WebFlood

Interactive Shallow-Water Simulations in City Environments

Anselm Eickhoff

Supervisor:
Prof. Nils Thuerey

Structure

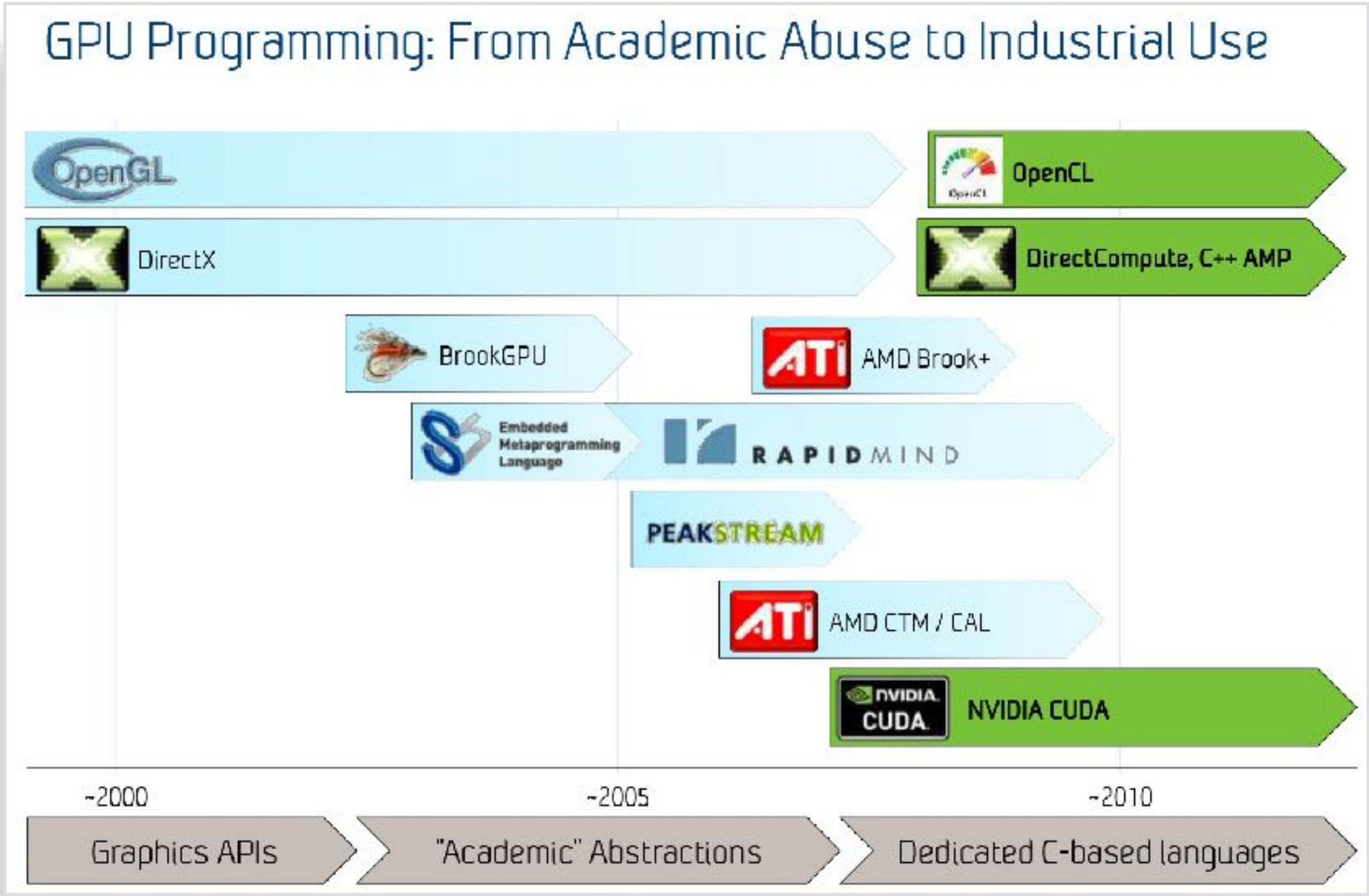
- GPGPU with WebGL
- Related Work & Models
- Shallow-Water Implementation
- Results
  - 2008 Flooding of Iowa
  - Classical Dam-Break Experiment
- Conclusion

Makeshift GPGPU

GPGPU in a browser

- WebGL widely supported
- Functionality level: OpenGL ES 2.0
- WebCL exists, but highly experimental

“How did people survive before CUDA/OpenCL?”



From: André R. Brodtkorb
"Compact Stencils for the Shallow Water Equations
on Graphics Processing Units"

computation problem

graphics API

Basic Technique

Framebuffer Object (FBO)

Texture
can be read from

+

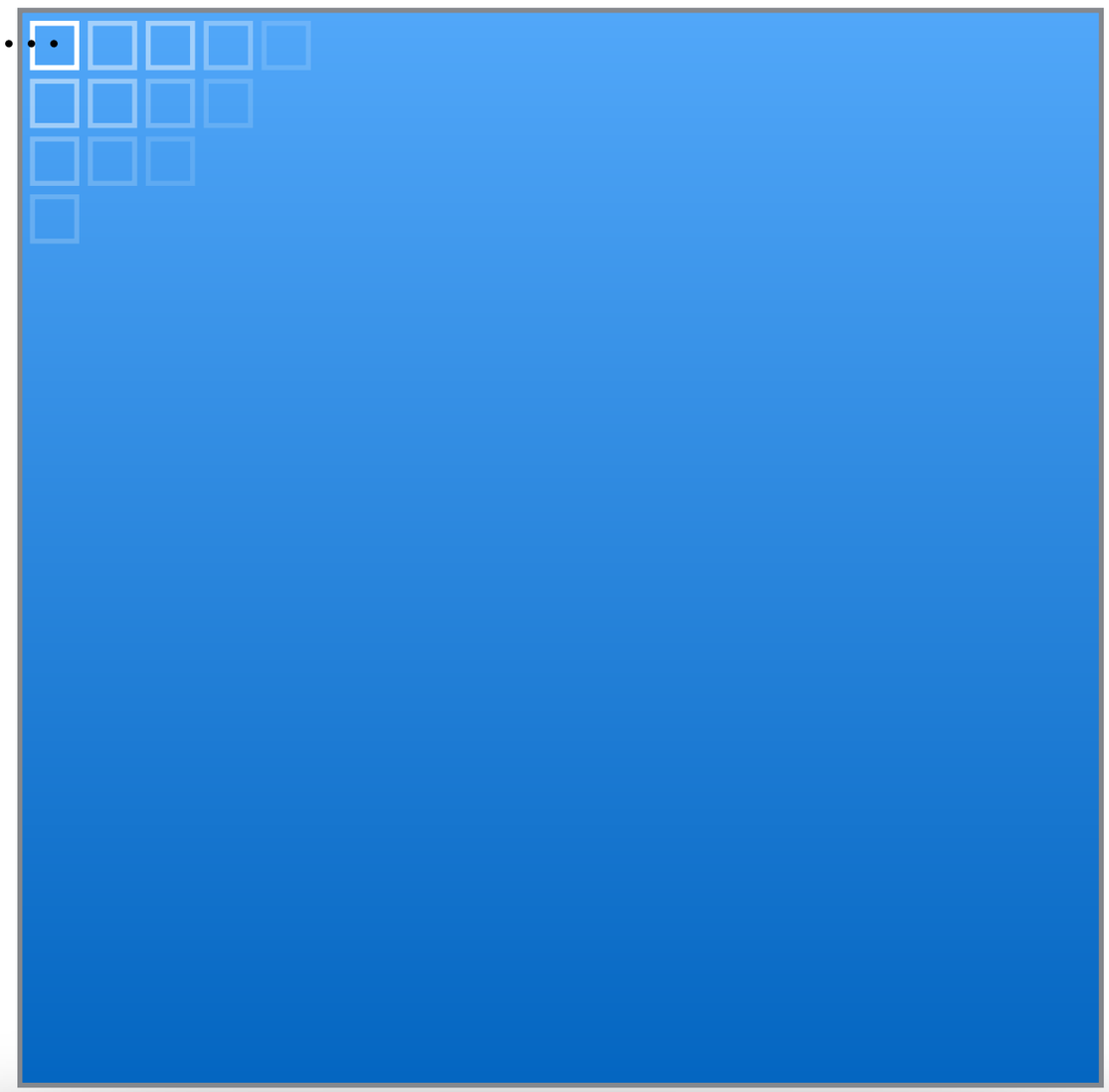
Render Target
can be written to

Basic Technique

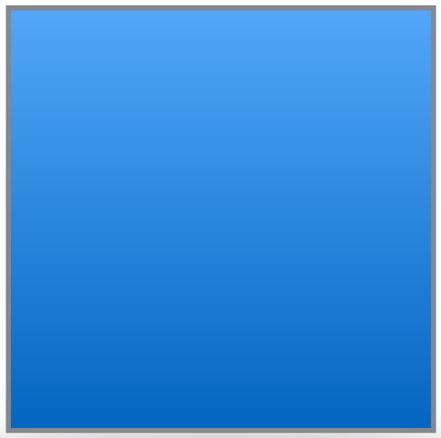
red
green
blue
alpha

4 data values
per pixel
(uInt or Float)

Encode simulation data in pixels

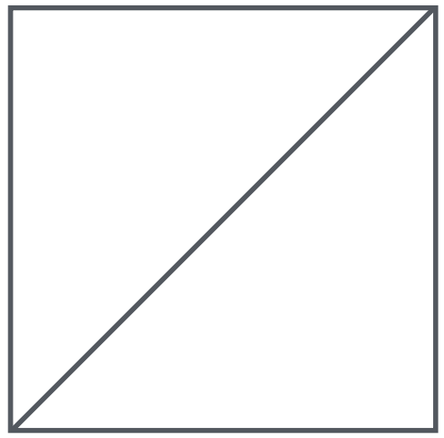


Basic Technique



FBO 1

initial
simulation state



Flat Geometry

fills entire screen

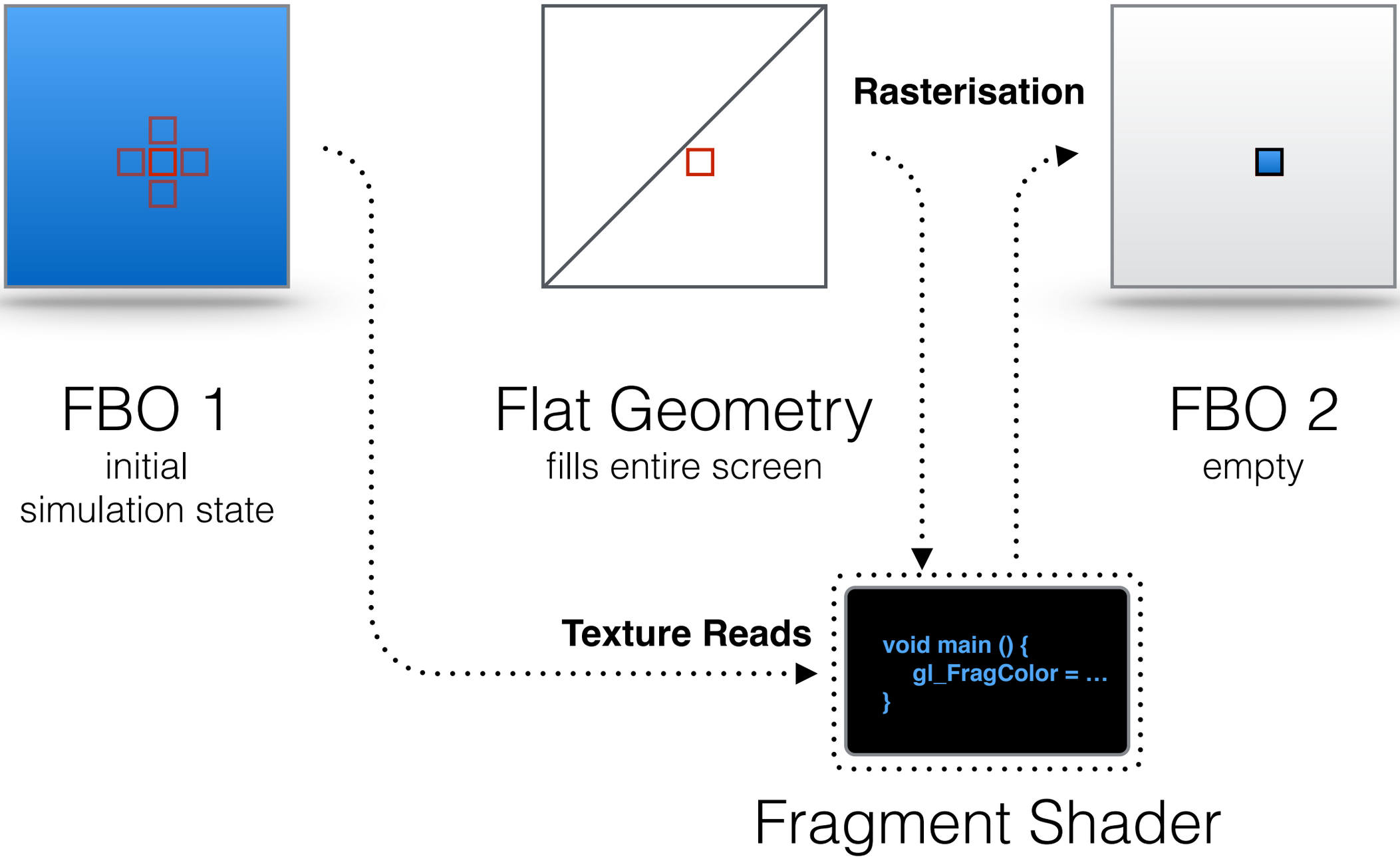
FBO 2

empty

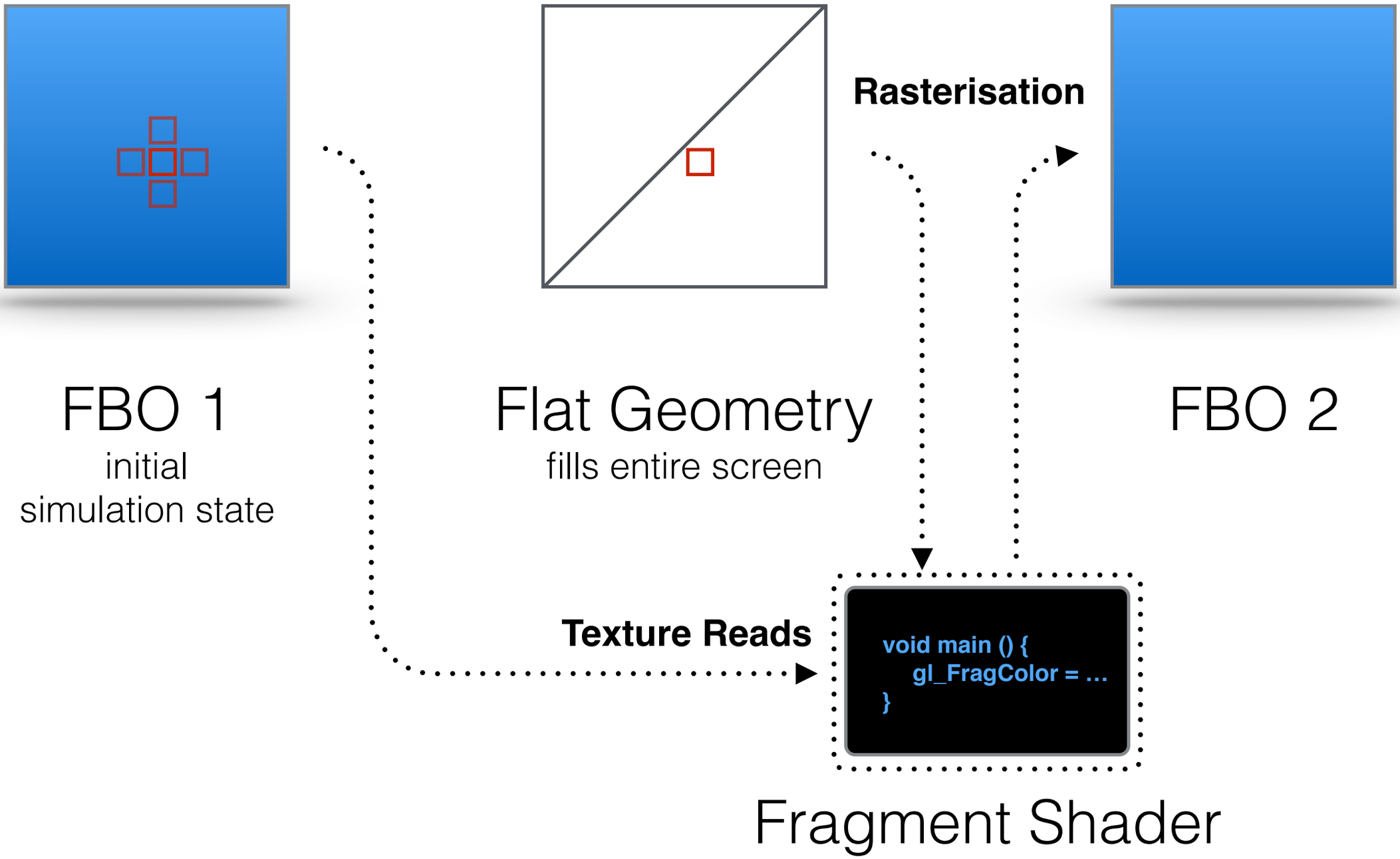
```
void main () {
    gl_FragColor = …
}
```

Fragment Shader

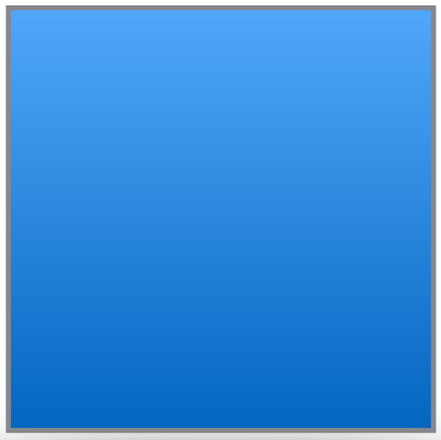
Basic Technique



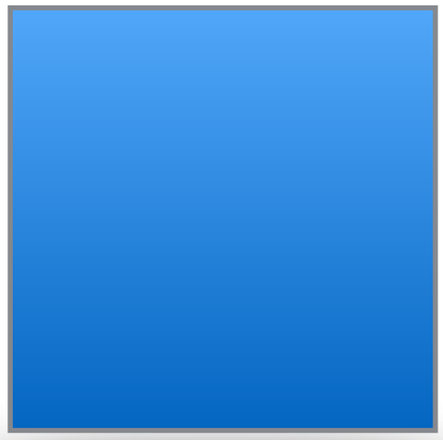
Basic Technique



Basic Technique

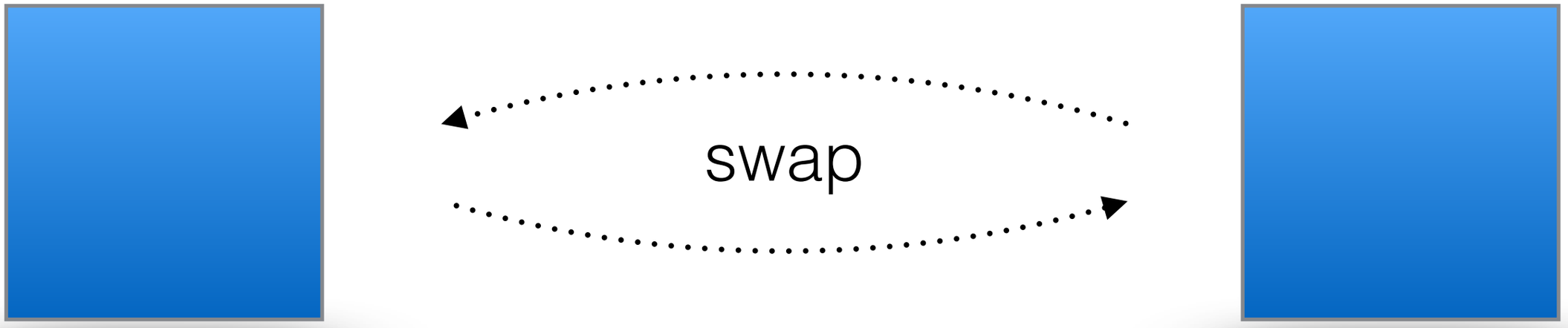


FBO 1
initial
simulation state



FBO 2
next
simulation state

Basic Technique



FBO 1
initial
simulation state

FBO 2
next
simulation state

Related Work

Computational Fluid Dynamics

$$\nabla \cdot \vec{u} = 0$$

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla p = \vec{g}$$

Incompressible & Inviscid
Navier-Stokes Equations

$$\nabla \cdot \vec{u} = 0$$

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla p = \boxed{\vec{g}}$$

1. Apply External Forces

$$\nabla \cdot \vec{u} = 0$$

$$\boxed{\frac{D\vec{u}}{Dt}} + \frac{1}{\rho}\nabla p = \vec{g}$$

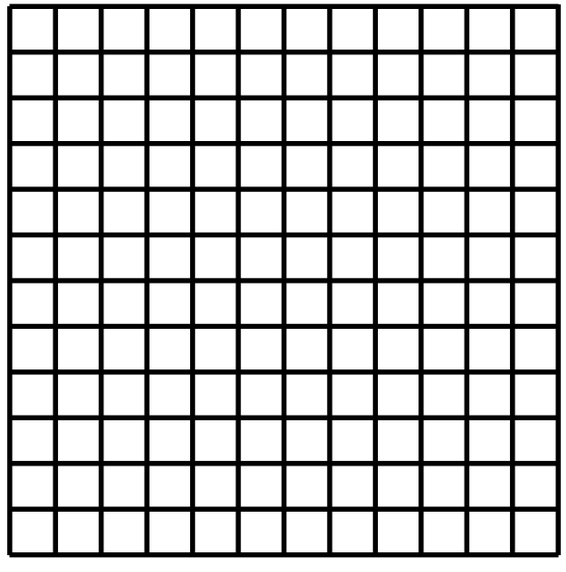
2. Advection

(move quantities along velocity field)

$$\nabla \cdot \vec{u} = 0$$

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla p = \vec{g}$$

3. Solve for Incompressibility



Grid-Based Methods

- “Eulerian Viewpoint”
- Changes of quantities at grid points are observed
- Fluid = Field

- NSE discretised, solved using finite differences
- Advection inexact and unstable
- Incompressibility requires solution to Poisson equation

THE PHYSICS OF FLUIDS VOLUME 8, NUMBER 12 DECEMBER 1965

Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface

FRANCIS H. HARLOW AND J. EDDIE WELCH
Los Alamos Scientific Laboratory, University of California, Los Alamos, New Mexico
(Received 16 April 1965; final manuscript received 3 September 1965)

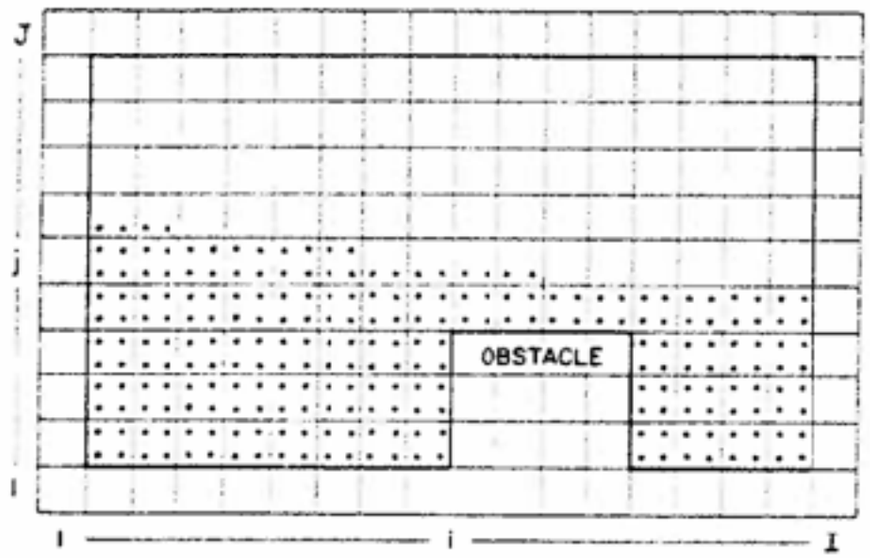


FIG. 1. Sketch of typical mesh and marker-particle layout. An actual calculation would be much more finely resolved.

Marker Particles

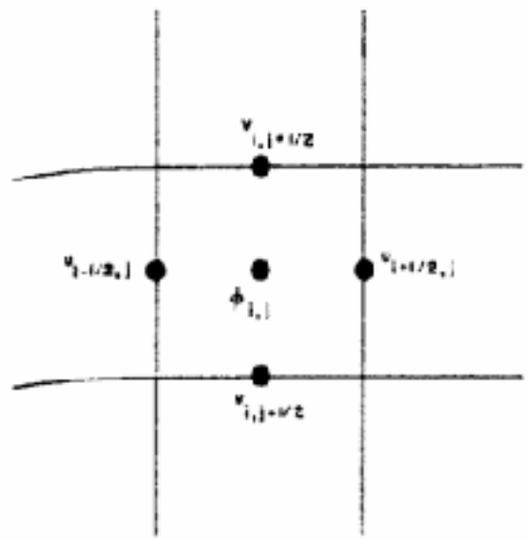
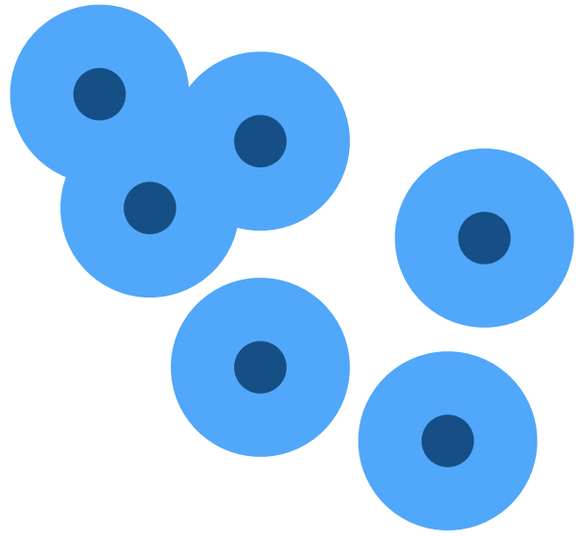


FIG. 2. Field variable value placement about a computational cell. Velocities are defined at cell boundaries while pressures are defined at cell centers.

MAC-Grid



Particle-Based Methods

- “Lagrangian Viewpoint”
- Particles are followed along their motion through the fluid
- Fluid = Collection of Particles
- Advection obvious and exact
- How to define fluid, ∇ and ∇^2 from particle properties?
- Incompressibility problematic

Smoothed particle hydrodynamics: theory and application to non-spherical stars

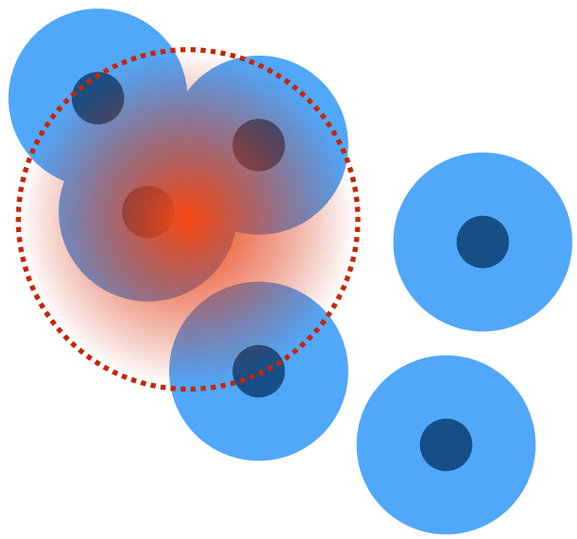
R. A. Gingold and J. J. Monaghan* *Institute of Astronomy, Madingley Road. Cambridge, CB3 0HA*

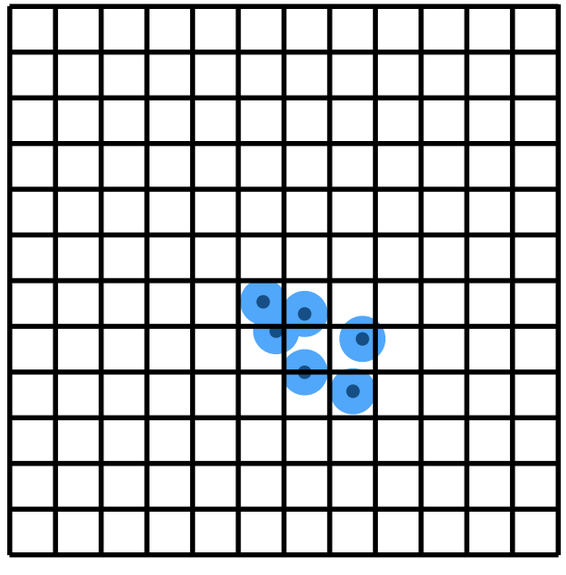
Received 1977 May 5, in original form February 2

$$A(x) = \sum_i m_i \frac{A_i}{\rho_i} W(x - x_i, h)$$

$$\nabla A(x) = \sum_i m_i \frac{A_i}{\rho_i} \nabla W(x - x_i, h)$$

$$\nabla^2 A(x) = \sum_i m_i \frac{A_i}{\rho_i} \nabla^2 W(x - x_i, h)$$





Hybrid Methods

Stable Fluids Jos Stam, 1999

- Lagrangian advection, otherwise grid-based
- Unconditionally stable

Lattice Boltzmann Method

- Average microscopic kinetics = macroscopic behaviour
- Converges to Navier-Stokes

Specialisations

Realtime Fluid Animation

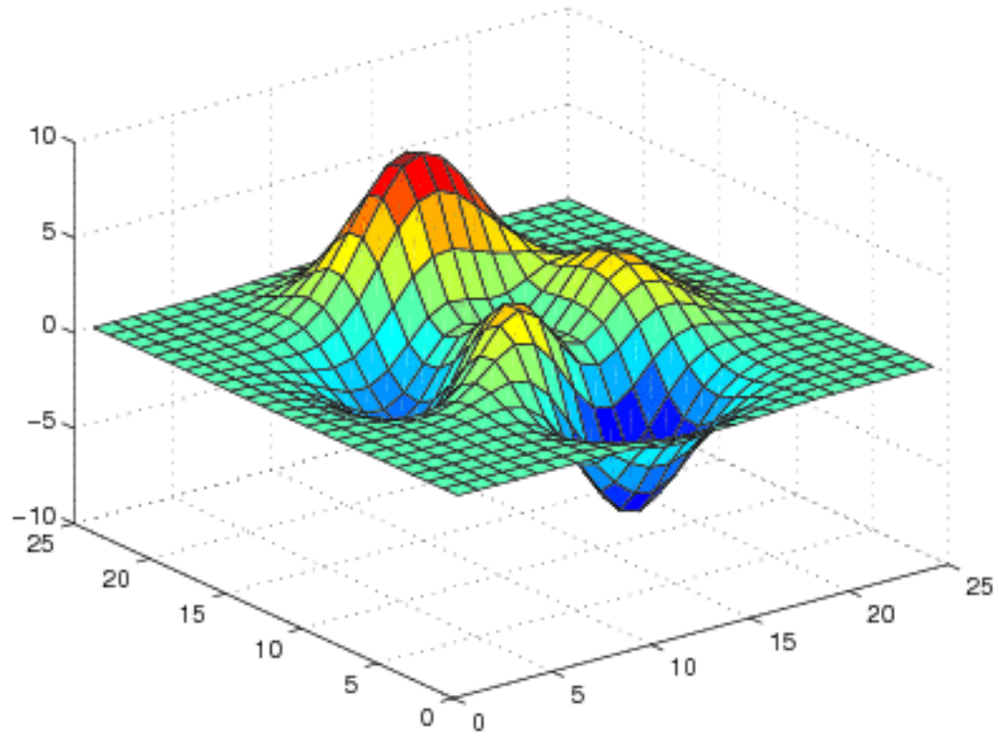
- Performance and Plausibility
- All Methods: Parallelisable!
- Realtime 3D: Achievable
- 2D Shallow-Water
  - less expensive
  - maps even better to GPU

Flood Simulation

- Correctness
- Offline Simulation: Acceptable
- Vast Simulation Domains
  - Optimise for topography with irregular meshes
  - 2D Shallow-Water

Shallow-Water

- Simplification of 3D Navier-Stokes
- Water Columns Hydrostatic Pressure
  - no vertical velocities
  - horizontal velocities equal over the whole column
- Water Height = Pressure
  - 2D Height Field Representation



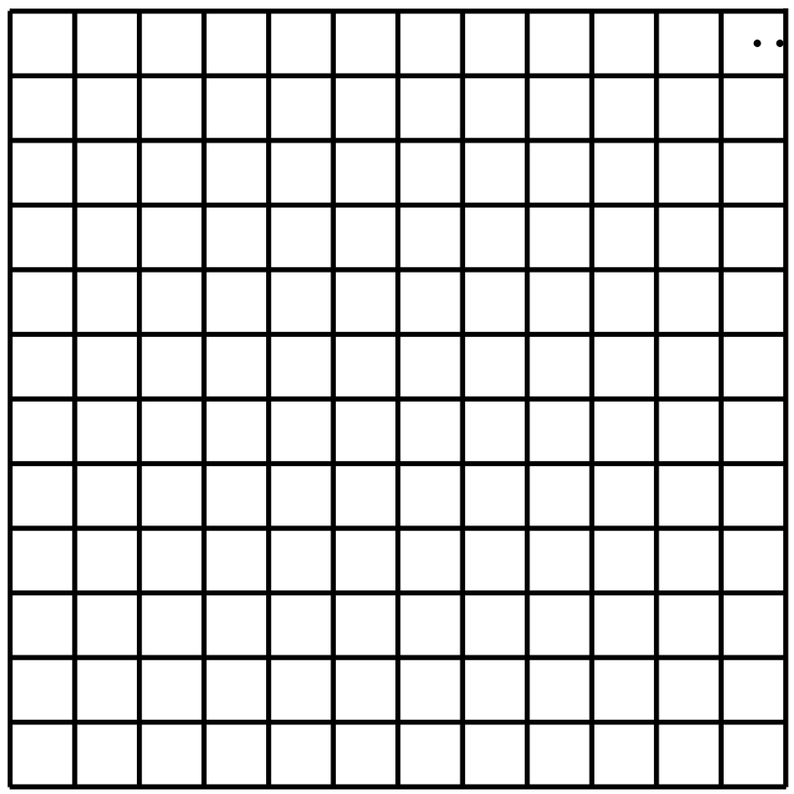
$$\frac{\partial \eta}{\partial t} + \vec{v} \cdot \nabla \eta = -\eta \nabla \cdot \vec{v}$$

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = g \nabla h$$

Grid-Based Shallow-Water

Makeshift GPGPU Implementation

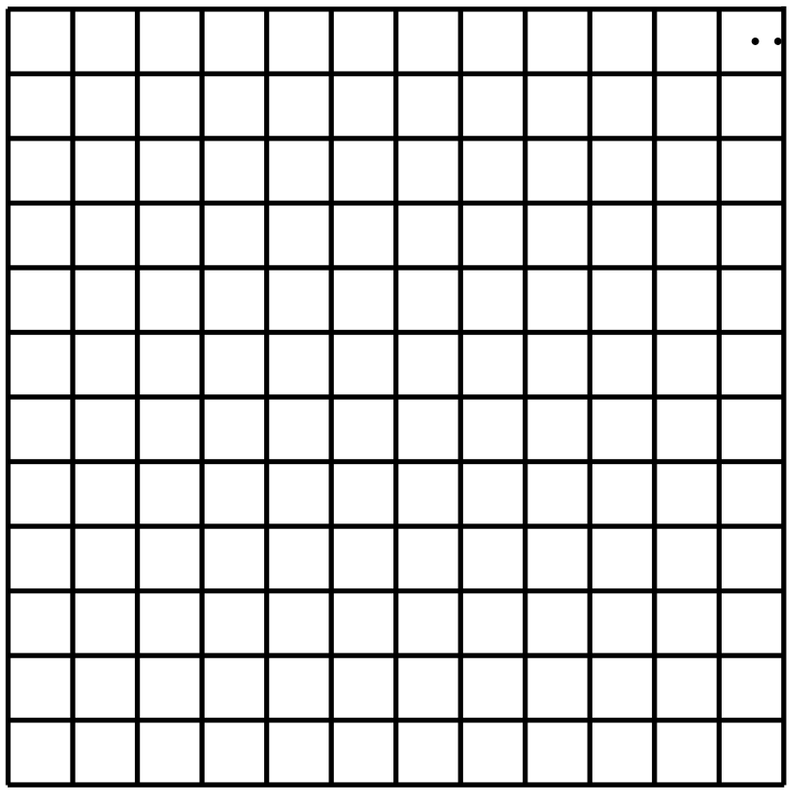
Data Representation



water velocity
water height
terrain height

Simulation Grid

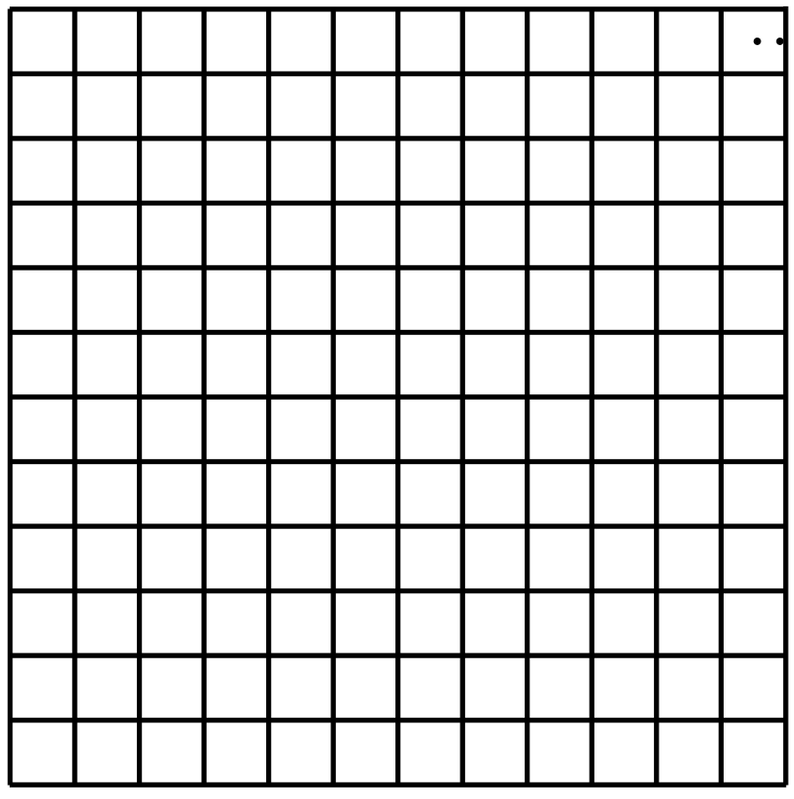
Data Representation



Simulation Grid

water velocity (x)
water velocity (y)
water height
terrain height

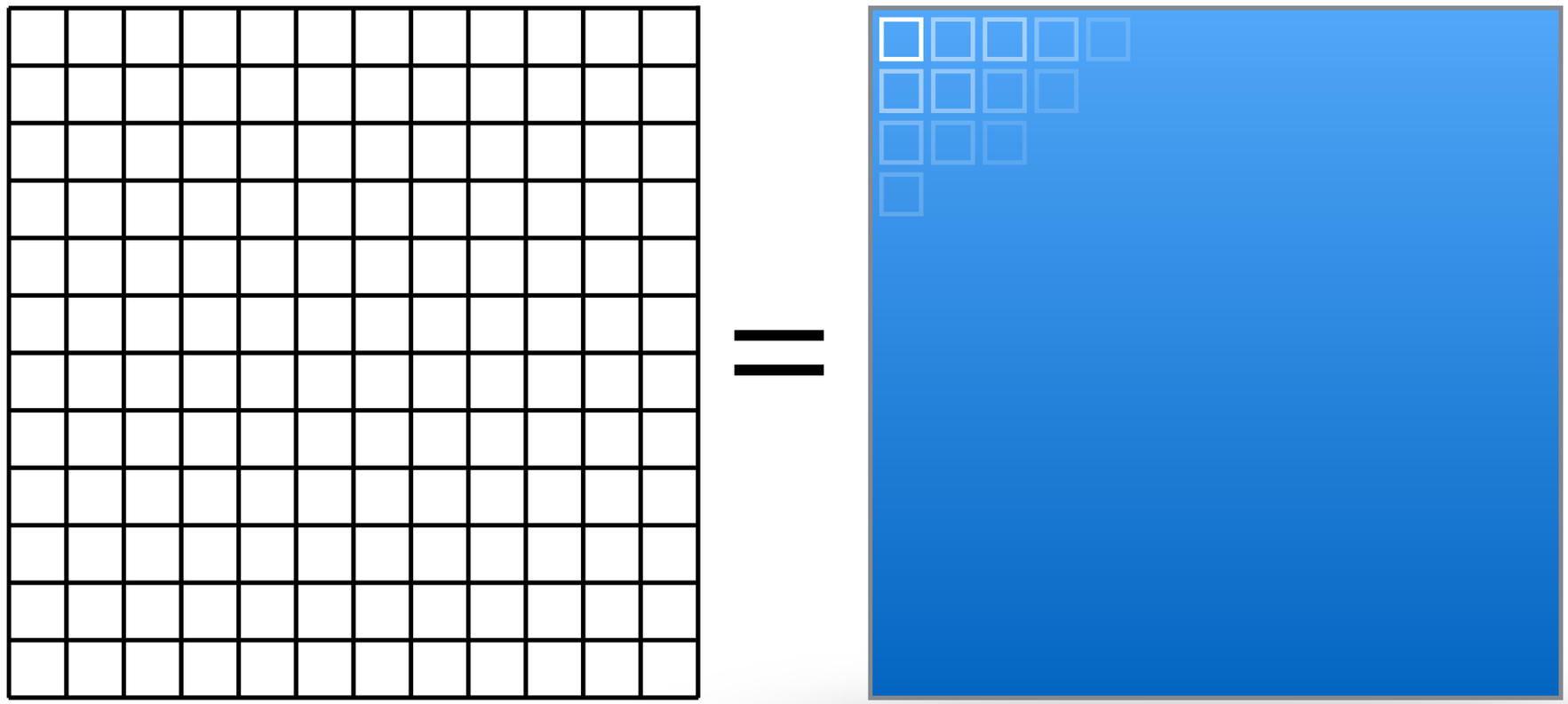
Data Representation



Simulation Grid

water velocity (x) **red**
water velocity (y) **green**
water height **blue**
terrain height **alpha**

Data Representation



Simulation Grid

FBO

Simulation Steps

Advection

Height Update

Velocity Update

Advection

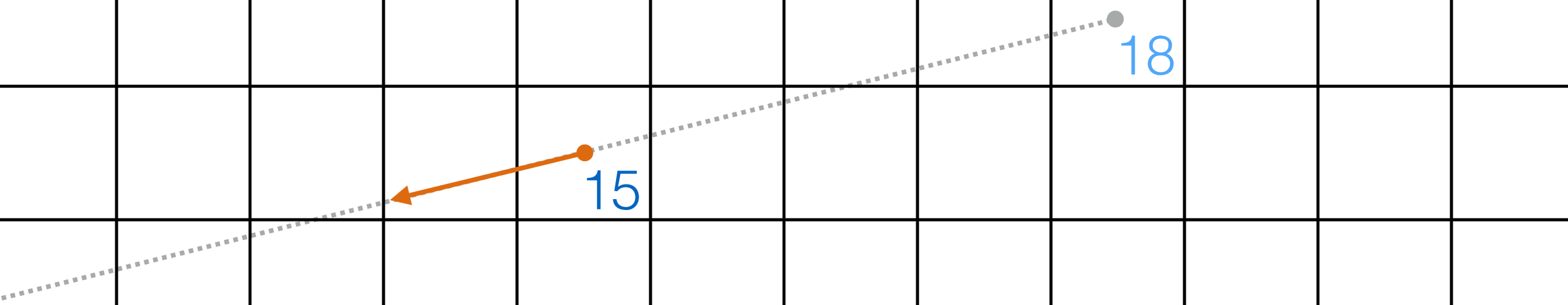
Move quantities along velocity field

$$\frac{\partial \eta}{\partial t} + \vec{v} \cdot \nabla \eta = -\eta \nabla \cdot \vec{v}$$
$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = g \nabla h$$

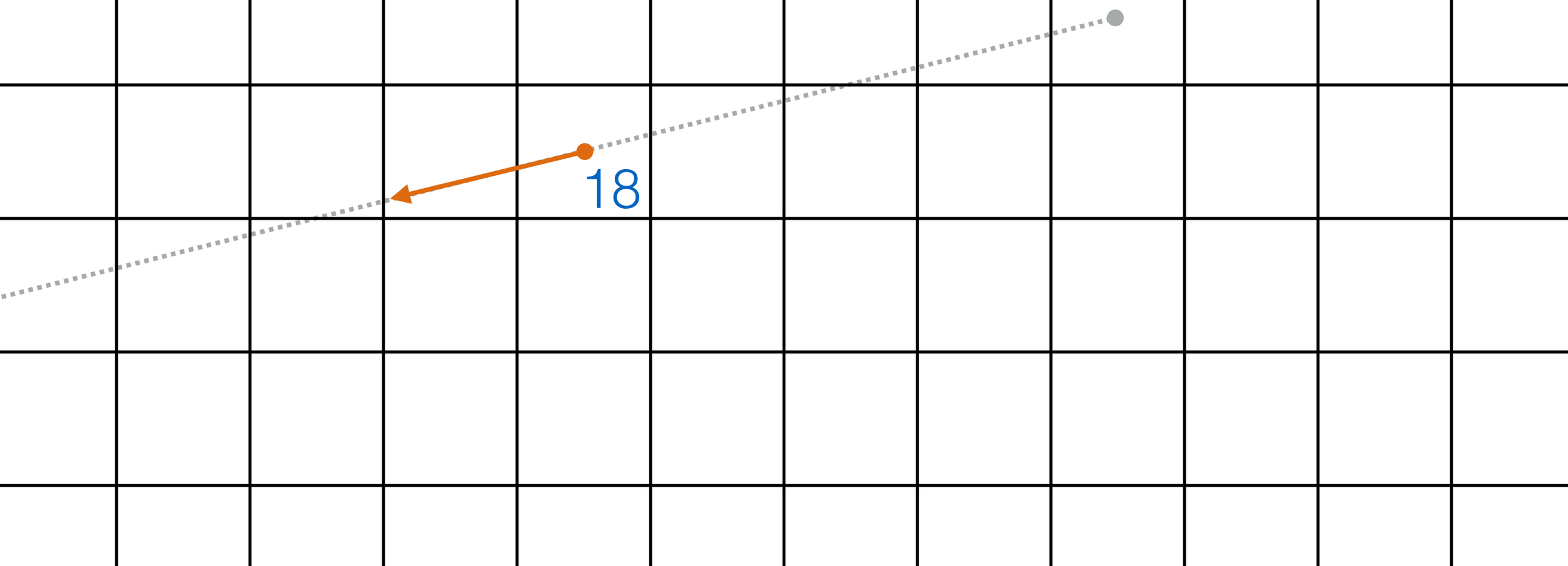
Semi-Lagrangian Advection

Trace a particle backwards in time
Copy value from origin

Semi-Lagrangian Advection

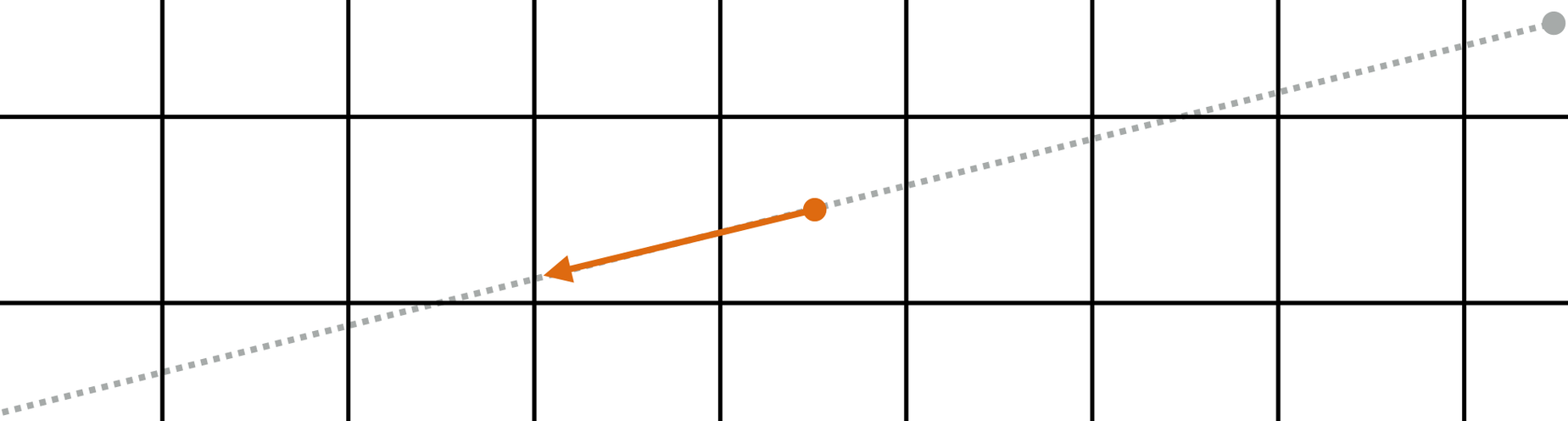


Semi-Lagrangian Advection

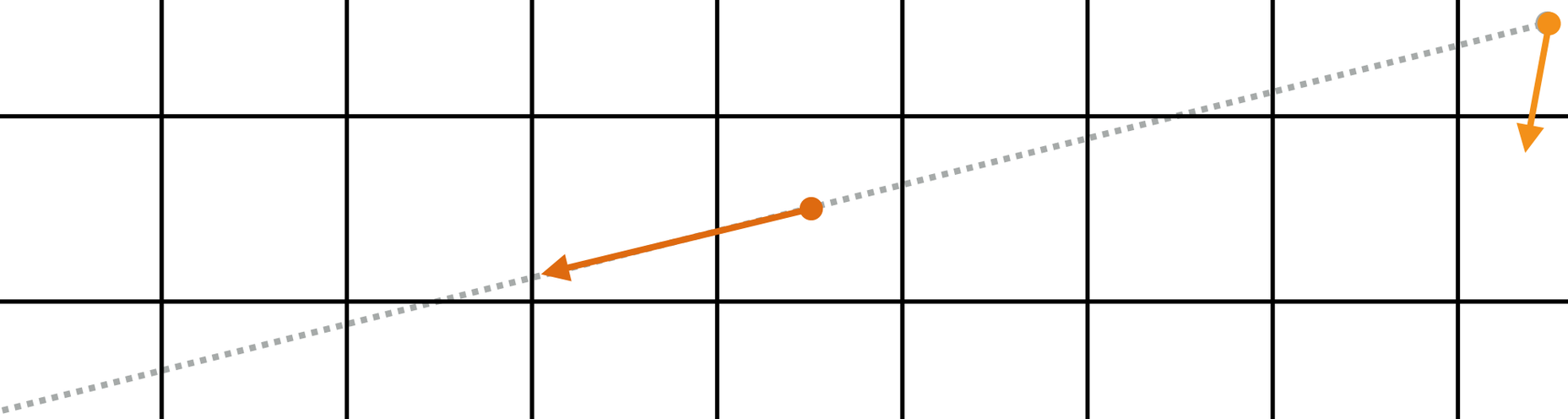


Semi-Lagrangian Advection

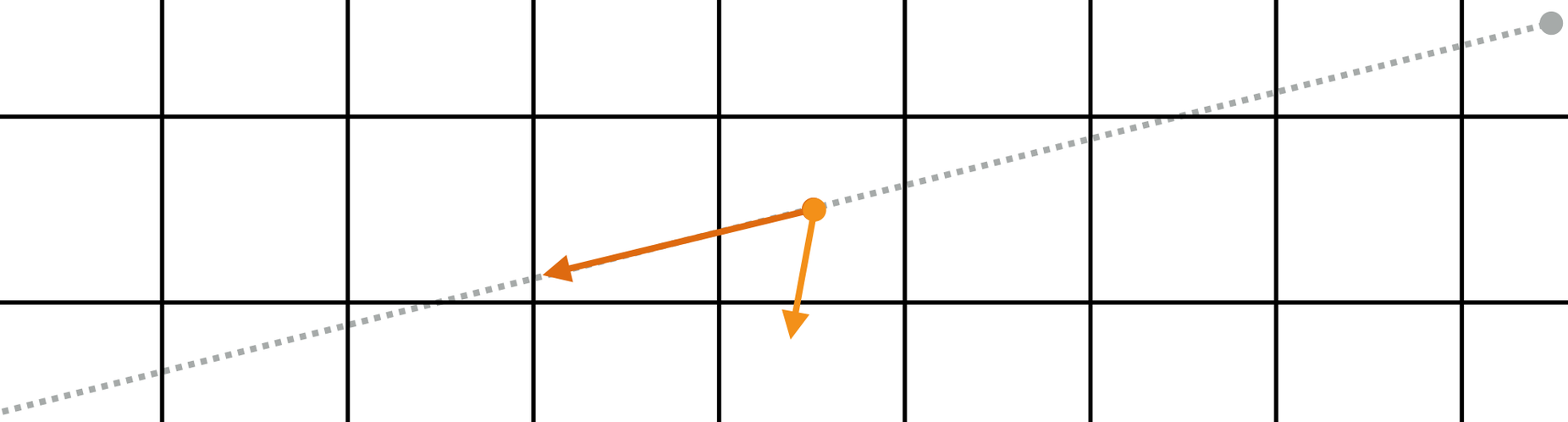
Semi-Lagrangian Advection



Semi-Lagrangian Advection



Semi-Lagrangian Advection



Semi-Lagrangian Advection

Semi-Lagrangian Advection

```
vec4 simulationStep() {
    vec4 here = simData(pos);

    vec4 origin = simData(pos - dt * V(here));
    float newHeight = H(origin);
    vec2 newVelocity = V(origin);

    return vec4(newVelocity, newHeight, T(here));
}
```

Height & Velocity Update

Advection only models movement, but not pressure

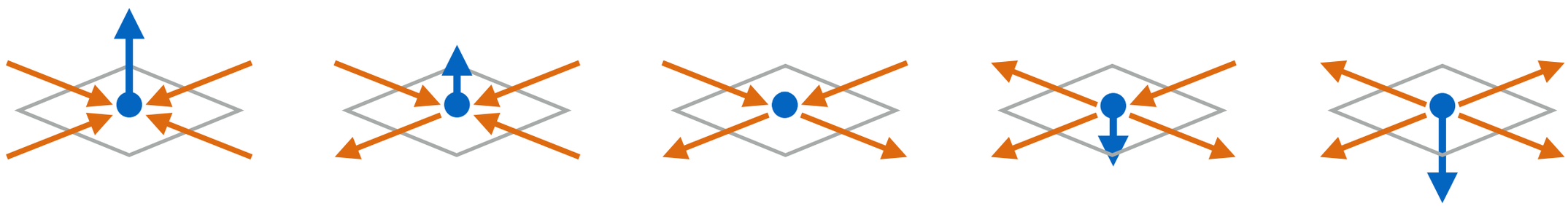
$$\frac{\partial \eta}{\partial t} + \vec{v} \cdot \nabla \eta = -\eta \nabla \cdot \vec{v}$$

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = g \nabla h$$

Height Update

$$\frac{\partial \eta}{\partial t} + \vec{v} \cdot \nabla \eta = -\eta \nabla \cdot \vec{v}$$

- In/Decrease water height based on **divergence** of velocity field
- Divergence: finite differences of ±1 cell, x and y separately



Inflow

$\nabla \cdot \vec{v} < 0$

Equilibrium

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

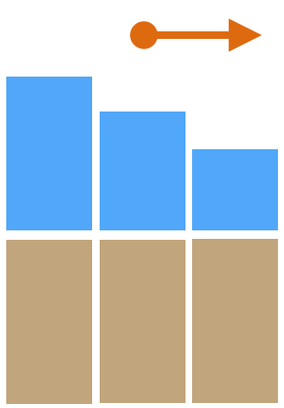
Outflow

$\nabla \cdot \vec{v} > 0$

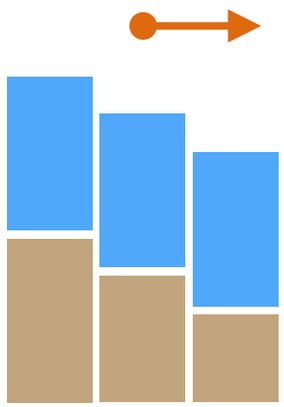
Velocity Update

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = \boxed{g \nabla h}$$

- Accelerate water velocity based on **slope** of water + terrain height
- Slope: finite differences of ±1 cell



Water Slope



Terrain Slope

Modifications

Modifications
For Friction

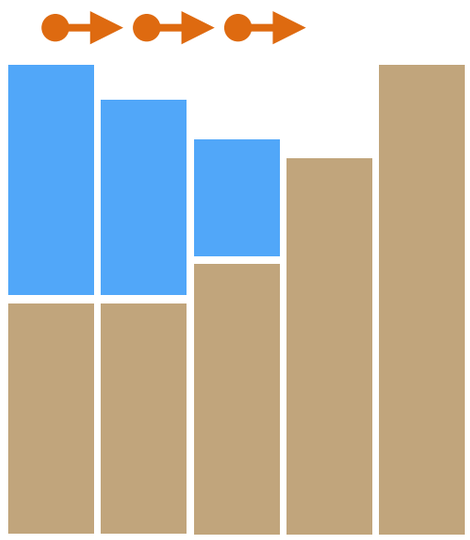
- “Friction Slope” that counteracts accelerating slope of water
- Depends on Water Velocity & Height, Local Manning Coefficient *n*

$$\vec{S}_f = n^2 \frac{\vec{v} \cdot |\vec{v}|}{h^{\frac{4}{3}}}$$

Friction Slope according to Manning’s Law

Modifications
For Wetting/Drying

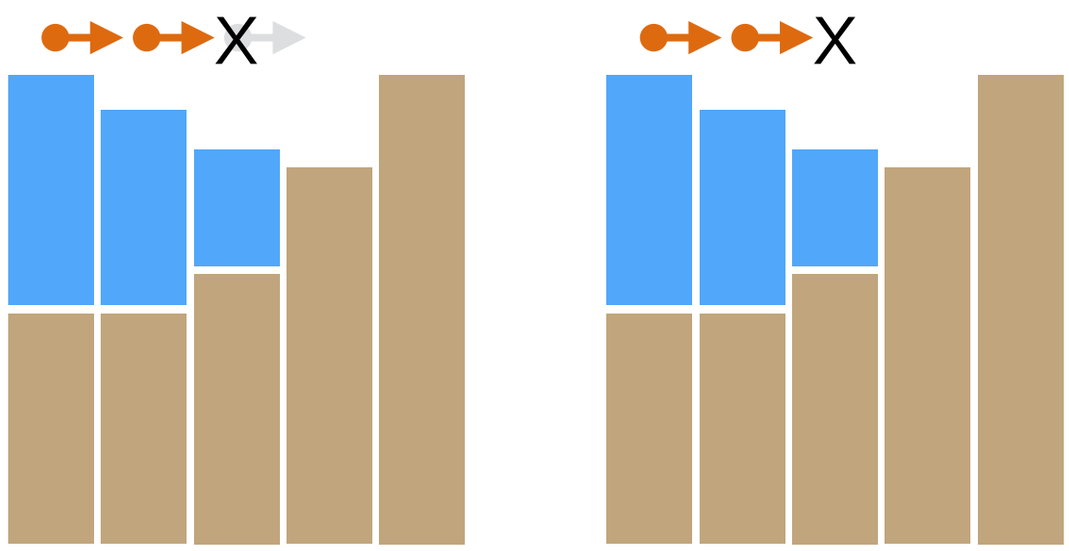
- Special boundary conditions at dry/wet front
- Dry/wet front needs to propagate realistically



wave towards shore

Modifications
For Wetting/Drying

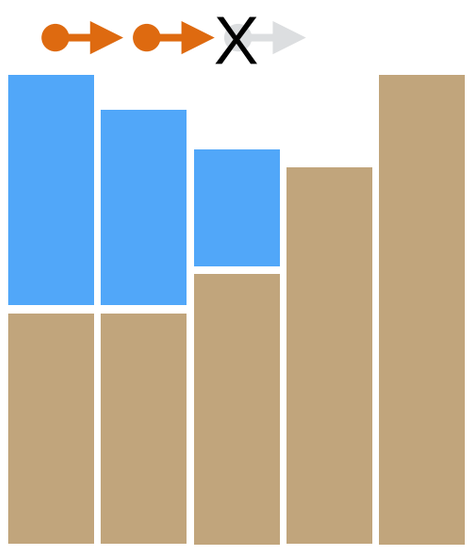
- Special boundary conditions at dry/wet front
- Dry/wet front needs to propagate realistically



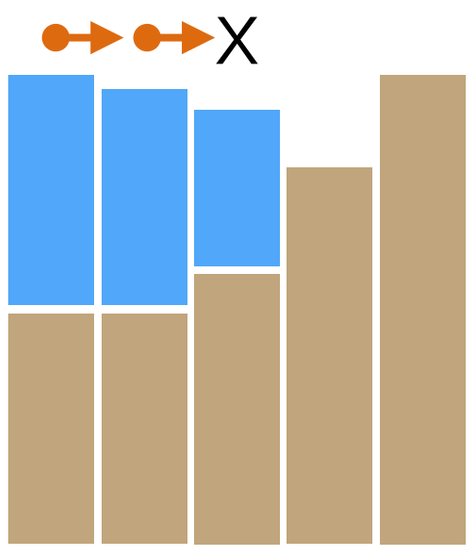
set velocity to 0

Modifications
For Wetting/Drying

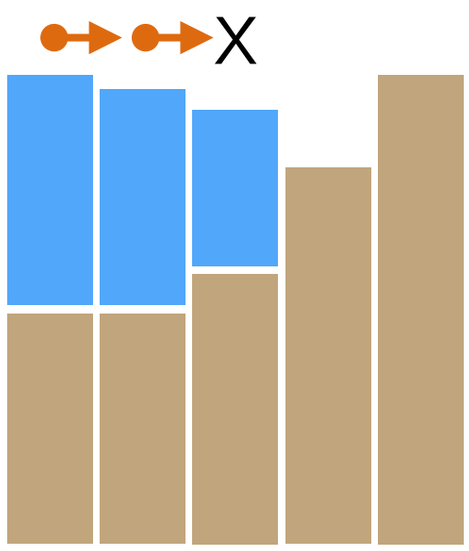
- Special boundary conditions at dry/wet front
- Dry/wet front needs to propagate realistically



set velocity to 0

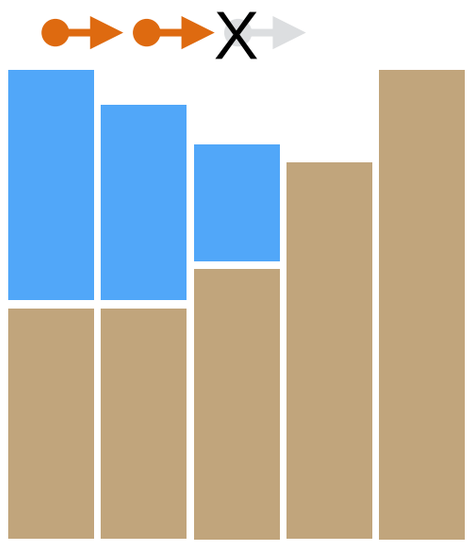


water builds up
because of velocity convergence

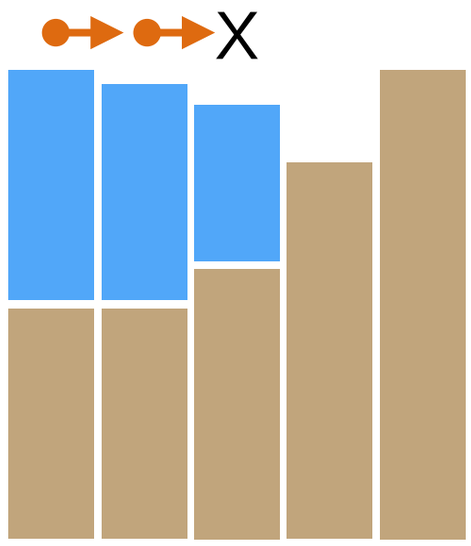


Modifications
For Wetting/Drying

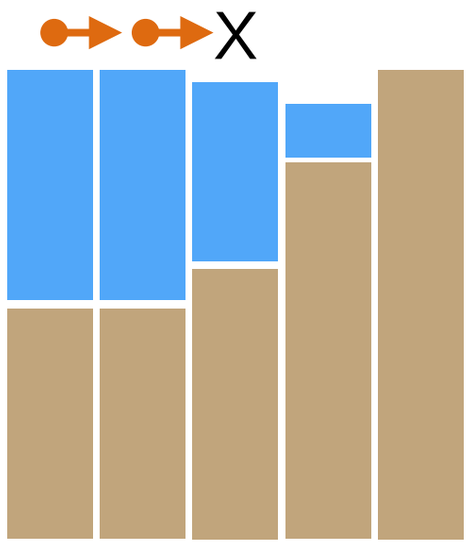
- Special boundary conditions at dry/wet front
- Dry/wet front needs to propagate realistically



set velocity to 0



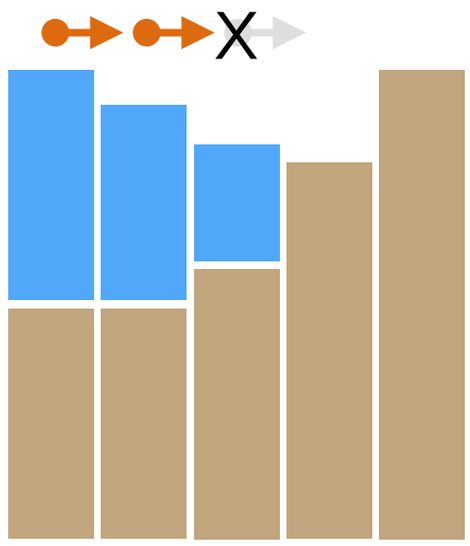
water builds up
because of velocity convergence



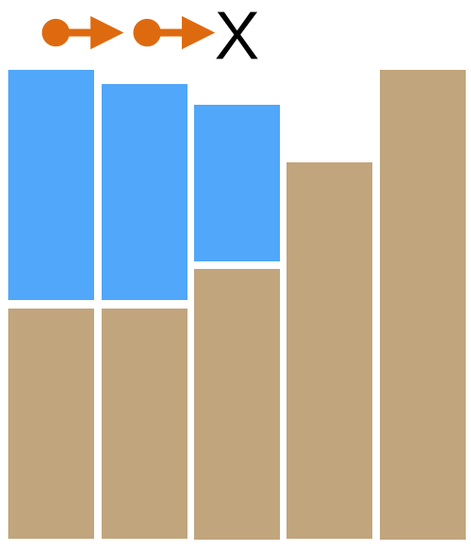
threshold reached
neighbouring cell is wetted

Modifications
For Wetting/Drying

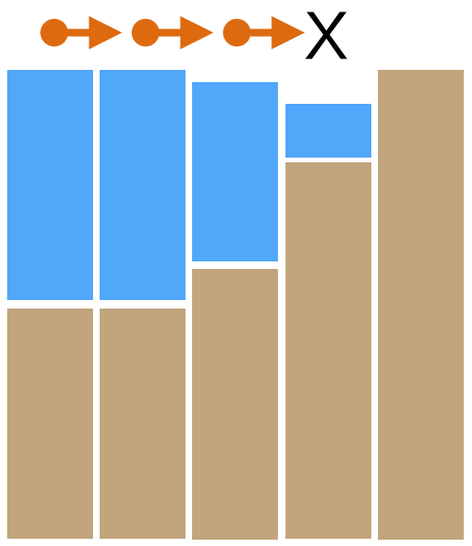
- Special boundary conditions at dry/wet front
- Dry/wet front needs to propagate realistically



set velocity to 0



water builds up
because of velocity convergence



threshold reached
neighbouring cell is wetted

Application & Results

2008 Flooding of Iowa

Image source: https://iowachaser.files.wordpress.com

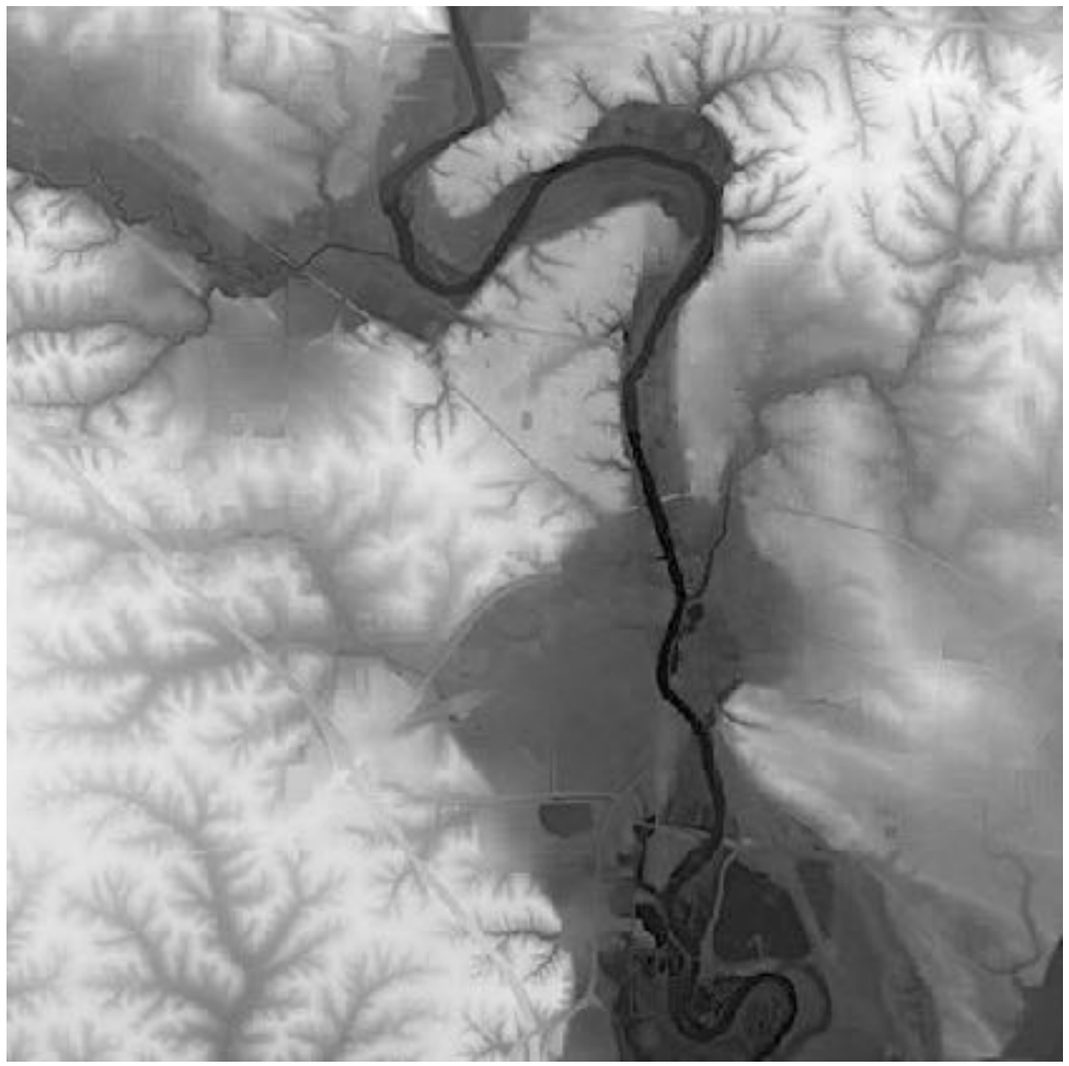
Iowa City

River through Urban Core
Interesting Topography

Image source: http://www.ryersonaircraft.com/

Good Data Availability

(in response to flooding)



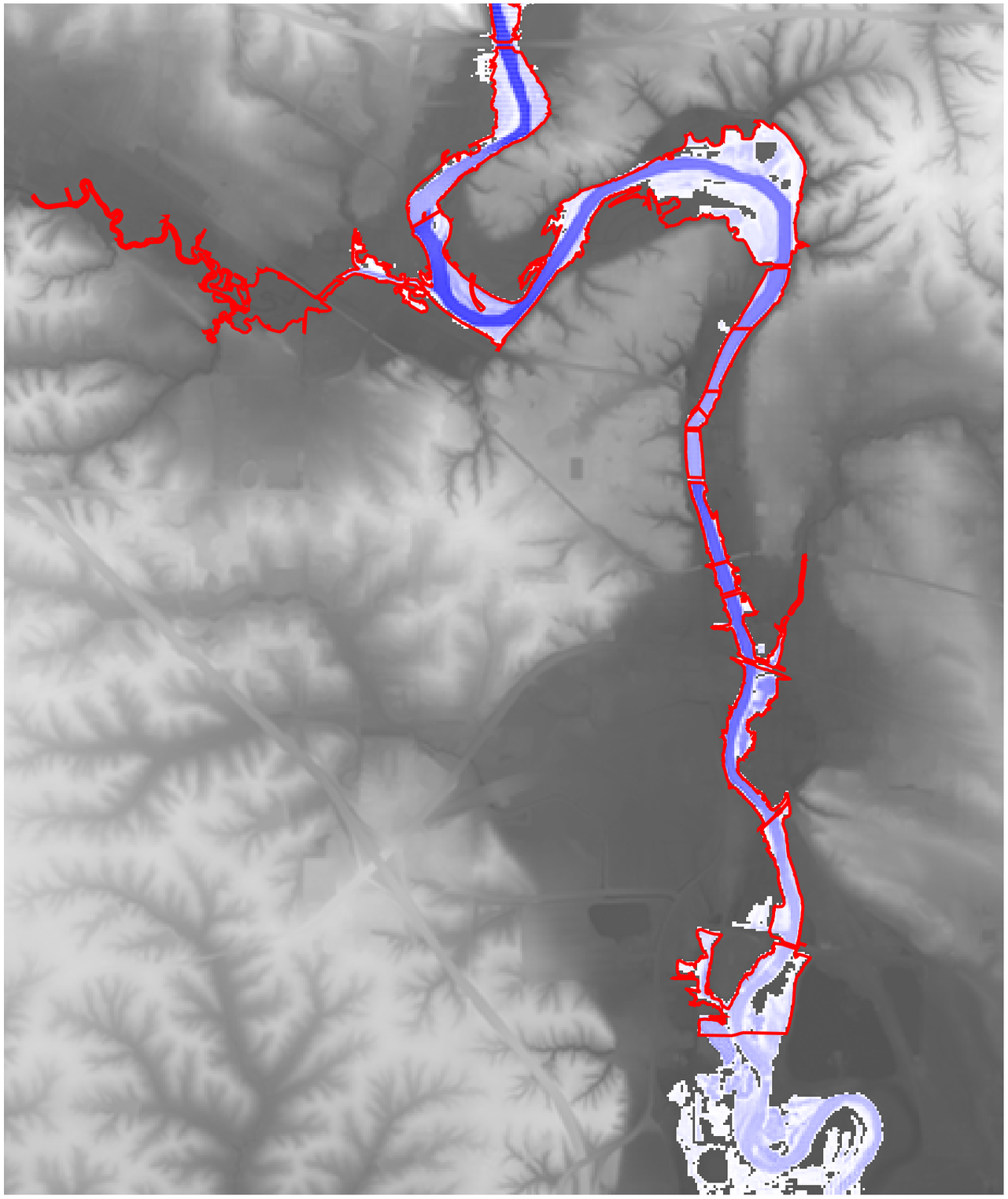
- LiDAR digital elevation maps (0.5m^2 resolution)

- Inundation maps for many real and simulated scenarios

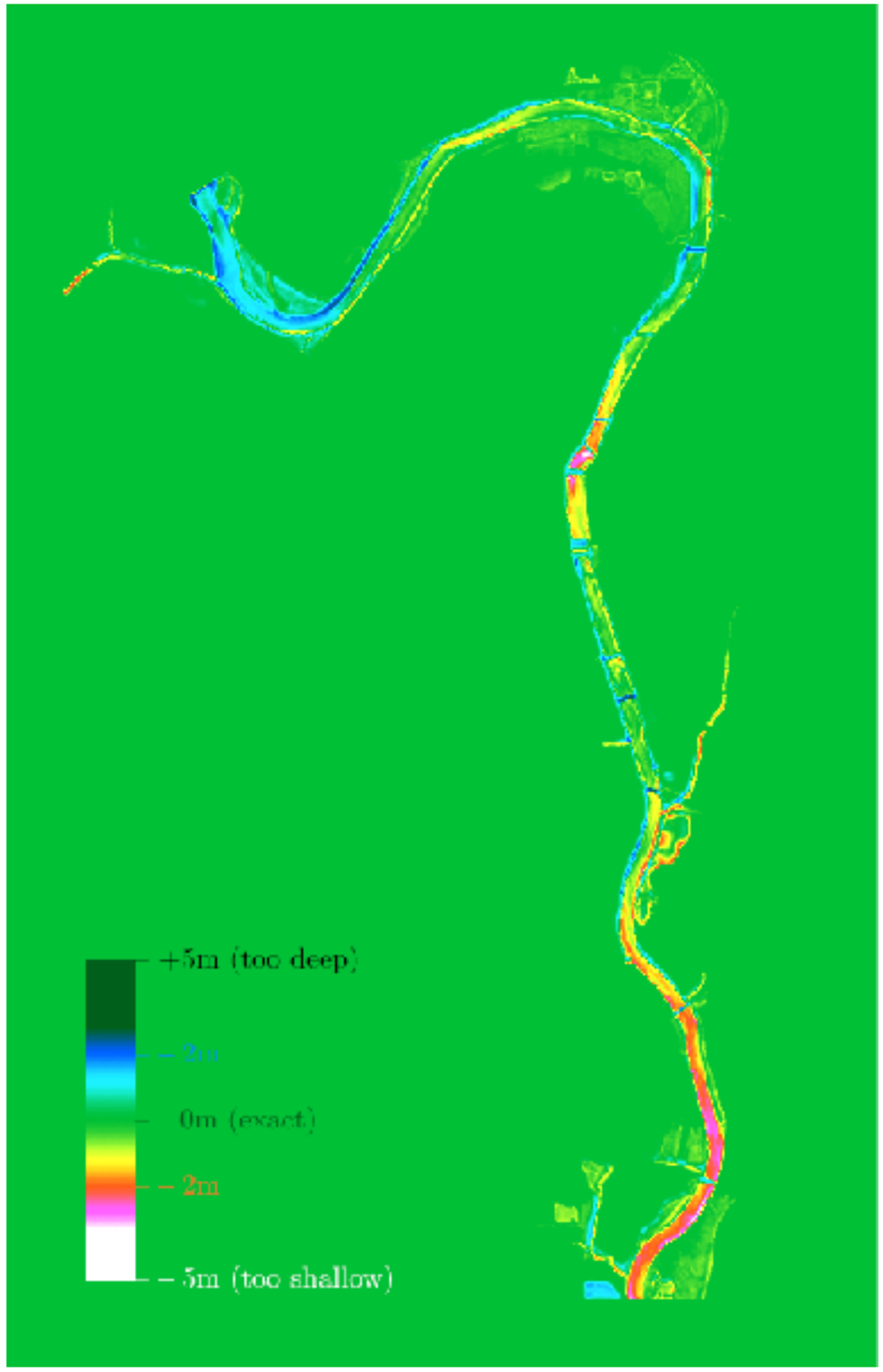
Inundation Maps Demo

WebFlood Demo Iowa

Source Height 7.5 / 25.5ft River Stage

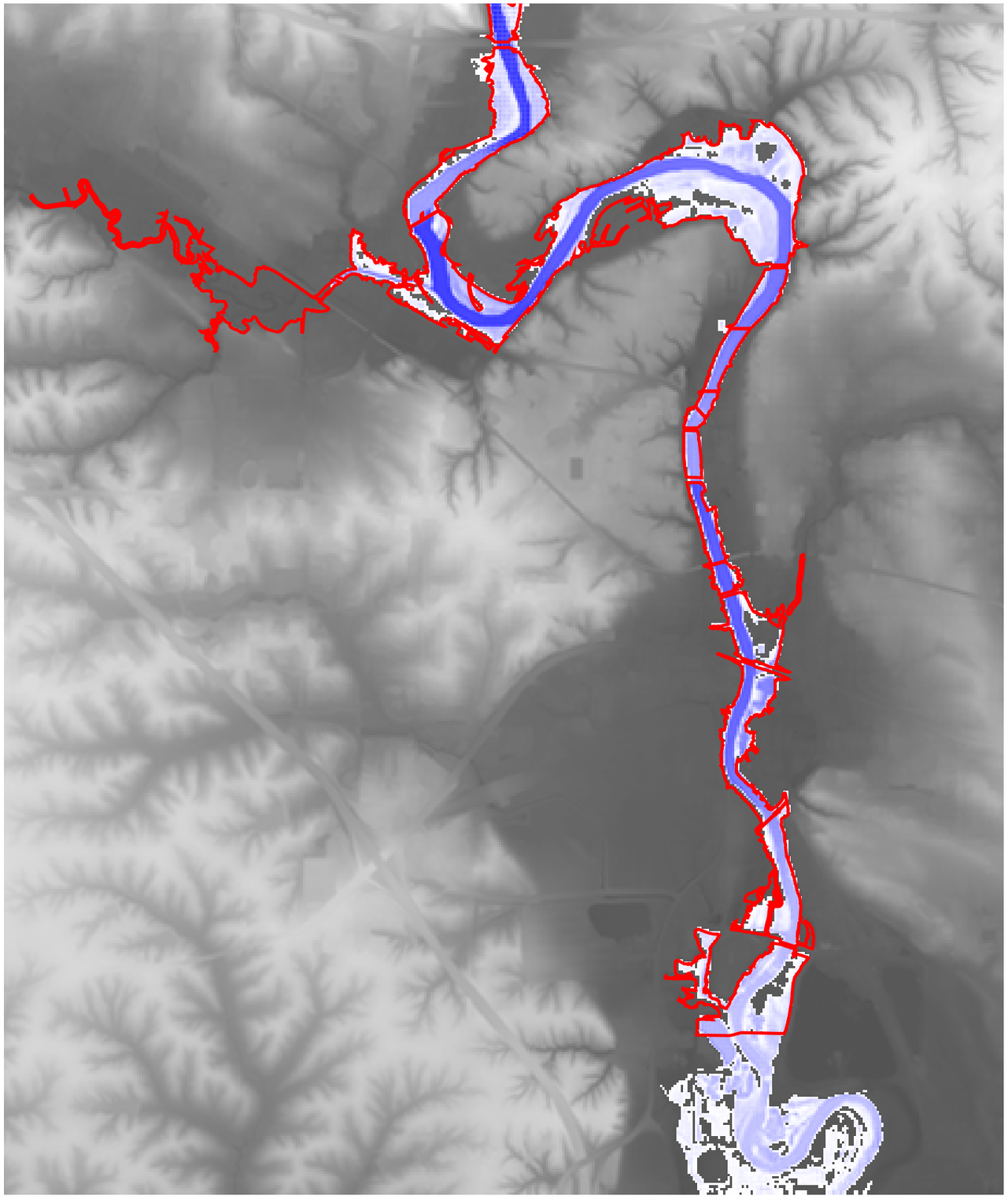


Inundation Extents

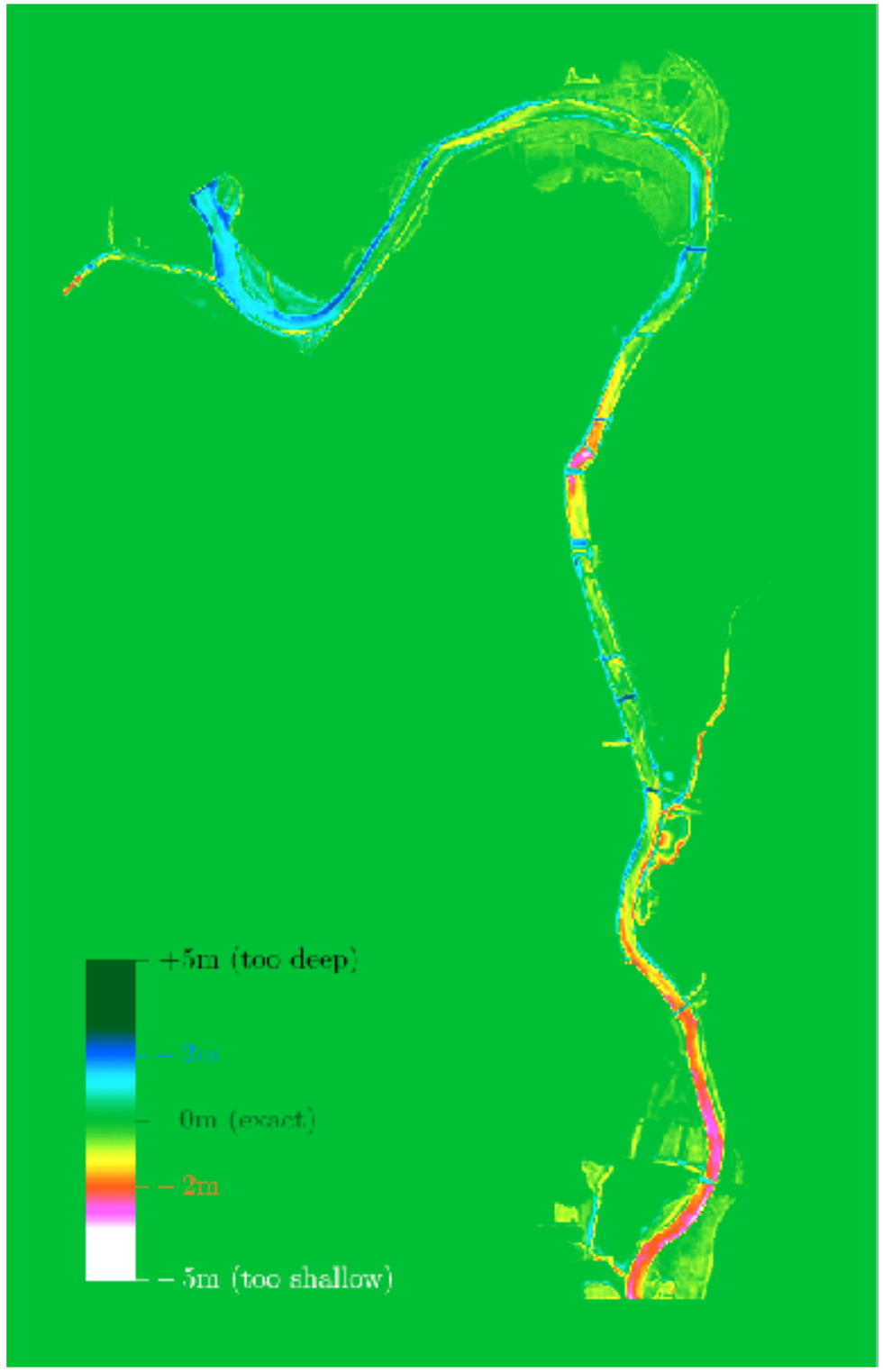


Inundation Depth Error

Source Height 7.7 / 26.5ft River Stage

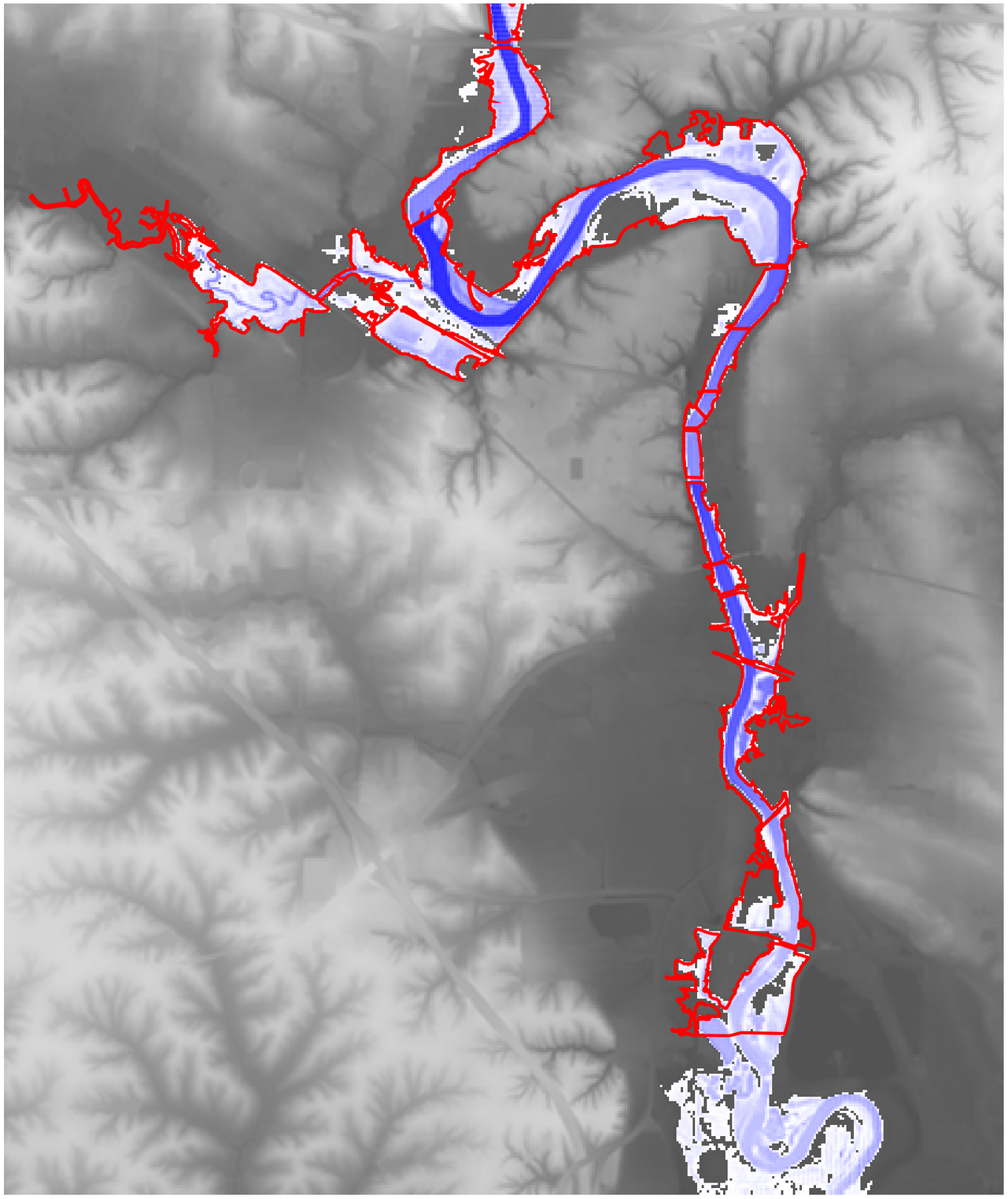


Inundation Extents

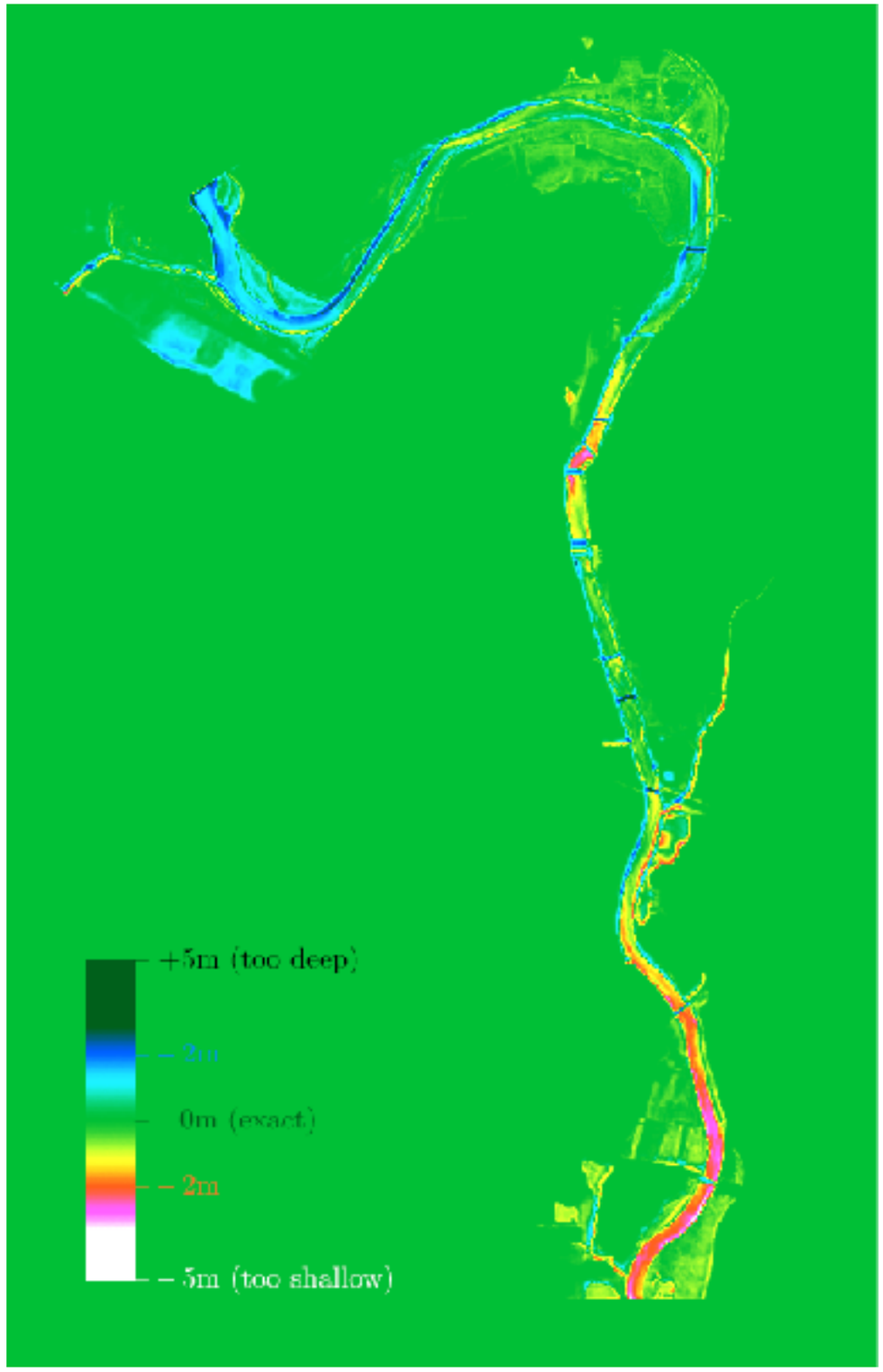


Inundation Depth Error

Source Height 7.9 / 27.5ft River Stage

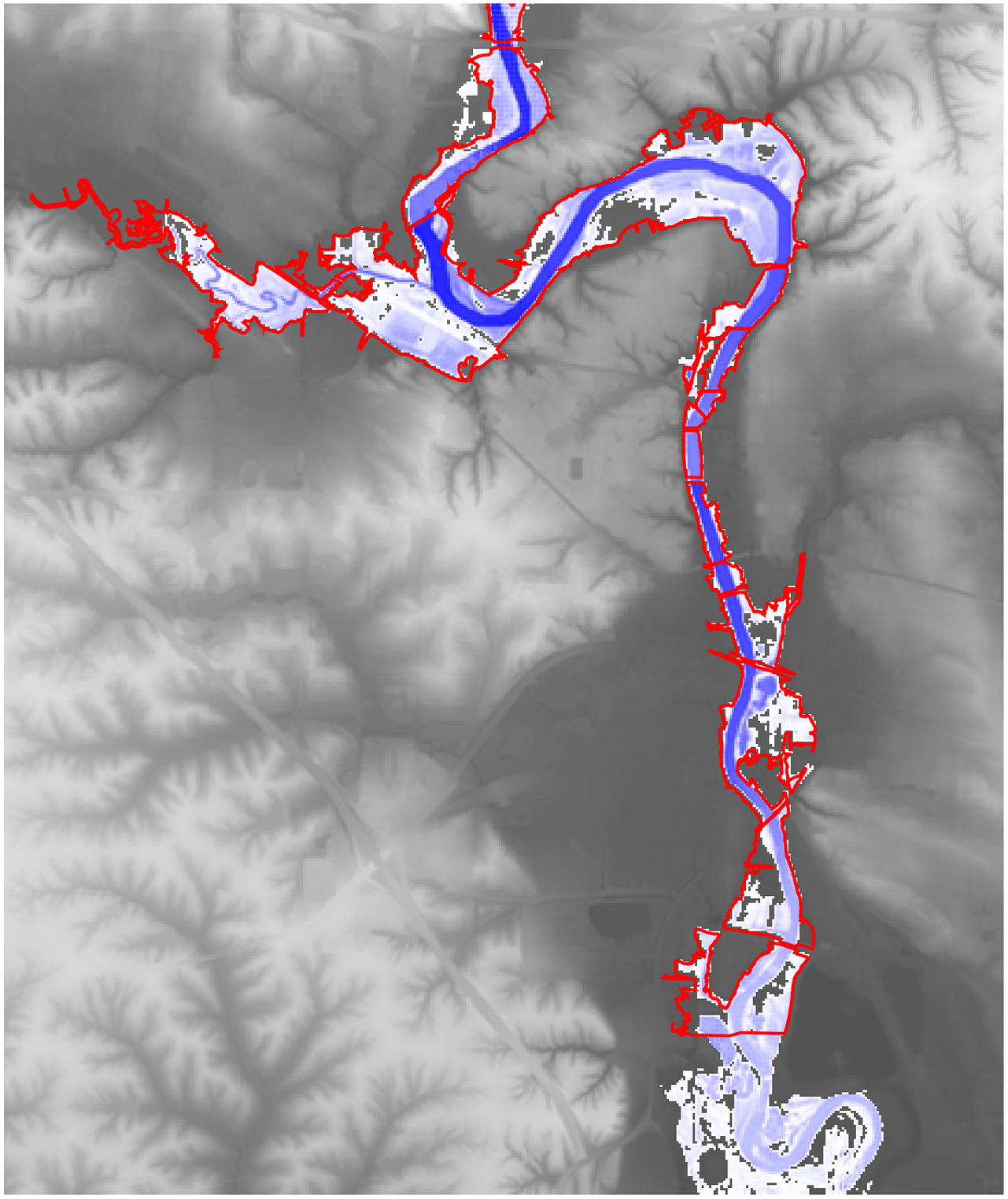


Inundation Extents

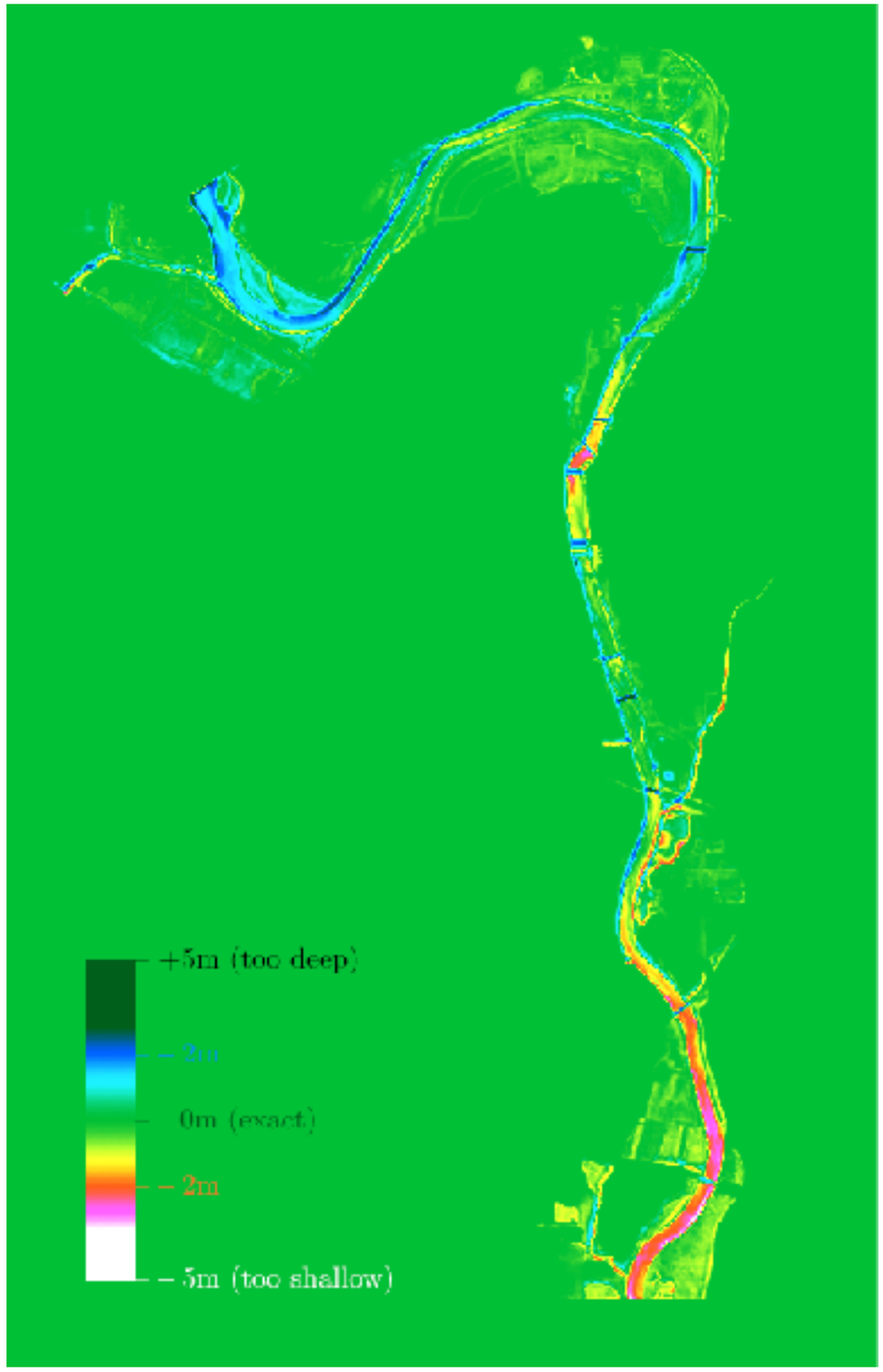


Inundation Depth Error

Source Height 8.1 / 28.5ft River Stage

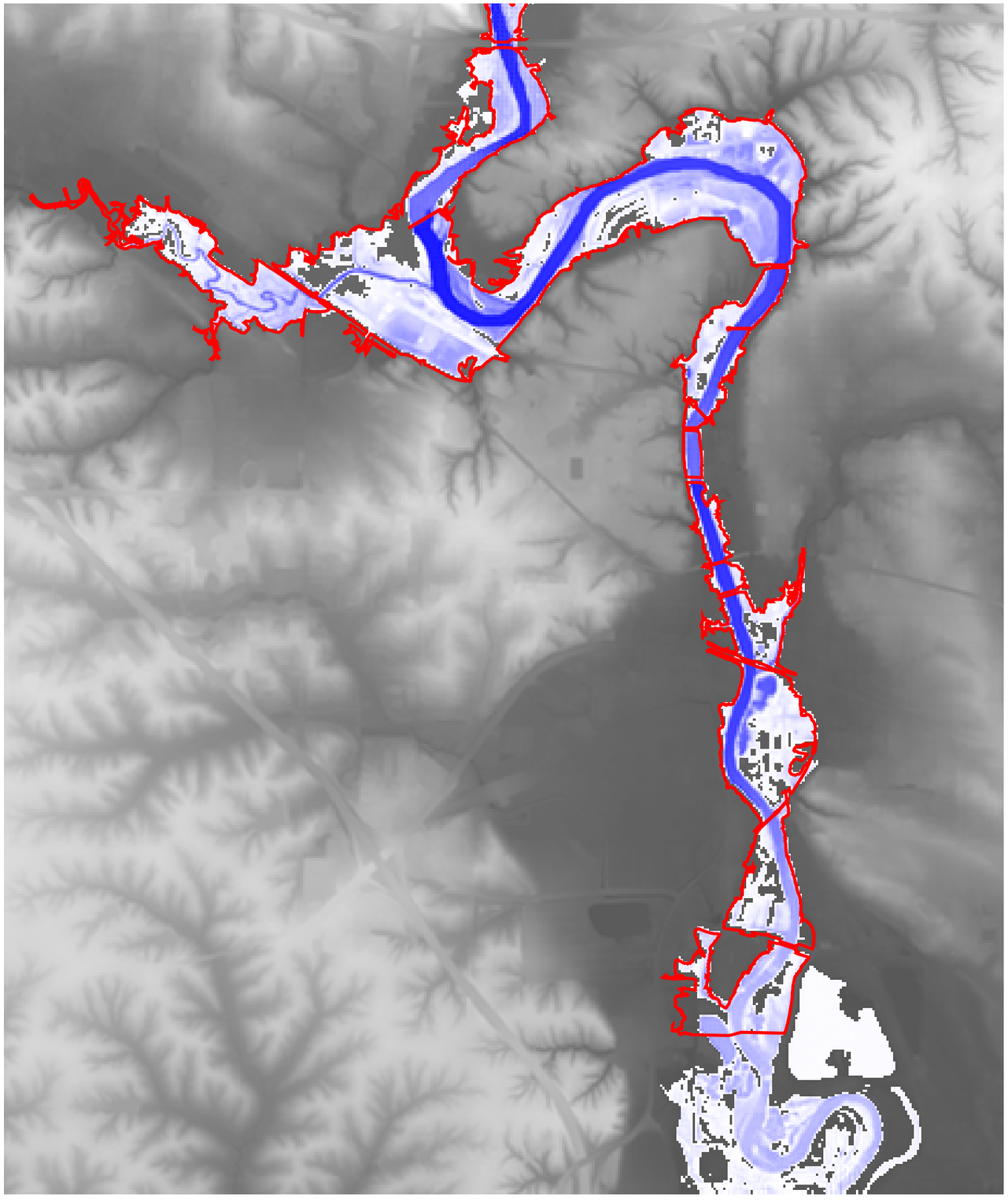


Inundation Extents

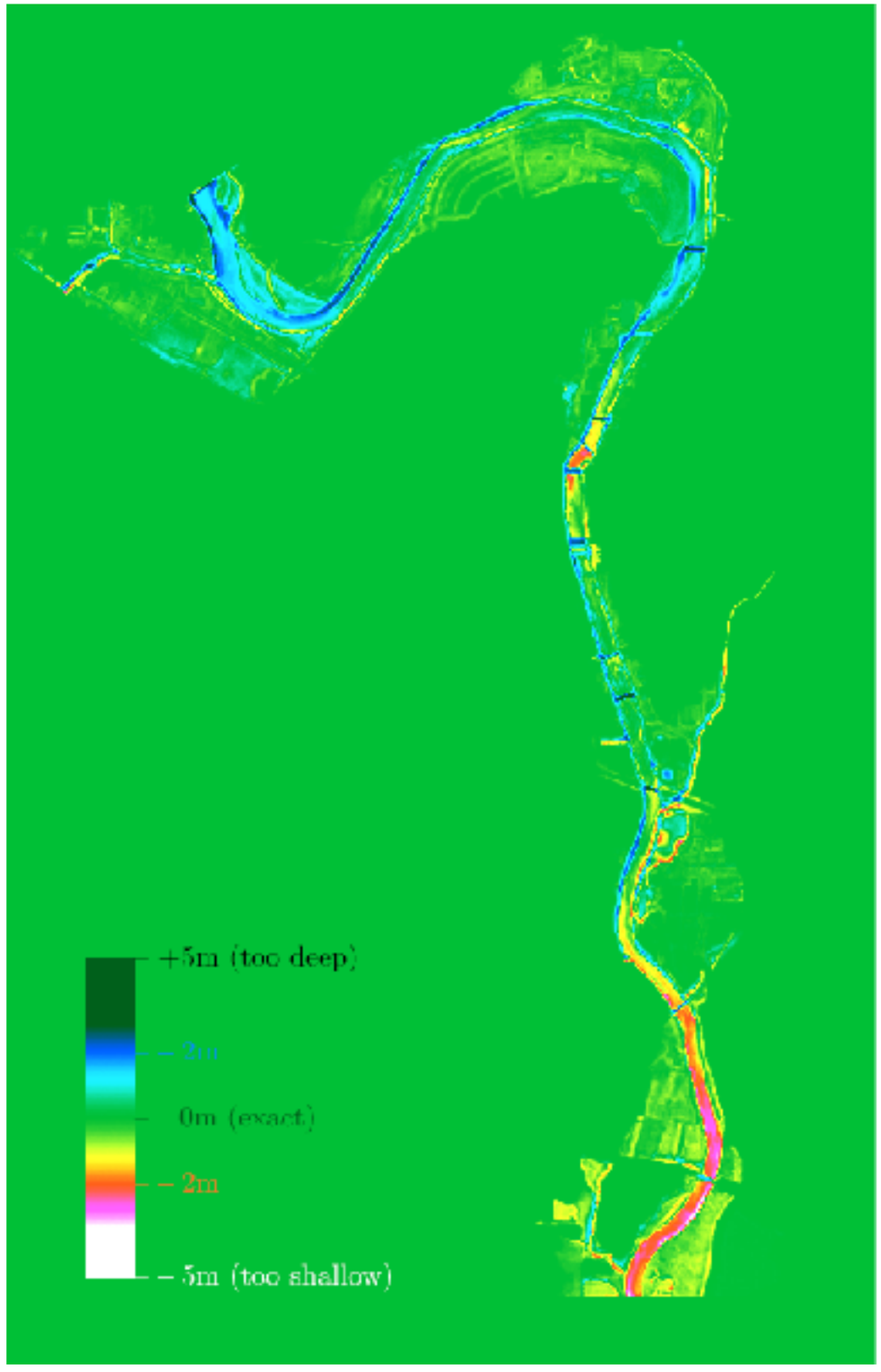


Inundation Depth Error

Source Height 8.3 / 29.5ft River Stage

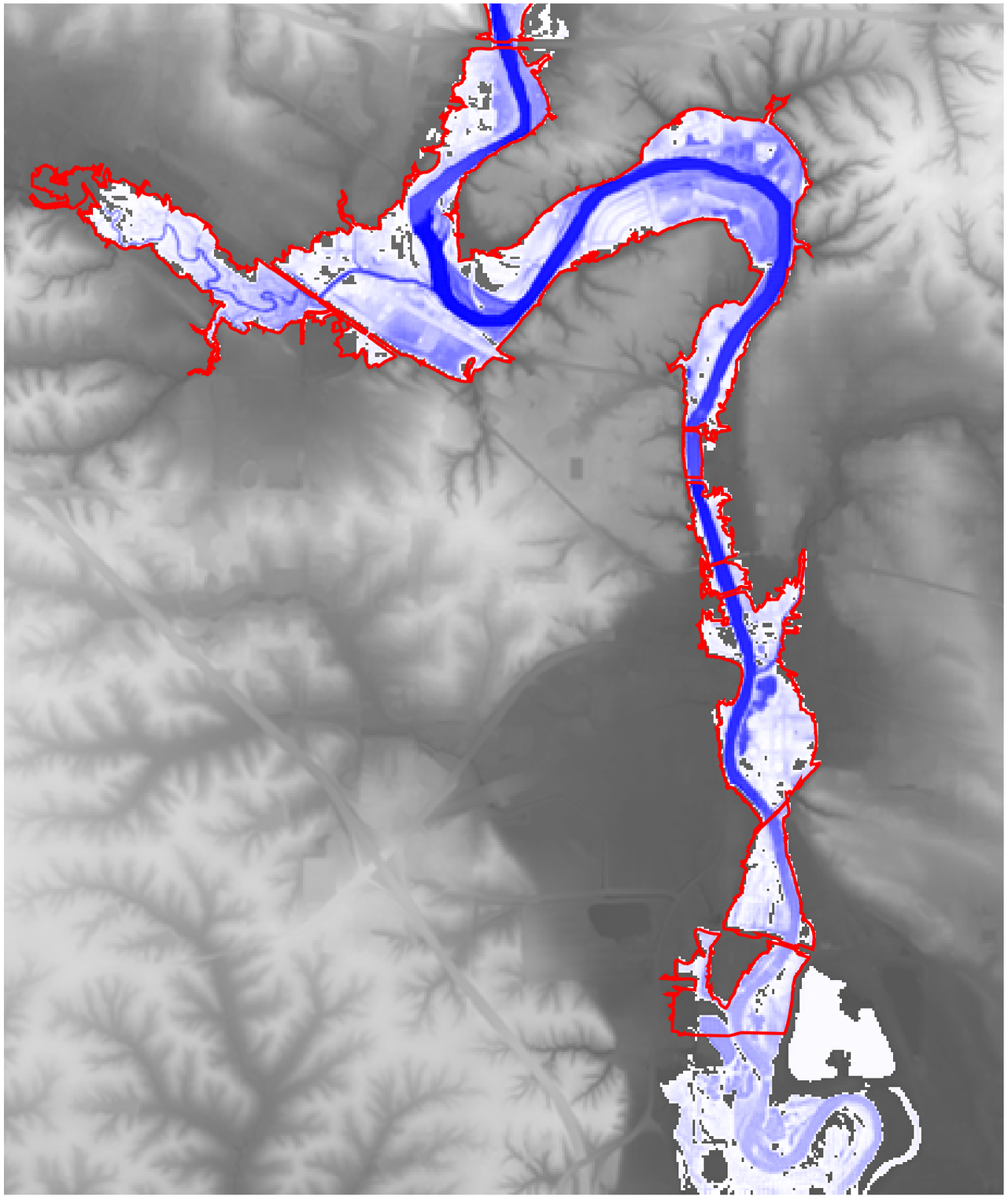


Inundation Extents

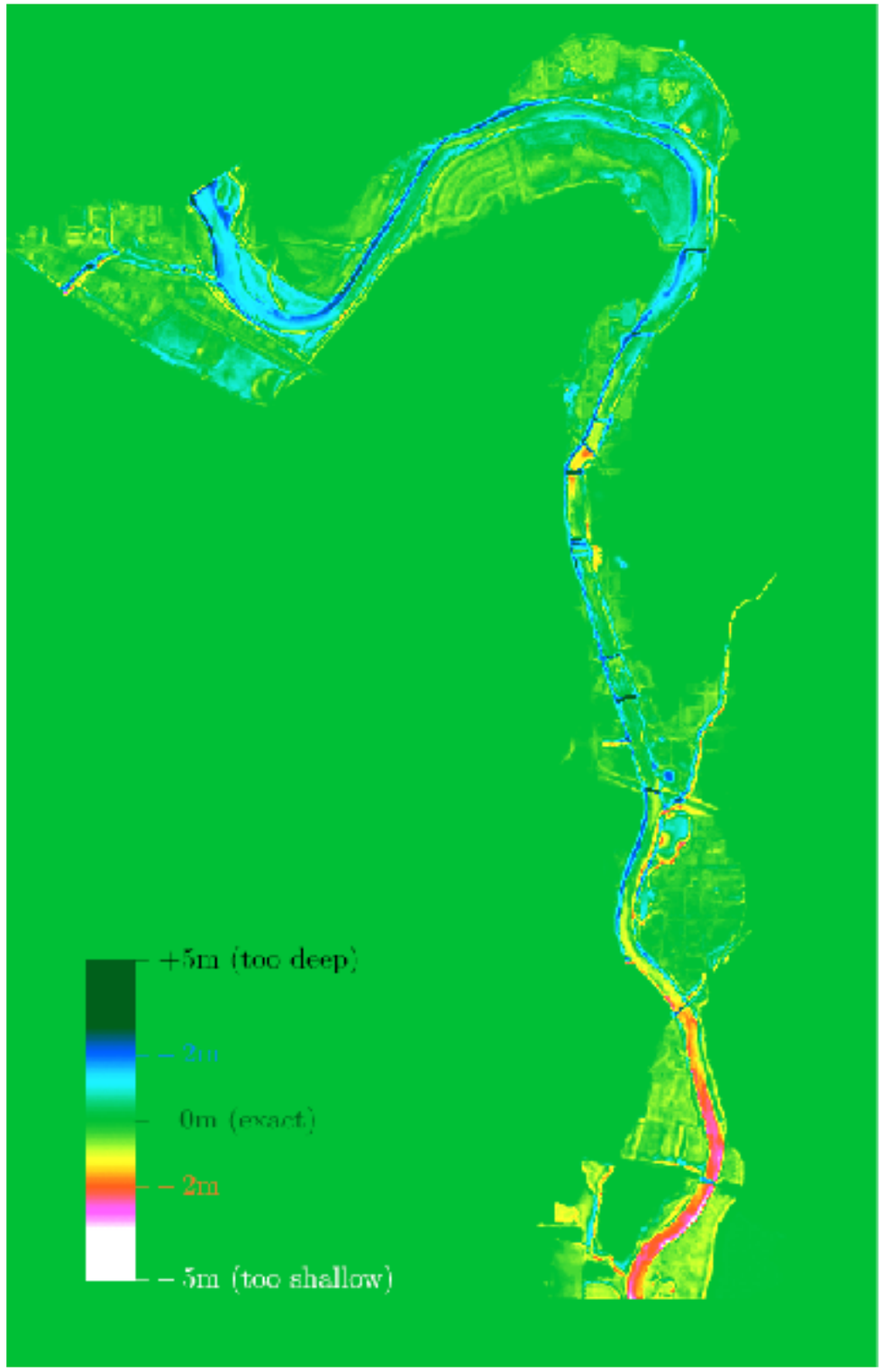


Inundation Depth Error

Source Height 8.9 / 31.5ft River Stage

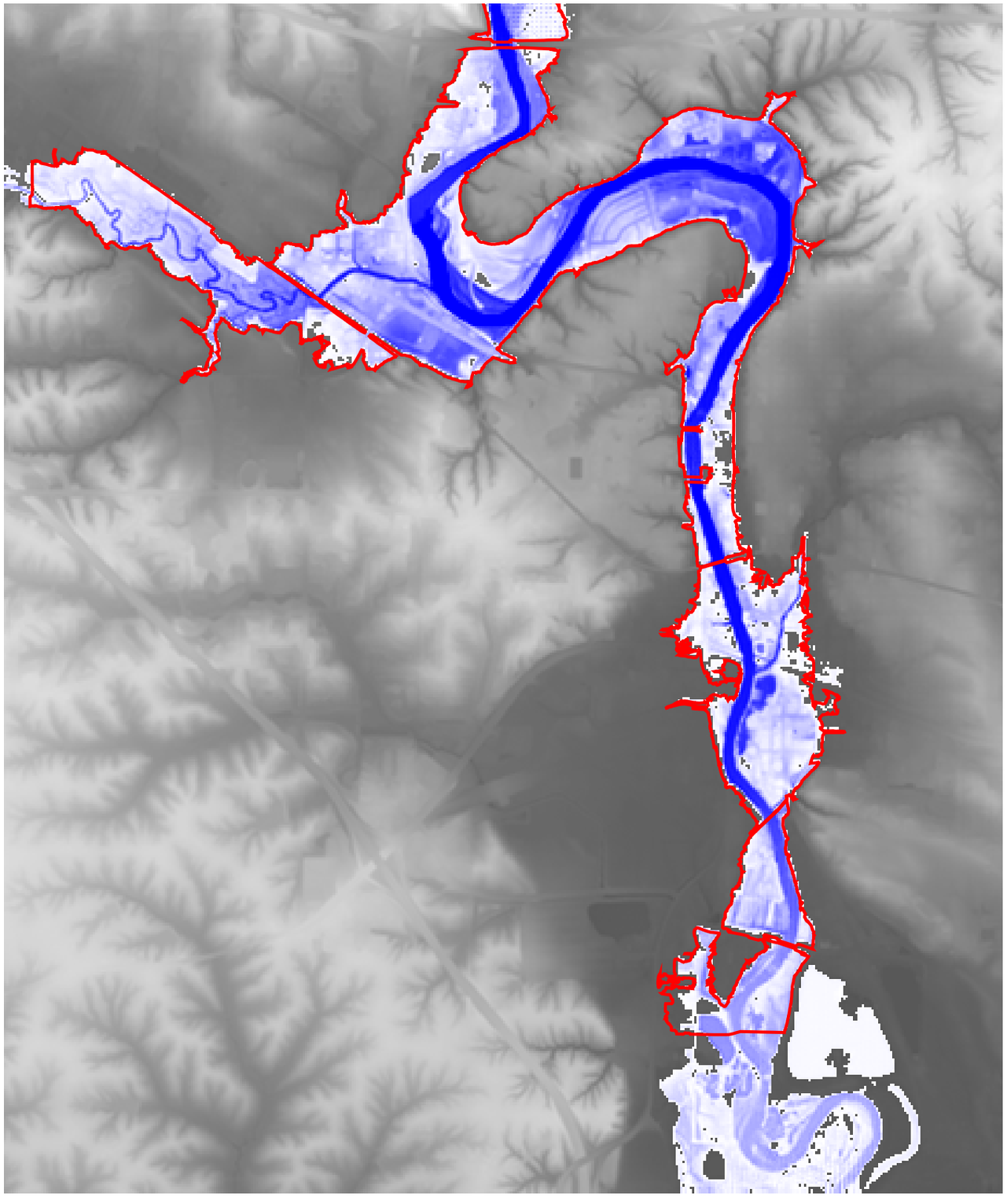


Inundation Extents

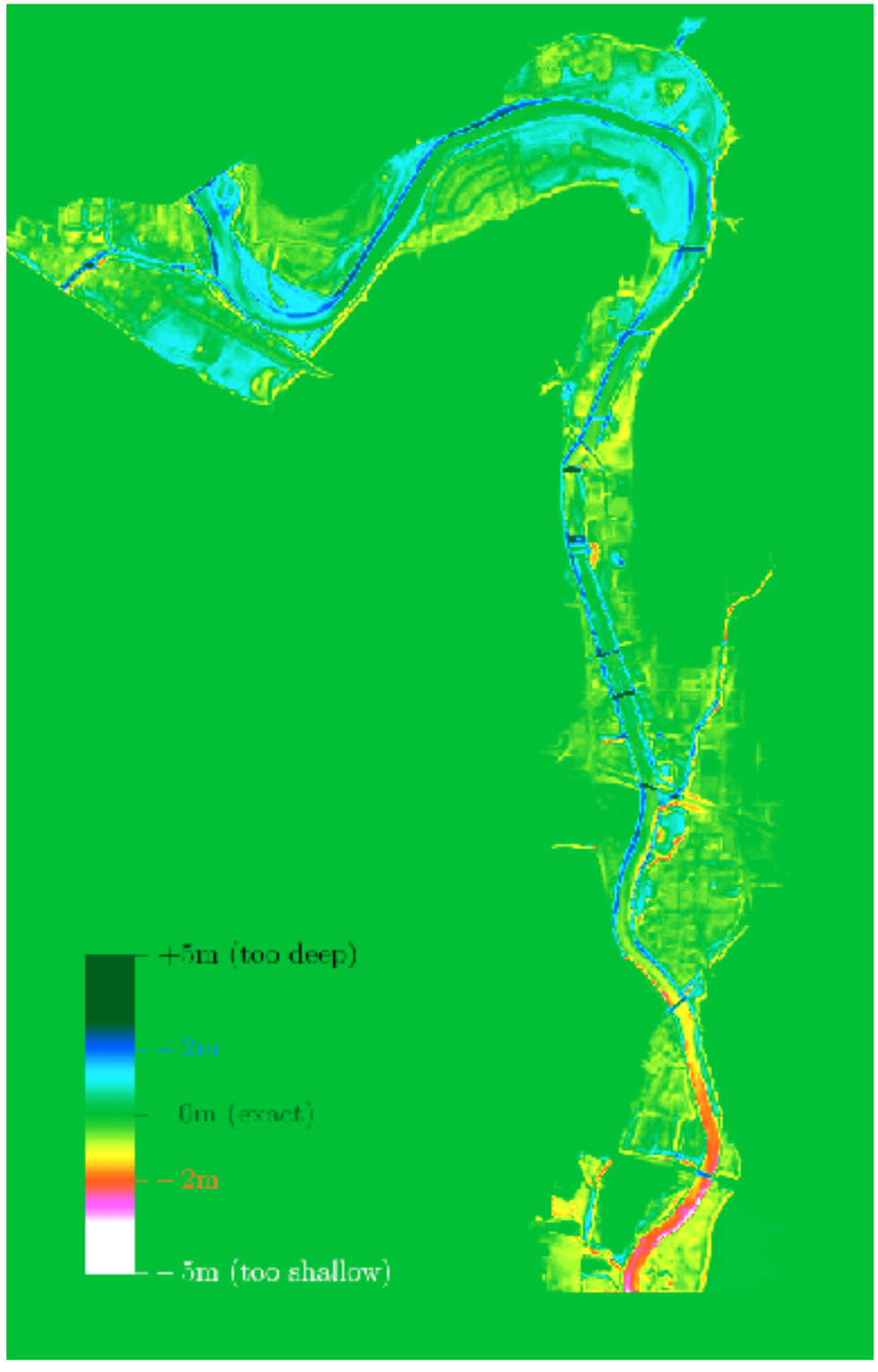


Inundation Depth Error

Source Height 9.5 / 34.0ft River Stage

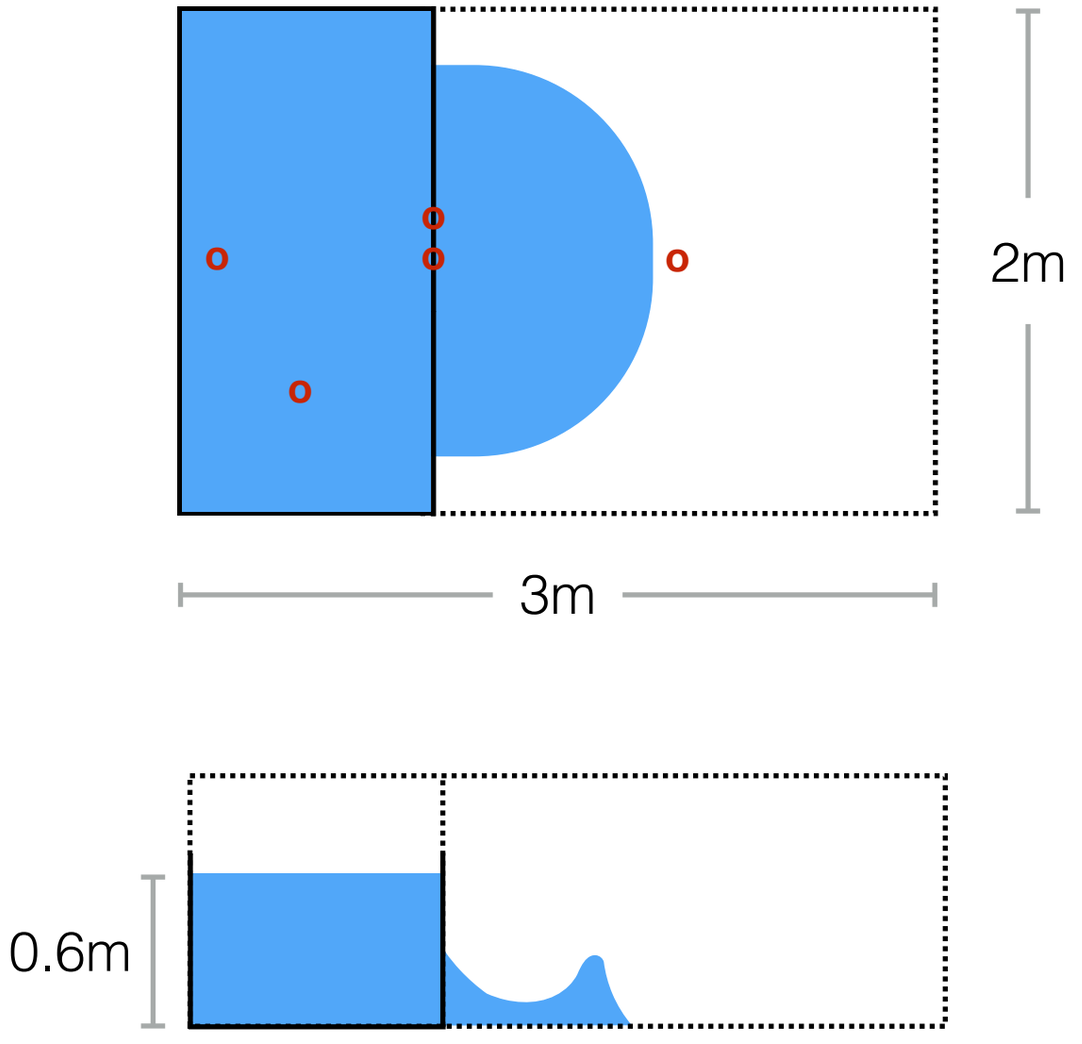


Inundation Extents



Inundation Depth Error

Dam-break experiment of Fraccarollo & Toro (1995)



- Classic Experiment
- Good Measured and Analytical Reference Data

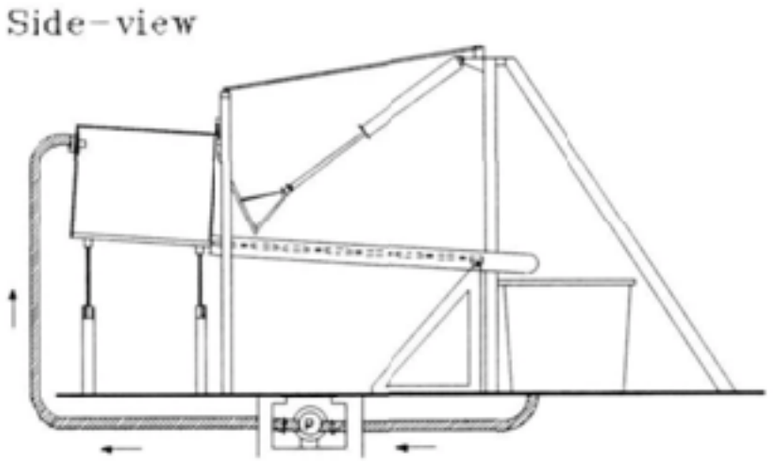
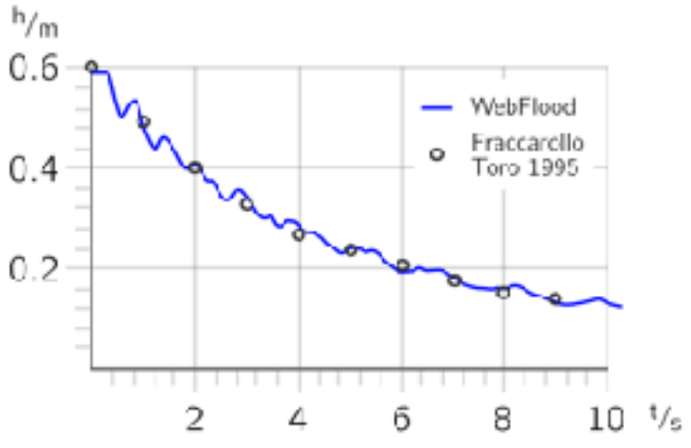
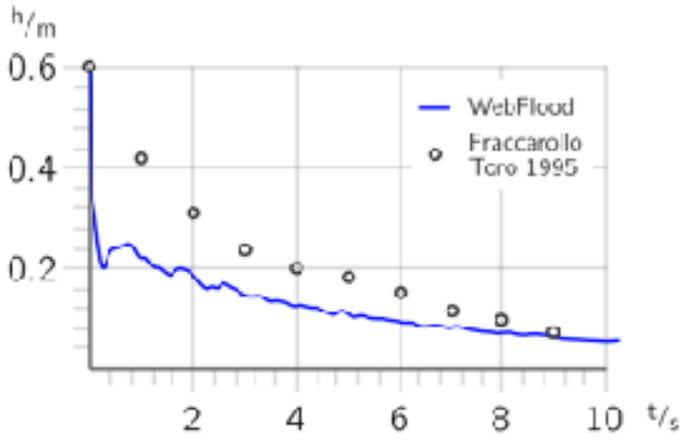


Fig.1. Schematic diagrams of the experimental set-up.

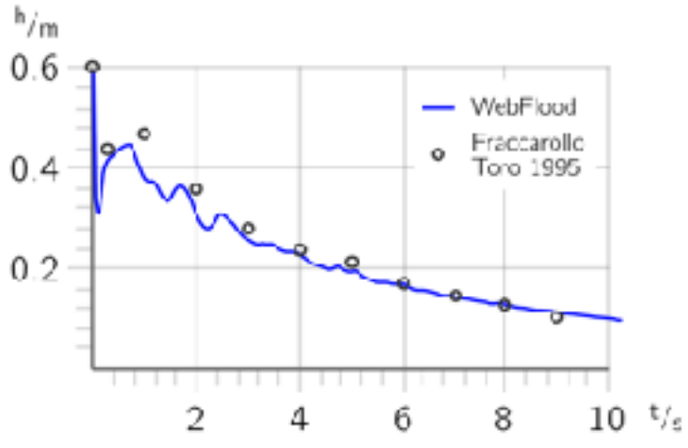
WebFlood Demo Dam-Break



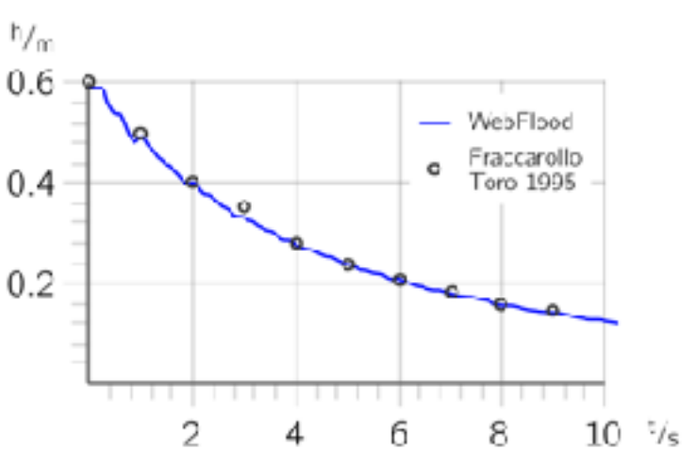
(a) Point -5A



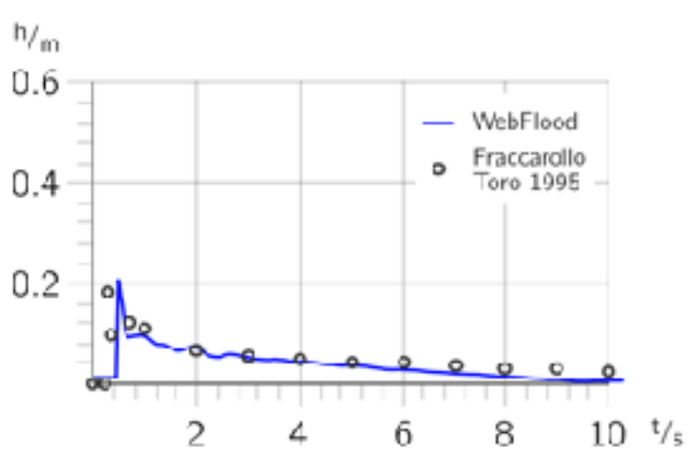
(b) Point 4



(c) Point 0



(d) Point C



(e) Point 8A

Conclusion

Possible Applications

- Public Information Systems
- Flood Response Decision-Making
- Simulation Aided Education

Improvements

- Damage Modelling/Prediction
- Optimisation (simulation domain: sparse!)
- More Rigorous Implementation
  - Full Mass Conservation
  - Better Stability
  - Less Artefacts

WebGL for GPGPU

- GPU → CPU Communication Difficult
  - synchronous, encoding issues
- Missing Features
  - Geometry Shaders
  - Transform Feedback
  - Array Textures
- … coming in WebGL 2!

Interactivity can make even complex things understandable

Bonus WebFlood Demo

Thank you

for your attention

Questions