1. Computational Fluid

a. Computational Fluid Dynamics is in the domain of Computational Science

b. Applications

   i. Computational Fluid Dynamics have many applications. Some applications include:

      1. Automotive Aerodynamics
      2. Designing HVAC Systems
      3. Water Flow Around Submarines
      4. Modeling Dams

c. The Physics of Fluids

   i. The Navier-Stokes equations describe the motion of fluid substances.

   ii. These equations arise from applying Newton's second law (F=ma) to fluid motion.

   iii. Velocity equation:

      1. The first term says that velocity is moved by itself.
      2. The second term says that velocity diffuses based on the viscosity constant V.
      3. The third term says that velocity is affected by an external force.

   iv. Density equation:

      1. The first term says that the density should follow the velocity field.
      2. The second term says that the density may diffuse at a constant rate K.
      3. The third term says that the density should increase due to an external source.

d. Fluid Representation

   i. Mathematical equations for fluids are useful when thinking about fluids in general. However, we need a **finite representation** for the fluid. The usual approach is to divide the fluid into a grid, a **lattice**, of identical cells where the center of each cell contains the density and velocity for a piece of fluid.

e. Implementing Navier-Stokes

   i. To implement the Navier-Stokes equations we need to break it up into discrete steps. These steps are:

      1. External Forces
      2. Diffusion
      3. Advection
      4. Projection

f. External Forces

   i. External forces applied to the fluid can be either **local forces** or **body forces**.

   ii. Local forces are applied to a specific region of the fluid – for example the force of a fan blowing air.

   iii. Body forces are forces that apply evenly to the entire fluid, like gravity.

   iv. Without external forces fluid will reach a steady state.

g. Diffusion

    i. Diffusion is how the density will spread across the grid cells. We first look at what happens at a single grid cell. We assume that cell **exchanges densities with its four direct neighbors.** The cell's density will decrease by losing density to its neighbors, but will also increase due to densities flowing in from the neighbors.

h. Advection

    i. The velocity of the fluid causes the fluid to transport objects, densities and other quantities along with the flow. Imagine squirting **dye into a moving fluid**. The dye is transported, or advected, along the fluid's velocity field.
    ii. The velocity of a cell is found by looking for the particles which end up exactly at the cell centers by tracing backwards in time from the cell centers.

i. Projection

    i. After the diffusion and advection steps the velocity field seldom **conserves mass**. The idea is to make it mass conserving in the last step. To achieve this result we use the Hodge decomposition, which is every velocity field, is the sum of a mass conserving field and a gradient field. To get the mass conserving field we subtract the gradient from our current field.

2. Introduction to CUDA

a. Why use CUDA?

    i. The reason for using CUDA is to gain increased performance.
    ii. A good example of this is the speed up gained from converting the CPU based fluid simulation program to the GPU based fluid simulation.

b. What is CUDA?

    i. Compute Unified Device Architecture
    ii. NVIDIA's software architecture for developing and running-data parallel programs
    iii. Programmed in an extension of the C language

c. CPU vs GPU

    i. CPU
        1. Fast caches
        2. Branching adaptability
        3. High performance
    ii. GPU
        1. Multiple Arithmetic Logic Units
        2. Fast onboard memory
        3. High throughput on parallel tasks
    iii. CPUs are great for task orientated parallelism
    iv. GPUs are great for data orientated parallelism

d. CPU vs GPU – Hardware
    i. Each stream processor in the GPU has its own control and cache along with many ALUs.
    ii. More transistors are devoted to data parallelism
e. Kernel Functions
    i. A kernel function is code that runs on the GPU
    ii. The code is downloaded and executed **simultaneously** on all stream processors on the GPU
f. Fluid Dynamics on the GPU
    i. To implement the Navier-Stokes equations on the a GPU we need to write kernel functions for:
        1. External Forces
        2. Diffusion
        3. Advection
        4. Projection
    ii. Copy data from RAM to GPU memory. Perform calculations. Copy data from GPU back to RAM.
    iii. Difficulties
        1. If the program has an error while running on the GPU, you don't get a helpful error message.
        2. Concurrency problems

3. Demonstration

a. Show adding density
b. Show moving fluid around
c. Show velocity field
d. Boundaries – bounce back