GAMES103: Intro to Physics-Based Animation

Waves: An Intro to Fluid Dynamics

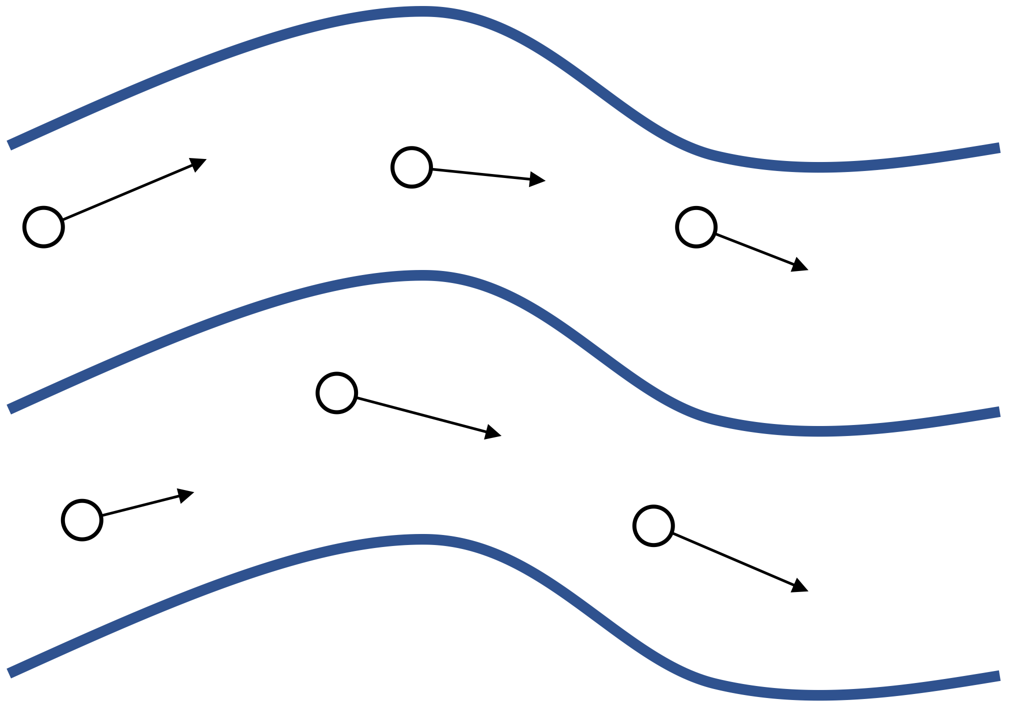
Huamin Wang

Dec 2021

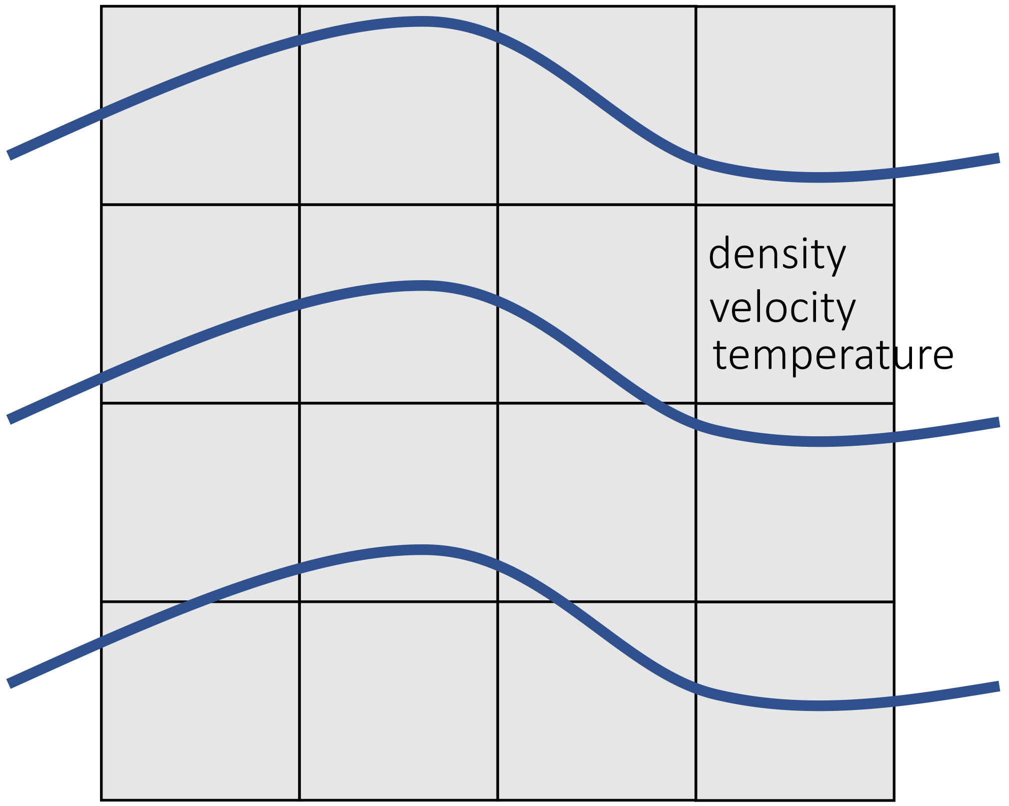
Fluid Effects

Unlike other bodies, Fluids exhibit highly volatile behaviors. As a result, it's difficult to come up with a single, efficient way for simulating all of fluid effects.

Two Types of Simulation Approaches



Lagrangian Approach
(dynamic particles or mesh)
Node movement carries physical quantities
(mass, velocity, …).

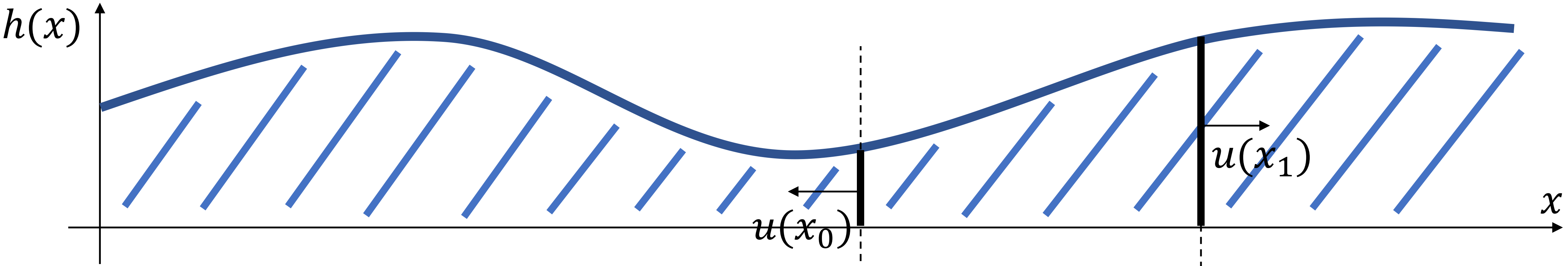


Eulerian Approach
(static grid or mesh)
Grid/Mesh doesn’t move. Stored physical
quantities change.

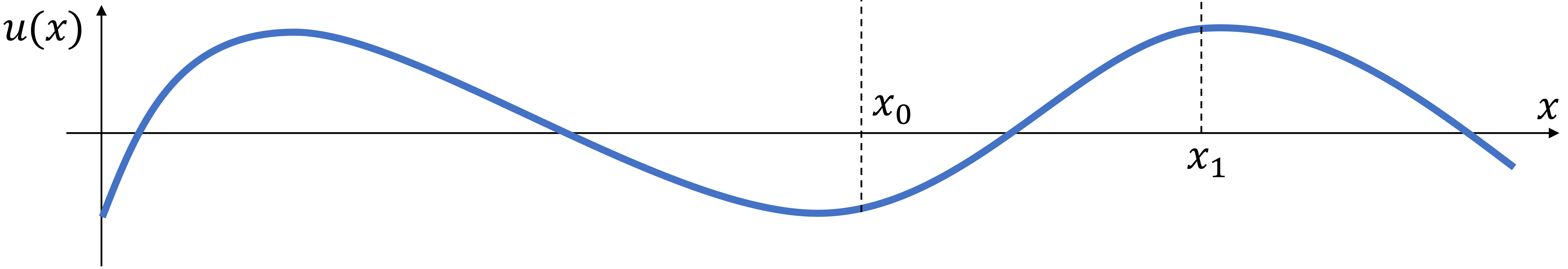
A Height Field Model

Height Field

In 2D, a (1.5D) height field is a height function $h(x)$.



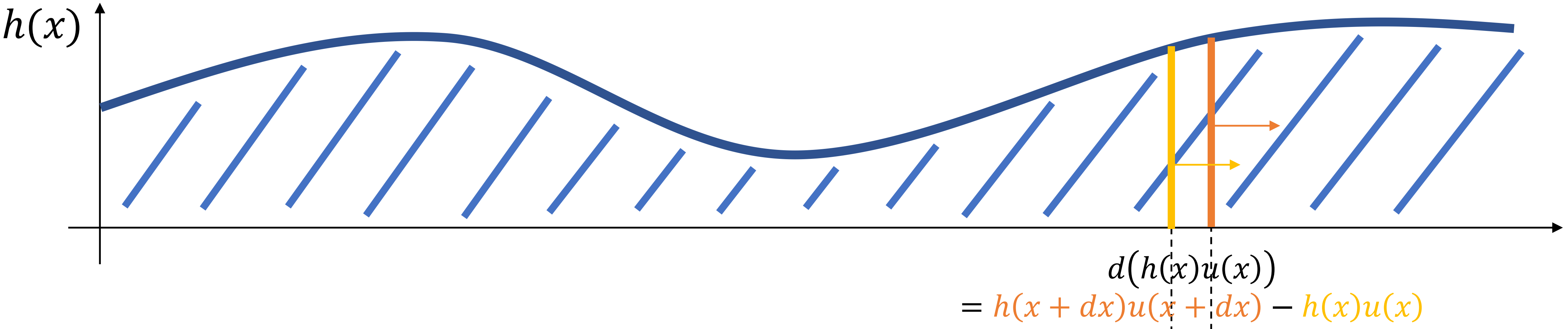
The velocity is also a function of x: $u(x)$.



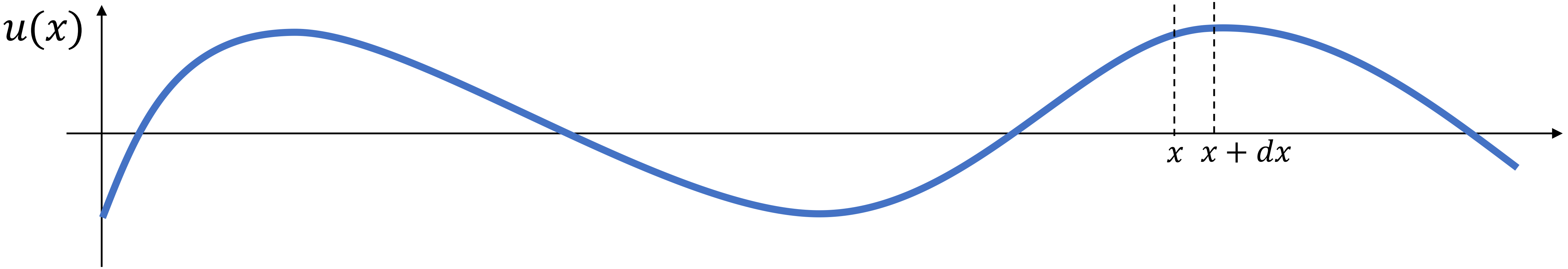
Height Field

In 2D, a height field is a height function $h(x)$.

$$\boxed{\frac{dh(x)}{dt} + \frac{d(h(x)u(x))}{dx} = 0}$$



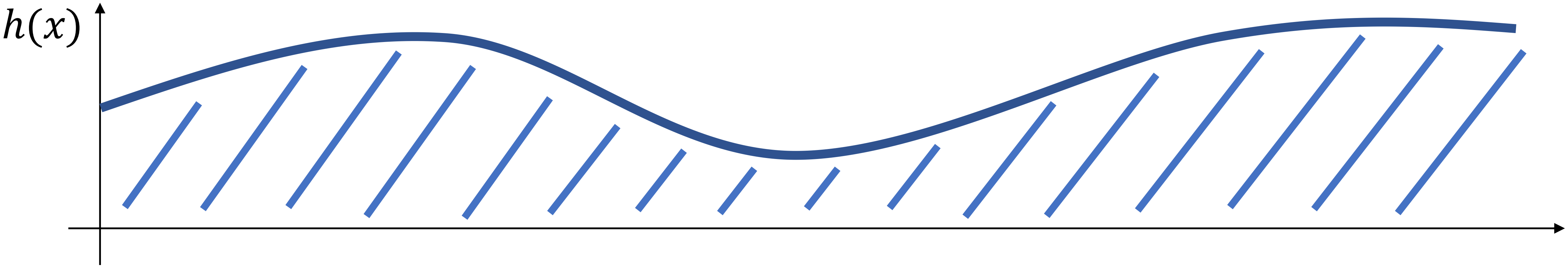
The velocity is also a function of x: $u(x)$.



Height Field

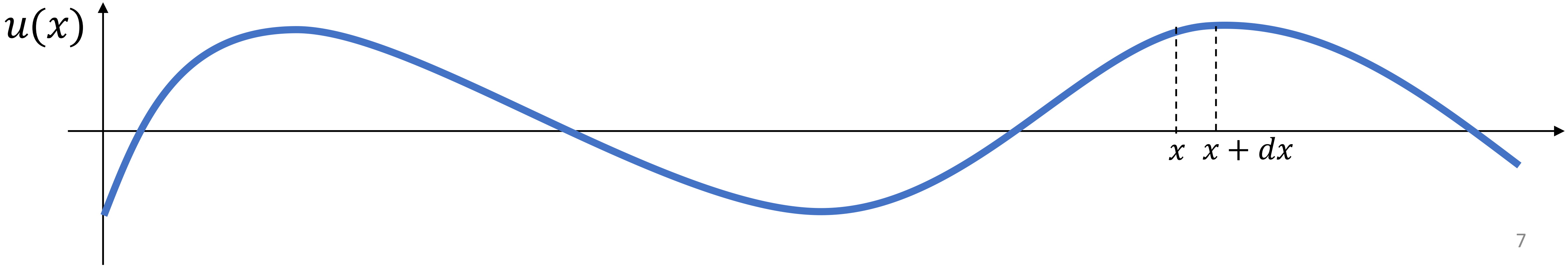
In 2D, a height field is a height function $h(x)$.

$$\frac{dh(x)}{dt}+\frac{d(h(x)u(x))}{dx}=0$$



The velocity is also a function of x: $u(x)$.

$$\frac{du(x)}{dt}=\underbrace{\boxed{-u(x)\frac{du(x)}{dx}}}_{\text{advection}}-\frac{1}{\rho}\frac{dP(x)}{dx}+\underbrace{\boxed{a(x)}}_{\text{external}}$$



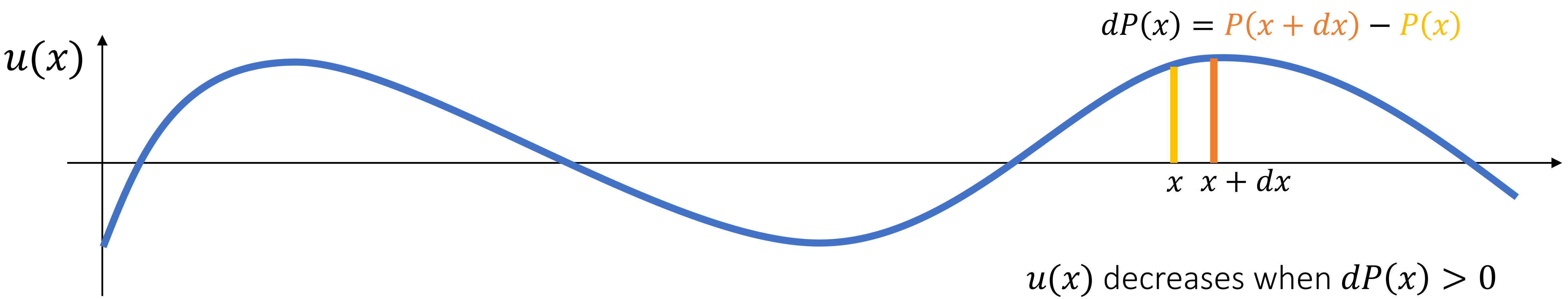
Height Field

Ignoring advection and external acceleration, we get a simple form:

$$\boxed{\frac{du(x)}{dt} = -\frac{1}{\rho}\frac{dP(x)}{dx}}$$

ρ: density

$P(x)$: pressure



$u(x)$ decreases when $dP(x) > 0$

$u(x)$ increases when $dP(x) < 0$

Shallow Wave Equation

We now have two equations:

$$\boxed{\frac{dh}{dt}+\frac{d(hu)}{dx}=0} \rightarrow \boxed{\frac{dh}{dt}+\cancel{u\frac{dh}{dx}}+h\frac{du}{dx}=0} \xrightarrow{dt} \boxed{\frac{d^2h}{dt^2}+h\frac{d^2u}{dxdt}=0}$$

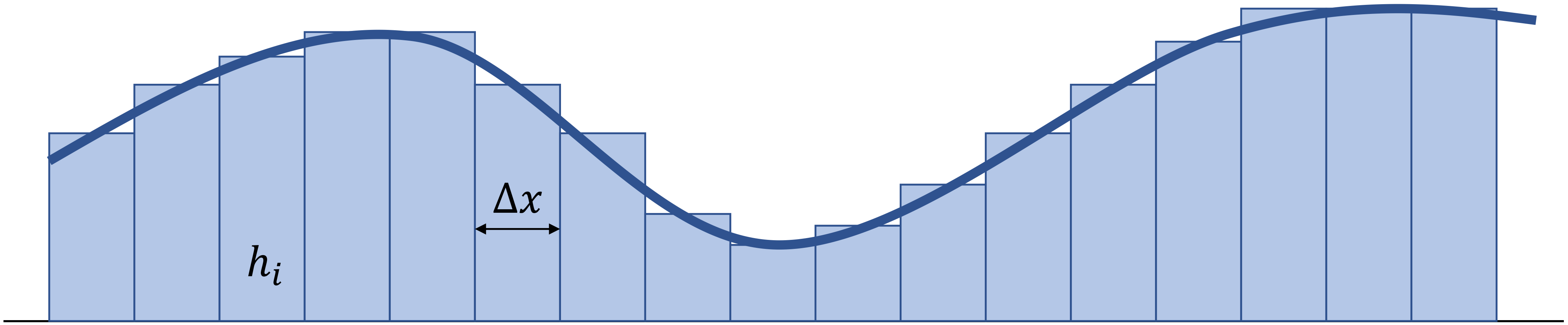
$$\boxed{\frac{du}{dt}=-\frac{1}{\rho}\frac{dP}{dx}} \xrightarrow{dx} \boxed{\frac{d^2u}{dxdt}=-\frac{1}{\rho}\frac{d^2P}{dx^2}}$$

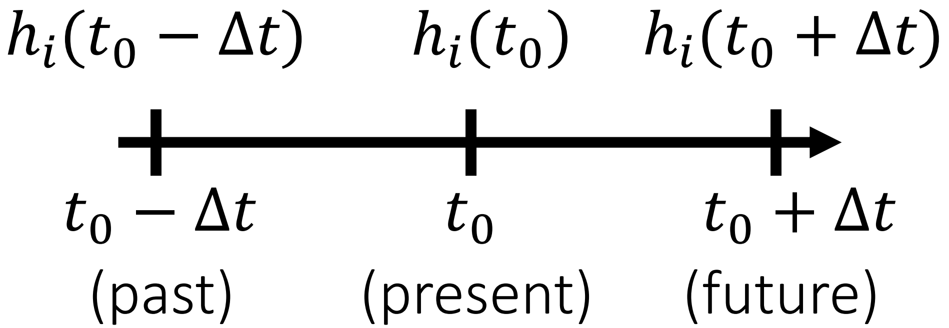
We can then eliminate u and formulate the shallow wave equation:

$$\boxed{\frac{d^2h}{dt^2}=\frac{h}{\rho}\frac{d^2P}{dx^2}}$$

Discretization

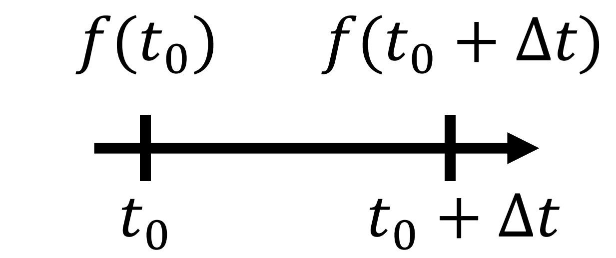
We discretize a continuous height field into a discrete set of height columns.





Finite Differencing

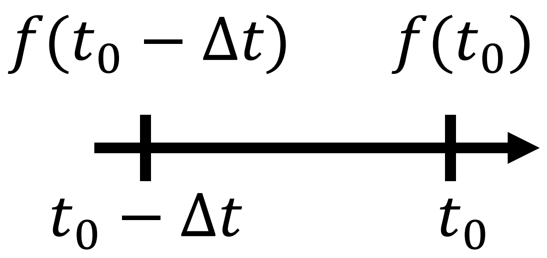
The idea of finite differencing is to use the difference to approximate the derivative.



$$f(t_0 + \Delta t) = f(t_0) + \Delta t \frac{df(t_0)}{dt} + \frac{\Delta t^2}{2}\frac{d^2 f(t_0)}{dt^2} + \ldots$$

Forward differencing (first-order)

$$\boxed{\frac{df(t_0)}{dt} \approx \frac{f(t_0 + \Delta t) - f(t_0)}{\Delta t}}$$



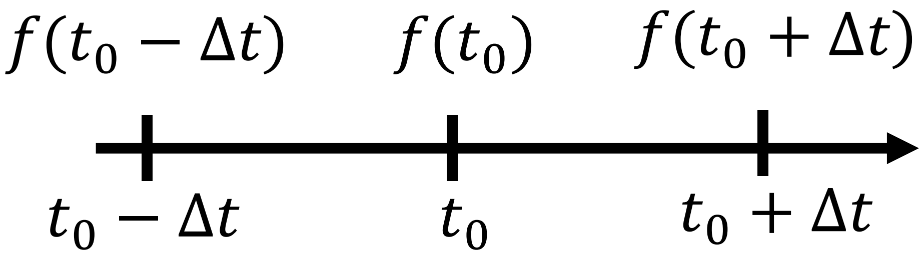
$$f(t_0 - \Delta t) = f(t_0) - \Delta t \frac{df(t_0)}{dt} + \frac{\Delta t^2}{2}\frac{d^2 f(t_0)}{dt^2} + \ldots$$

Backward differencing (first-order)

$$\boxed{\frac{df(t_0)}{dt} \approx \frac{f(t_0) - f(t_0 - \Delta t)}{\Delta t}}$$

Finite Differencing

The idea of finite differencing is to use the difference to approximate the derivative.



$$f(t_0 + \Delta t) = f(t_0) + \Delta t \frac{df(t_0)}{dt} + \frac{\Delta t^2}{2}\frac{d^2 f(t_0)}{dt^2} + \ldots$$

$$f(t_0 - \Delta t) = f(t_0) - \Delta t \frac{df(t_0)}{dt} + \frac{\Delta t^2}{2}\frac{d^2 f(t_0)}{dt^2} + \ldots$$

Central differencing (second-order)

$$\boxed{\frac{df(t_0)}{dt} \approx \frac{f(t_0 + \Delta t) - f(t_0 - \Delta t)}{2\Delta t}}$$

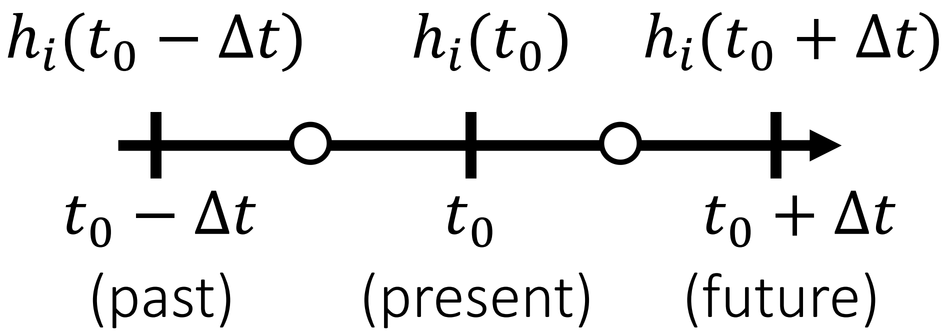
Second-Order Derivatives

We apply central differencing twice to estimate d^2h_i/dt^2.

$$\frac{dh_i(t_0 + 0.5\Delta t)}{dt} \approx \frac{h_i(t_0 + \Delta t) - h_i(t_0)}{\Delta t} \qquad \frac{dh_i(t_0 - 0.5\Delta t)}{dt} \approx \frac{h_i(t_0) - h_i(t_0 - \Delta t)}{\Delta t}$$

$$\boxed{\frac{d^2h_i(t_0)}{dt^2} \approx \frac{\frac{dh_i(t_0+0.5\Delta t)}{dt} - \frac{dh_i(t_0-0.5\Delta t)}{dt}}{\Delta t} \approx \frac{h_i(t_0 + \Delta t) + h_i(t_0 - \Delta t) - 2h_i(t_0)}{\Delta t^2}}$$

Laplacian

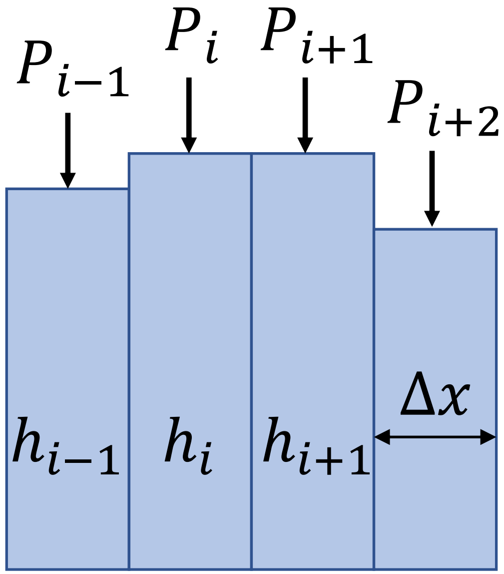


Second-Order Derivatives

Similarly, we apply central differencing twice to estimate d^2P/dx^2.

$$\frac{dP_{i+0.5}}{dt} \approx \frac{P_{i+1} - P_i}{\Delta x} \qquad \frac{dP_{i-0.5}}{dx} \approx \frac{P_i - P_{i-1}}{\Delta x}$$

$$\frac{d^2P_i}{dx^2} \approx \frac{\frac{dP_{i+0.5}}{dx} - \frac{dP_{i-0.5}}{dx}}{\Delta x} \approx \frac{P_{i+1} + P_{i-1} - 2P_i}{\Delta x^2}$$



Discretized Shallow Wave Equation

We can now discretize the shallow wave equation $\frac{d^2h}{dt^2} = \frac{h}{\rho}\frac{d^2P}{dx^2}$.

$$\frac{d^2h_i(t_0)}{dt^2} \approx \frac{h_i(t_0+\Delta t)+h_i(t_0-\Delta t)-2h_i(t_0)}{\Delta t^2}$$

$$\frac{d^2P_i}{dx^2} \approx \frac{P_{i+1}+P_{i-1}-2P_i}{\Delta x^2}$$

$$\frac{h_i(t_0+\Delta t)+h_i(t_0-\Delta t)-2h_i(t_0)}{\Delta t^2} = \frac{h_i}{\rho}\left(\frac{P_{i+1}+P_{i-1}-2P_i}{\Delta x^2}\right)$$

$$h_i(t_0+\Delta t) = 2h_i(t_0) - h_i(t_0-\Delta t) + \frac{\Delta t^2 h_i}{\Delta x^2 \rho}(P_{i+1}+P_{i-1}-2P_i)$$

Volume Preservation

We want the volume to stay the same. Suppose that $\sum h_i(t) = \sum h_i(t - \Delta t) = V$. But,

$$h_i(t_0 + \Delta t) = 2h_i(t_0) - h_i(t_0 - \Delta t) + \frac{\Delta t^2 h_i}{\Delta x^2 \rho}(P_{i+1} + P_{i-1} - 2P_i)$$

$$\sum h_i(t + \Delta t) = 2\sum h_i(t_0) - \sum h_i(t_0 - \Delta t) + \sum \frac{\Delta t^2 h_i}{\Delta x^2 \rho}(P_{i+1} + P_{i-1} - 2P_i)$$

$$= V + \boxed{\sum \frac{\Delta t^2 h_i}{\Delta x^2 \rho}(P_{i+1} + P_{i-1} - 2P_i)}$$

This may not be zero!!!

Volume Preservation – Solution 1

One way to preserve volume is to modify scheme into:

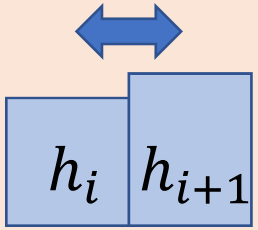
$$h_i(t_0+\Delta t) = 2h_i(t_0) - h_i(t_0-\Delta t) + \frac{\Delta t^2 h_i}{\Delta x^2 \rho}(P_{i+1}+P_{i-1}-2P_i)$$

$$h_i(t_0+\Delta t) = 2h_i(t_0) - h_i(t_0-\Delta t) + \frac{\Delta t^2}{\Delta x^2 \rho}\left(\left(\frac{h_{i-1}+h_i}{2}\right)(P_{i-1}-P_i) + \left(\frac{h_{i+1}+h_i}{2}\right)(P_{i+1}-P_i)\right)$$

$$\sum h_i(t+\Delta t) = V + \frac{\Delta t^2}{\Delta x^2 \rho}\boxed{\sum\left(\left(\frac{h_{i-1}+h_i}{2}\right)(P_{i-1}-P_i) + \left(\frac{h_{i+1}+h_i}{2}\right)(P_{i+1}-P_i)\right)}$$

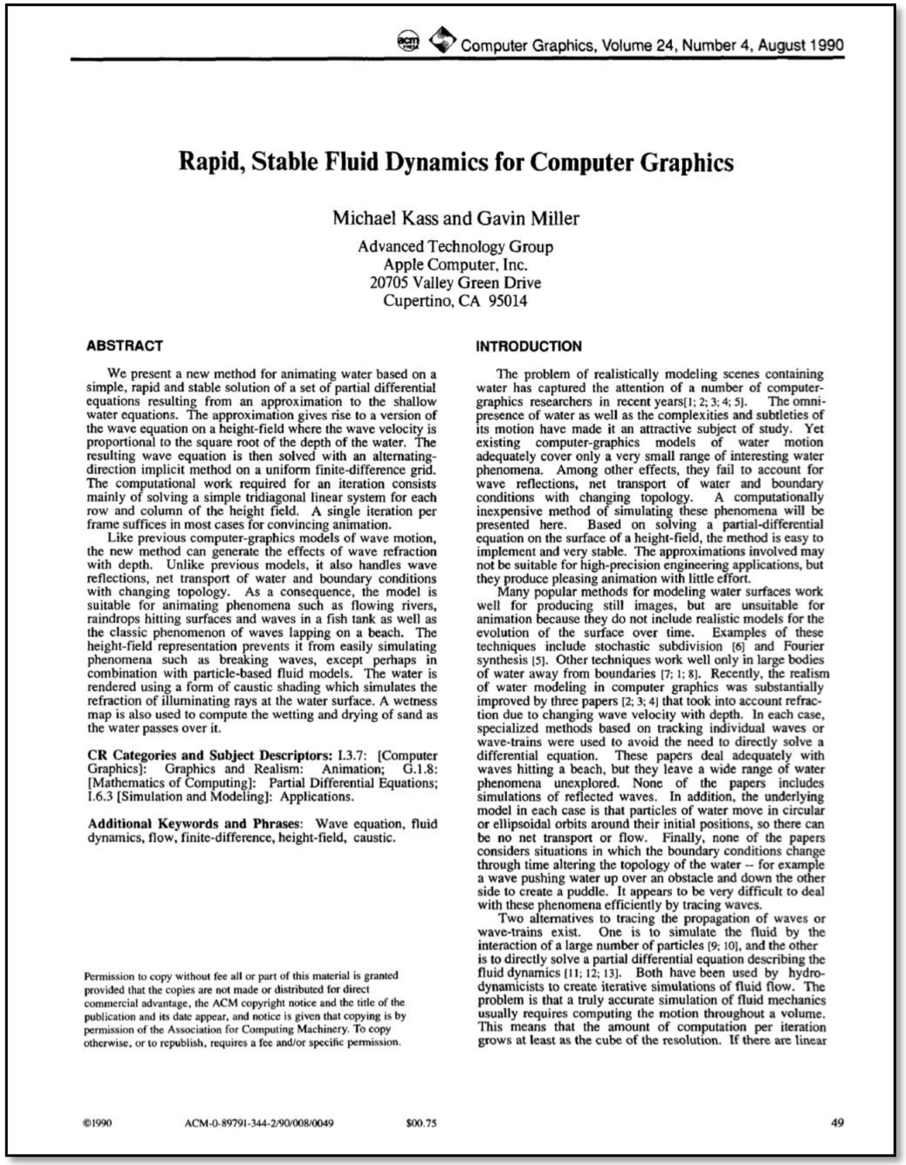
This must be zero!!!

This is because water exchanges between h_i and h_{i+1}.



After-Class Reading

Kass and Miller. 1990. *Rapid, Stable Fluid Dynamics for Computer Graphics*. Computer Graphics.



Computer Graphics, Volume 24, Number 4, August 1990

Rapid, Stable Fluid Dynamics for Computer Graphics

Michael Kass and Gavin Miller

Advanced Technology Group
Apple Computer, Inc.
20705 Valley Green Drive
Cupertino, CA 95014

ABSTRACT

We present a new method for animating water based on a simple, rapid and stable solution of a set of partial differential equations resulting from an approximation to the shallow water equations. The approximation gives rise to a version of the wave equation on a height-field where the wave velocity is proportional to the square root of the depth of the water. The resulting wave equation is then solved with an alternating-direction implicit method on a uniform finite-difference grid. The computational work required for an iteration consists mainly of solving a simple tridiagonal linear system for each row and column of the height field. A single iteration per frame suffices in most cases for convincing animation.

Like previous computer-graphics models of wave motion, the new method can generate the effects of wave refraction with depth. Unlike previous models, it also handles wave reflections, net transport of water and boundary conditions with changing topology. As a consequence, the model is suitable for animating phenomena such as flowing rivers, raindrops hitting surfaces and waves in a fish tank as well as the classic phenomenon of waves lapping on a beach. The height-field representation prevents it from easily simulating phenomena such as breaking waves, except perhaps in combination with particle-based fluid models. The water is rendered using a form of caustic shading which simulates the refraction of illuminating rays at the water surface. A wetness map is also used to compute the wetting and drying of sand as the water passes over it.

CR Categories and Subject Descriptors: I.3.7: [Computer Graphics]: Graphics and Realism: Animation; G.1.8: [Mathematics of Computing]: Partial Differential Equations; I.6.3 [Simulation and Modeling]: Applications.

Additional Keywords and Phrases: Wave equation, fluid dynamics, flow, finite-difference, height-field, caustic.

INTRODUCTION

The problem of realistically modeling scenes containing water has captured the attention of a number of computer-graphics researchers in recent years[1; 2; 3; 4; 5]. The omnipresence of water as well as the complexities and subtleties of its motion have made it an attractive subject of study. Yet existing computer-graphics models of water motion adequately cover only a very small range of interesting water phenomena. Among other effects, they fail to account for wave reflections, net transport of water and boundary conditions with changing topology. A computationally inexpensive method of simulating these phenomena will be presented here. Based on solving a partial-differential equation on the surface of a height-field, the method is easy to implement and very stable. The approximations involved may not be suitable for high-precision engineering applications, but they produce pleasing animation with little effort.

Many popular methods for modeling water surfaces work well for producing still images, but are unsuitable for animation because they do not include realistic models for the evolution of the surface over time. Examples of these techniques include stochastic subdivision [6] and Fourier synthesis [5]. Other techniques work well only in large bodies of water away from boundaries [7; 1; 8]. Recently, the realism of water modeling in computer graphics was substantially improved by three papers [2; 3; 4] that took into account refraction due to changing wave velocity with depth. In each case, specialized methods based on tracking individual waves or wave-trains were used to avoid the need to directly solve a differential equation. These papers deal adequately with waves hitting a beach, but they leave a wide range of water phenomena unexplored. None of the papers includes simulations of reflected waves. In addition, the underlying model in each case is that particles of water move in circular or ellipsoidal orbits around their initial positions, so there can be no net transport or flow. Finally, none of the papers considers situations in which the boundary conditions change through time altering the topology of the water -- for example a wave pushing water up over an obstacle and down the other side to create a puddle. It appears to be very difficult to deal with these phenomena efficiently by tracing waves.

Two alternatives to tracing the propagation of waves or wave-trains exist. One is to simulate the fluid by the interaction of a large number of particles [9; 10], and the other is to directly solve a partial differential equation describing the fluid dynamics [11; 12; 13]. Both have been used by hydrodynamicists to create iterative simulations of fluid flow. The problem is that a truly accurate simulation of fluid mechanics usually requires computing the motion throughout a volume. This means that the amount of computation per iteration grows at least as the cube of the resolution. If there are linear

Volume Preservation – Solution 2

An easier way to preserve volume is to simply assume h_i in the right term is constant.

$$h_i(t_0 + \Delta t) = 2h_i(t_0) - h_i(t_0 - \Delta t) + \frac{\Delta t^2 h_i}{\Delta x^2 \rho}(P_{i+1} + P_{i-1} - 2P_i)$$

$$h_i(t_0 + \Delta t) = 2h_i(t_0) - h_i(t_0 - \Delta t) + \frac{\Delta t^2 H}{\Delta x^2 \rho}(P_{i+1} + P_{i-1} - 2P_i)$$

$$\sum h_i(t + \Delta t) = V + \frac{\Delta t^2 H}{\Delta x^2 \rho} \boxed{\sum \left((P_{i-1} - P_i) + (P_{i+1} - P_i)\right)}$$

This must be zero!!!

Pressure

The pressure is related to the water height: $P_i = \rho g h_i$.

$$h_i(t_0 + \Delta t) = 2h_i(t_0) - h_i(t_0 - \Delta t) + \frac{\Delta t^2 H}{\Delta x^2 \rho}(P_{i+1} + P_{i-1} - 2P_i)$$

$$h_i(t_0 + \Delta t) = 2h_i(t_0) - h_i(t_0 - \Delta t) + \boxed{\frac{\Delta t^2 H g}{\Delta x^2}}(h_{i+1}(t_0) + h_{i-1}(t_0) - 2h_i(t_0))$$

Replaced by a constant α

Viscosity

Like damping, viscosity tries to slow down the waves.

$$h_i(t_0 + \Delta t) = h_i(t_0) + \boxed{\left(h_i(t_0) - h_i(t_0 - \Delta t)\right)} + \alpha(h_{i+1}(t_0) + h_{i-1}(t_0) - 2h_i(t_0))$$

Momentum is here.

$$h_i(t_0 + \Delta t) = h_i(t_0) + \boxed{\beta}\left(h_i(t_0) - h_i(t_0 - \Delta t)\right) + \alpha(h_{i+1}(t_0) + h_{i-1}(t_0) - 2h_i(t_0))$$

Viscosity constant

Algorithm

A Shallow Wave Simulator

For every cell *i*

$$h_i^{new} \leftarrow h_i + \beta(h_i - h_i^{old})$$

$$h_i^{new} \leftarrow h_i^{new} + \alpha(h_{i-1} - h_i)$$

$$h_i^{new} \leftarrow h_i^{new} + \alpha(h_{i+1} - h_i)$$

For every cell *i*

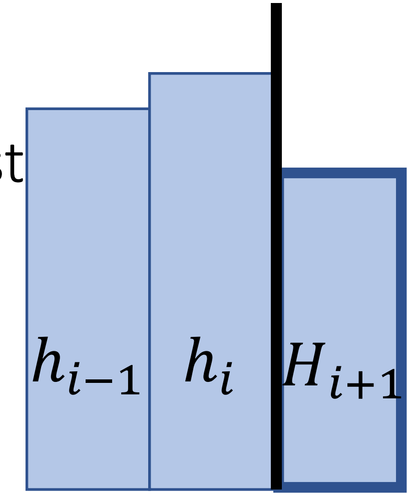
$$h_i^{old} \leftarrow h_i$$

$$h_i \leftarrow h_i^{new}$$

Boundary Conditions

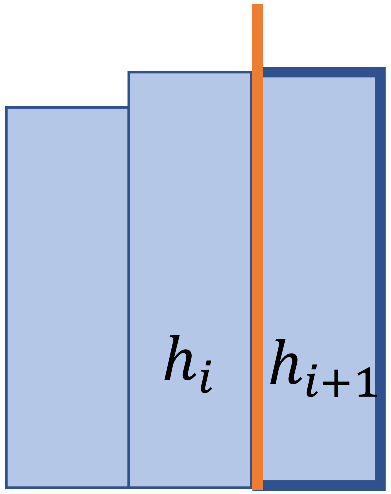
A Dirichlet boundary assumes that the boundary height H_{i+1} is const
It's considered as an open boundary.

$$h_{i+1} - h_i + h_{i-1} - h_i = H_{i+1} - h_i + h_{i-1} - h_i$$



A Neumann boundary specifies the boundary derivatives. For example, a zero-derivative boundary means $h_{i+1} \equiv h_i$. It's considered as a closed boundary.

$$h_{i+1} - h_i + h_{i-1} - h_i = h_{i-1} - h_i$$



No water exchange through the boundary

Algorithm with Neumann Boundaries

A Shallow Wave Simulator

For every cell *i*

$$h_i^{new} \leftarrow h_i + \beta(h_i - h_i^{old})$$

If h_{i-1} exists, then $h_i^{new} \leftarrow h_i^{new} + \alpha(h_{i-1} - h_i)$

If h_{i+1} exists, then $h_i^{new} \leftarrow h_i^{new} + \alpha(h_{i+1} - h_i)$

For every cell *i*

$$h_i^{old} \leftarrow h_i$$

$$h_i \leftarrow h_i^{new}$$

Algorithm with Neumann Boundaries

Extending the simulator to 3D is also straightforward.

A Shallow Wave Simulator

For every cell *i, j*

$$h_{i,j}^{new} \leftarrow h_{i,j} + \beta(h_{i,j} - h_{i,j}^{old})$$

If $h_{i-1,j}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i-1,j} - h_{i,j})$

If $h_{i+1,j}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i+1,j} - h_{i,j})$

If $h_{i,j-1}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i,j-1} - h_{i,j})$

If $h_{i,j+1}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i,j+1} - h_{i,j})$

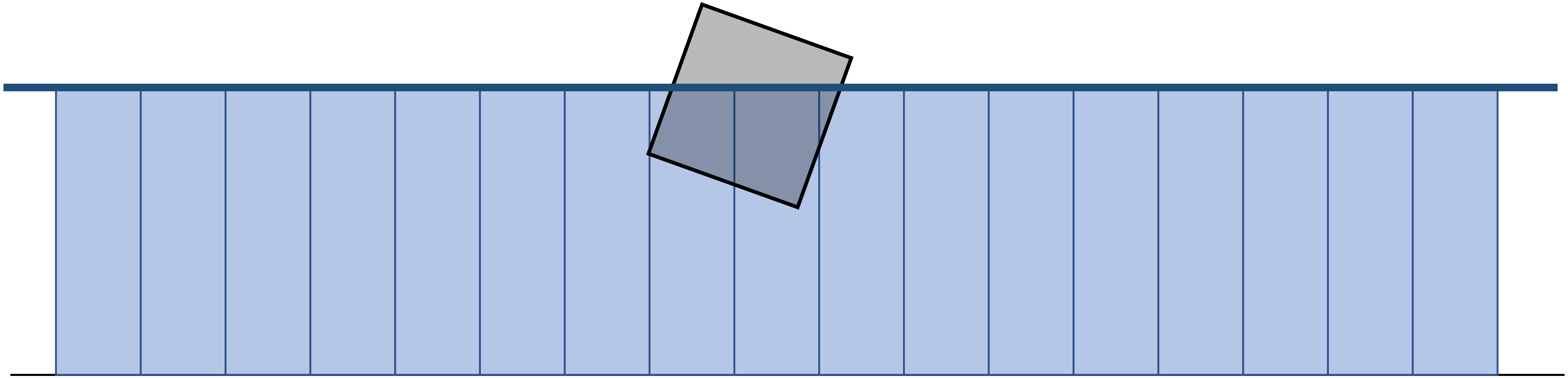
For every cell *i, j*

$$h_{i,j}^{old} \leftarrow h_{i,j}$$

$$h_{i,j} \leftarrow h_{i,j}^{new}$$

Two-Way Coupling

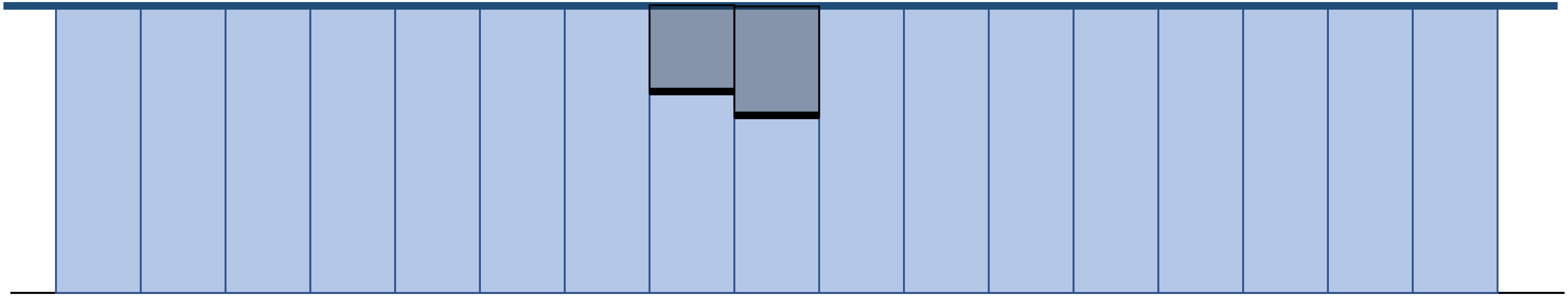
The coupling between a solid and a liquid should be two-way, i.e., liquid->solid and solid->liquid.



Two-Way Coupling

The coupling between solid and water should be two-way, i.e., water>solid and solid->water.

The key question is how to expel water out of the gray cell regions???

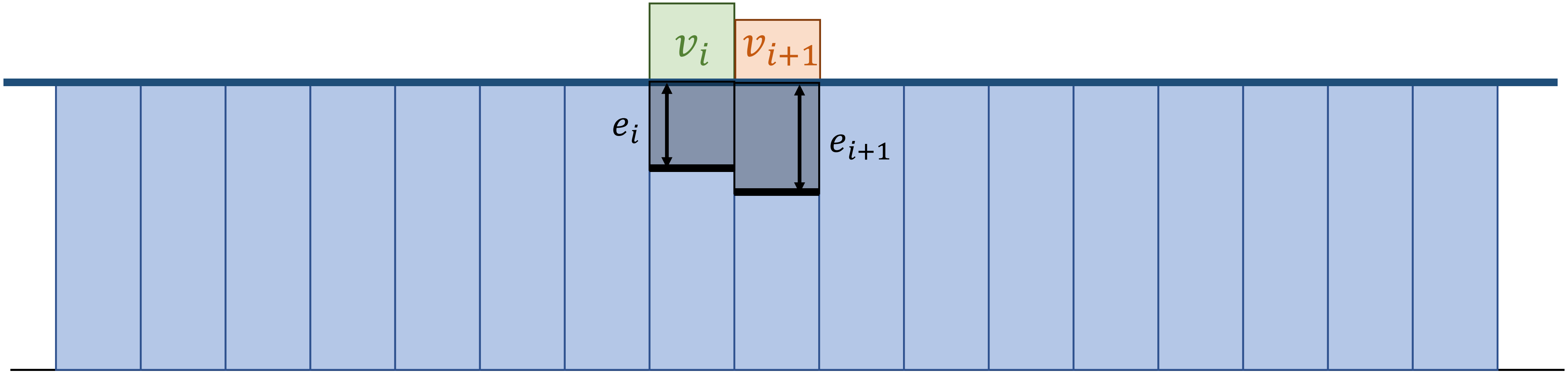


Virtual Height

The idea is to set up a virtual height v_i, so that $h_i^{real\_new} = h_i - e_i$.

$$h_i - e_i = h_i + \beta\left(h_i - h_i^{old}\right) + \alpha(v_{i+1} + h_{i+1} + h_{i-1} - 2v_i - 2h_i) = h_i^{new} + \alpha(v_{i+1} - 2v_i)$$

$$h_{i+1} - e_{i+1} = h_{i+1} + \beta\left(h_{i+1} - h_{i+1}^{old}\right) + \alpha(h_{i+2} + v_i + h_i - 2v_{i+1} - 2h_{i+1}) = h_{i+1}^{new} + \alpha(v_i - 2v_{i+1})$$

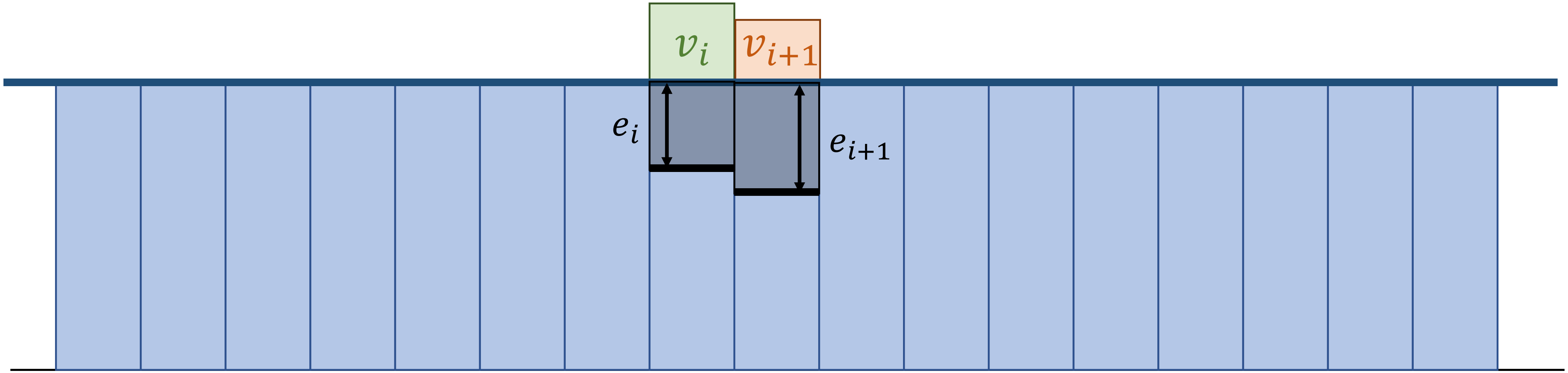


Poisson's Equation

The outcome is Poisson's equation, with v_i and v_{i+1} being unknowns.

$$2v_i - v_{i+1} = \frac{1}{\alpha}(h_i^{new} - h_i + e_i) = b_i$$

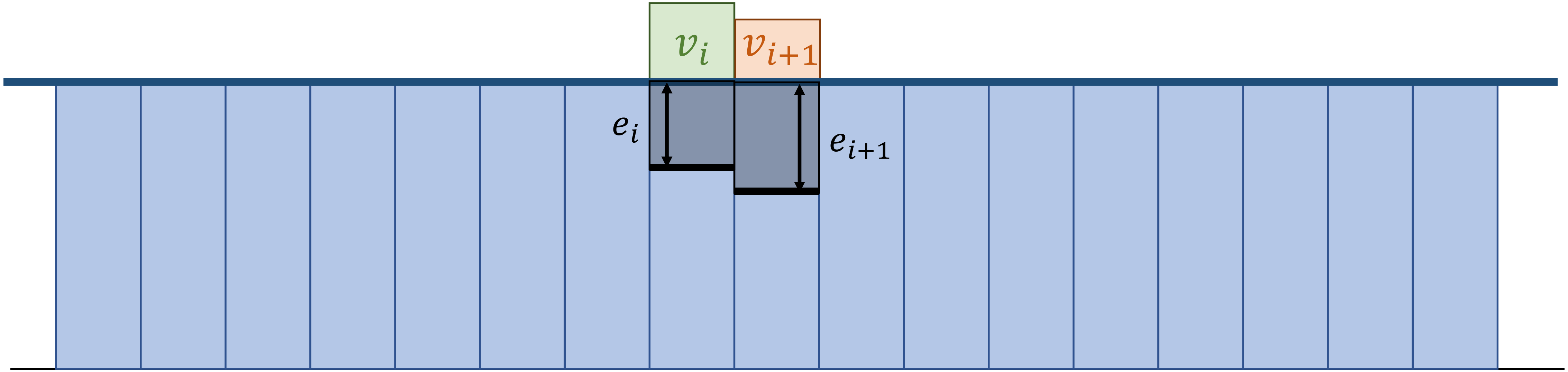
$$-v_i + 2v_{i+1} = \frac{1}{\alpha}(h_{i+1}^{new} - h_{i+1} + e_{i+1}) = b_{i+1}$$



Poisson's Equation

The outcome is Poisson's equation, with v_i and v_{i+1} being unknowns.

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} v_i \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} b_i \\ b_{i+1} \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} v_{i-1} \\ v_i \\ v_{i+1} \\ v_{i+2} \end{bmatrix} = \begin{bmatrix} 0 \\ b_i \\ b_{i+1} \\ 0 \end{bmatrix}$$



Algorithm with Coupling

For every cell *i, j*

if in contact

$b_{i,j} \leftarrow \frac{1}{\alpha}(h_{i,j}^{new} - h_{i,j} + e_{i,j})$

$tag_{i,j} \leftarrow true$

else

$v_{i,j} \leftarrow 0$

$tag_{i,j} \leftarrow false$

$\text{PCG\_Solve}(v,\ b,\ tag)$

γ is a relaxation factor.

A Shallow Wave Simulator

For every cell *i, j*

$h_{i,j}^{new} \leftarrow h_{i,j} + \beta(h_{i,j} - h_{i,j}^{old})$

If $h_{i-1,j}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i-1,j} - h_{i,j})$

If $h_{i+1,j}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i+1,j} - h_{i,j})$

If $h_{i,j-1}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i,j-1} - h_{i,j})$

If $h_{i,j+1}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha(h_{i,j+1} - h_{i,j})$

Get *v*

For every cell *i, j*

If $h_{i-1,j}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha\gamma(v_{i-1,j} - v_{i,j})$

If $h_{i+1,j}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha\gamma(v_{i+1,j} - v_{i,j})$

If $h_{i,j-1}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha\gamma(v_{i,j-1} - v_{i,j})$

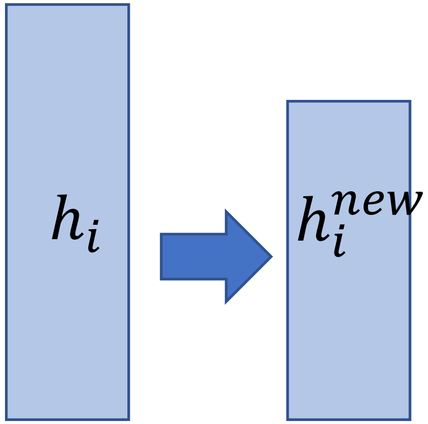
If $h_{i,j+1}$ exists, then $h_{i,j}^{new} \leftarrow h_{i,j}^{new} + \alpha\gamma(v_{i,j+1} - v_{i,j})$

$h_{i,j}^{old} \leftarrow h_{i,j}$

$h_{i,j} \leftarrow h_{i,j}^{new}$

Rigid Body Update

We estimate the floating force by the actual water expelled in every column.



$$f_i = \rho g \Delta x (h_i - h_i^{new})$$

Or in 3D,

$$f_{i,j} = \rho g \Delta A (h_{i,j} - h_{i,j}^{new})$$

A Summary For the Day

- The shallow wave model simulates waves over a height field.
- It's based on a lot of simplification. We will discuss what fluid dynamics really looks like without simplification.
- The strength of the shallow wave model is its simplicity and efficiency. It can easily simulate water-solid coupling too.
- See Lab 4 for more details.