Fluid Dynamics

CSE169: Computer Animation

Steve Rotenberg

UCSD, Spring 2016

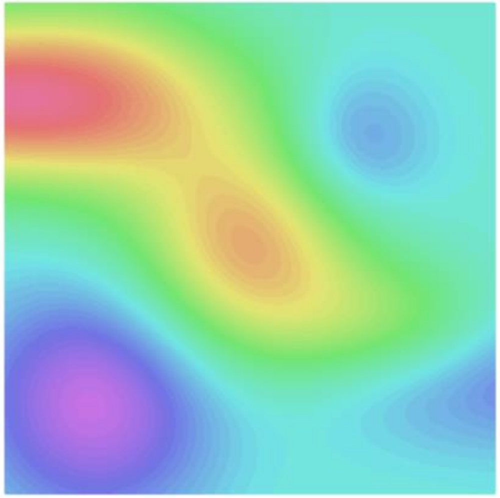
Fluid Dynamics

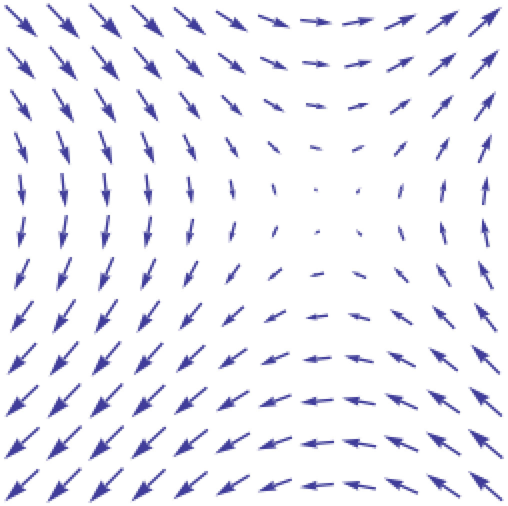
- *Fluid dynamics* refers to the physics of fluid motion
- The *Navier-Stokes equation* describes the motion of fluids and can appear in many forms
- Note that ‘fluid’ can mean both liquids and gasses, as both are described by the same equations
- *Computational fluid dynamics* (CFD) refers to the large body of computational techniques involved in simulating fluid motion. CFD is used extensively in engineering for aerodynamic design and analysis of vehicles and other systems. Some of the techniques have been borrowed by the computer graphics community
- In computer animation, we use fluid dynamics for visual effects such as smoke, fire, water, liquids, viscous fluids, and even semi-solid materials

Differential Vector Calculus

Fields

- A *field* is a function of position **x** and may vary over time *t*
- A *scalar field* such as *s*(**x**,*t*) assigns a scalar value to every point in space. A good example of a scalar field would be the temperature in a room
- A *vector field* such as **v**(**x**,*t*) assigns a vector to every point in space. An example of a vector field would be the velocity of the air





Del Operator

- The *Del* operator ∇ is useful for computing several types of derivatives of fields

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{bmatrix}$$

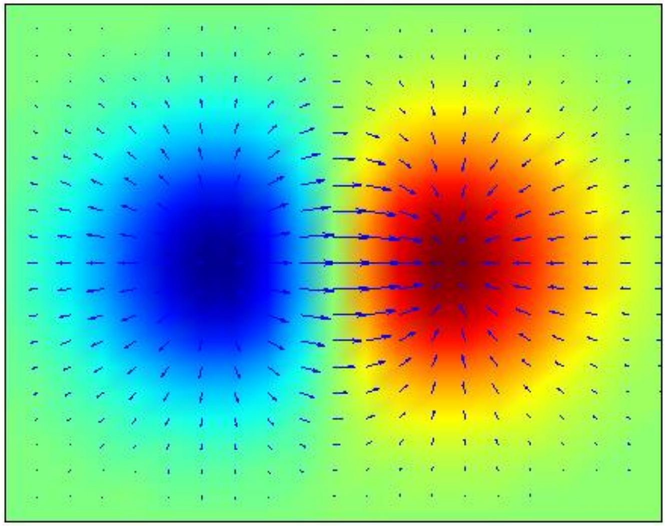
- It looks and acts a lot like a vector itself, but technically, its an operator

Gradient

- The *gradient* is a generalization of the concept of a derivative

$$\nabla s = \begin{bmatrix} \frac{\partial s}{\partial x} & \frac{\partial s}{\partial y} & \frac{\partial s}{\partial z} \end{bmatrix}$$

- When applied to a *scalar* field, the result is a *vector* pointing in the direction the field is increasing
- In 1D, this reduces to the standard derivative (slope)



Divergence

- The *divergence* of a vector field is a measure of how much the vectors are expanding

$$\nabla \cdot \mathbf{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

- For example, when air is heated in a region, it will locally expand, causing a positive divergence in the area of expansion
- The divergence operator works on a vector field and produces a scalar field as a result

Curl

- The *curl* operator produces a new vector field that measures the rotation of the original vector field

$$\nabla \times \mathbf{v} = \begin{bmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} & \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} & \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{bmatrix}$$

- For example, if the air is circulating in a particular region, then the curl in that region will represent the axis of rotation
- The magnitude of the curl is twice the angular velocity of the vector field

Laplacian

- The *Laplacian* operator is a measure of the second derivative of a scalar or vector field

$$\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

- Just as in 1D where the second derivative relates to the curvature of a function, the Laplacian relates to the curvature of a field
- The Laplacian of a scalar field is another scalar field:

$$\nabla^2 s = \frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2}$$

- And the Laplacian of a vector field is another vector field

$$\nabla^2 \mathbf{v} = \frac{\partial^2 \mathbf{v}}{\partial x^2} + \frac{\partial^2 \mathbf{v}}{\partial y^2} + \frac{\partial^2 \mathbf{v}}{\partial z^2}$$

Del Operations

- Del: $\nabla = \left[\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \quad \frac{\partial}{\partial z}\right]$
- Gradient: $\nabla s = \left[\frac{\partial s}{\partial x} \quad \frac{\partial s}{\partial y} \quad \frac{\partial s}{\partial z}\right]$
- Divergence: $\nabla \cdot \mathbf{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$
- Curl: $\nabla \times \mathbf{v} = \left[\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \quad \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \quad \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right]$
- Laplacian: $\nabla^2 s = \frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2}$

Navier-Stokes Equation

Frame of Reference

- When describing fluid motion, it is important to be consistent with the frame of reference
- In fluid dynamics, there are two main ways of addressing this
- With the *Eulerian* frame of reference, we describe the motion of the fluid from some fixed point in space
- With the *Lagrangian* frame of reference, we describe the motion of the fluid from the point of view moving with the fluid itself
- Eulerian simulations typically use a fixed grid or similar structure and store velocities at every point in the grid
- Lagrangian simulations typically use particles that move with the fluid itself. Velocities are stored on the particles that are irregularly spaced throughout the domain
- We will stick with an Eulerian point of view today, but we will look at Lagrangian methods in the next lecture when we discuss particle based fluid simulation

Velocity Field

- We will describe the equations of motion for a basic incompressible fluid (such as air or water)
- To keep it simple, we will assume uniform density and temperature
- The main field that we are interested in therefore, is the velocity $\mathbf{v}(\mathbf{x}, t)$
- We assume that our field is defined over some domain (such as a rectangle or box) and that we have some numerical representation of the field (such as a uniform grid of velocity vectors)
- We will effectively be applying Newton's second law by computing a force everywhere on the grid, and then converting it to an acceleration by $\mathbf{f} = m\mathbf{a}$, however, as we are assuming uniform density (mass/volume), then the m term is always constant, and we can assume that it is just 1.0
- Therefore, we are really just interested in computing the acceleration $\frac{d\mathbf{v}}{dt}$ at every point on the grid

Transport Equations

- Before looking at the full Navier-Stokes equation, we will look at some simpler examples of *transport equations* and related concepts
  - Advection
  - Convection
  - Diffusion
  - Viscosity
  - Pressure gradient
  - Incompressibility

Advection

- Let us assume that we have a velocity field **v**(**x**) and we have some scalar field *s*(**x**) that represents some scalar quantity that is being transported through the velocity field
- For example, **v** might be the velocity of air in the room and *s* might represent temperature, or the concentration of some pigment or smoke, etc.
- As the fluid moves around, it will transport the scalar field with it. We say that the scalar field is *advected* by the fluid
- The rate of change of the scalar field at some location is:

$$\frac{ds}{dt} = -\mathbf{v} \cdot \nabla s$$

Convection

- The velocity field **v** describes the movement of the fluid down to the molecular level
- Therefore, all properties of the fluid at a particular point should be advected by the velocity field
- This includes the property of velocity itself!
- The advection of velocity through the velocity field is called *convection*

$$\frac{d\mathbf{v}}{dt} = -\mathbf{v} \cdot \nabla \mathbf{v}$$

- Remember that $d\mathbf{v}/dt$ is an acceleration, and since **f**=*m***a**, we are really describing a force

Diffusion

- Lets say that we put a drop of red food coloring in a motionless water tank. Due to random molecular motion, the red color will gradually diffuse throughout the tank until it reaches equilibrium
- This is known as a *diffusion* process and is described by the diffusion equation

$$\frac{ds}{dt} = k\nabla^2 s$$

- The constant *k* describes the rate of diffusion
- Heat diffuses through solids and fluids through a similar process and is described by a diffusion equation

Viscosity

- *Viscosity* is essentially the diffusion of velocity in a fluid and is described by a diffusion equation as well:

$$\frac{d\mathbf{v}}{dt} = \mu \nabla^2 \mathbf{v}$$

- The constant μ is the coefficient of viscosity and describes how viscous the fluid is. Air and water have low values, whereas something like syrup would have a relatively higher value
- Some materials like modeling clay or silly putty are extremely viscous fluids that can behave similar to solids
- Like convection, viscosity is a force. It resists relative motion and tries to smooth out the velocity field

Pressure Gradient

- Fluids flow from high pressure regions to low pressure regions in the direction of the *pressure gradient*

$$\frac{d\mathbf{v}}{dt} = -\nabla p$$

- The difference in pressure acts as a force in the direction from high to low (thus the minus sign)

Transport Equations

- Advection: $\frac{ds}{dt} = -\mathbf{v} \cdot \nabla s$
- Convection: $\frac{d\mathbf{v}}{dt} = -\mathbf{v} \cdot \nabla \mathbf{v}$
- Diffusion: $\frac{ds}{dt} = k\nabla^2 s$
- Viscosity: $\frac{d\mathbf{v}}{dt} = \mu\nabla^2 \mathbf{v}$
- Pressure: $\frac{d\mathbf{v}}{dt} = -\nabla p$

Navier-Stokes Equation

- The complete *Navier-Stokes equation* describes the strict conservation of mass, energy, and momentum within a fluid
- Energy can be converted between potential, kinetic, and thermal states
- The full equation accounts for fluid flow, convection, viscosity, sound waves, shock waves, thermal buoyancy, and more
- However, simpler forms of the equation can be derived for specific purposes. Fluid simulation, for example, typically uses a limited form known as the incompressible flow equation

Incompressibility

- Real fluids have some degree of compressibility. Gasses are very compressible and even liquids can be compressed some
- Sound waves in a fluid are caused by compression, as are supersonic shocks, but generally, we are not interested in modeling these fluid behaviors
- We will therefore assume that the fluid is incompressible and we will enforce this as a constraint
- Incompressibility requires that there is zero divergence of the velocity field everywhere

$$\nabla \cdot \mathbf{v} = 0$$

- This is actually very reasonable, as compression has a negligible affect on fluids moving well below the speed of sound

Navier-Stokes Equation

- The incompressible Navier-Stokes equation describes the forces on a fluid as the sum of convection, viscosity, and pressure terms:

$$\frac{d\mathbf{v}}{dt} = -\mathbf{v} \cdot \nabla\mathbf{v} + \mu\nabla^2\mathbf{v} - \nabla p$$

- In addition, we also have the incompressibility constraint:

$$\nabla \cdot \mathbf{v} = 0$$

Computational Fluid Dynamics

Computational Fluid Dynamics

- Now that we’ve seen the equations of fluid dynamics, we turn to the issue of computer implementation
- The Del operations and the transport equations are defined in terms of general calculus fields
- We must address the issue of how we represent fields on the computer and how we perform calculus operations on them

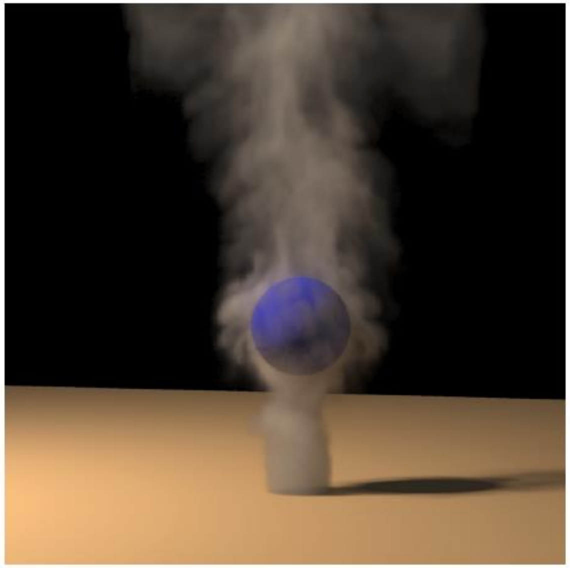
Numerical Representation of Fields

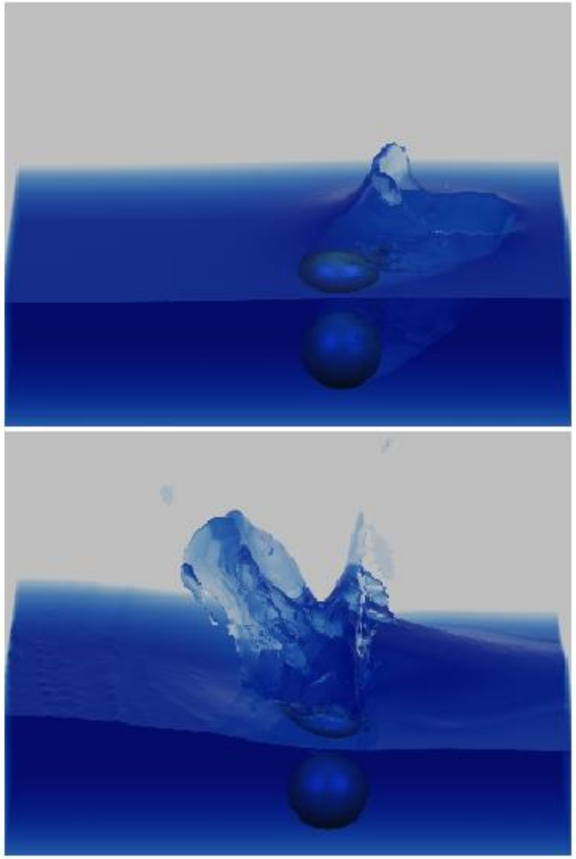
- A scalar or vector field represents a continuously variable value across space that can have infinite detail
- Obviously, on the computer, we can’t truly represent the value of the field everywhere to this level, so we must use some form of approximation
- A standard approach to representing a continuous field is to sample it at some number of discrete points and use some form of interpolation to get the value between the points
- There are several choices of how to arrange our samples:
  - Uniform grid
  - Hierarchical grid
  - Irregular mesh
  - Particle based

Uniform Grids

- *Uniform grids* are easy to deal with and tend to be computationally efficient due to their simplicity
- It is very straightforward to compute derivatives on uniform grids
- However, they require large amounts of memory to represent large domains
- They don’t adapt well to varying levels of detail, as they represent the field to an even level of detail everywhere

Uniform Grids

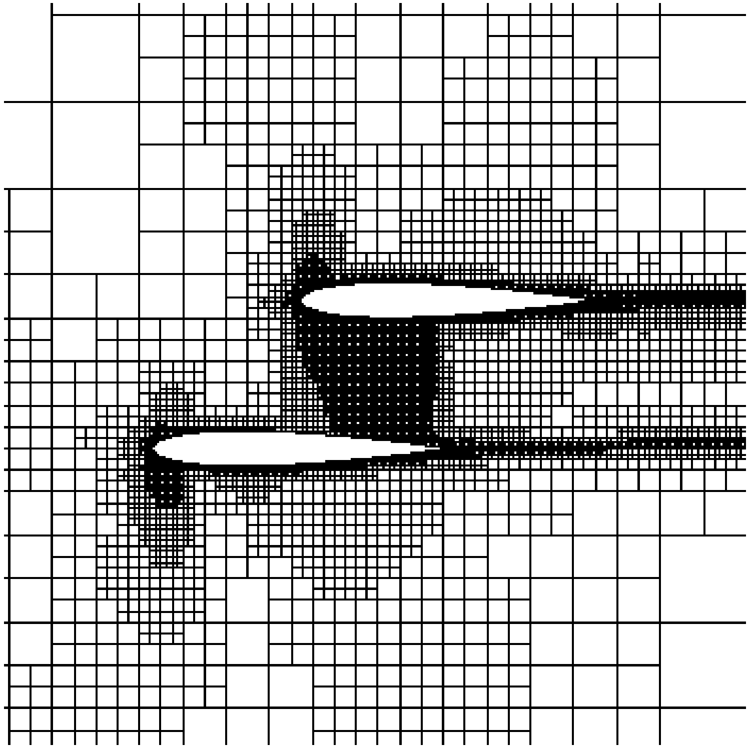


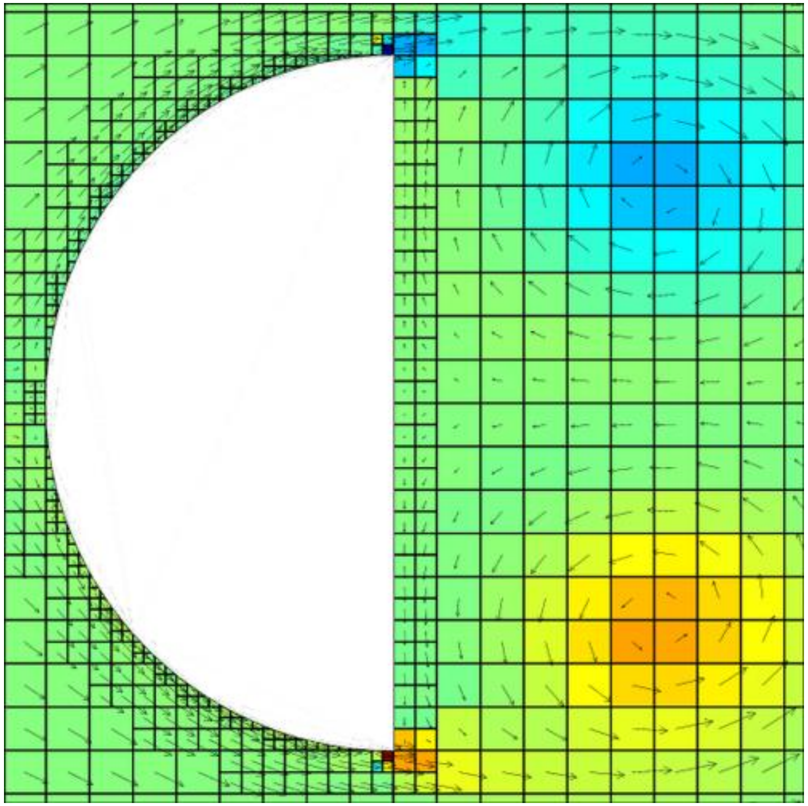


Hierarchical Grids

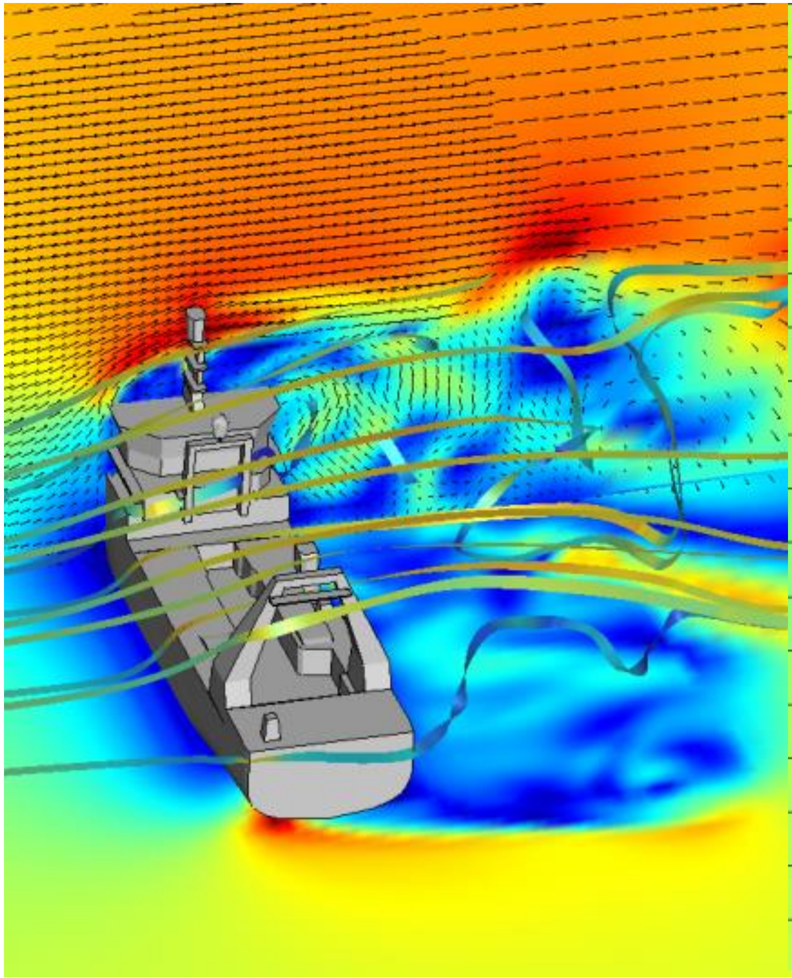
- *Hierarchical grids* attempt to benefit from the simplicity of uniform grids, but also have the additional benefit of scaling well to large problems and varying levels of detail
- The grid resolution can locally increase to handle more detailed flows in regions that require it
- This allows both memory and compute time to be used efficiently and adapt automatically to the problem complexity

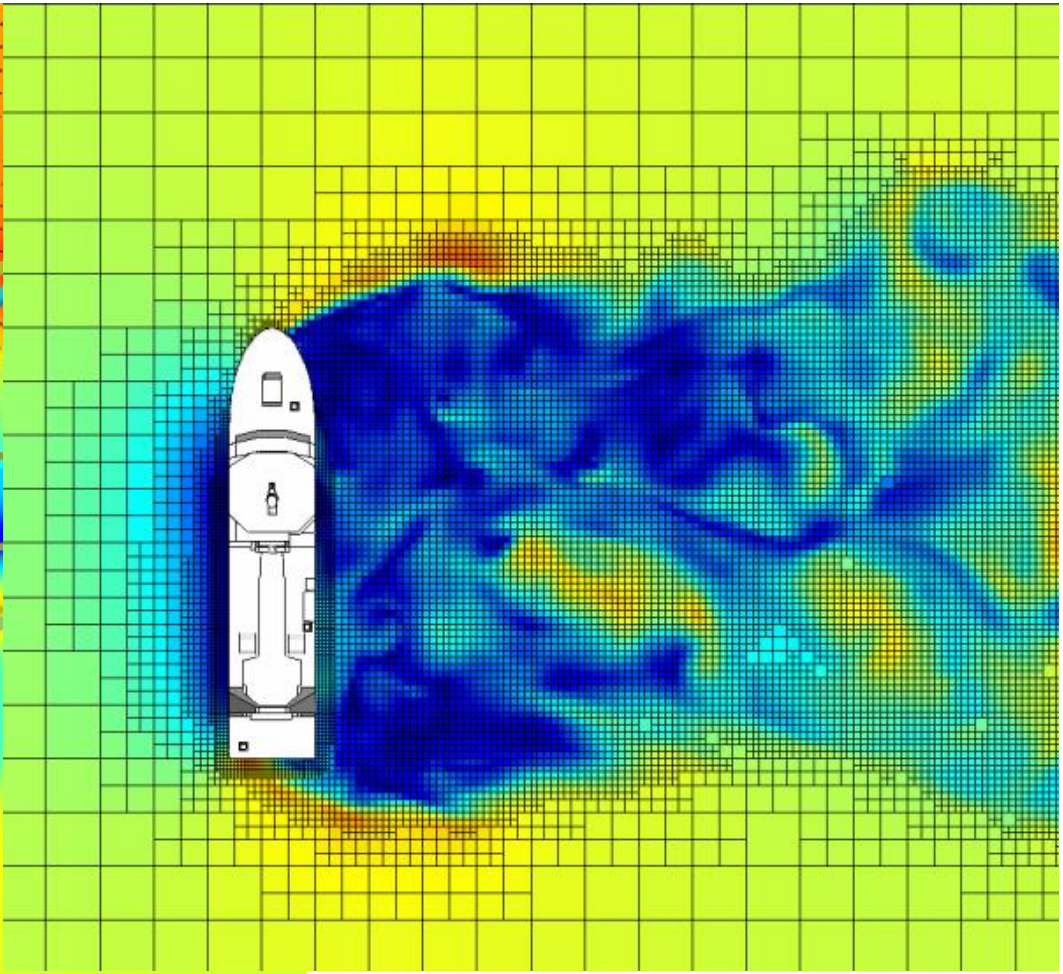
Hierarchical Grids





Hierarchical Grids

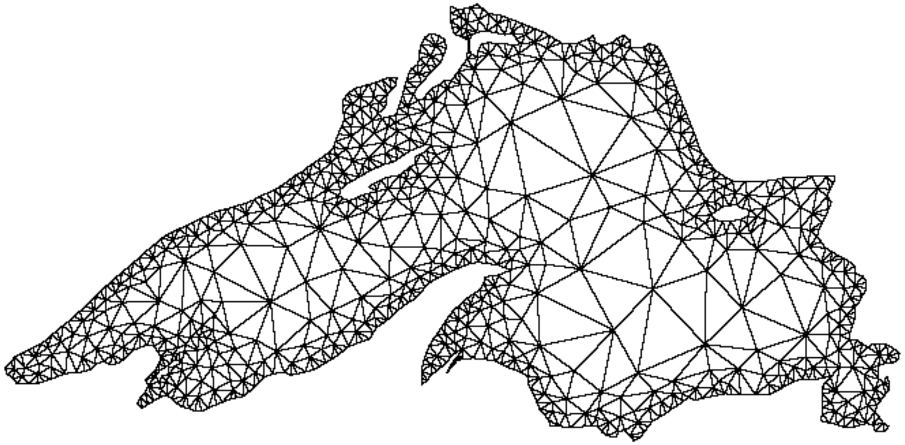


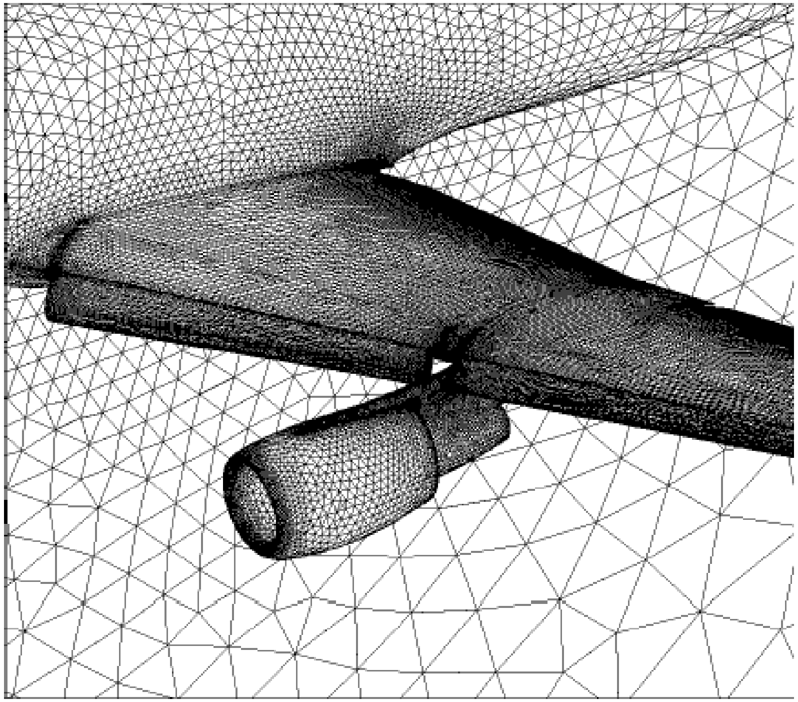


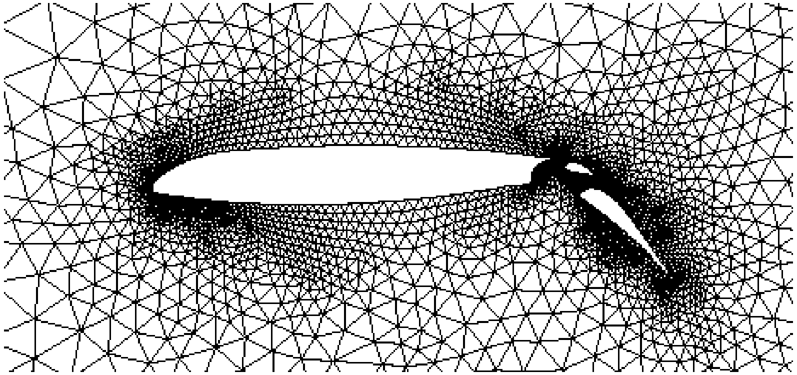
Irregular Meshes

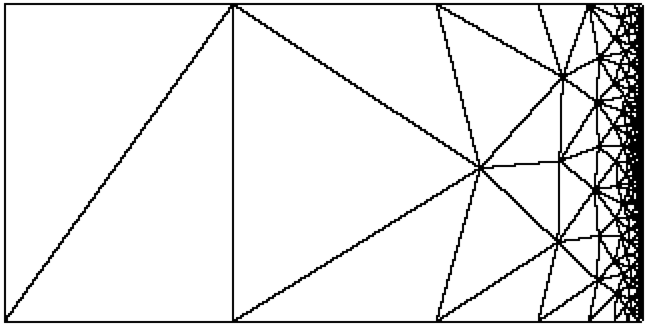
- *Irregular meshes* are built from triangles in 2D and tetrahedra in 3D
- Irregular meshes are used extensively in engineering applications, but less so in computer animation
- One of the main benefits of irregular meshes is their ability to adapt to complex domain geometry
- They also adapt well to varying levels of detail
- They can be quite complex to generate however and can have a lot of computational overhead in highly dynamic situations with moving objects
- If the irregular mesh changes over time to adapt to the problem complexity, it is called an *adaptive mesh*

Irregular Mesh









Adaptive Meshes

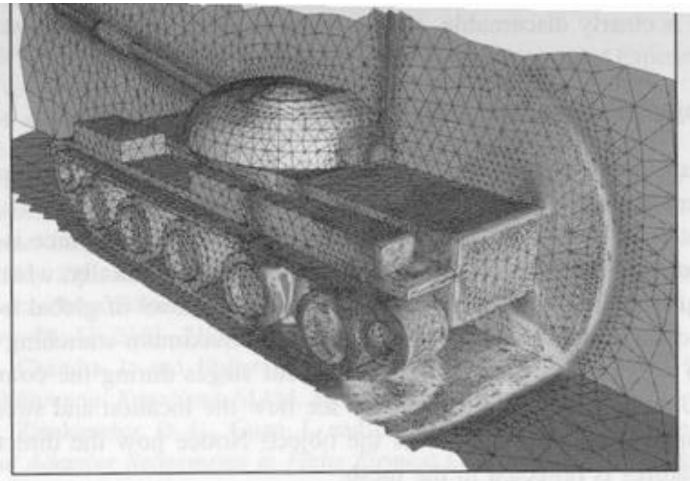
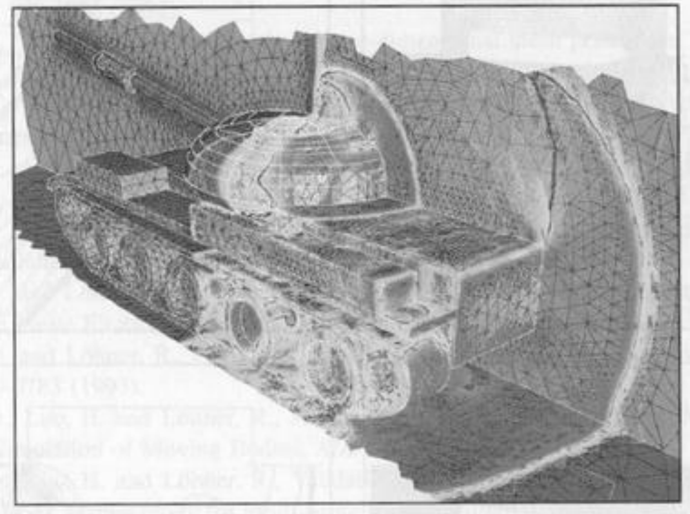
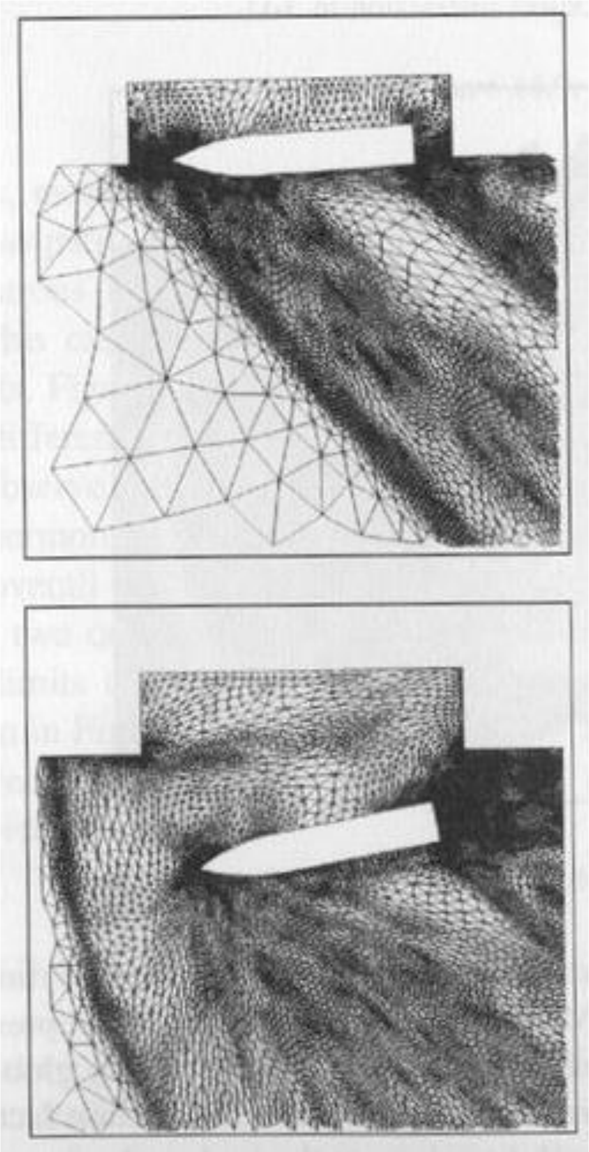


Figure 14.10a Shock-object interaction in 3-D

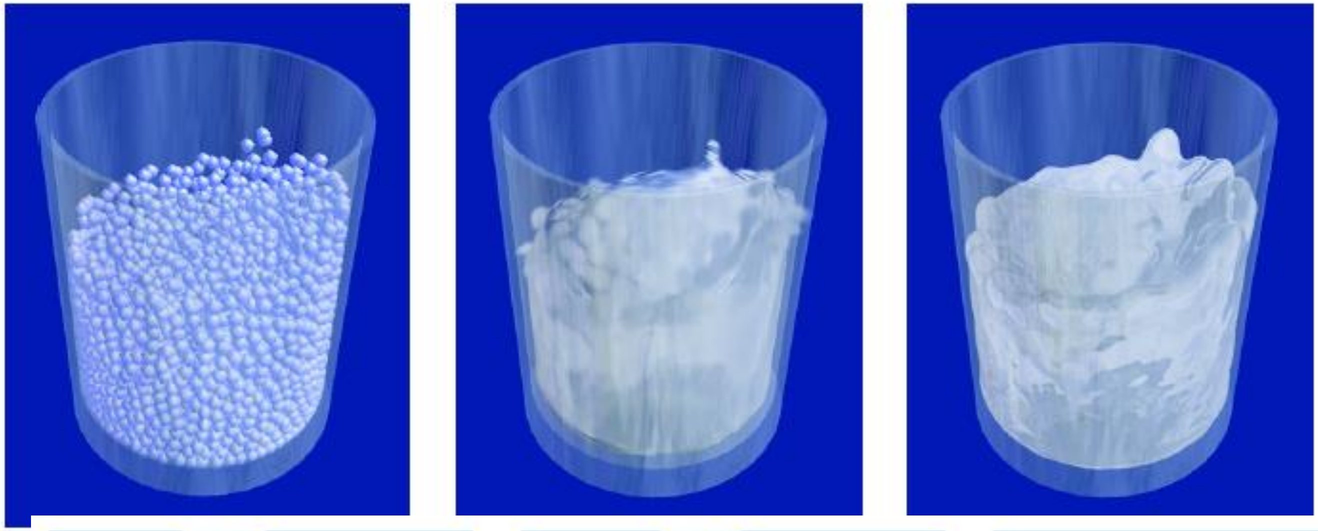


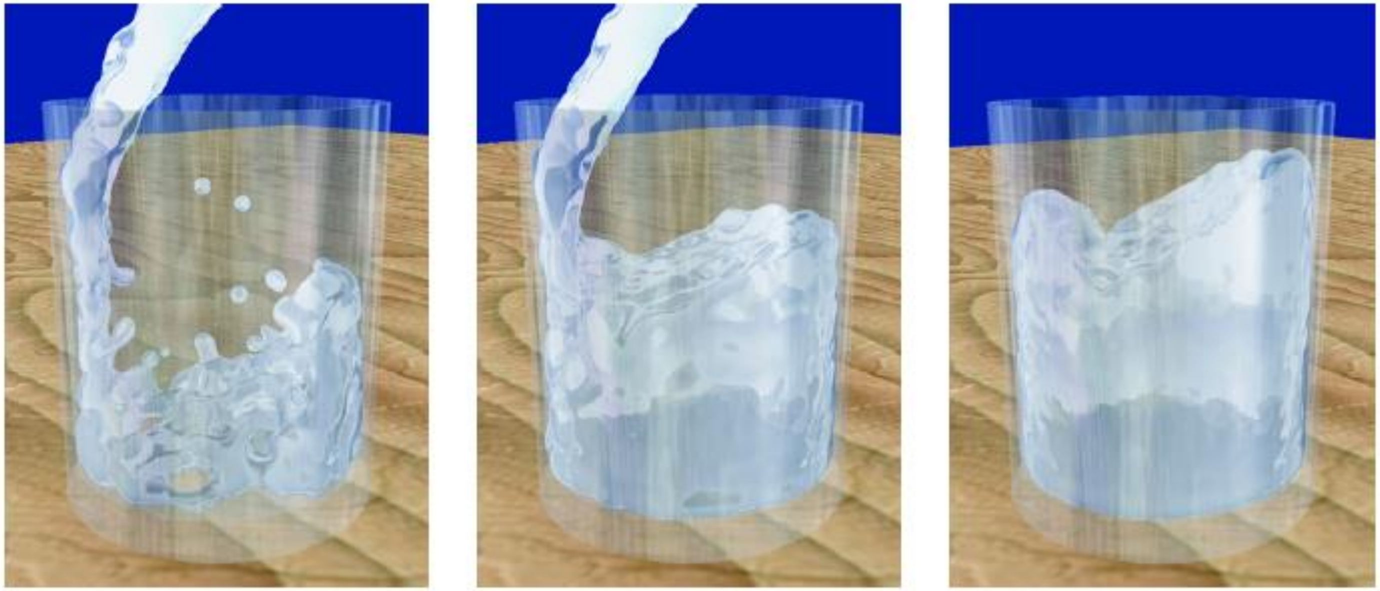


Particle-Based (Meshless)

- Instead of using a mesh with well defined connectivity, particle methods sample the field on a set of irregularly distributed particles
- Particles aren’t meant to be 0 dimensional points- they are assumed to represent a small ‘smear’ of the field, over some radius, and the value of the field at any point is determined by several nearby particles
- Calculating derivatives can be tricky and there are several approaches
- Particle methods are very well suited to water and liquid simulation for a variety of reasons and have been gaining a lot of popularity in the computer graphics industry recently

Particle Based





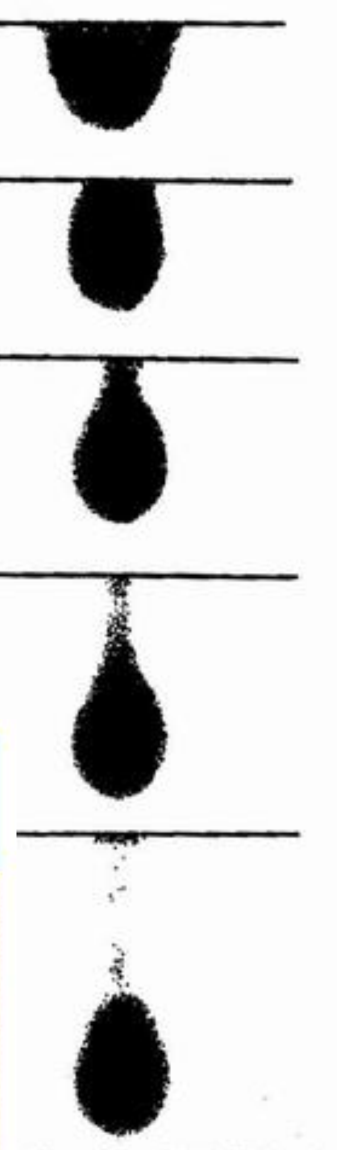


Figure 9.2 — Drop fall.

Uniform Grids & Finite Differencing

- For today, we will just consider the case of uniform grid
- A scalar field is represented as a 2D/3D array of floats and a vector field is a 2D/3D array of vectors
- We will use a technique called finite differencing to compute derivatives of the fields

Finite Differencing

- Lets say we have a scalar field $s(\mathbf{x},t)$ stored on a uniform grid and we want to compute a new vector field $\mathbf{v}(\mathbf{x},t)$ which is the gradient of s
- For every grid cell, we will calculate the gradient (slope) by using the values of the neighboring cells

Finite Difference First Derivatives

- If we have a scalar field *s*(**x**,*t*) stored on a uniform grid, we can approximate the partial derivative along *x* at grid cell *i* as:

$$\frac{\partial s_i}{\partial x} \approx \frac{s_{i+1} - s_{i-1}}{x_{i+1} - x_{i-1}} = \frac{s_{i+1} - s_{i-1}}{2\Delta x}$$

- Where cell *i*+1 is the cell in the +*x* direction and cell *i*-1 is in the –*x* direction
- Also Δ*x* is the cell size in the *x* direction
- All of the partial derivatives in the gradient, divergence, and curl can be computed in this way

Finite Difference Second Derivative

- The second derivative can be computed in a similar way:

$$\frac{\partial^2 s_i}{\partial x^2} \approx \frac{s_{i+1} - 2s_i + s_{i-1}}{\Delta x^2}$$

- This can be used in the computation of the Laplacian
- Remember, these are based on the assumption of a uniform grid. To calculate the derivatives on irregular meshes or with particle methods, the formulas get more complex

Boundary Conditions

- Finite differencing requires examining values in neighboring cells to compute derivatives
- However, for cells on the boundary of the domain, they may not have any neighbors
- Therefore, we need to assign some sort of boundary conditions to sort out how they are treated
- In fluid dynamics, we might want to treat a boundary as a wall, or as being open to the outside environment. If a wall, it might have friction or be smooth, or have other relevant properties. If open, it might act as a source or sink, or neither
- There are a lot of options on how to deal with boundaries, so we will not worry about the details for today
- Just understand that they define some sort of case-specific modification to how the derivatives are computed along the boundaries

Solving the Navier-Stokes Equation

- Now that we know how to represent a field and compute derivatives, lets proceed to solving the Navier-Stokes equation

$$\frac{d\mathbf{v}}{dt} = -\mathbf{v} \cdot \nabla\mathbf{v} + \mu\nabla^2\mathbf{v} - \nabla p$$

- We will use a two-step method called a *projection method*
- In the first step, we advance the velocity field according to the convection and viscosity terms, generating a new velocity field that will probably violate the incompressibility constraint
- In the second step, we calculate a pressure field that corrects the divergence caused in the first step, and the gradient of the pressure field is added to the velocity, thus maintaining incompressibility
- The pressure field essentially projects the velocity field onto the space of divergence-free vector fields, and so is known as a projection method

Pressure Projection

- Step 1: Advance velocity according to convection & viscosity terms

$$\mathbf{v}^* = \mathbf{v}_0 + \Delta t(\mu \nabla^2 \mathbf{v}_0 - \mathbf{v}_0 \cdot \nabla \mathbf{v}_0)$$

- Step 2: Solve for unknown pressure field *p*

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}^*$$

- Step 2.5: Add pressure gradient term to get new velocity

$$\mathbf{v}_1 = \mathbf{v}^* - \Delta t(\nabla p)$$

Step 1: Convection and Viscosity

- In the first step, we compute a new candidate velocity field $\mathbf{v}^*$ according to the convection and viscosity forces

$$\mathbf{v}^* = \mathbf{v}_0 + \Delta t(\mu \nabla^2 \mathbf{v}_0 - \mathbf{v}_0 \cdot \nabla \mathbf{v}_0)$$

- We use the finite differencing formulas to compute the gradient and Laplacian of the original velocity field $\mathbf{v}_0$ at every cell

Step 2: Solve Pressure

- After step 1, the candidate velocity field $\mathbf{v}^*$ will not be divergence free

$$\nabla \cdot \mathbf{v}^* \neq 0$$

- We assume that a pressure field exists that will counteract the divergence, such that when its effects are added, the new field will be divergence free

$$\nabla \cdot (\mathbf{v}^* - \Delta t(\nabla p)) = 0$$

- Rearranging this, we get:

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}^*$$

- Which is known as a *Poisson equation*

Step 2: Solve Pressure

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}^*$$

- Finite differencing the Poisson equation creates a large number of simultaneous algebraic equations that must be solved
- Several options exist for solving these systems
  - Direct solution
  - Iterative relaxation scheme
  - Conjugate gradient solver
  - Multi-grid solver
- Solving the Poisson equation is really the key computational step in fluid dynamics… however… we won’t get into the details today

Advanced Topics

- Multi-phase flows
- Fluid interfaces
- Surface tension
- Fluid-solid interaction
- Phase transitions
- Thermal buoyancy
- Compressible flow
- Supersonic shocks
- Turbulence & mixing