Wind Visualizations

The second physics problem that modelled using the AR sandbox was the trajectory of wind over a surface. Essentially, we were looking to create something similar to Figure 3.3 using the sandbox as both input and output. The model selected was the one presented in (C. et al., 1949), which used simple numerical methods, making it easier to implement.

3.3.1 Algorithmic Methods

In order to build a full wind visualization on the sandbox, we needed to start with something smaller. A full C++ implementation of the Charney & Eliassen paper required a lot of planning, so we worked on creating visualizations using algorithmic techniques instead.

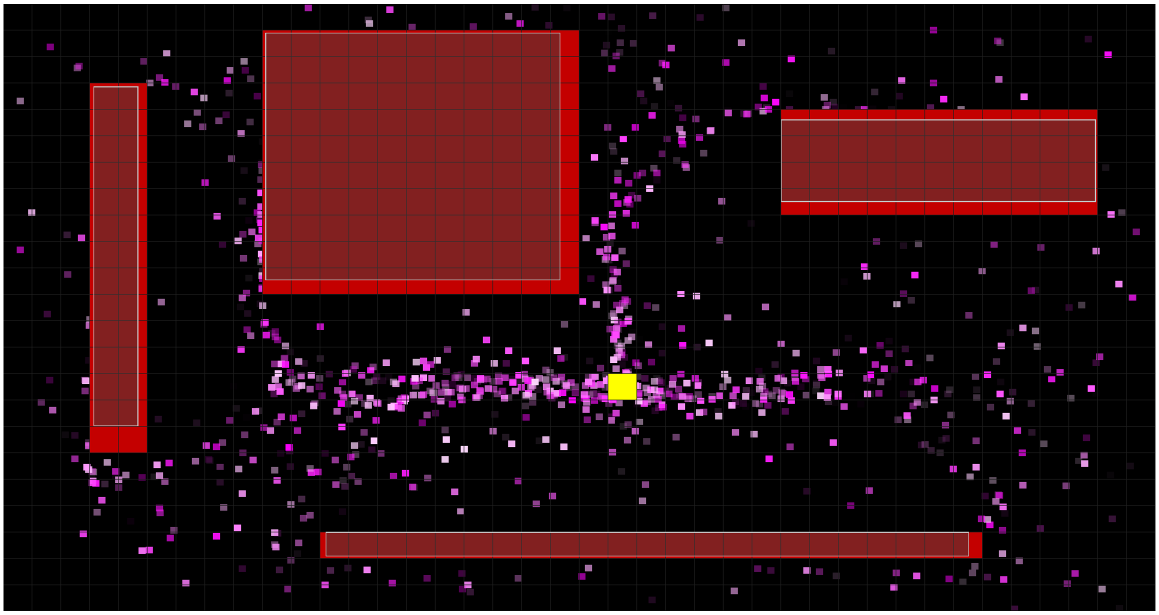

Figure 3.4: The visualization (without modifications), including particle simulation (Churchill, 2022)

We started by using previous work done by one of my supervisors (Churchill, 2022) as a basis for future efforts. This previous work, a particle simulation over a discrete grid, was a perfect start because it was very similar to the data sent by the sandbox. A vector field is created based on obstacles present in the grid space, and then particles flow in the direction of the cell they occupy (note the grid in Figure 3.4, which creates cells, as well as the magenta particles).

The vector field was the key reason why this was chosen as the starting point — it is created using a breadth-first search (BFS) approach, but we knew that it would be easy to “swap out the engine” and replace the code that generated the BFS with methods taken from physics.

Although we had an excellent starting point, there was a lot that needed to change to make it work on the sandbox. The original version only considered discrete cells which could be either be traversable or solid. However, the physical sandbox has so many cells (represented by a 2D array of floating point values between zero and one) that it is inefficient (and unnecessary) to compute a vector for every single cell. Therefore, a "cell size" was introduced (typically 4×4 or 8×8) which takes the average of each of the "real" cells and converts it to a "grid cell". This is done by taking the average of the real cells, ignoring any out of bounds positions (so the bottom-right grid cell of an 1003×1003 array with 8×8 cells would only take the average of a 3×3 subsection).

Another problem was the obstacles. The assignment used discrete obstacles, such that every point was either traversable or solid. However, the sandbox's depth allowed for a greater range of values, so we opted to use weights instead. Essentially, the BFS was modified to be a weighted BFS, ensuring that wind could flow both "uphill" and "downhill" and not get trapped in valleys etc.

Additionally, the assignment used cardinal vectors, which take one of eight directions (up, down, left, right, and their combinations). The vectors for each cell were chosen based on the distances assigned by BFS to their neighbours, with preference given to those with shorter distances. This was serviceable, but there were notable issues where strands would abruptly take 90° angles.

Since cardinal vectors limited precision (only eight directions were available) and they lead to undesirable visuals, it was necessary to abandon them in favour of angular vectors. Using the method outlined in (Durant, 2013), we were able to create a smoother and more continuous visualization. This method replaced selecting a vector with a simple computation:

$$\begin{bmatrix} x \\ \\ y \end{bmatrix} = \eta \begin{bmatrix} C(x-1,y) - C(x+1,y) \\ \\ C(x,y-1) - C(x,y+1) \end{bmatrix}$$

where $C(x,y)$ is the cost of that cell and η is a normalizing factor. This method of computing the vector was taken from (Durant, 2013)

3.3.2 Physics Methods

The forecast method in (C. et al., 1949) was fairly straightforward once we digested the paper. The key function (no. 35 in the paper), is below:

$$z(x) = \kappa\lambda^2 \int_0^{2\pi} h(\alpha)\Phi_\sigma(x-\alpha)d\alpha$$

Here, $z(x)$ represents the height of the 500 mbar pressure surface at some longitude x, $h(\alpha)$ is the height of the land at longitude α, κ is a reduction factor, and $\Phi_\sigma(x-\alpha)$ is the Green's function with friction coefficient σ at longitude $x-\alpha$. The Green's function is given below:

$$\Phi_\sigma(x) = \frac{1}{2\pi} \sum_{n=-\infty}^{+\infty} \frac{e^{inx}}{n^2 - s^2 - i\sigma(n + \frac{m^2}{m})}$$

Additional constants were necessary to determine λ^2 and m^2, which were defined as $\lambda^2 = 2.5\sin^2 2\phi$ and $m^2 = \frac{\beta}{U} - s^2$. ϕ, the representative latitude, was set at 45°. This

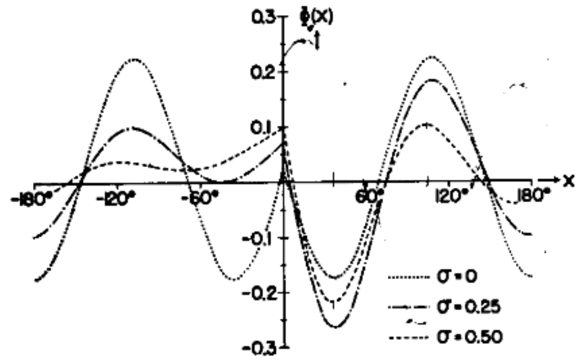

Fig. 2. The GREEN's function Φ_σ (x) for the stationary perturbation. The dotted curve represents the function for the case of no friction, the dash-dotted curve for the case of moderate friction, and the dashed curve for the case of strong friction.

Figure 3.5: The Green's function from (C. et al., 1949)

was used to derive β, so that $\beta = 4\pi \cos^2 \phi$. U, the overall speed of the wind around the Earth, was set at 0.29 radians per day. All angles refer to an arc around the Earth, equivalent to degrees of longitude.

Converting these methods from mathematical notation to workable code was not an overly difficult task. However, validating the correctness of the implementation was much more difficult, as the paper did not provide any numerical data, only images of results. Therefore, before implementing the model in C++, we decided to first implement it in a Jupyter notebook instead. This was much faster, allowed access to tools like NumPy, and provided a useful basis for validation, as my physics co-supervisor who was working on the notebook along with me is much more familiar with Python than C++.

The first function we implemented was the Green's function, which looked like

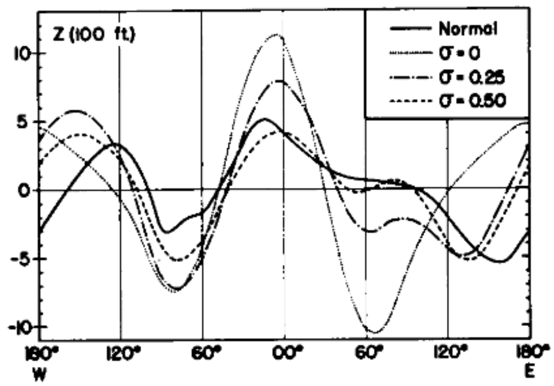

Fig. 4. Normal height profile of the 500 mb surface at 45° N for January together with computed stationary profiles for $\sigma = 0$ (no friction), for $\sigma = 0.25$ (moderate friction) and for $\sigma = 0.50$ (strong friction). For purposes of comparison the heights are represented as deviations from their respective means.

Figure 3.6: The computed height profile from (C. et al., 1949)

Figure 3.5 in the paper. Our results (shown later in the paper) were very close to this plot. We then implemented $z(x)$ (Figure 3.6), which was different than the paper as we used a different $h(x)$ but still yielded similar results. The code for this can be found at https://github.com/EthanDenny/charney-eliassen-simulations (Denny, 2025). This code had two parts, a Jupyter notebook and a C++ implementation. The notebook attempted to simulate the model as closely as possible, with the addition of code for extending the one-dimensional result to two dimensions. This used the following math:

$$Z(x, y) = z(x)\sin(y)$$

where $Z(x, y)$ is the third-dimensional z-coordinate given some two-dimensional x and y, and $z(x)$ is the function from the original paper. This is then used to compute

u and v:

$$u(x,y) = -\frac{\eta g}{f}(\frac{Z(x,y+1) - Z(x,y-1)}{dy}) + w$$
$$v(x,y) = \frac{\eta g}{f}(\frac{Z(x+1,y) - Z(x+1,y)}{dx})$$

where η is a normalizing factor, dx and dy are the distance between sampled points along their respective axes, g is the gravity constant, and w is a constant wind velocity added to u. f is given as $f = 2\omega \sin\phi$, where $\omega = 7.27 \times 10^{-5}$ is the angular velocity of the Earth's rotation around its axis. u and v serve as the components of a velocity vector for the position p of a given projectile, so given $p = \begin{bmatrix} x \\ y \end{bmatrix}$, then $\dot{p} = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix}$, and therefore any projectile with position p will move along the vector $\dot{p}$.

This algorithm (including implementations for $z(x)$ and the Green's function) was first implemented in Python, which was used to check correctness. Python was a strong first choice, but it was slow and not the language that the rest of the sandbox code was implemented in. Therefore, we rewrote the simulation in C++. This rewritten code was then added to the main sandbox code base, with small interface changes to allow for selecting whether to use algorithmic- or physics-based computations.

3.3.3 Particles

In our wind visualizations, particles represent a single, continuous point in the grid. The velocity of each particle is always taken from the cell or pixel that it occupies, and this leads to both a small memory footprint and efficient processing, meaning many tens of thousands of particles can be simulated at the same time. However, particles

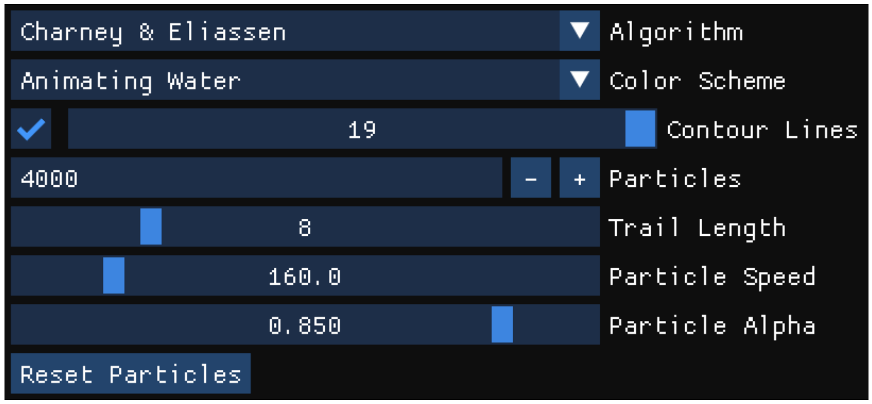

Figure 3.7: The interface that controls the algorithmic wind visualizations

are not a single visual entity. Instead, a "trail" of past positions is drawn, creating a strand-like entity (creating an effect similar to a gust of wind) that moves smoothly over the sand. Every particle is placed on the "left" of the simulation ($x = 0$) and then moves right. When particles reach the far edge, they loop back around, and are placed at a random y-coordinate, leading to a constant flow of particles.

3.3.4 Interface

The wind visualizations have several customizable features, to allow us to test different parameters and settings. Figure 3.7 displays the interface for these, with individual settings described below:

- Algorithm: What creates the vector field. Either "Charney & Eliassen" or "BFS".
- Color Scheme: The underlying colours given to the terrain, before the wind visualization is applied.

- Contour Lines: Controls the contour lines on the display, including whether to display them and the number of levels.
- Particles: The number of particles, i.e. ”gusts of wind”.
- Cell Size: The cell size for the BFS; higher numbers mean less precision but more performance.
- Trail Length: The number of past positions to store for each particle, which are displayed as a “trail”, helping to create a strand-like effect.
- Terrain Weight: How much the terrain affects the particles' movement. In the BFS calculation, the cost of a given cell will be $D \times (1 + T \times W)$, where D is the distance (normal cost), T is the terrain cost (a value between 0 and 1, where 0 is a lower averaged height of the cell and 1 is higher), and W is the weight.
- Particle Speed: How fast particles move across the sand.
- Particle Alpha: The transparency of the particles (purely a visual effect).

Chapter 4

Results

4.1 The Heat Transfer Equation

4.1.1 Visual Results

The heat transfer simulation, while colourful (Figure 4.1), was not necessarily a strong fit for the sandbox. The input system was confusing - while developing it, we often had to clarify to each other whether greater elevation or greater depth led to greater conductivity - and it could not take full advantage of the projector, as the output did not really correspond with the height of the sand. This is a limitation of the current state of our AR sandbox project - the input and output are limited in scope compared to other forms of augmented reality (e.g. holographics, AR glasses, etc.). Therefore, we feel that the physics problems best suited for AR sandboxes are those which operate on a larger, more physical scale, which is why we moved to visualizing wind trajectories instead.

4.1.2 “Whale of a Day”

The heat transfer simulation was presented at MUN’s “Whale Of A Day!”, an outreach event focused on raising the public’s awareness of physical and ocean sciences. Many of the displays and presentations were aimed at children, but enjoyable and interesting for all ages. The team at the AI & Games Lab presented the sandbox at this event, which was a massive success for raising the profile of our project and awareness of the technology. We observed that children especially were drawn to the bright and engaging display, which quickly turned to joy when they realized that the projection responded to their actions. This led to many opportunities to engage older children and adults in conversations about the technologies and topographical concepts.

Although most of the demonstration time was dedicated to the terrain / topographical simulations, there were many opportunities to show off the heat simulation, which were well received by persons both from Memorial and the public in general. We attempted to take a survey asking a few simple questions of attendees, by placing a sign with a QR code that linked to a Google Form (the content of which was described in the methodology section). We only received two responses, which we believe was a result of the difficulty those interacting with the sandbox had to see the QR code (which was often blocked by those standing around the display), and by the amount of “work” required to fill out the survey - scanning a QR code, then filling

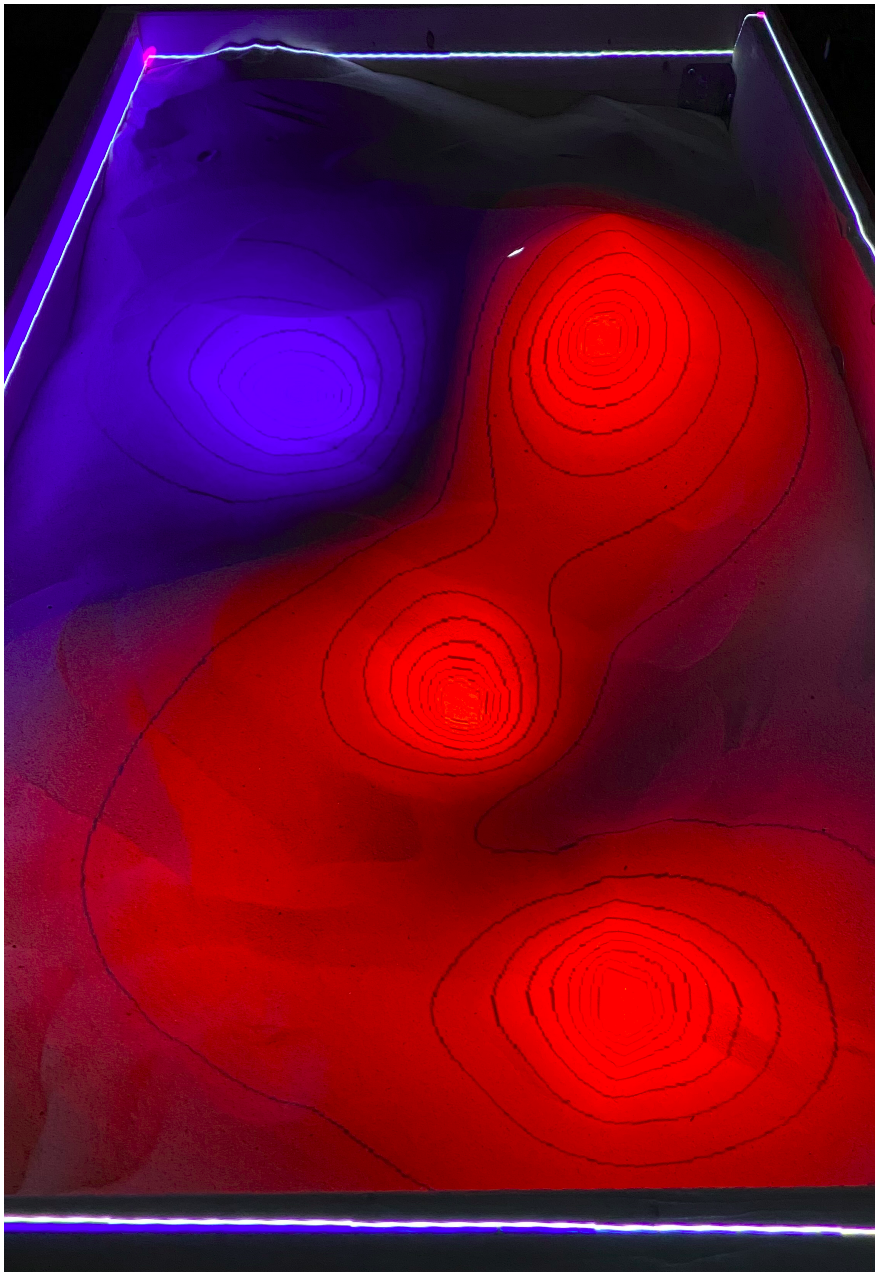

Figure 4.1: An image of the sandbox simulating the heat transfer equation. Note the striking visual qualities of the sandbox under low-light conditions, and how the contour lines accentuate the temperatures' spread

out a survey - is difficult while standing at a busy event, especially if one is trying to supervise children at the same time.

4.2 Wind Visualizations

4.2.1 Algorithmic Methods

While the visuals of the algorithmic approach were impressive - the wind particles looked good, and were very configurable - the underlying algorithm left a lot to be desired. Despite being fast and smooth, wind would get siloed into small corridors with any terrain weight greater than about 0.15, and the results did not look convincingly like what real wind patterns would be expected to.

Figure 4.2 demonstrates the final visual look of the algorithmic approach. A few things should be noted here. There is some stretching and warping, because a logical pixel or square in the grid does not necessarily correspond to a visual pixel being sent to the projector. This is because the projection's calibration process distorts the image slightly, requiring a projection that does not easily allow for a clean look with the shader setup we were using. Additionally, the wind visualization is not specific to any particular terrain shading, but rather sits on top of the terrain using a different channel to pass wind information.

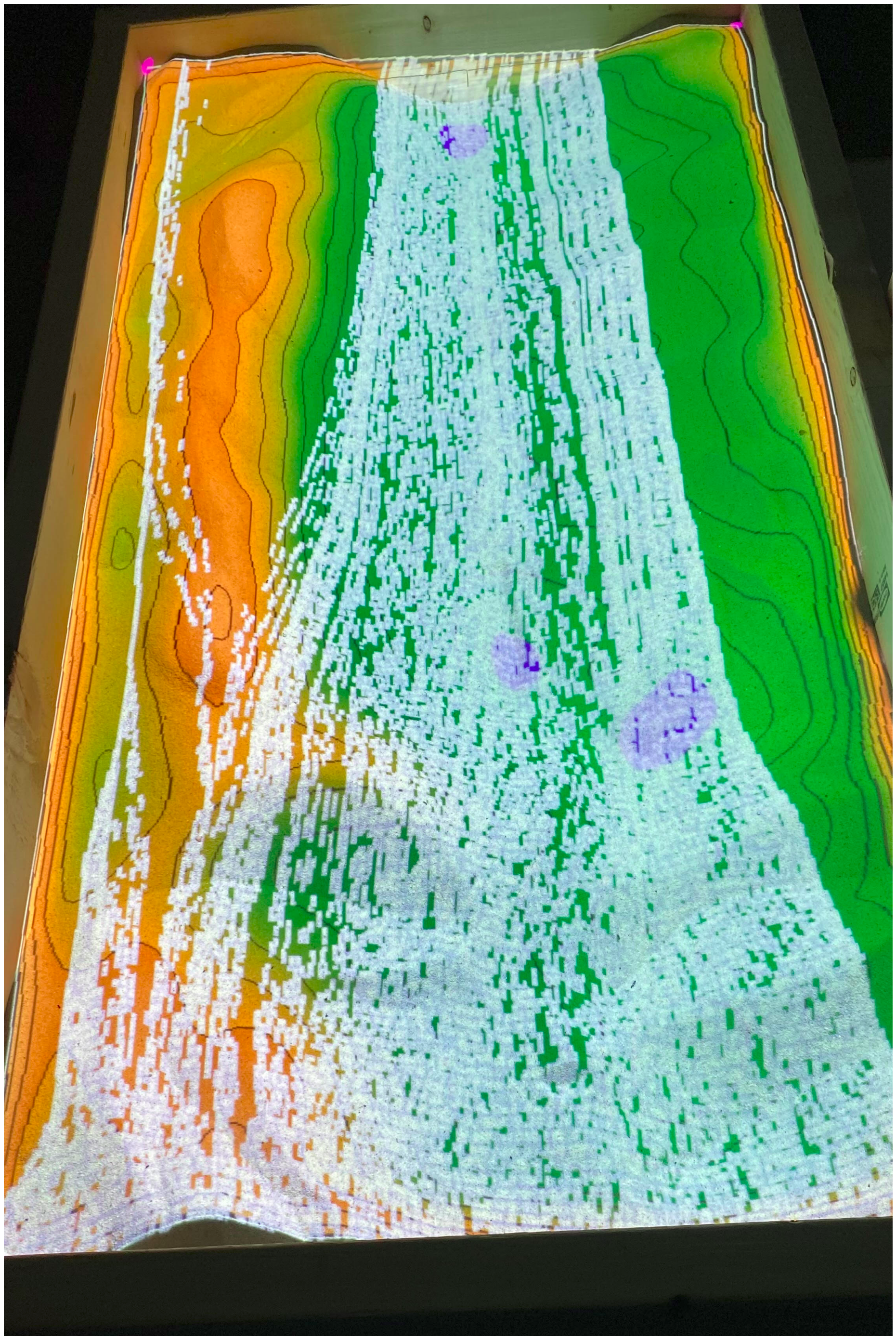

Figure 4.2: Visualization of vector fields, made to look like wind

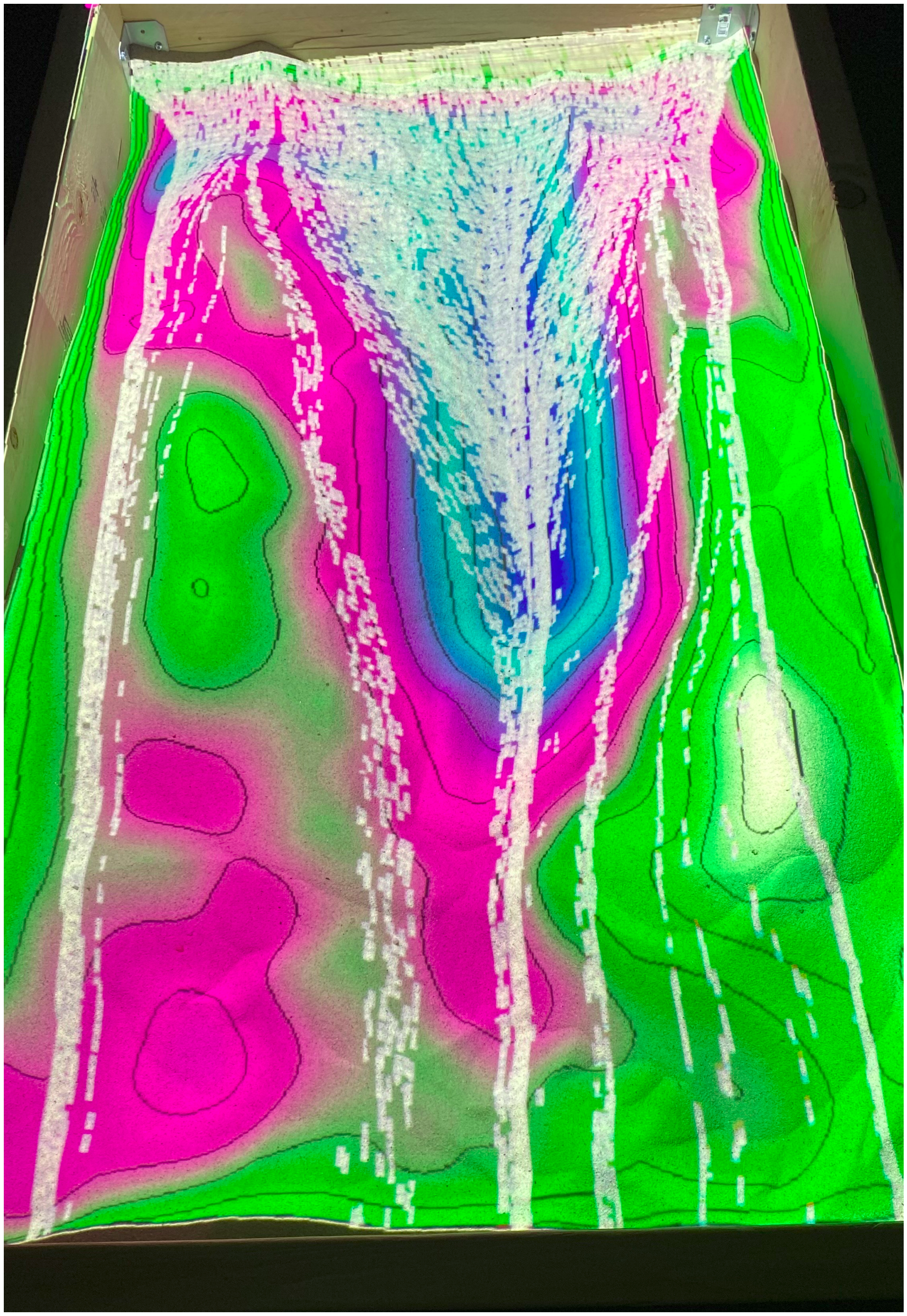

Figure 4.3: The wind visualization shown over the “Popsicle” visualization

4.2.2 Jupyter Notebook

Before implementing the physics-based wind visualizations on the sandbox, we felt it would be best to simulate them in a Jupyter notebook and then an independent C++ project based on that notebook. The code used to generate these figures can be found at https://github.com/EthanDenny/charney-eliassen-simulations (Denny, 2025).

Figure 4.4 shows the results of plotting the Green's function, which is very similar to the plot from the original paper (Figure 3.5). Figure 4.5 shows the results of plotting $z(x)$ over $h(x)$, with $h(x)$ being a single Gaussian. Figure 4.6 shows the results of computing the wind trajectories in Python, and Figure 4.7 shows the same computation done in C++ (which also implemented the Green's function and $z(x)$ in C++). Note that the two figures are also very similar, demonstrating that the computation done with C++ was similarly accurate to that in Python.

The Python implementation was very slow. On my ThinkPad T480s laptop with an Intel i5-8250U CPU, it ran in about 9.64 seconds. When implemented with C++, it was much faster, reaching times as low as 0.68 seconds. Figure 4.8 shows a chart comparing the performance of different implementation approaches.

4.2.3 Sandbox

The final visual results on the sandbox were a long time in the making, but the effort paid off. Overall, we were very happy with the visualizations. Vortices, waves, and swirls combined to make this visualization much more full and satisfying than the

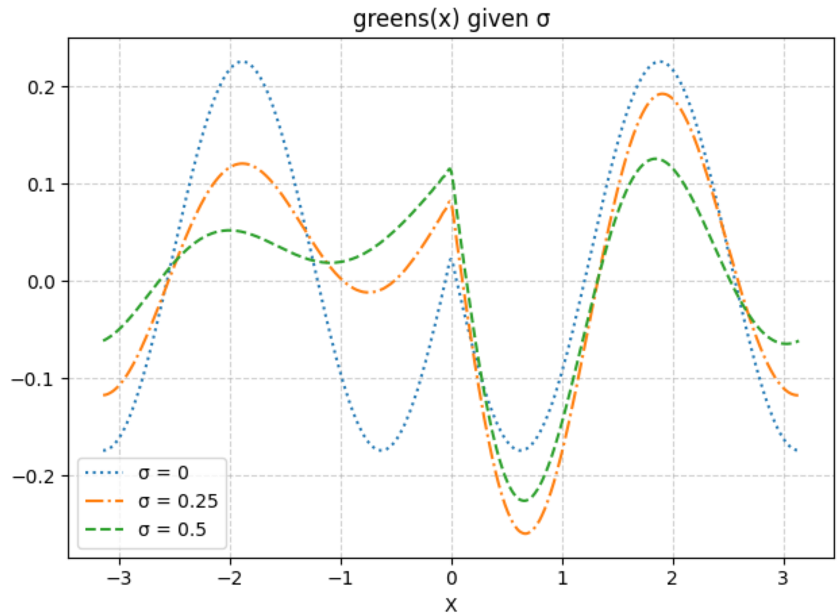

Figure 4.4: Green's function

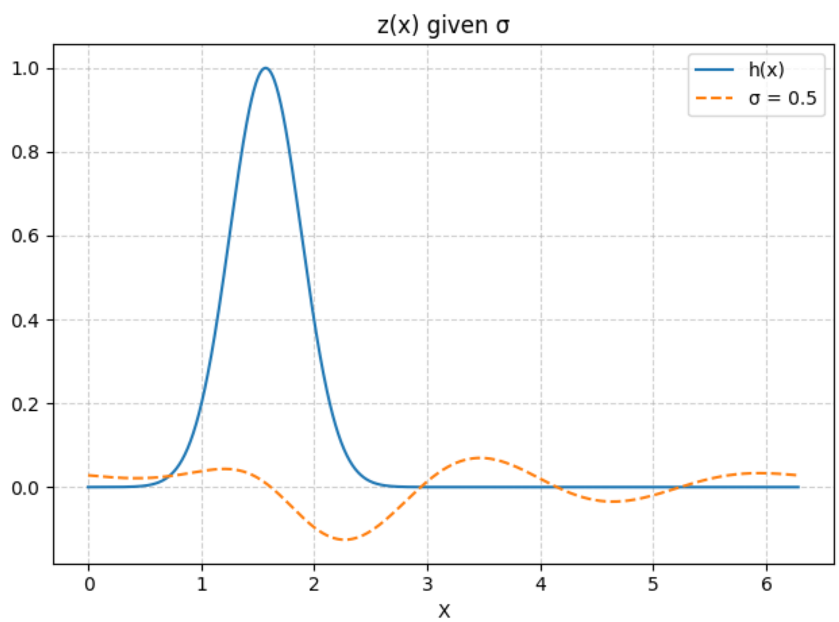

Figure 4.5: z(x)

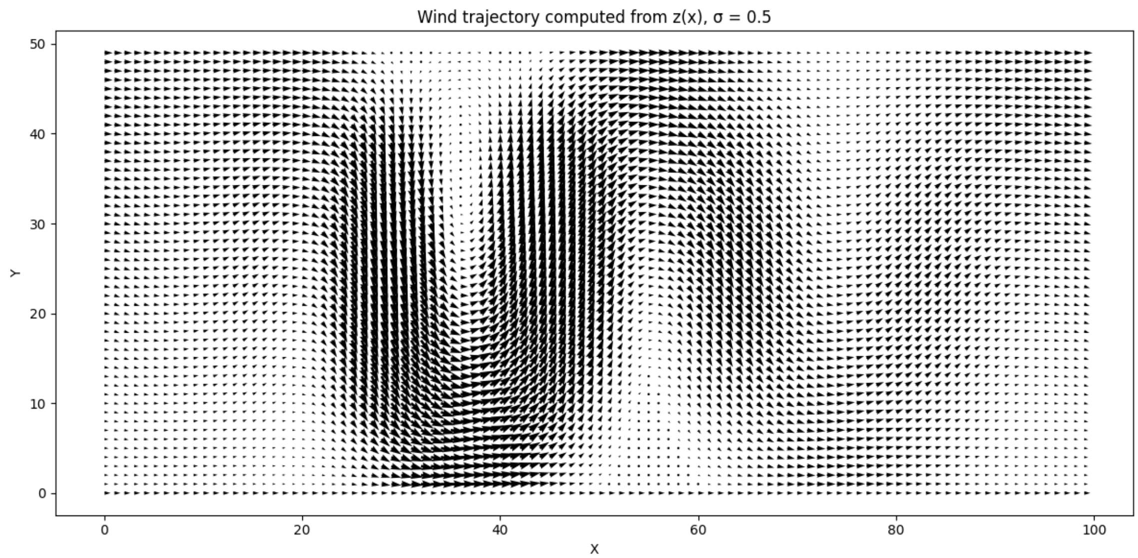

Figure 4.6: Wind trajectories, Python

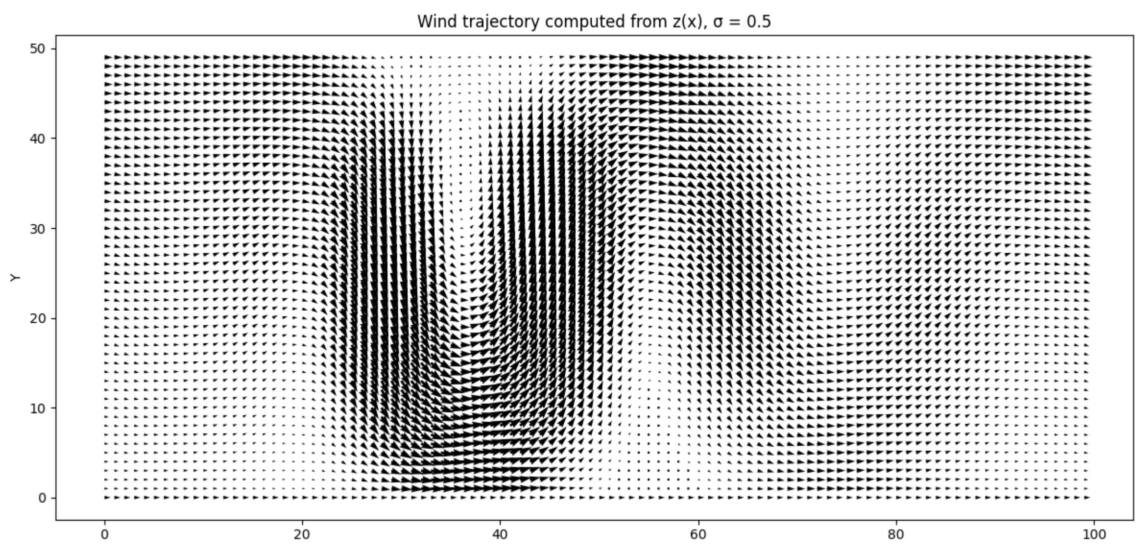

Figure 4.7: Wind trajectories, C++

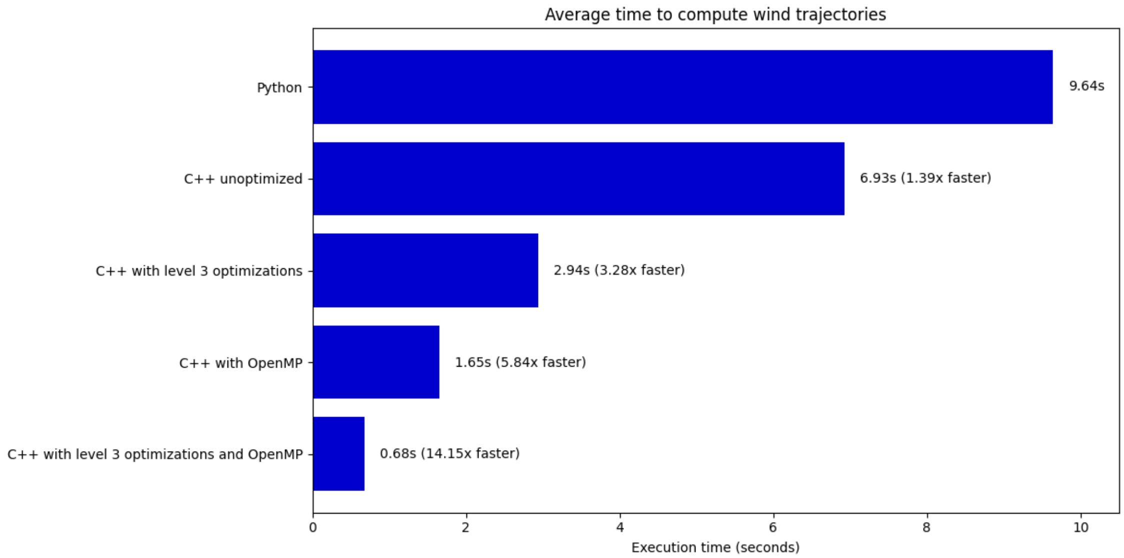

Figure 4.8: Time to compute wind trajectories. An average was taken for a range of values between $\sigma = 0$ and $\sigma = 1$

algorithmic approach, and the closer adherence to real world physics meant that its use as an outreach and educational tool would be much more appropriate. Given the nature of the vector field, this model benefitted from a smaller number of particles with longer trails and faster speeds - just 10-12% of the default particle count for BFS still looked great.

A notable problem with the results was the unresponsiveness of the model to small changes in the sand. Because the model takes a mean across one dimension, processed in 1D and then extended back to 2D, small changes, such as a single scoop of sand or a hand, did not make large changes. This was in contrast to the algorithmic approach, which could update and respond in near-real-time. Furthermore, because we were only able to update every two seconds, "hacky" solutions such as running multiple instances of the model in parallel for different bands (thus reducing the effect of averaging), were not computationally plausible and would lead to a decrease in mathematical and physical accuracy. To solve this problem, we brainstormed other solutions. A promising but high effort approach would be to create our own model, drawing on more recent research; unfortunately, a lack of time prevented this from being explored.

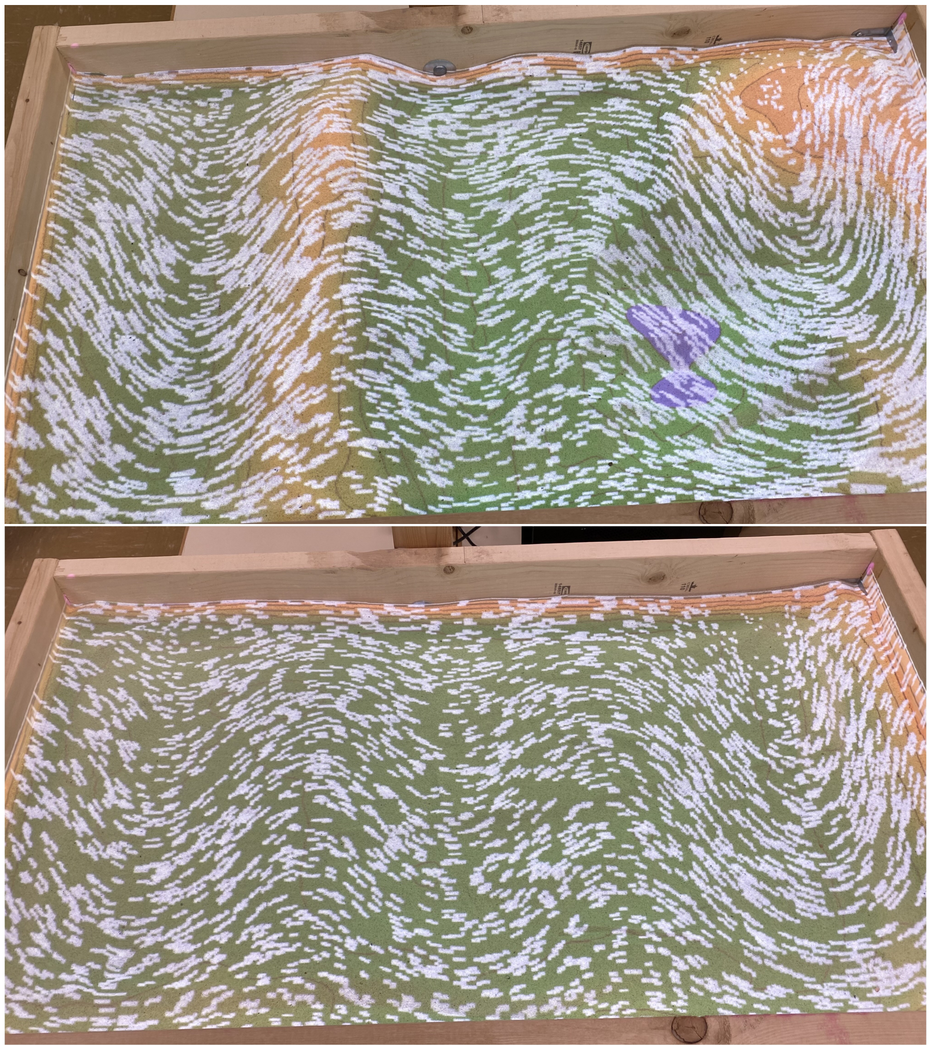

Figure 4.9: The final visualization using the Charney & Eliassen model. The top image has bumpy terrain; the bottom image has flat terrain

Chapter 5

Conclusion

We believe that augmented reality sandboxes have a strong role to play in the future of physics education, specifically in outreach to the public and prospective students. While the current evidence shows that they have had limited benefit to direct education, most of this research focuses on topographical geoscience labs, with little to no investigation into their applications in physics. Certainly, we believe that we may be the first to attempt to centre physics in this way. Other work has included physical equations: the Navier-Stokes equations, for example, have been used to improve the appearance of ”water” in landscape visualizations. But our work takes this a step further through work like modelling the heat equation, which required a radical reimagining of how the sandbox should be viewed as both an input and output device.

In addition to the novelty of our visualizations, there has been a major focus on high performance as a goal both for this thesis work and the sandbox project at the AI

& Games Lab as a whole. Previous sandbox offerings have relied on older sensors and decade-old software - our work, built from scratch using modern languages, tools, and equipment - updates over 90 times per second in most cases, removing a key barrier to immersion, and allowing the focus to be placed on the scientific concepts at hand.

Overall, this has been an extremely productive contribution to the sandbox project here at MUN, and we believe that the visualizations we have created will be useful both in demonstrating the value of our specific project, and in demonstrating how physics can be as useful a target for AR sandbox education, as geoscience, if not more so.

Bibliography

AccuWeather. (2025). United States Wind Maps. Retrieved February 23, 2025, from https://www.accuweather.com/en/us/national/wind-flow

Churchill, D. (2022). COMP4303 - AI for Video Games - Lecture 07 - Assignment 2. Retrieved February 23, 2025, from https://www.youtube.com/watch?v=oXiNwUVvOYE

Denny, E. (2025). Charney & eliassen simulations. Retrieved March 25, 2025, from https://github.com/EthanDenny/charney-eliassen-simulations

Durant, S. (2013). Understanding Goal-Based Vector Field Pathfinding. Retrieved November 25, 2024, from https://code.tutsplus.com/understanding-goal-based-vector-field-pathfinding--gamedev-9007t

et al., B. (2023). Augmented reality technology used for developing topographic map-reading skills in an earth science course and its potential implications in broader learning venues. *Journal of Science Education and Technology*, *32*(2), 284–293.

et al., C. (1949). A numerical method for predicting the perturbations of the middle latitude westerlies. *Tellus*, *1*(2), 38–54.

et al., C. (2024). Sandbox. Retrieved March 11, 2025, from https://github.com/EthanDenny/sandbox

et al., G. (2020). Young children's use of an augmented reality sandbox to enhance spatial thinking. *Children's Geographies*, *18*(2), 209–221.

et al., G. (2017). Instructor-led approach to integrating an augmented reality sandbox into a large-enrollment introductory geoscience course for nonmajors produces no gains. *Journal of Geoscience Education*, *65*(3), 283–291.

et al., I. (2004). Tangible user interfaces (tuis): A novel paradigm for gis. *Transactions in GIS*, *8*(4), 407–421.

et al., M. (2020). A multi-institutional study of inquiry-based lab activities using the augmented reality sandbox: Impacts on undergraduate student learning. *Journal of Geography in Higher Education*, *44*(1), 85–107.

et al., S. (2020). A novel approach to measuring student engagement while using an augmented reality sandbox. *Journal of Geography in Higher Education*, *44*(4), 512–531.

et al., Y. (2020). Distributed spatial sensemaking on the augmented reality sandbox. *International Journal of Computer-Supported Collaborative Learning*, *15*(1), 115–141.

Kreylos, O. (2022). SARndbox Download. Retrieved March 11, 2025, from https://web.cs.ucdavis.edu/~okreylos/ResDev/SARndbox/Download.html

Kreylos, O. (2023). Oliver Kreylos' Research and Development Homepage - Augmented Reality Sandbox. Retrieved October 4, 2024, from https://web.cs.ucdavis.edu/~okreylos/ResDev/SARndbox/

Minolta, K. (2006). Introducing Konica Minolta's VIVID Line of 3D Scanners. Retrieved February 24, 2025, from https://www.virvig.eu/services/Minolta.pdf

Smagorinsky, J. (1983). The beginnings of numerical weather prediction and general circulation modeling: Early recollections. *Advances in Geophysics*, *25*, 3–37.

UC Davis. (n.d.). Ar (augmented reality) sandbox. Retrieved February 24, 2025, from https://datalab.ucdavis.edu/project/ar-sandbox/

Whitworth, D. (2010). Kinect gets UK release date. *The BBC*. Retrieved February 24, 2025, from https://www.bbc.com/news/newsbeat-10996389