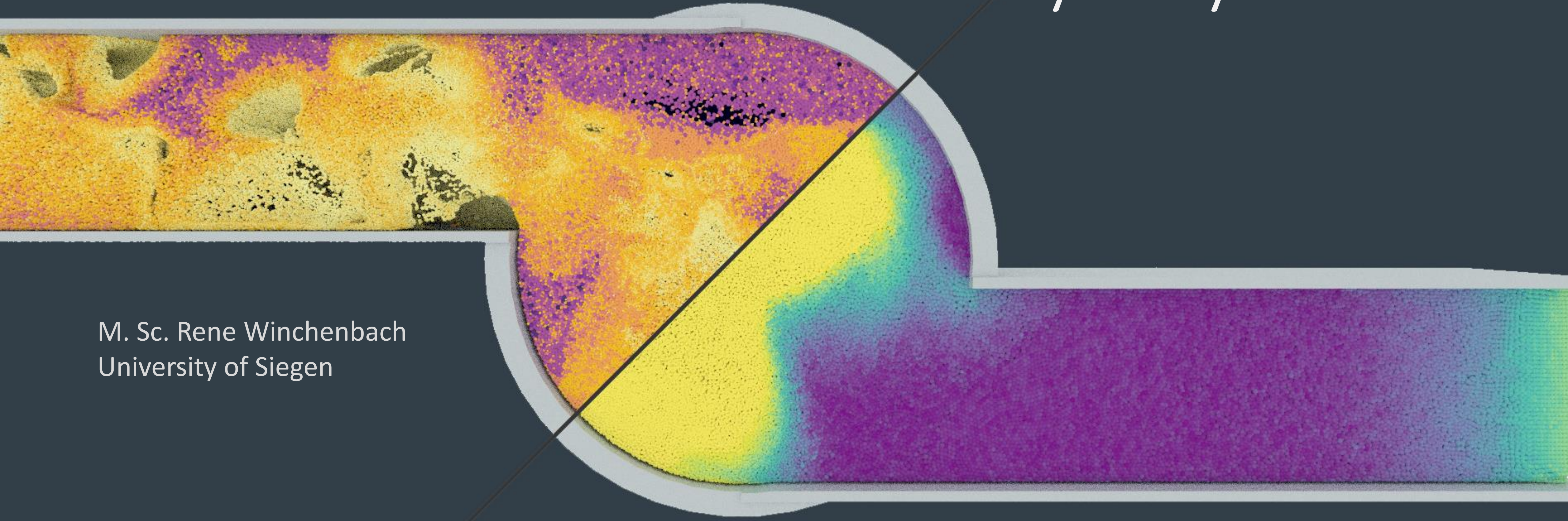
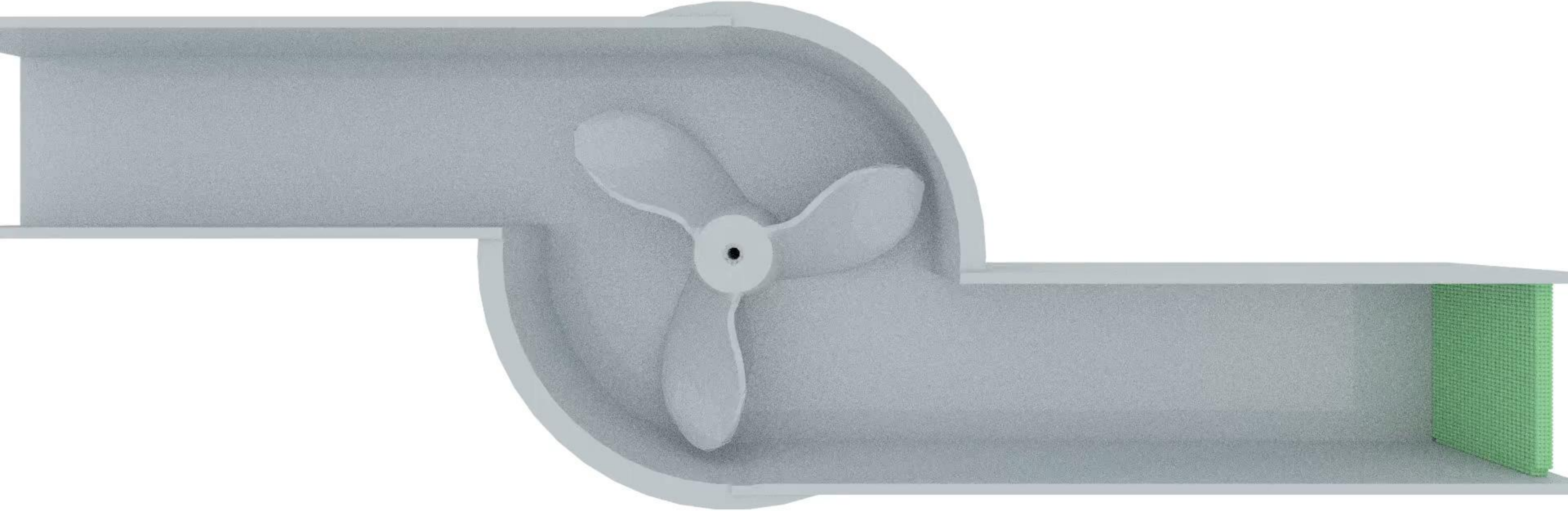
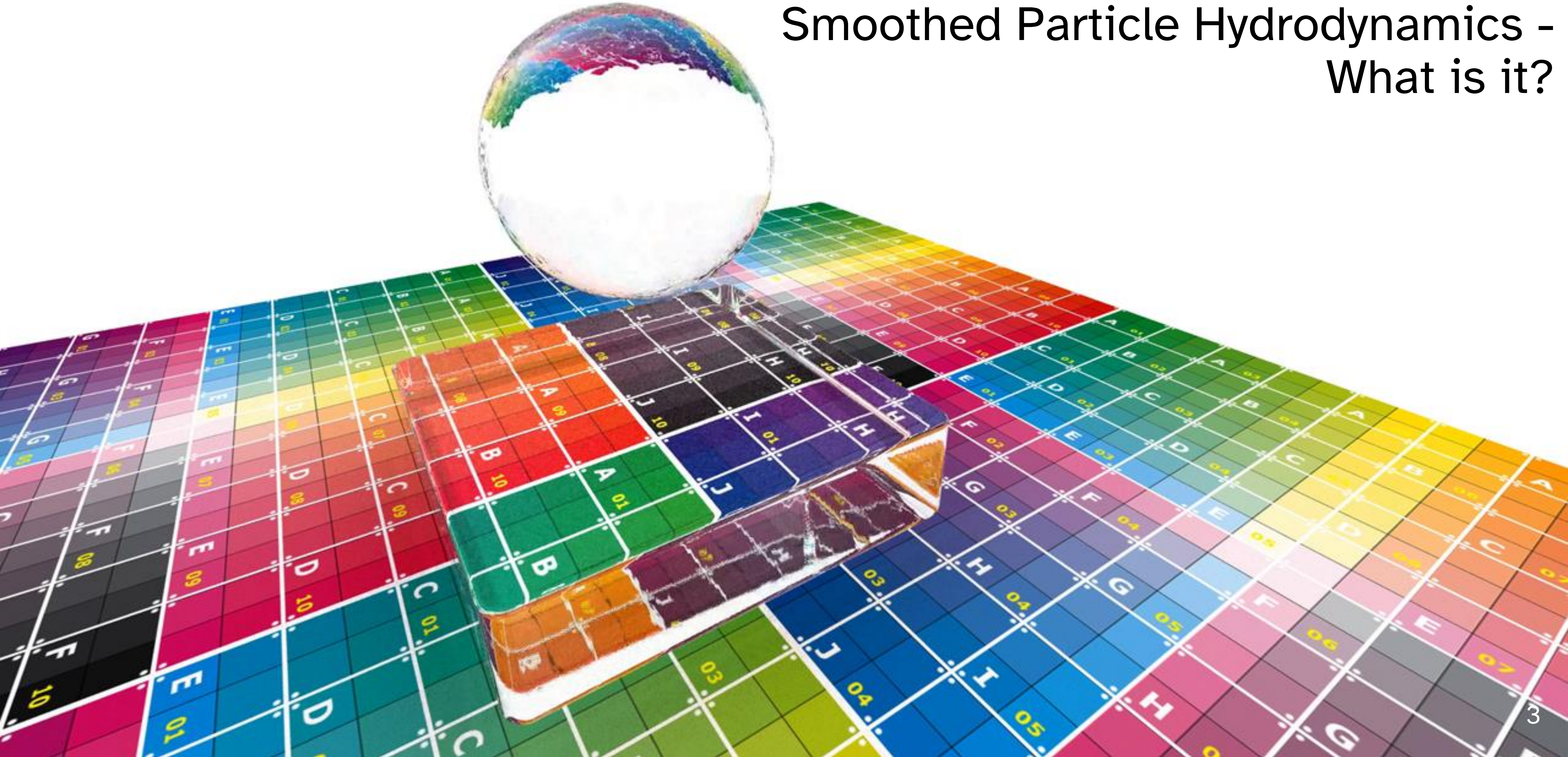
# Spatially Adaptive Smoothed Particle Hydroydnamics



M. Sc. Rene Winchenbach
University of Siegen



# Smoothed Particle Hydrodynamics - What is it?



# Fluid Simulations

- What is a *fluid simulation*?
- Fluid simulations is not just fluid mechanics
- Roughly five sub-fields (for the purposes of this talk)
    1. Fluid mechanics, e.g., the Navier-Stokes equations
    2. Boundary Conditions, e.g., coupling fluids and solids
    3. External Physics, e.g., moving rigid bodies
    4. Discretization, e.g., Eulerian or Lagrangian approaches
    5. Implementation, e.g., how computational complexity can be reduced
- All these sub-fields are interdependent and important
- Changes in one sub-field can affect all other sub-fields

# Fluid Simulations

- Fluid simulations used in two broad fields (for our purposes)
  - Computer Animation (CA)
  - Computational Fluid Dynamics (CFD)
- These fields have very different requirements:
  - Computer Animation focuses on visual appearance
  - Computational Fluid Dynamics focused on realism
- A visually unappealing simulation is easy to spot and adjust
- A wholly unrealistic simulation is easy to spot
- A subtly unrealistic simulation is virtually impossible to spot
- Simulations can be visually appealing but not realistic
- Simulations can be realistic but not visually appealing

# Smoothed Particle Hydrodynamics

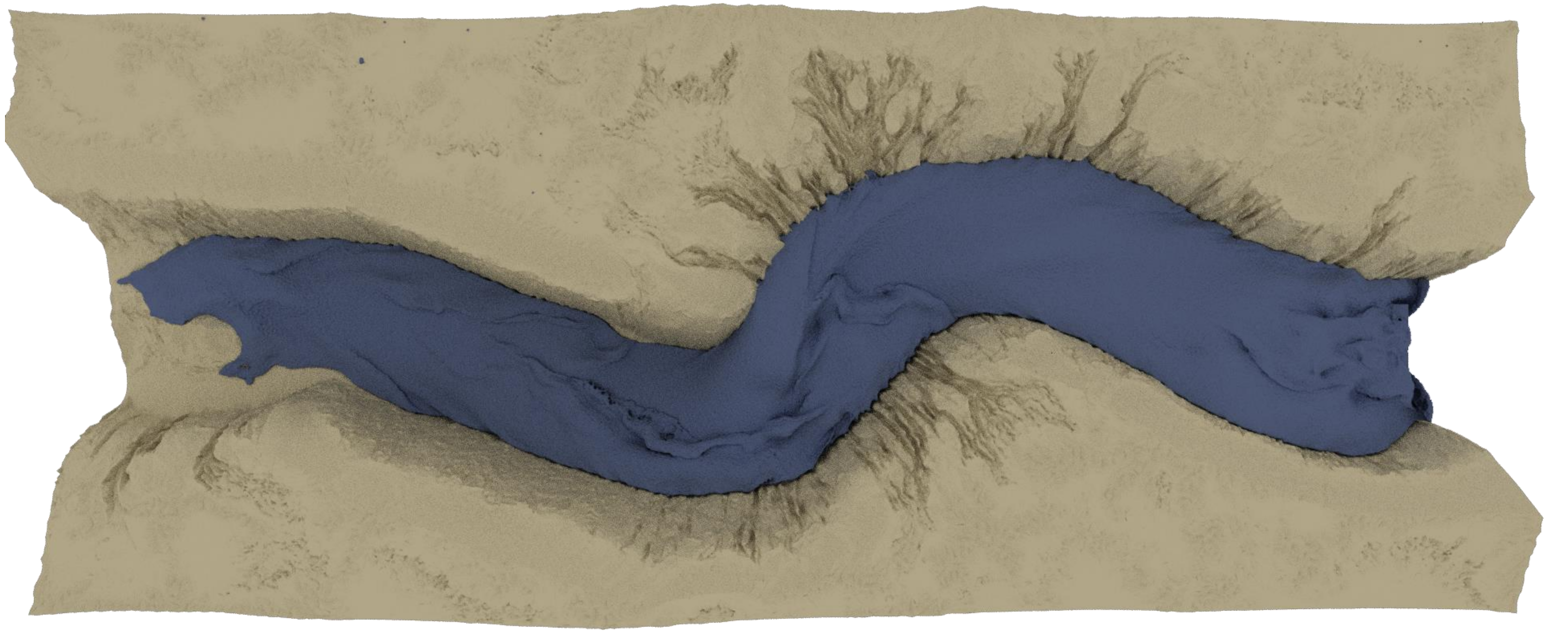
- Lagrangian simulation method for hydrodynamic problems
- SPH established in astrophysics[GM77]
- Mathematically based on density functions[Par62]
- Widely used nowadays in Computer Animation and CFD

Used for:

- Free-surface modelling[Kos+19]
- Conservation of mass[WHK17]
- Computational efficiency[UHT17]

Biggest challenges:

- Computational efficiency
- Incompressibility



Large scale fluid simulation (1km long)

# Core Mathematical Model

- Basic SPH equation is discretization of field identities[Pri12]:

$$A(\boldsymbol{x}) = \int_{\mathbb{R}^d} A(\boldsymbol{x}')\delta(\boldsymbol{x} - \boldsymbol{x}')d\boldsymbol{x}'$$

- Dirac Delta $\delta$ replaced using Kernel function $W(\boldsymbol{r}, h) = \frac{C_d}{h^d}\widehat{W}\left(\frac{\|\boldsymbol{r}\|}{h}\right)$
- Wendland-4 Kernel[DA12]: $\widehat{W}(q) = [1 - q]_+^6\left(1 - 6q + \frac{35}{3}\right)$
- $\mathbb{R}^d$ is limited to compact domain around points $\Omega_{\boldsymbol{x}}$ using support $h$
- Integral discretized using particles:

$$\langle A(\boldsymbol{x})\rangle = \sum_j \frac{m_j}{\rho_j} A_j W(\boldsymbol{x} - \boldsymbol{x}_j, h)$$

- Density interpolant using SPH[Mon95]:

$$\langle \rho(\boldsymbol{x})\rangle = \sum_j \frac{m_j}{\rho_j} \rho_j W(\boldsymbol{x} - \boldsymbol{x}_j, h) = \sum_j m_j W(\boldsymbol{x} - \boldsymbol{x}_j, h)$$

# Gradients

- Gradient of SPH Interpolant:

$$\langle \nabla_{\mathbf{x}} A(\boldsymbol{x}) \rangle = \sum_j \frac{m_j}{\rho_j} A_j \nabla_{\boldsymbol{x}} W(\boldsymbol{x} - \boldsymbol{x}_j, h)$$

- Kernel function gradient depends on derivative of distance $\boldsymbol{x} - \boldsymbol{x}_j$ and support $h$
- Assumption in Computer Animation: $\nabla_{\boldsymbol{x}} h = 0 \wedge \frac{dh}{dt} = 0$ (grad-h terms)
- Gradients not accurate for constant fields or symmetric interactions[Pri12]
  - For constant: $\langle \nabla_{\mathbf{x}} A(\boldsymbol{x}) \rangle \approx \frac{1}{\rho(\boldsymbol{x})} \left[ \left\langle \nabla_{\mathbf{x}} \left( \rho(\boldsymbol{x}) A(\boldsymbol{x}) \right) \right\rangle - A(\boldsymbol{x}) \langle \nabla_{\mathbf{x}} \rho(\boldsymbol{x}) \rangle \right]$
  - For symmetry: $\langle \nabla_{\mathbf{x}} A(\boldsymbol{x}) \rangle \approx \rho(\boldsymbol{x}) \left[ \frac{A(\boldsymbol{x})}{\rho(\boldsymbol{x})^2} \langle \nabla_{\mathbf{x}} \rho(\boldsymbol{x}) \rangle + \left\langle \nabla_{\mathbf{x}} \frac{A(\boldsymbol{x})}{\rho(\boldsymbol{x})} \right\rangle \right]$
- Pressure gradient using SPH[Kos+19]:

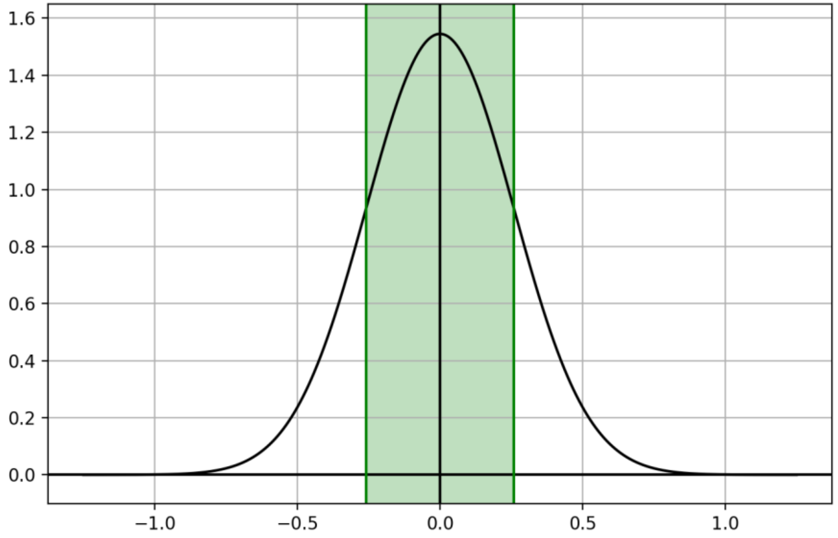
$$\nabla p(\boldsymbol{x}) = \rho(\boldsymbol{x}) \sum_j m_j \left( \frac{p(\boldsymbol{x})}{\rho(\boldsymbol{x})^2} + \frac{p_j}{\rho_j^2} \right) \nabla_{\mathbf{x}} \mathrm{W}(\mathbf{x} - \mathbf{x}_{\mathrm{j}}, \mathrm{h})$$

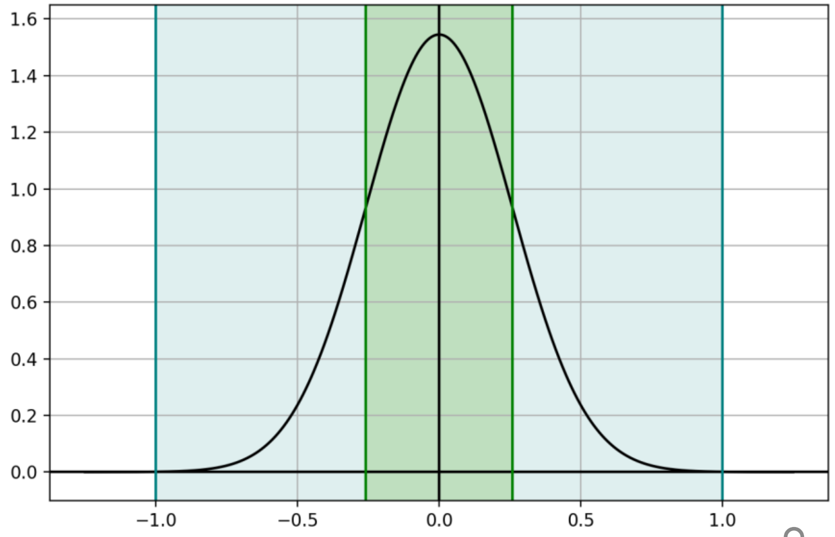
# Kernel Functions

- The kernel function replaces a Dirac delta function
- *Obvious* choice: Gaussian kernel with infinite support:

$$W(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{r^2}{2\sigma^2}}$$

- $\sigma$ describes how quickly influence fades
- Note that $\lim_{\sigma \to 0} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{r^2}{2\sigma^2}} = \delta(r)$
- Due to infinite support SPH would be $\mathcal{O}(n^2)$
- But influence after some distance practically 0
- Accordingly: cut off kernel at arbitrary *support radius* H
- Reduces complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$
- However, as $W(H) \neq 0$, this yields a $C^{-1}$ kernel



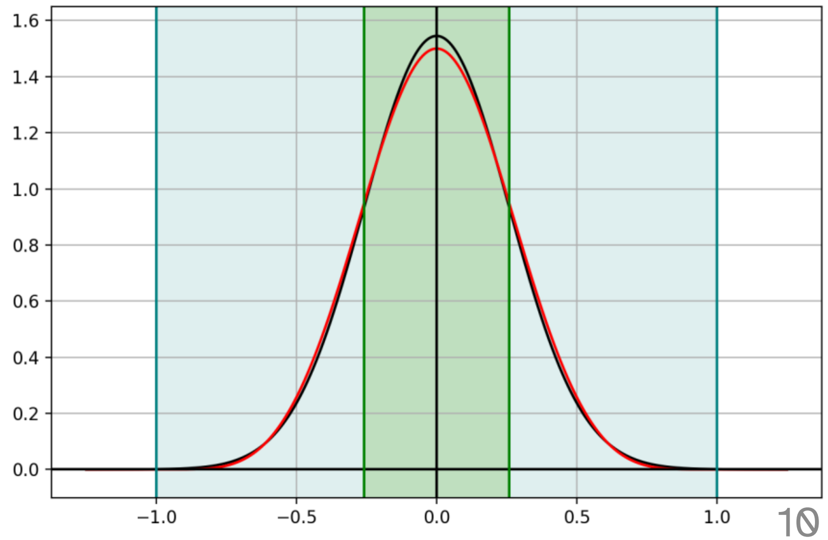



# Kernel Functions

- Solution: Construct kernel that is 0 at H:

$$\widehat{W}(q) = [1-q]_+^6 \left(1 - 6q + \frac{35}{3}\right)$$

- This kernel with support radius H approximates a Gaussian kernel with deviation $\sigma$
- Accordingly, there is a factor $H/\sigma$ to convert between them
- One approach: define smoothing scale $h = 2\sigma$ and evaluate standard deviation of W
- For Wendland-4[DA12]: $\frac{H}{h} = 1.934592\ldots$
- However, definition of $h$ is mostly arbitrary
- One could set $h = 3.869184\ldots\sigma$, yielding $\frac{H}{h} = 1$
- In CA this is a very common approach
- Consequently, $h$ refers to the support radius[DA12]



# Support Radius

- Support radius determines number of neighbors for a particle at rest
- Each kernel function has a numerically ideal neighborhood size $N_h$ [DA12]
- Particles have volume based on their mass and density: $V = \frac{m}{\rho}$
- Support domain is spherical (for isotropic SPH) and of volume: $V = \frac{4}{3}\pi h^3$
- Expected neighbors $N = \rho \frac{4\pi h^3}{3m} \rightarrow h = \sqrt[3]{\frac{3}{4\pi}\frac{m}{\rho}N}$
- Inserting $N_h$ for particle $i$ yields:

$$h_i = \sqrt[3]{\frac{3}{4\pi}\frac{m_i}{\rho_i}N_h} \leftrightarrow \rho_i = \sum_j m_j W\left(x_i - x_j, \frac{h_i + h_j}{2}\right)$$

- This creates a cycle where $\rho$ and $h$ are interdependent
- Solution (often used in CA): $\rho_i = \rho_0$ when determining $h$
- Alternatively, only evaluate this cycle once

# Incompressibility & Divergence-Freedom

- Divergence-Freedom based on continuity equation[BK15]:

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot \boldsymbol{v}$$

- An incompressible fluid is at its rest density in the bulk
- Consequently, an incompressible fluid is also divergence-free
- However, a divergence-free fluid need not be incompressible
- Free surfaces in SPH are below rest density due to incomplete neighborhoods
- In standard approaches the continuity equation is discretized as[Ihm+13]:

$$\rho_i^{t+\Delta t} = \rho_i^t + \Delta t \sum_j m_j \boldsymbol{v}_{ij}^{t+\Delta t} \cdot \nabla_i W_{ij}$$

# Incompressibility & Divergence-Freedom

- Discretized using semi-implicit Euler and standard SPH[Ihm+13]:

$$\rho_i^{t+\Delta t} = \rho_i^t + \Delta t \sum_j m_j \boldsymbol{v}_{ij}^{t+\Delta t} \cdot \nabla_i W_{ij}$$

- Standard SPH approaches use predicted velocities[Ihm+13,BK15]:

$$\boldsymbol{v}_i^{t+\Delta t} = \boldsymbol{v}_i^t + \Delta t \frac{D\boldsymbol{v}_i^t}{Dt}$$

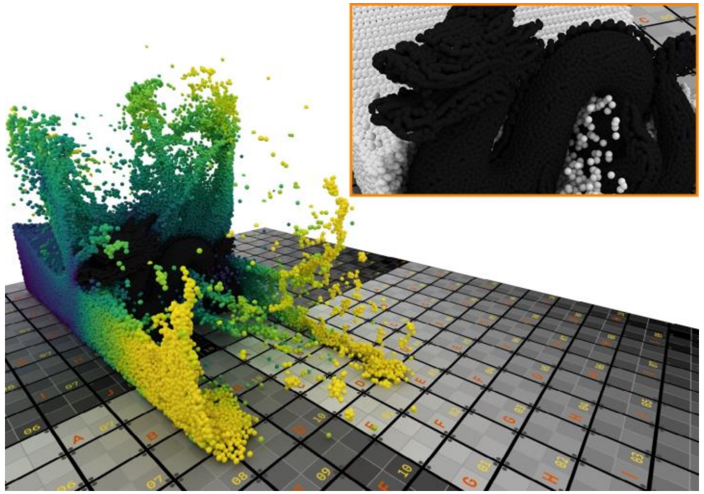
- Use Navier-Stokes for material derivative:

$$\rho \frac{Dv_i}{Dt} = -\nabla p + \mu \nabla^2 \boldsymbol{v} + \boldsymbol{f}^{ext}$$

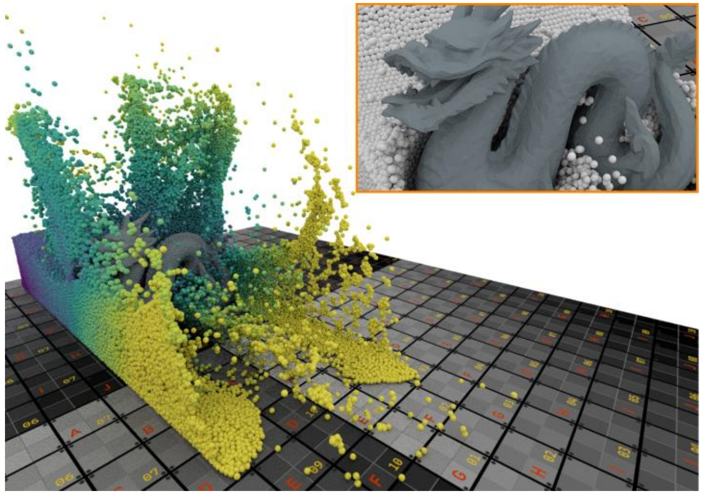
- Yields system of equations $\mathrm{A}\boldsymbol{p} = \boldsymbol{s}$ with unknown pressure values
- $\rho_i^{t+\Delta t} = \rho_0$ models incompressibility
- $\rho_i^{t+\Delta t} = \rho_i^t$ models divergence-freedom[Ban+18]

# Boundary Handling

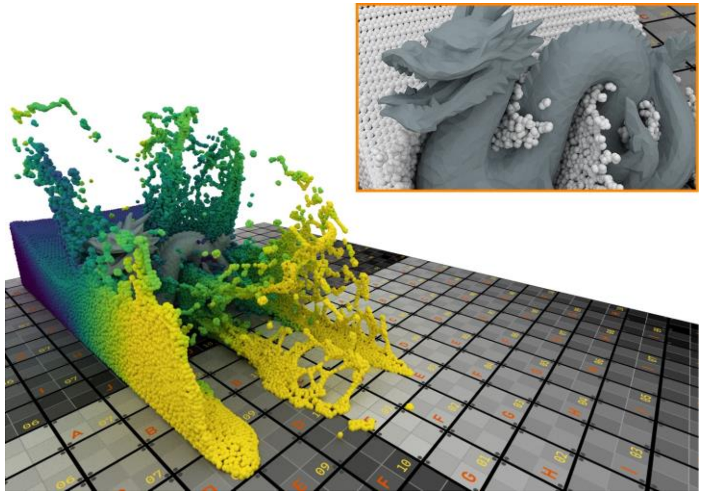
- Simulations are rarely involve only fluids
- Boundaries to the simulation are usually not Lagrangian → need discretization
- Three (recent) approaches to boundary handling:
    1. Boundary surfaces sampled with ghost-particles
    2. Integrals directly evaluated over the boundary
    3. Semi-analytic models using heuristics



Particle-based[Gis+19]



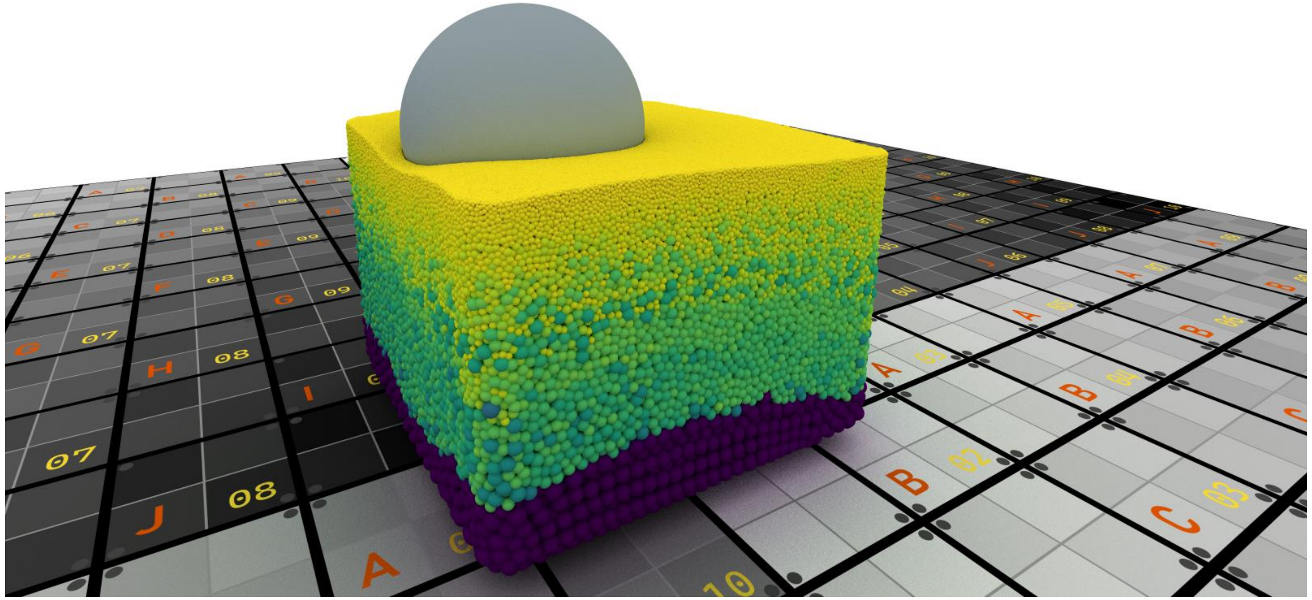
Integral-based[Ben+19]



Semi-Analytic[WAK20]

# Spatial Adaptivity

- Global resolution changes scale computationally with $\mathcal{O}\left(\frac{1}{r^3}\right)$
- Accordingly, focusing resources on small regions highly beneficial
- In Computer Animation generally the focus is on the fluid surface
- Increase resolution through *splitting*
- Splitting particles introduces significant errors
- Impact of errors reduced by:
  - Removing badly behaving particles
  - Adding artificial viscosity
  - Temporal blending
- Decrease resolution through *merging*
- Generally, not error prone

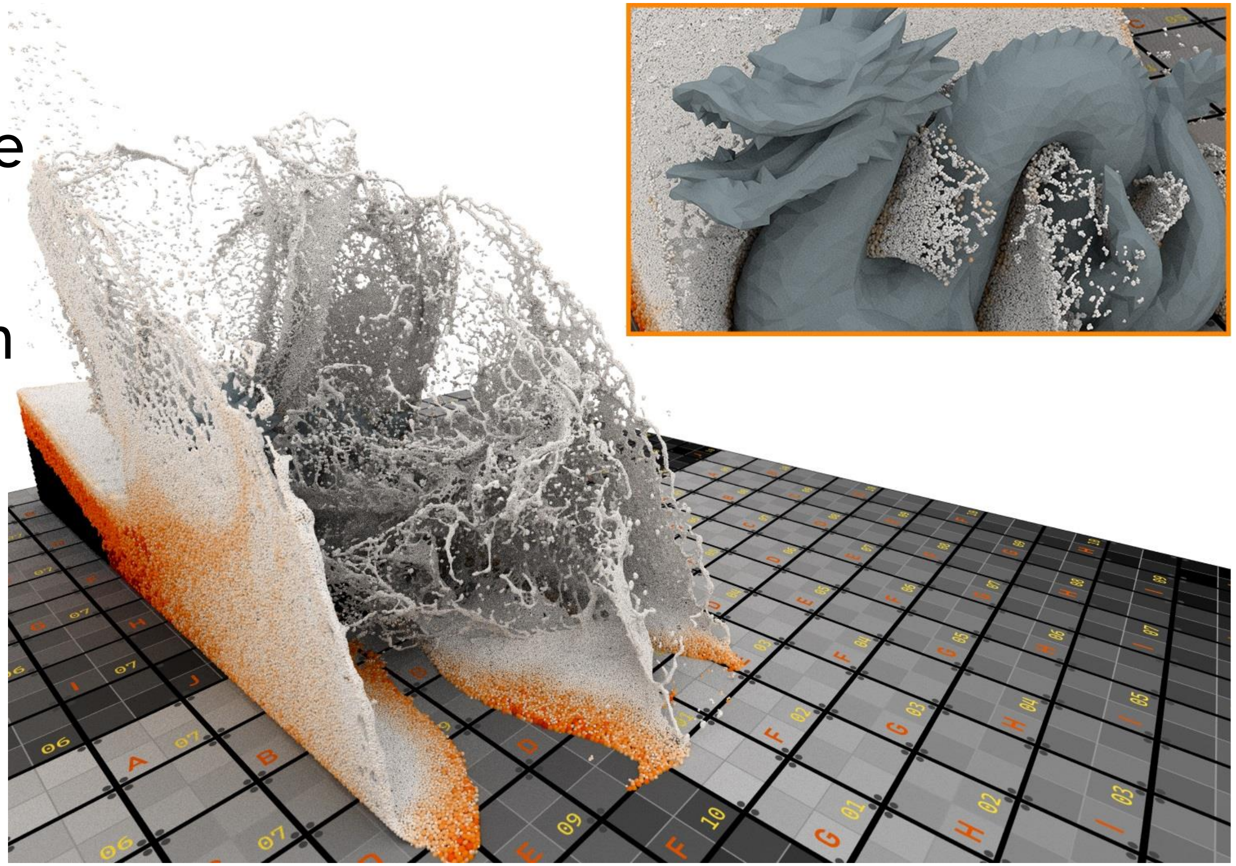


An example adaptive simulation

# Spatial Adaptivity in Smoothed Particle Hydrodynamics

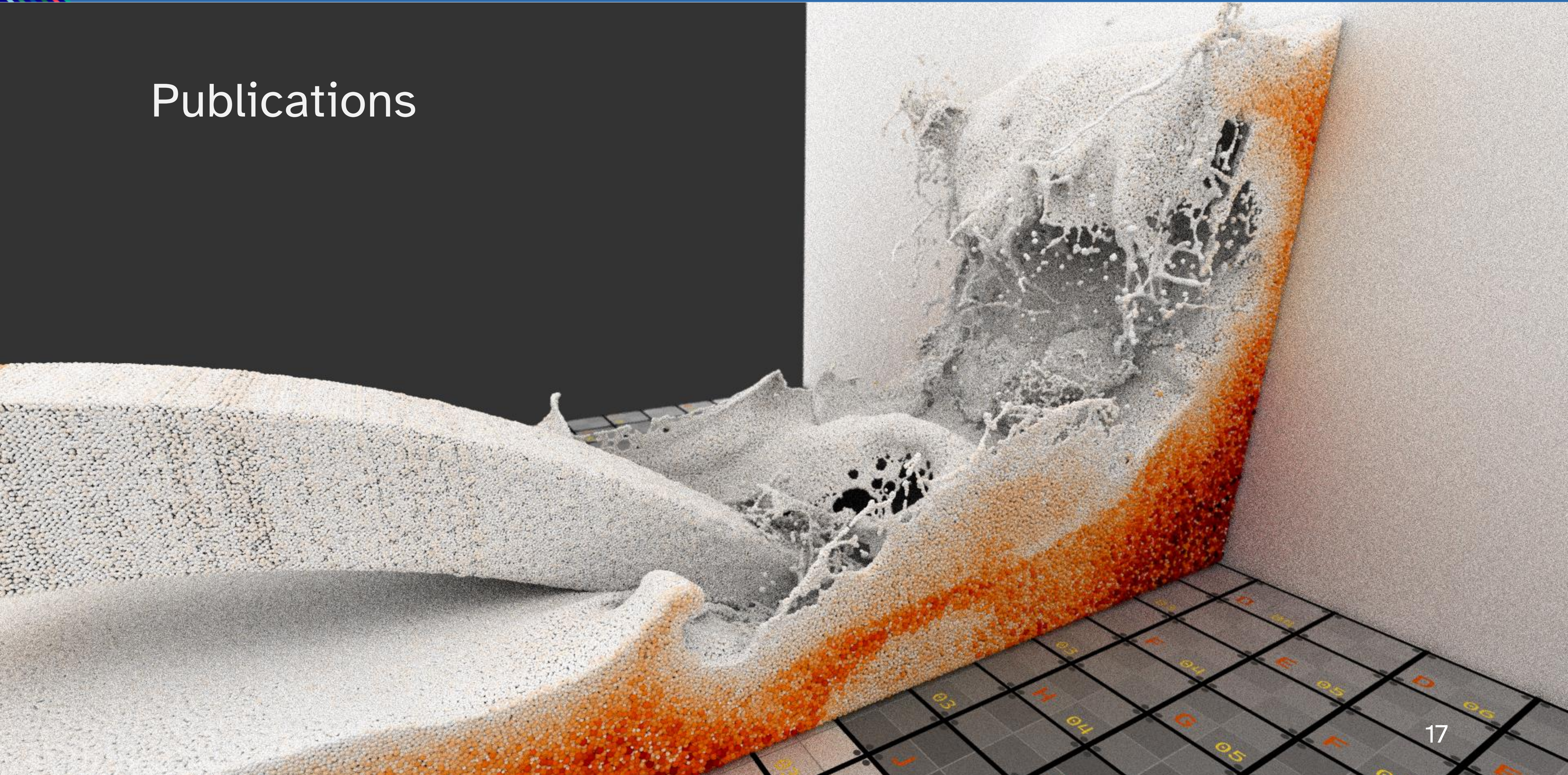
In the research as part of my PhD the following challenges were investigated:

- Determining the ideal size for a particle
- Improving particle refinement processes
- Reducing errors induced by adaptivity
- Boundary handling for uniform and adaptive
- Rendering of spatially adaptive simulations
- Data handling and efficient implementation



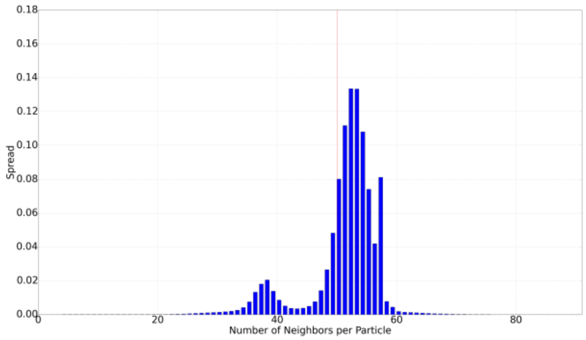
An example adaptive simulation

# Publications

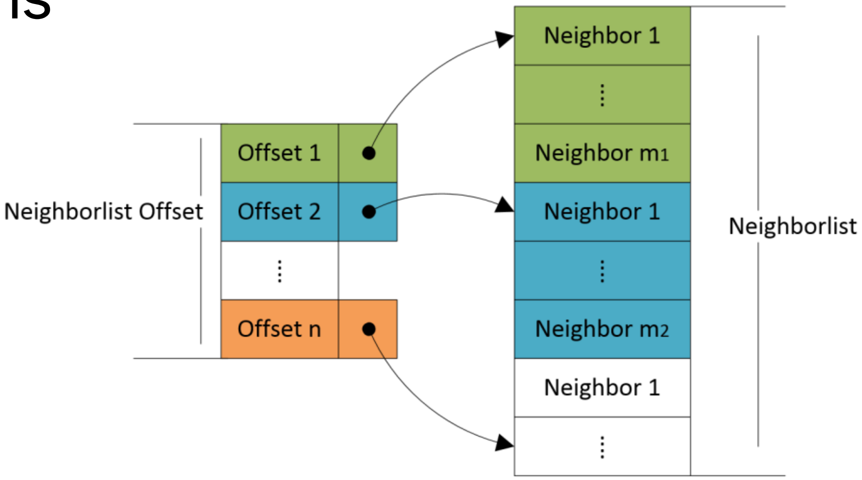


# The problem with neighbors

- SPH scales with $\mathcal{O}(nm)$ where $m$ is the number of neighbors of a particle
- Number of neighbors can vary significantly between particles
- Sequential per-particle lists require $n \cdot m_{avg}$ memory
- Varying numbers of neighbors pose computational challenges:
    1. Bounding the maximum memory consumption
    2. Optimizing access patterns for GPU implementations
- These problems are exacerbated for adaptive simulations



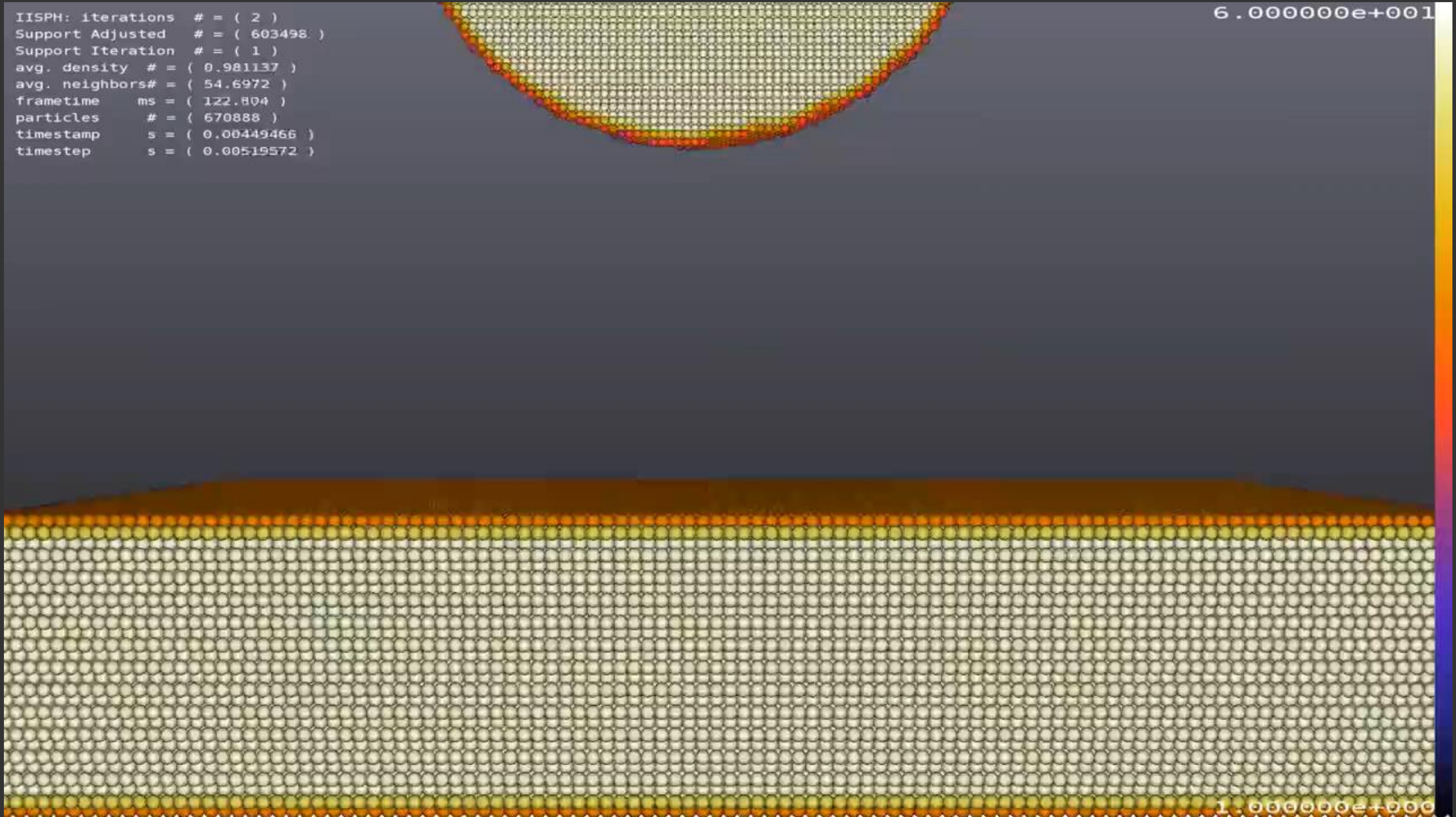

Distribution of neighborhood sizes




Per-particle sequential lists

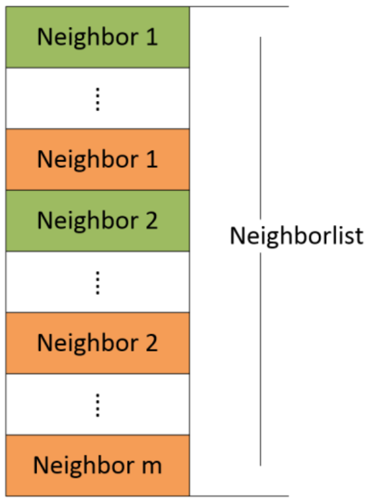
# Constrained Neighbor Lists for SPH-based Fluid Simulations

Authors: **Rene Winchenbach**, Hendrik Hochstetter, Andreas Kolb
Symposium on Computer Animation 2016 - Zurich

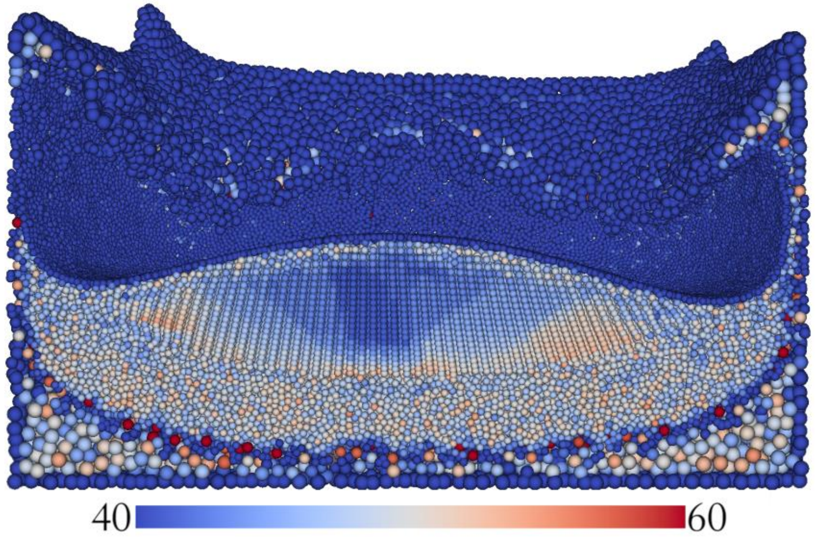


# Constrained Neighbor Lists for SPH-based Fluid Simulations [WHK16]

- Coalesced memory accesses are significantly faster
- Coalesced structures require $n \cdot m_{max}$ entries
- Particle count limited by maximum neighborhood size over simulation
- Particle resolution boundaries have particles with large neighborhoods
- Excluding neighbors arbitrarily leads to significant instabilities
- Instead, reduce support radius of particles to reduce neighbors
- Large reductions yield instabilities, instead:
  - Set $m_{max} \approx 1.1 \cdot N_h$ (for the neighborlist)
  - Use non-coalesced overflow list for violating particles
  - Only if overflow list full support radii will get modified



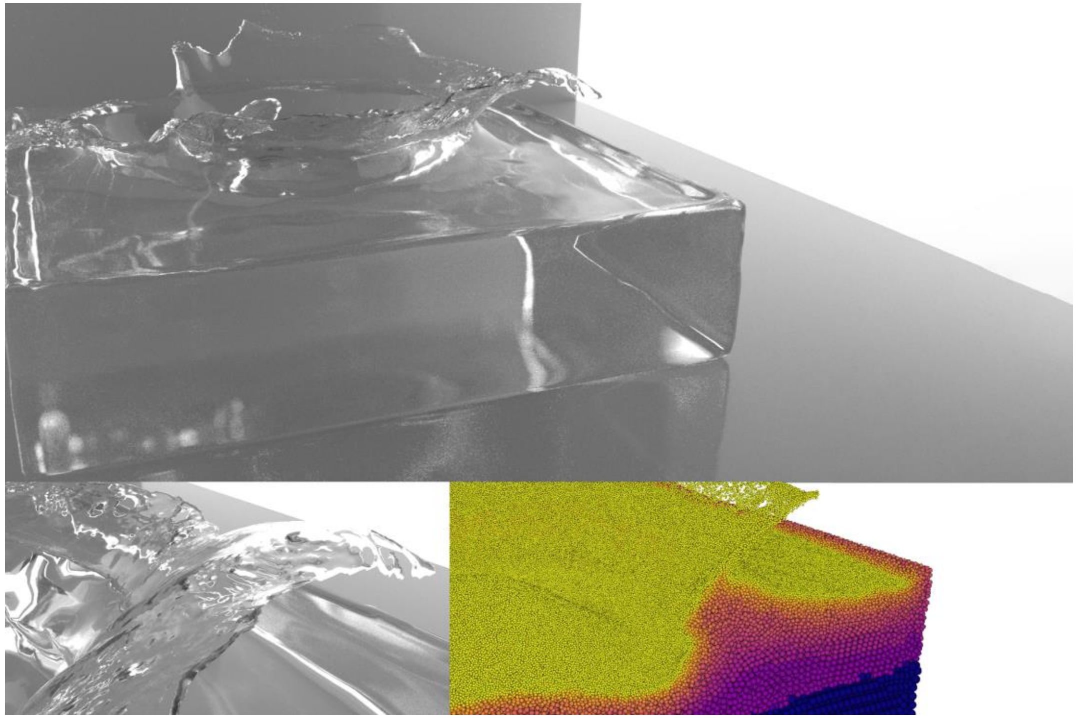

Coalesced memory layout




Particle neighborhood sizes color coded

# Finding interesting places and changing resolution

- Resolution should reflect interest
- How can we define *interest* for a particle?
- Computer Animation concerned with surfaces
- Idea: Use surface distance to determine interest
- Requires continuous changes of resolution
- Mixing different resolution causes instabilities
- Finding partners for merging is challenging
- Temporal blending to reduce errors is inefficient

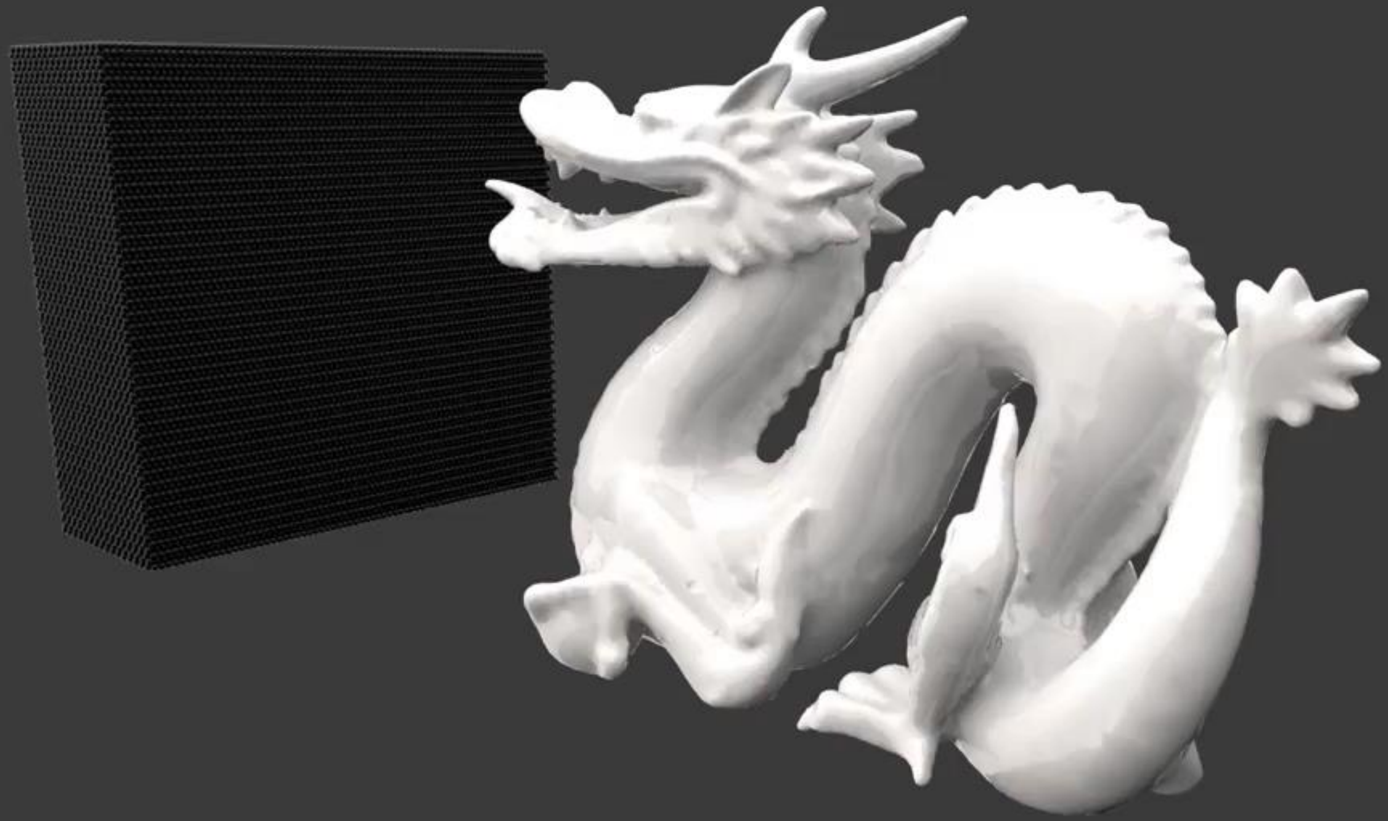


Spatially adaptive simulation, particle volume colored

# Infinite Continuous Adaptivity for Incompressible SPH

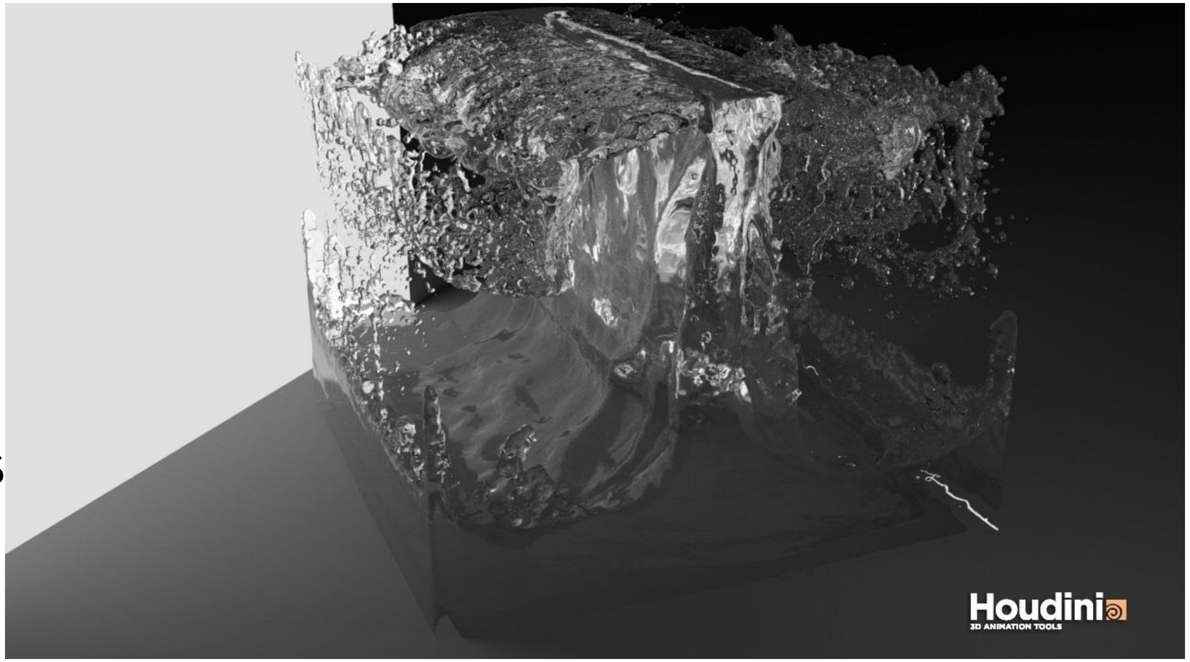
Authors: **Rene Winchenbach**, Hendrik Hochstetter, Andreas Kolb

ACM SIGGRAPH 2017 – Los Angeles



# Infinite Continuous Adaptivity for Incompressible SPH [WHK17]

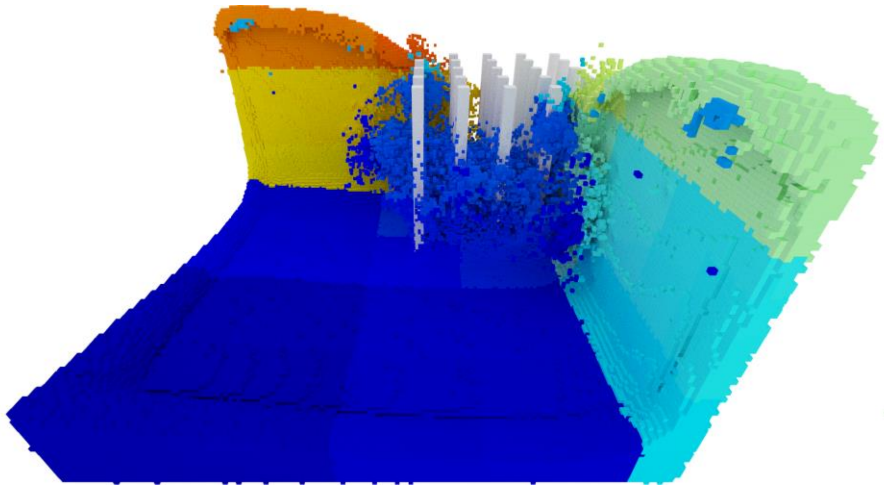
- Use sizing function $S(x)$ to determine ideal size based on linear distance to surface
- Surface distance measure needs to be temporally smooth and stable [BTN13,HS13]
- Classify particles based on size relative to sizing function
- Particle refinement not stable → avoid it where possible
- Sharing splits a particle that is *slightly* too large and merges the excess portion away
- Perform temporal blending with an implicit parent particle after splitting
- Temporal blending only applied to density
- Explicitly model support derivatives to improve simulation stability
- Use global viscosity to reduce residual errors




1000:1 spatially adaptive double dam break

# Particles and where to find them

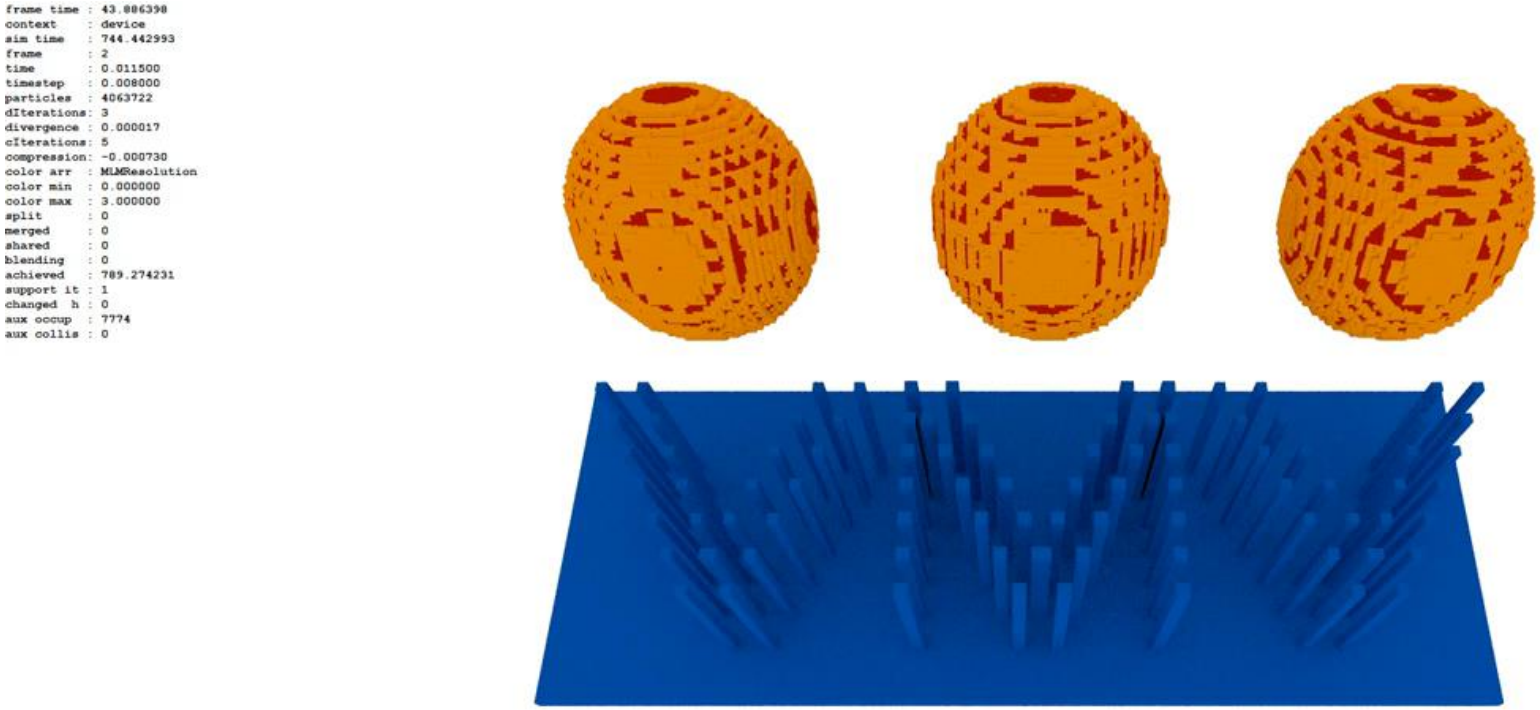
- Finding neighbors using a trivial approach is $\mathcal{O}(n^2)$
- Connecting spatial and memory locations through cells
- Cell size based on support radius
- Average cell contains:
  - For uniform: $\frac{81}{4\pi}N_h$ particles
  - For adaptive: $\alpha\frac{81}{4\pi}N_h$particles
- For $\alpha \gg 100$ finding neighbors is computationally inefficient
- Different cell sizes for different resolutions required
- Creating multiple datastructures expensive
- Dense cell grid scales with domain size and resolution



Occupied cells color coded by spatial index

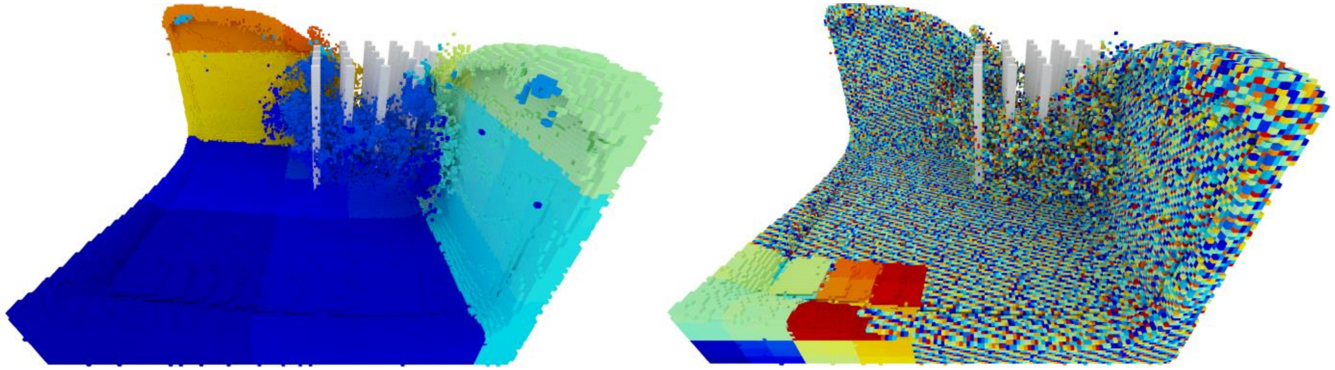
# Multi-Level-Memory Structures for Adaptive SPH Simulations

Authors: **Rene Winchenbach**, Andreas Kolb
Vision Modeling & Visualization 2019 - Rostock

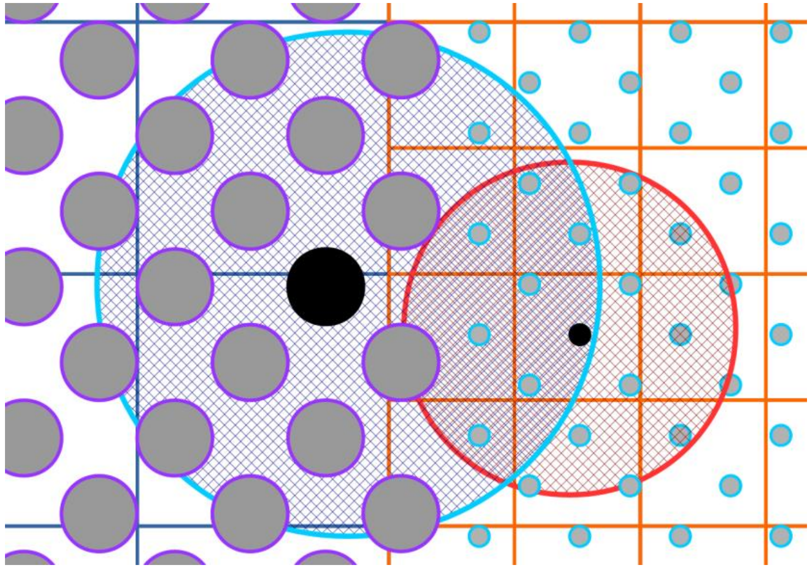


# Multi-Level-Memory Structures for Adaptive SPH Simulations [WK19]

- Use compact hashing to create a three-layered datastructure
    1. Hashmap based on cell location pointing to occupied cells
    2. List of occupied cells sorted by hash and pointing to particles in cell
    3. Particles sorted by Morton code
- Morton code is a self-similar space-filling curve that can be efficiently calculated
- First and second layers can be created for multiples of a base cell size
- Create appropriate structures for different particle resolutions
- Particles assigned to next coarsest level
- Requires symmetrization process to avoid visibility issues



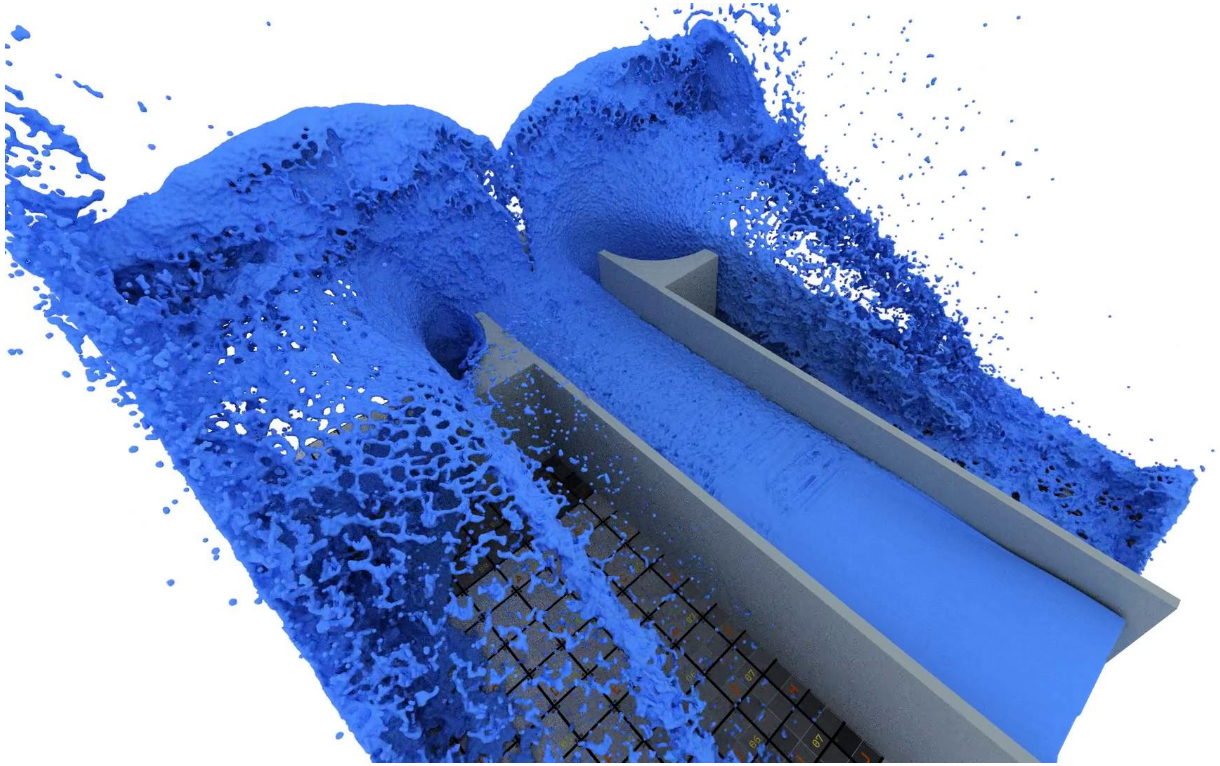
Occupied cells color coded by spatial index and hashed index



Visualization of visibility issues

# Voxel Traversal – or how I learned to stop worrying and love the integer

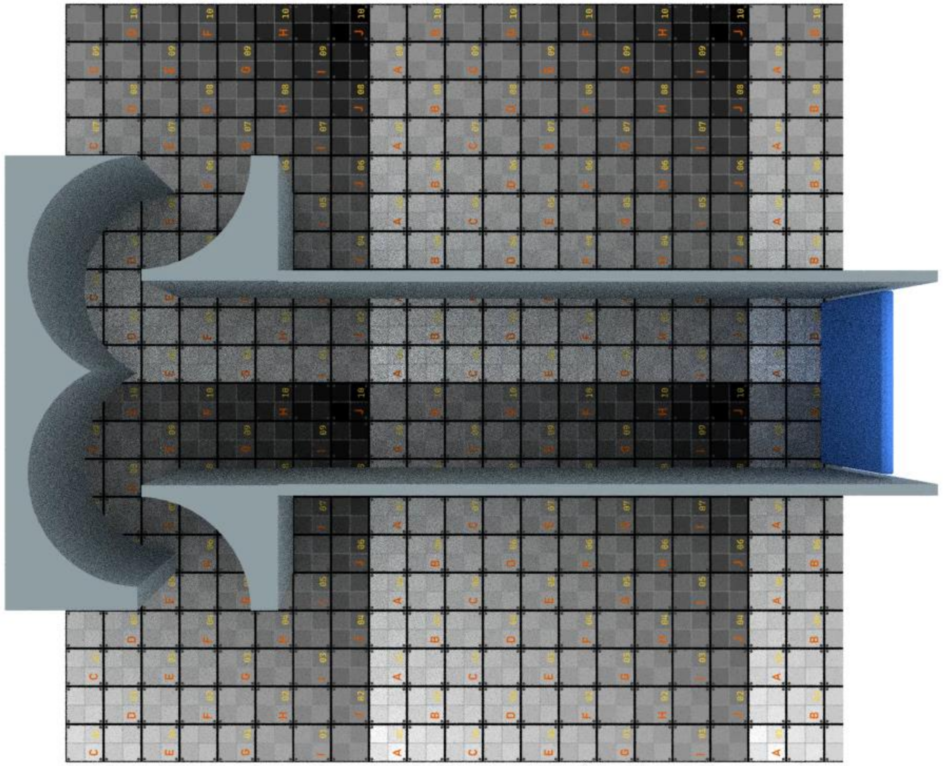
- Rendering fluid simulations is generally relevant in Computer Animation
- Extracting explicit meshes requires significant memory
- On-the-fly ray tracing is memory-efficient
- Reusing existing datastructures preferrable
- Datastructure optimized for simulation
- Anisotropic SPH used for rendering



On-the-fly rendered fluid surface

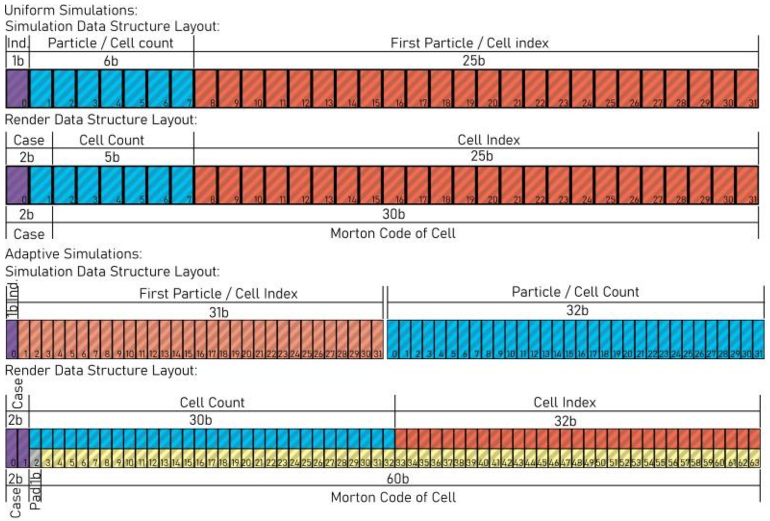
# Multi-Level-Memory Structures for Simulating and Rendering SPH

Authors: **Rene Winchenbach**, Andreas Kolb
Computer Graphics Forum 2020 - Journal

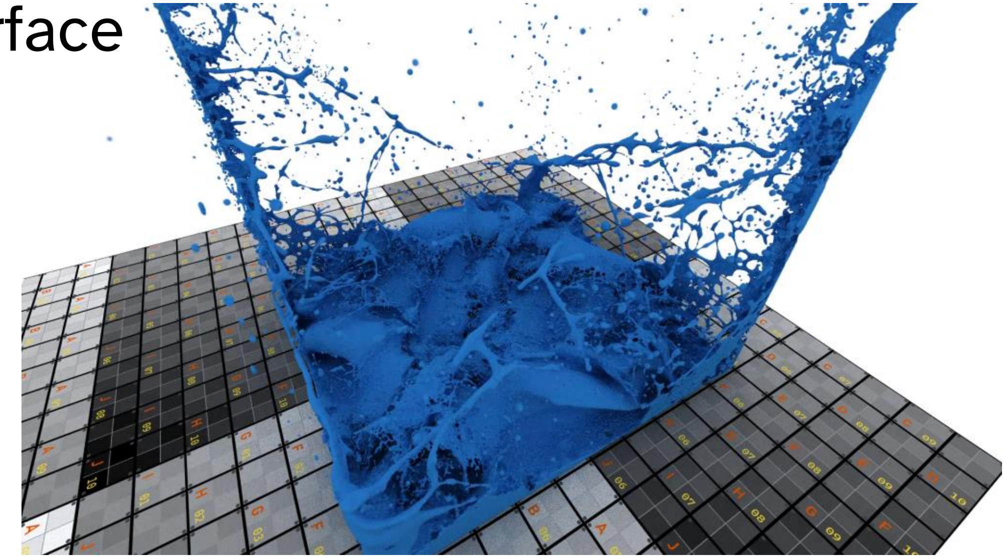


# Multi-Level-Memory Structures for Simulating and Rendering SPH [WK20]

- Rendering and simulation have very different requirements
- However, these are mostly algorithmic
- Reuse the simulation memory for rendering
- Only requires reconstructing layers 1 and 2
- Cells can be treated and iterated like voxels
- Voxel traversal long solved problem
- Auxilliary structure to mark voxels as containing surface
- Cells marked based on dilation of occupied cells
    1. Cells that are empty
    2. Cells that contain surface fluid particles
    3. Cells that contain only bulk fluid particles
    4. Cells that are adjacent to 2
- Anisotropic rendering requires an additional resort



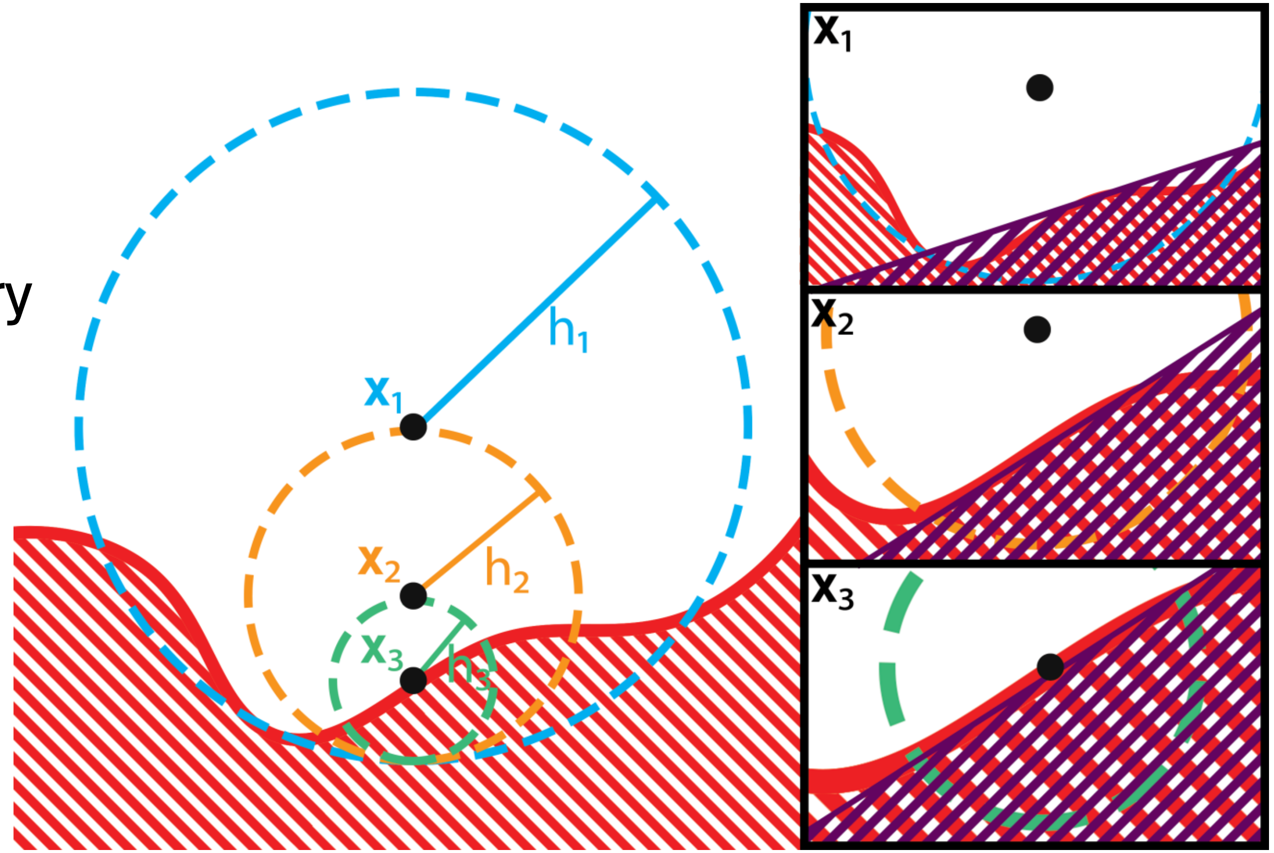

Memory layout diagrams



Rendering of an isotropic fluid surface

# When all you have is an analytic solution, everything is a wall

- Boundary handling for spatial adaptivity is difficult
- Requires consistent behavior across resolutions
- Ideally no re-sampling of the boundary
- Finding analytic solutions is very difficult
- However, special cases can be solved
- For a continuous boundary: the higher the fluid resolution the flatter the boundary
- Idea: Treat all boundaries as flat

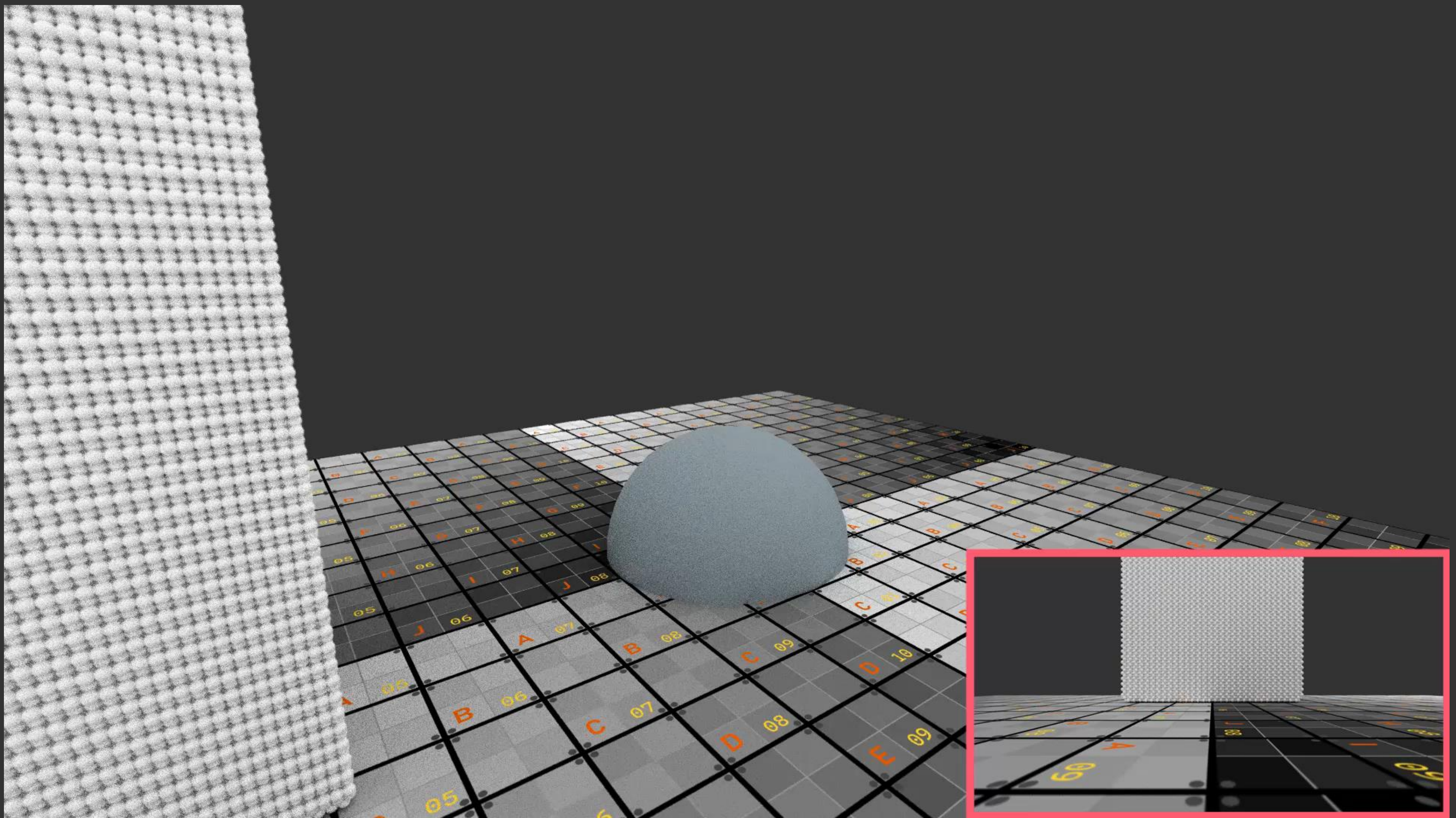



A curved boundary appearing flatter as resolution increases

# Semi-Analytic Boundary Handling Below Particle Resolution for SPH

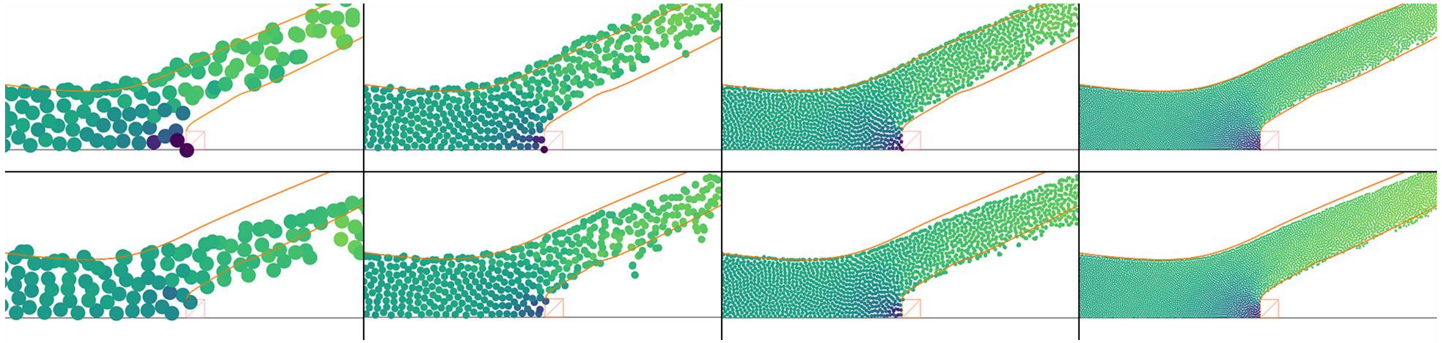
Authors: **Rene Winchenbach**, Rustam Akhunov, Andreas Kolb

ACM SIGGRAPH Asia 2020 - Daegu

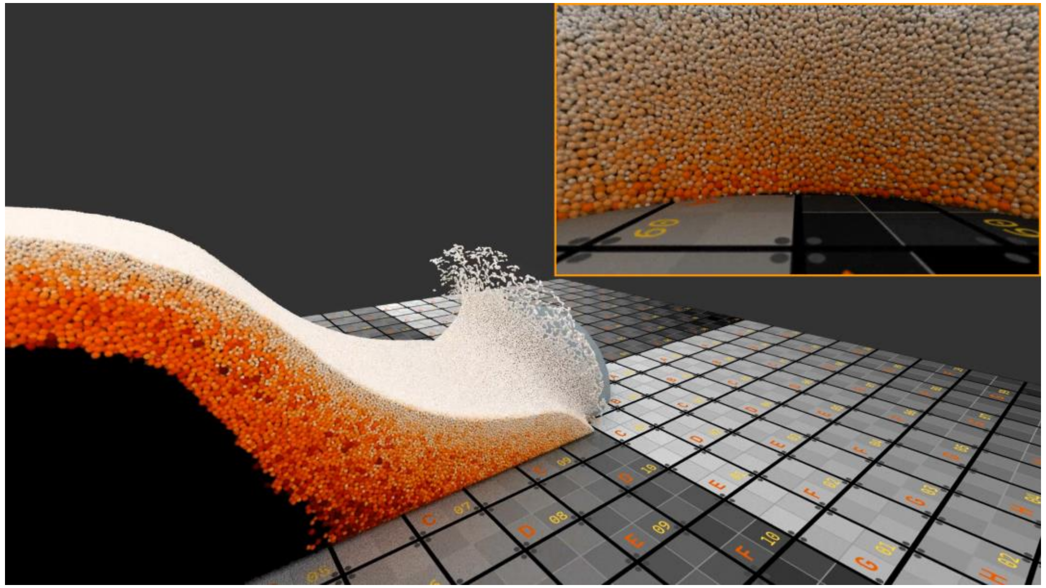


# Semi-Analytic Boundary Handling Below Particle Resolution for SPH [WAK20]

- Planar boundaries can be analytically integrated in 2D and 3D
- Signed Distance Fields yield distance to the closest point on a boundary surface
- Use numerical gradient of SDF to find approximate planar boundary
- This is consistent for different resolutions which yields a consistent boundary shape
- This yields stable and consistent behavior for adaptive resolutions (right)
- This also yields consistent behavior for different fixed resolutions (left)
- Distance and gradient also yield contact point for two-way coupling effects and drag forces



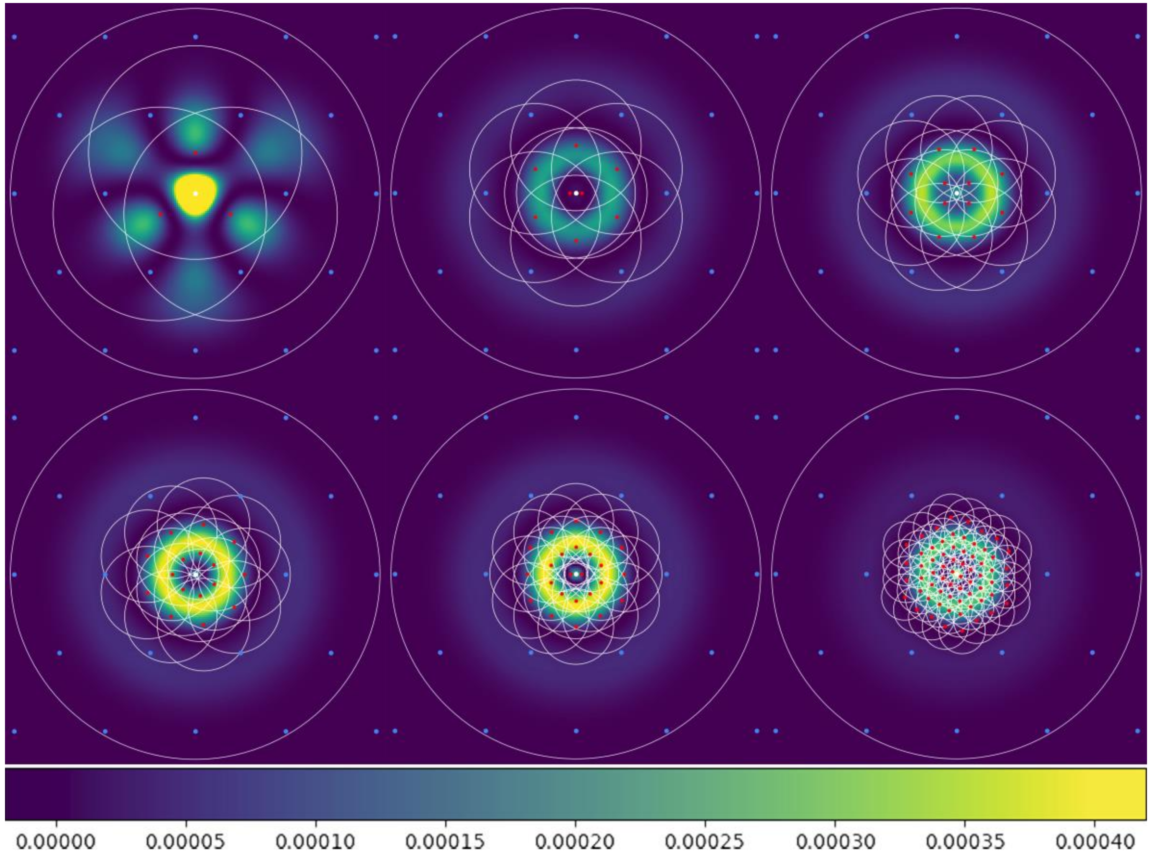
Different fluid resolutions interacting with a fixed boundary, our approach on top



Adaptive simulation with closeup view of the boundary

# Always online: optimizing errors on the go

- Increasing global viscosity to dampen errors is not ideal
- Error damping only applied where necessary
- Errors should be minimized in general
- Refinement requires patterns for replacement
- Patterns should be replicable and optimal
- Particle neighborhood should have influence
- Boundaries should also influence refinement

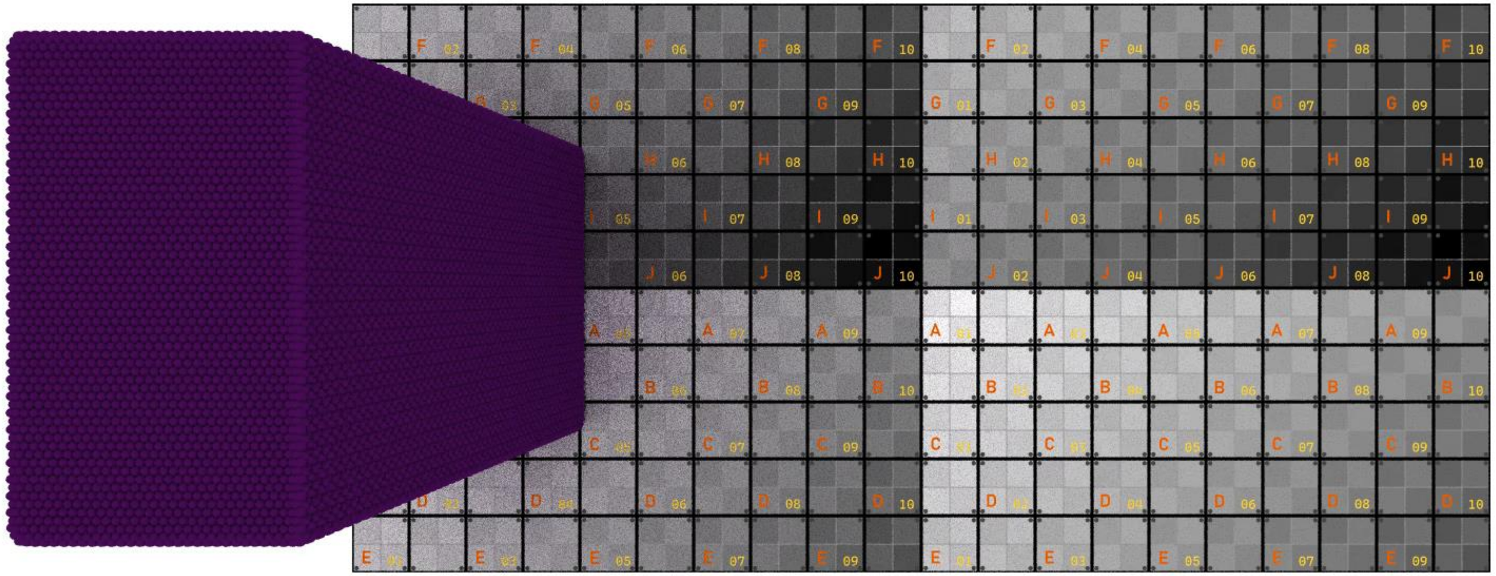



2D refinement patterns generated automatically

# Optimized Refinement for Spatially Adaptive SPH

Authors: **Rene Winchenbach**, Andreas Kolb

ACM Transactions on Graphics 2021 - Journal

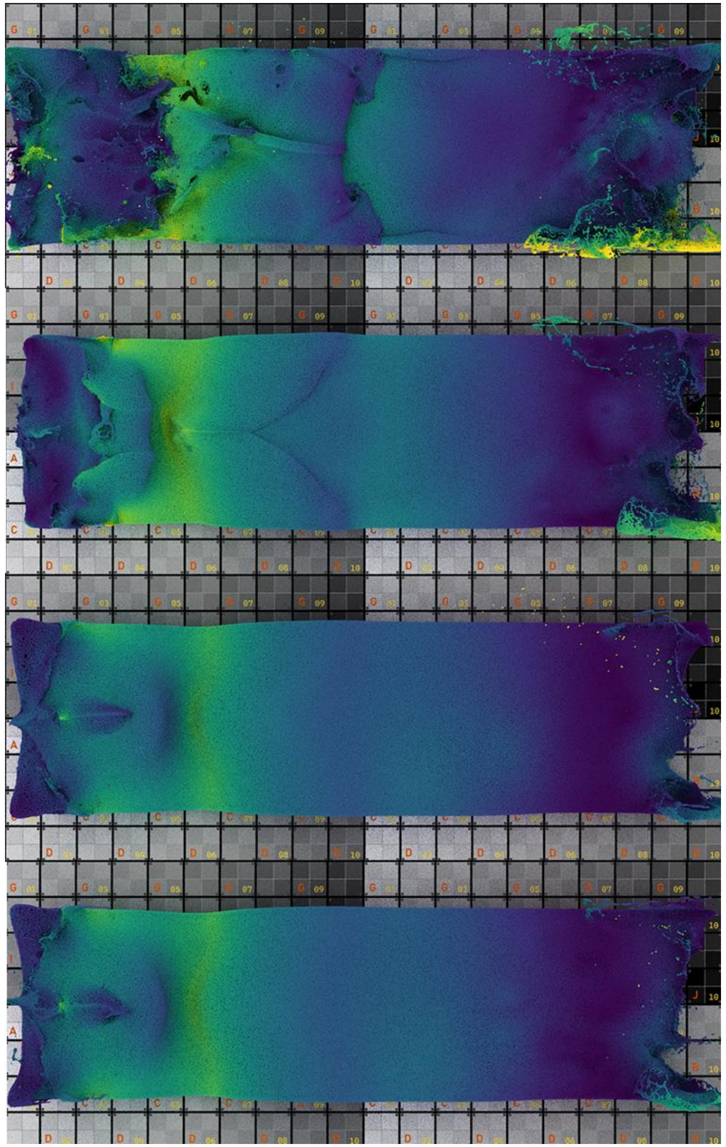


# Optimized Refinement for Spatially Adaptive SPH[WK21]

- Particle refinement can be modeled as divergence-free:

$$\iiint_{\mathbb{R}^3} \|\rho^\star(\boldsymbol{x}') - \rho(\boldsymbol{x}')\| \, d\boldsymbol{x}' = 0$$

- Optimize for positions and masses of inserted particles
- Evaluated only on positions of actual particles for efficiency
- Split optimization before and during a simulation:
  - A priori optimization using LBFGS-B and SLSQP
  - Online during parallel evolutionary optimization on GPU
- However, error never becomes 0
- Implicit temporal blending to dampen errors
- Add local viscosity based on progress of temporal blending
- Online optimization also considers boundary interactions



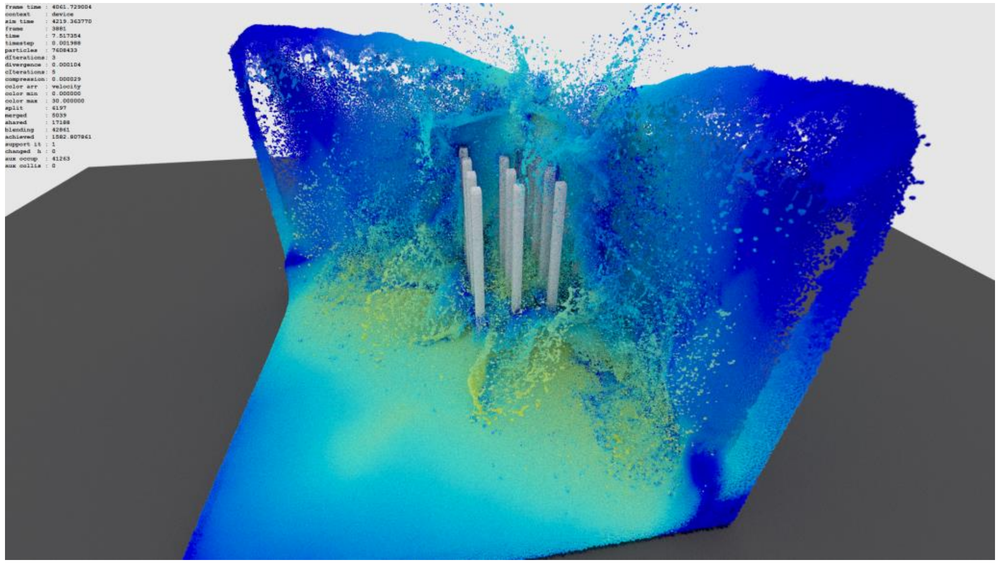
From bottom to top more corrections are applied

# Future Work

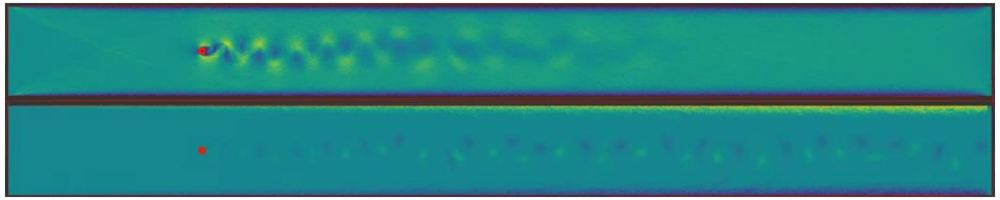
(no result found in terms of standard mathematical functions)

# Finding other interesting things

- Using only the surface distance is not an ideal metric for general simulations:
  - If the surface is smooth then having a high resolution does not improve the results
  - Spray particles don't need to be *fluid* particles to improve the visual appearance
  - Interesting parts of the fluid flow may be inside the bulk (especially in CFD)
- Interest is difficult to describe explicitly
- Changing resolution takes time
- Accordingly, an ideal sizing function would evaluate if a region of the fluid will be interesting *soon*
- Predictions are very difficult to model traditionally
- Idea:
  - Create dataset of different uniform resolution simulations
  - Predict difference in flow based on resolution
  - Use this information to adjust resolution



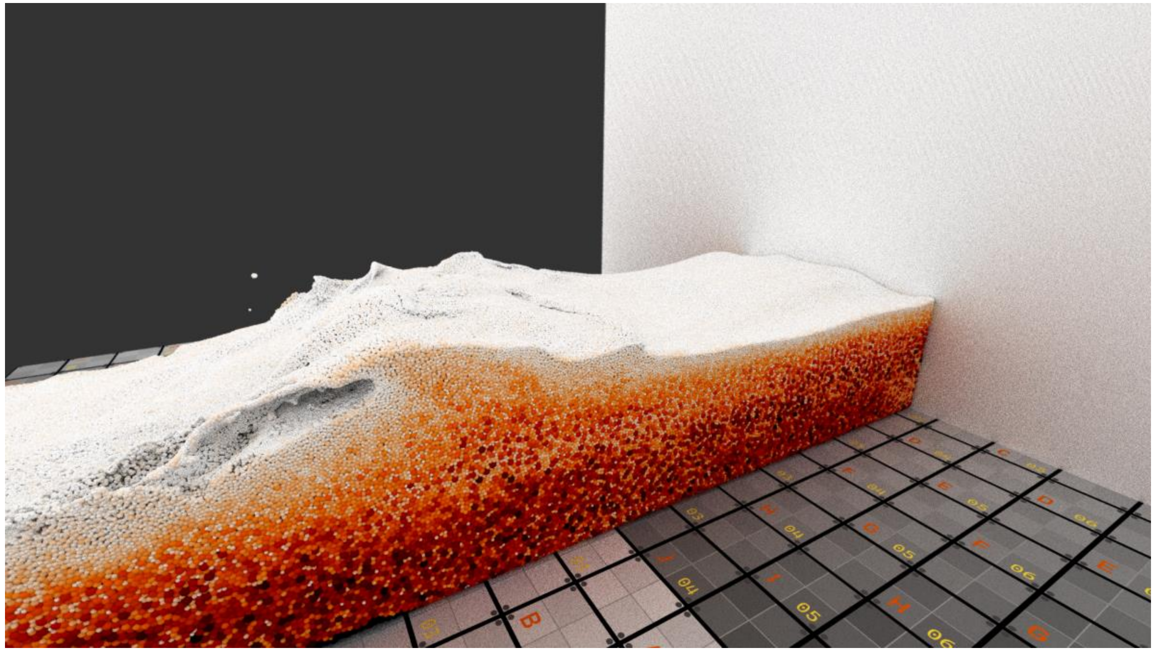
Adaptive Corner Dambreak simulation



Classic uniform flow past cylinder simulation in 2D

# Global resolution changes

- So far changing resolution is only considered to be a local process
- Primary challenge is the optimization algorithm
- Each refined particle involves $5$ degrees of freedom (position, mass, support)
- Ideally larger regions change resolution simultaneously to match sizing function
- Constrained optimization of many particles:
  - Obey conservation of mass
  - Obey conservation of energy and momentum
  - Preserve the underlying scalar fields
- Traditional optimization algorithms are reaching their practical limitations



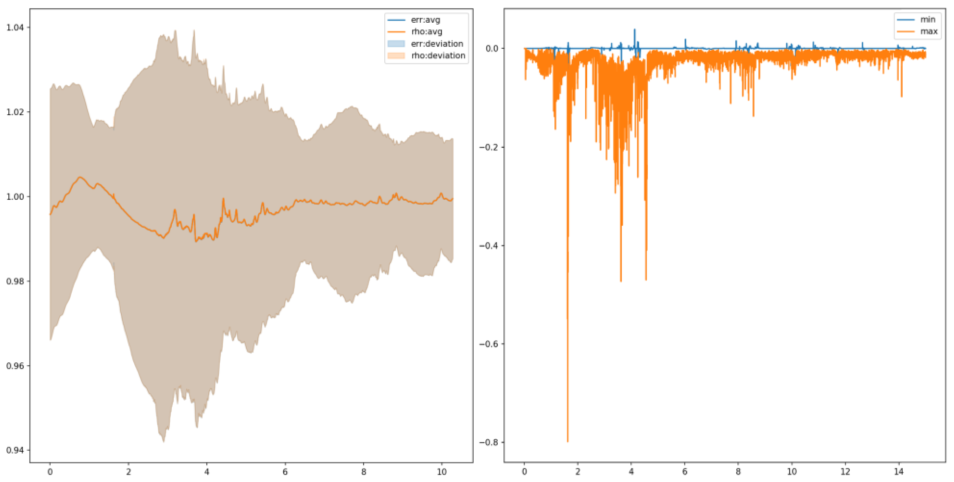
Adaptive simulation with non-smooth resolution gradient

# Looking ahead: Beyond the Taylor Series

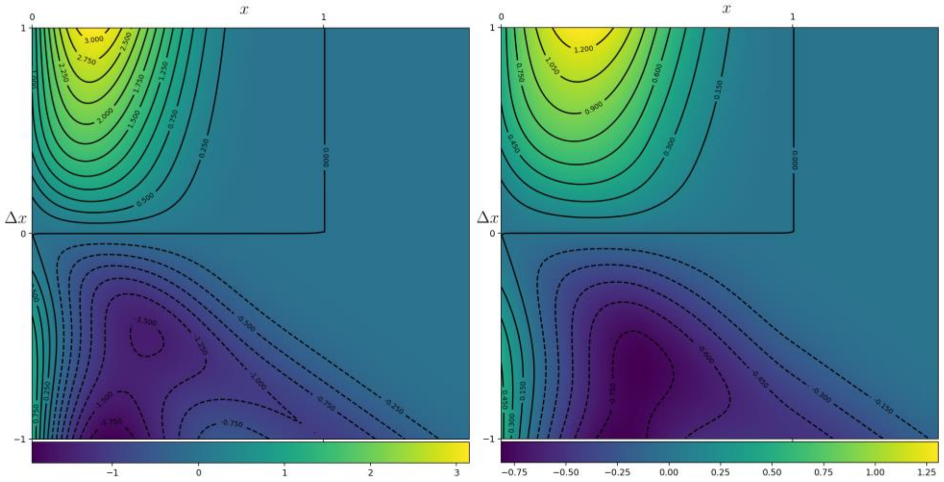
- Learning useful for predictions
- SPH relies on a prediction for incompressibility:

$$\rho_i^{t+\Delta t} = \rho_i^t + \Delta t \sum_j m_j \boldsymbol{v}_{ij}^{t+\Delta t} \cdot \nabla_i W_{ij}$$

- Accuracy of this estimate is essential
- However, the compact nature of the kernel and the first order approximation can yield significant errors
- Maximally observed error about 60%
- Modelling the effects explicitly impractical:
  - Requires information from non-neighboring particles
  - Requires higher order explicit prediction schemes
  - Requires non-compact kernels for continuity in prediction
- Estimate error via learning, keep the solver[Um+20]



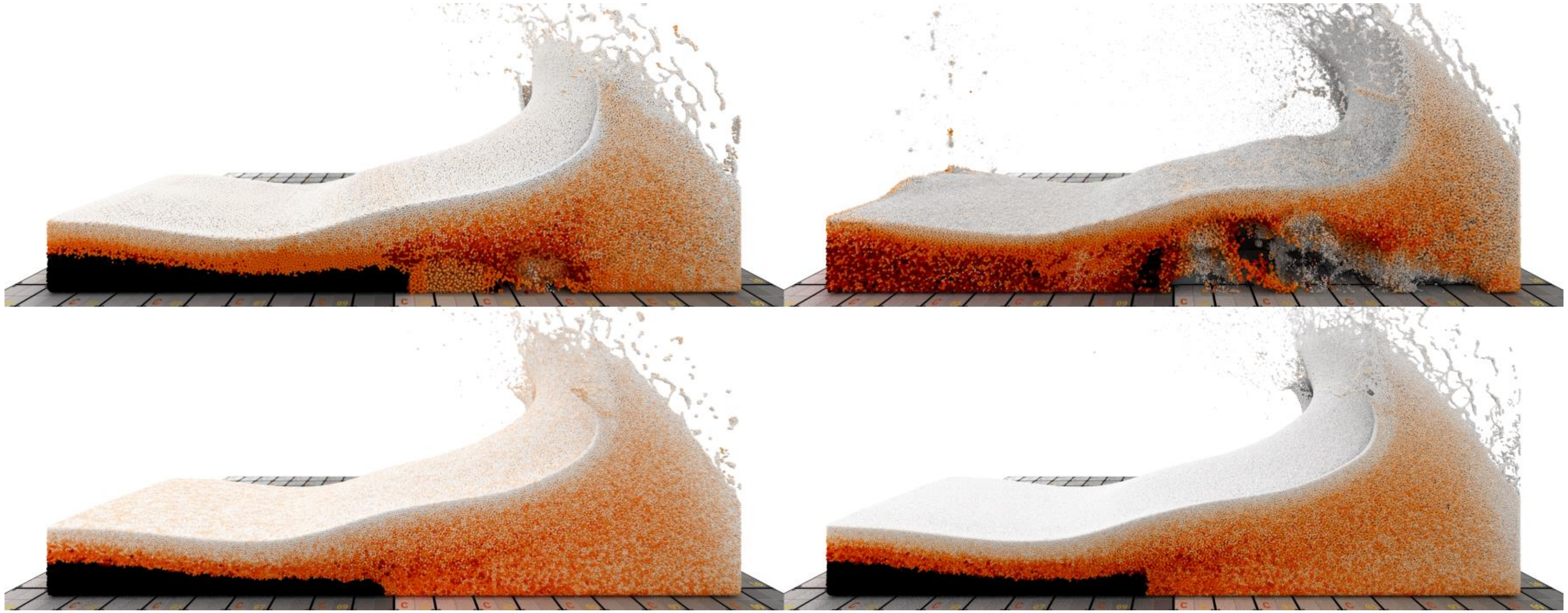

Average density and error of a simulation (left) minimum and maximum difference (right)




Evaluation of the kernel Taylor series error

# Conclusions

- Spatially adaptive methods have been significantly improved in recent research
- Several problems still exist to further improve the simulation models
- Explicit evaluations of these problems is not practical, generating data is not
- Use data-driven approaches to improve existing models
- Intuitive models are not always replicable, stable or reliable



Unstable (top) simulations using intuition were improved using intuition-free optimization (below)

# Bibliography

[GM77] Robert A. Gingold and Joseph J. Monaghan. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: Monthly Notices of the Royal Astronomical Society181.3 (1977),pp. 375–389
[Par62] Emanuel Parzen. “On estimation of a probability density function and mode”. In: The Annals of Mathematical Statistics33.3 (1962),pp. 1065–1076
[Kos+19] Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. “Smoothed Particle Hydrodynamics Techniques for thePhysics Based Simulation of Fluids and Solids”. In: Eurographics 2019- Tutorials. The Eurographics Association, 2019, pp. 1–41
[DA12] Walter Dehnen and Hossam Aly. “Improving convergence in smoothed particle hydrodynamics simulations without pairing instability”. In: Monthly Notices of the Royal Astronomical Society425.2 (2012),pp. 1068–1082
[Mon05] Joseph J. Monaghan. “Smoothed particle hydrodynamics”. In: Reports on progress in physics68.8 (2005), p. 1703 |
[Pri12] Daniel J. Price. “Smoothed particle hydrodynamics and magneto-hydrodynamics”. In: Journal of Computational Physics231.3 (2012),pp. 759–794
[Gis+19] Christoph Gissler, Andreas Peer, Stefan Band, Jan Bender, and Matthias Teschner. “Interlinked SPH Pressure Solvers for Strong Fluid-Rigid Coupling”. In: ACM Transactions on Graphics (TOG)38.1(2019), p. 5
[Ben+19] Jan Bender, Tassilo Kugelstadt, Marcel Weiler, and Dan Koschier. “Volume Maps: An Implicit Boundary Representation for SPH”. In: Motion, Interaction and Games. ACM. 2019, p. 26
[Ihm+13] Markus Ihmsen, Jens Cornelis, Barbara Solenthaler, Christopher Horvath, and Matthias Teschner. “Implicit incompressible SPH”. In: IEEE Transactions on Visualization and Computer Graphics20.3 (2013),pp. 426–435
[BK15] Jan Bender and Dan Koschier. “Divergence-free smoothed particle hydrodynamics”. In: Proceedings of the 14th ACM SIG-GRAPH/Eurographics Symposium on Computer Animation. 2015,pp. 147–155
[Ban+18a] Stefan Band, Christoph Gissler, Markus Ihmsen, Jens Cornelis, Andreas Peer, and Matthias Teschner. “Pressure boundaries for implicit incompressible SPH”. In: ACM Transactions on Graphics (TOG)37.2(2018), p. 14
[UHT17] Kiwon Um, Xiangyu Hu, and Nils Thuerey. “Perceptual evaluation of liquid simulation methods”. In: ACM Transactions on Graphics (TOG)36.4 (2017), p. 143
[Thu+20] Nils Thuerey, K. Weissenow, L Prantl and Xiangyu Hu. “Deep Learning Methods for Reynolds-Averaged Navier Stokes Simulations of Airflow Flows”. In: AIAA Journal 58.1 (2020)
[Um+20] Kiwon Um, Raymond Fei, Robert Brand, Philipp Holl and Nils Thuerey. “Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers”. In: Advances in Neural Information Processing (NeurIPS) (2020)
[WHK16] Rene Winchenbach, Hendrik Hochstetter, and Andreas Kolb. “Con-strained Neighbor Lists for SPH-based Fluid Simulations”. In: Proceedings of the 15th ACM SIGGRAPH/Eurographics symposium on computer animation. July 2016
[WHK17] Rene Winchenbach, Hendrik Hochstetter, and Andreas Kolb. “Infinite Continuous Adaptivity for Incompressible SPH”. In: ACM Transactions on Graphics (TOG)36.4 (2017), 102:1–102:1|
[WK19] Rene Winchenbach and Andreas Kolb. “Multi-Level-Memory Structures for Adaptive SPH Simulations”. In: (2019)
[WK20] Rene Winchenbach and Andreas Kolb. “Multi-Level Memory Structures for Simulating and Rendering Smoothed Particle Hydrodynamics”. In: Computer Graphics Forum39.6 (2020), pp. 527–541
[WK21] Rene Winchenbach and Andreas Kolb. “Optimized Refinement for Spatially Adaptive SPH”. In: ACM Transactions on Graphics (TOG)40.1(Jan. 2021)

# The end

Thank you for your attention