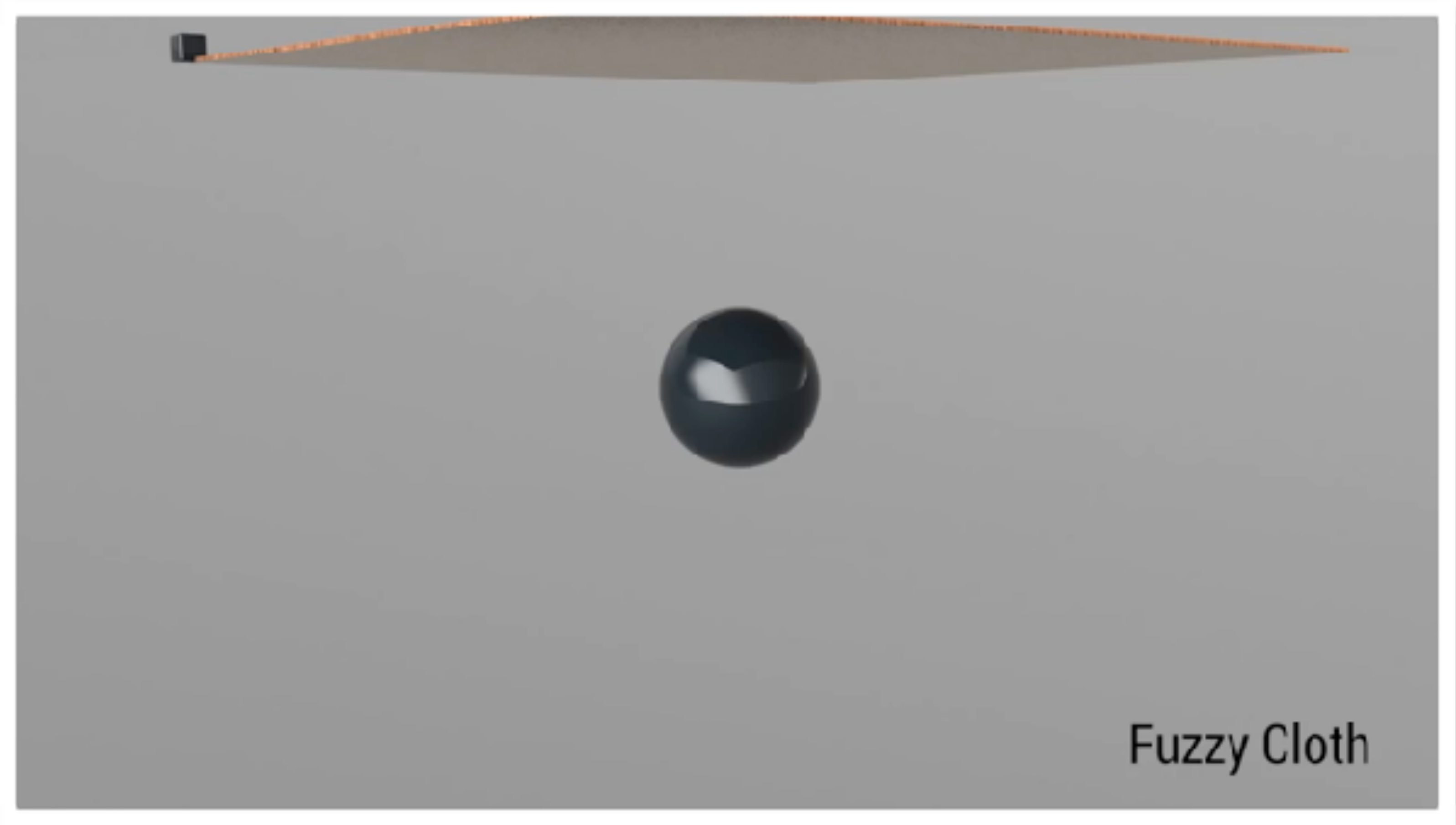
Sparse Paged Grid and its Applications to Adaptivity and Material Point Method

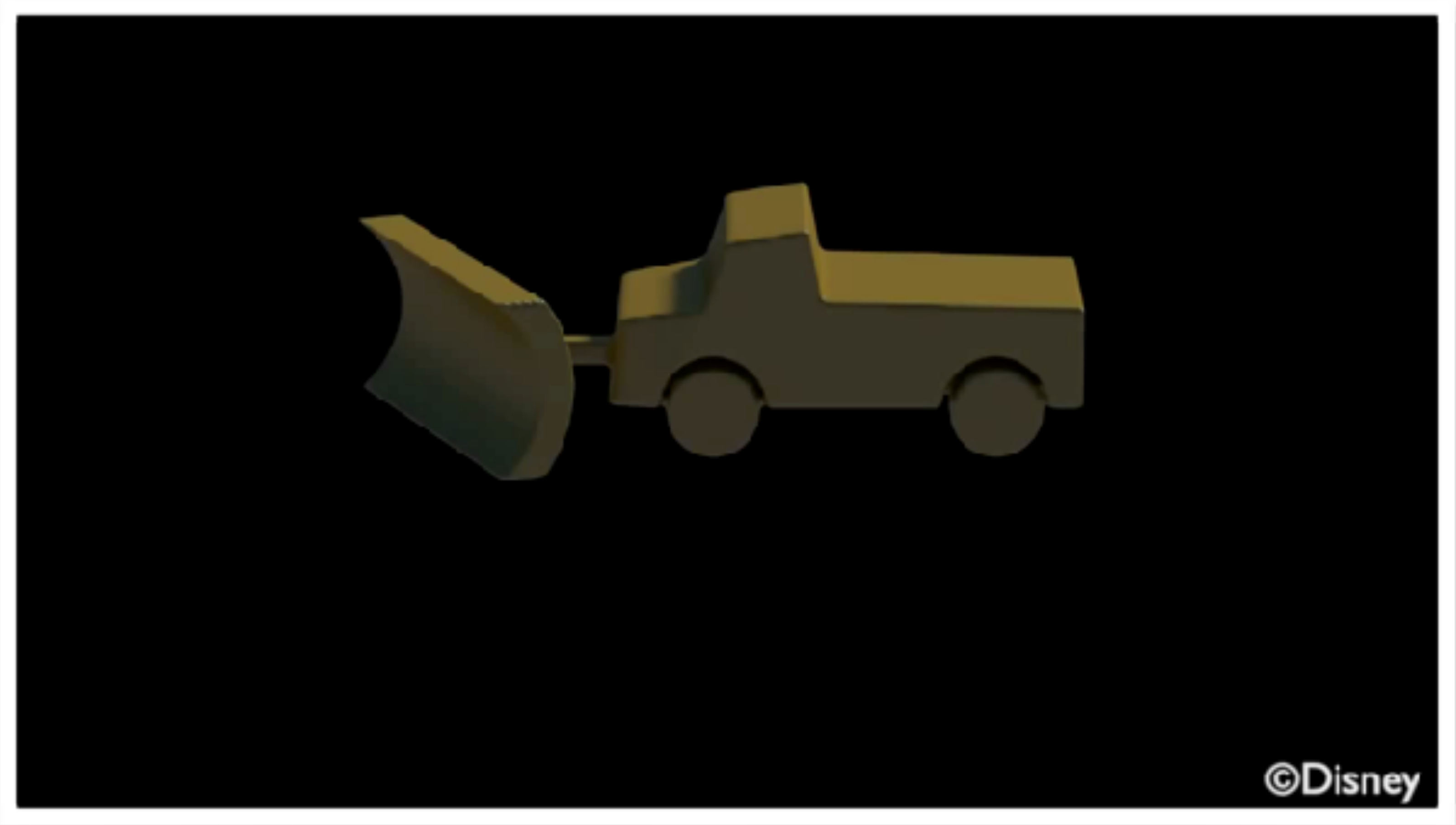
Ming Gao

Motivation - expanding feature set

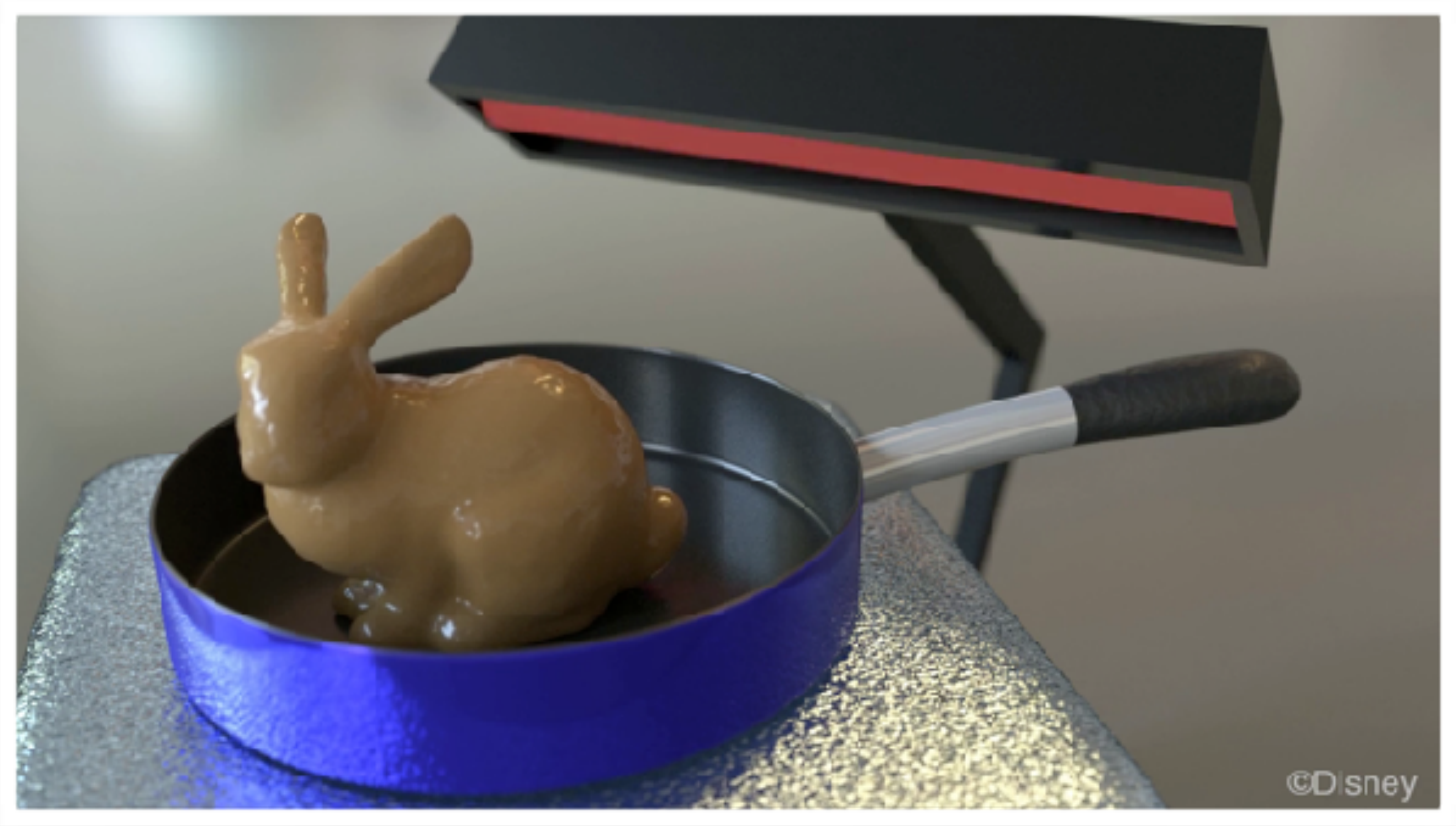


Wet cloth - [Fei et al. 18]

Viscous fluid - [Larionov et al. 17]

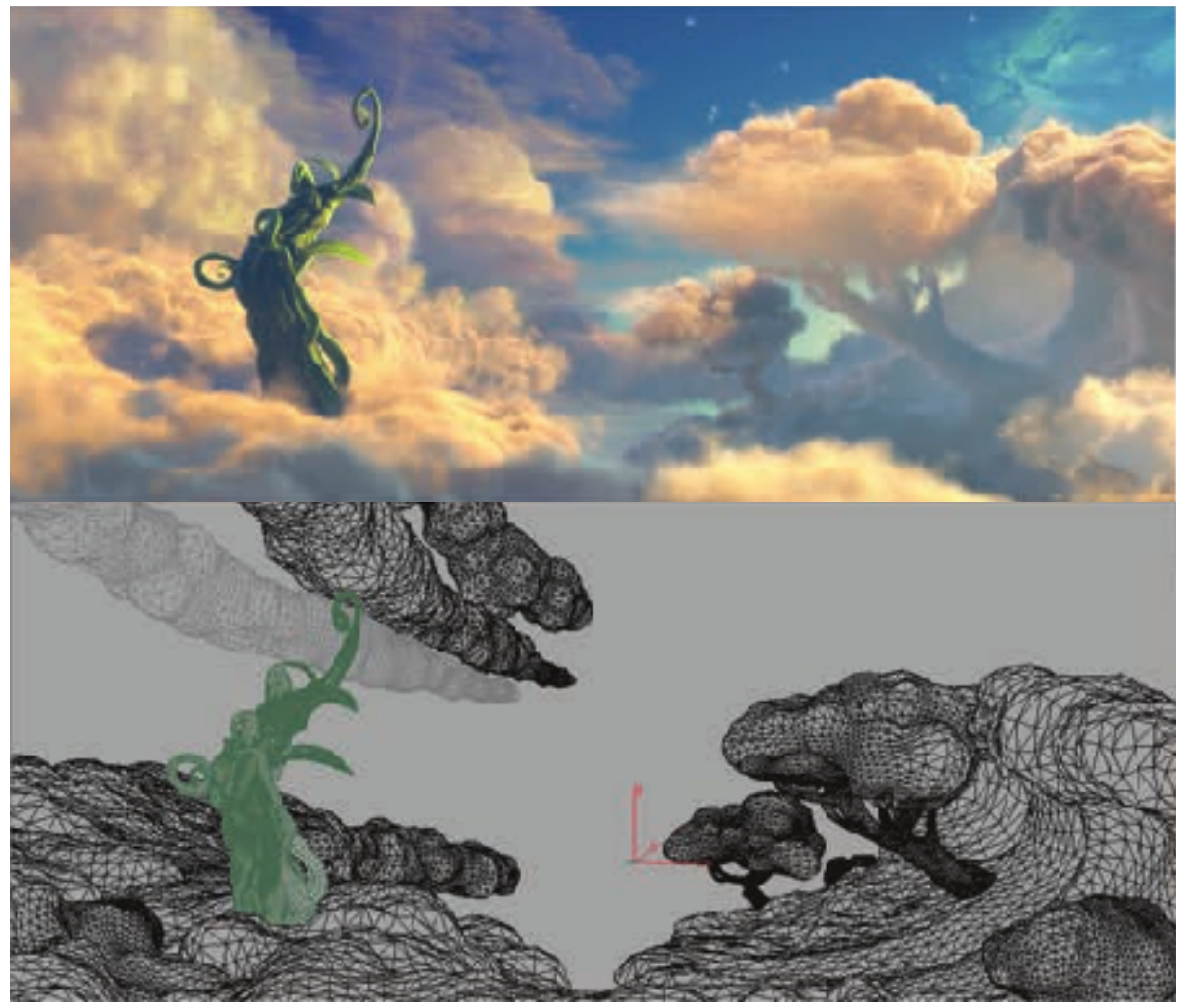


Snow - [Stomakhin et al. 13]



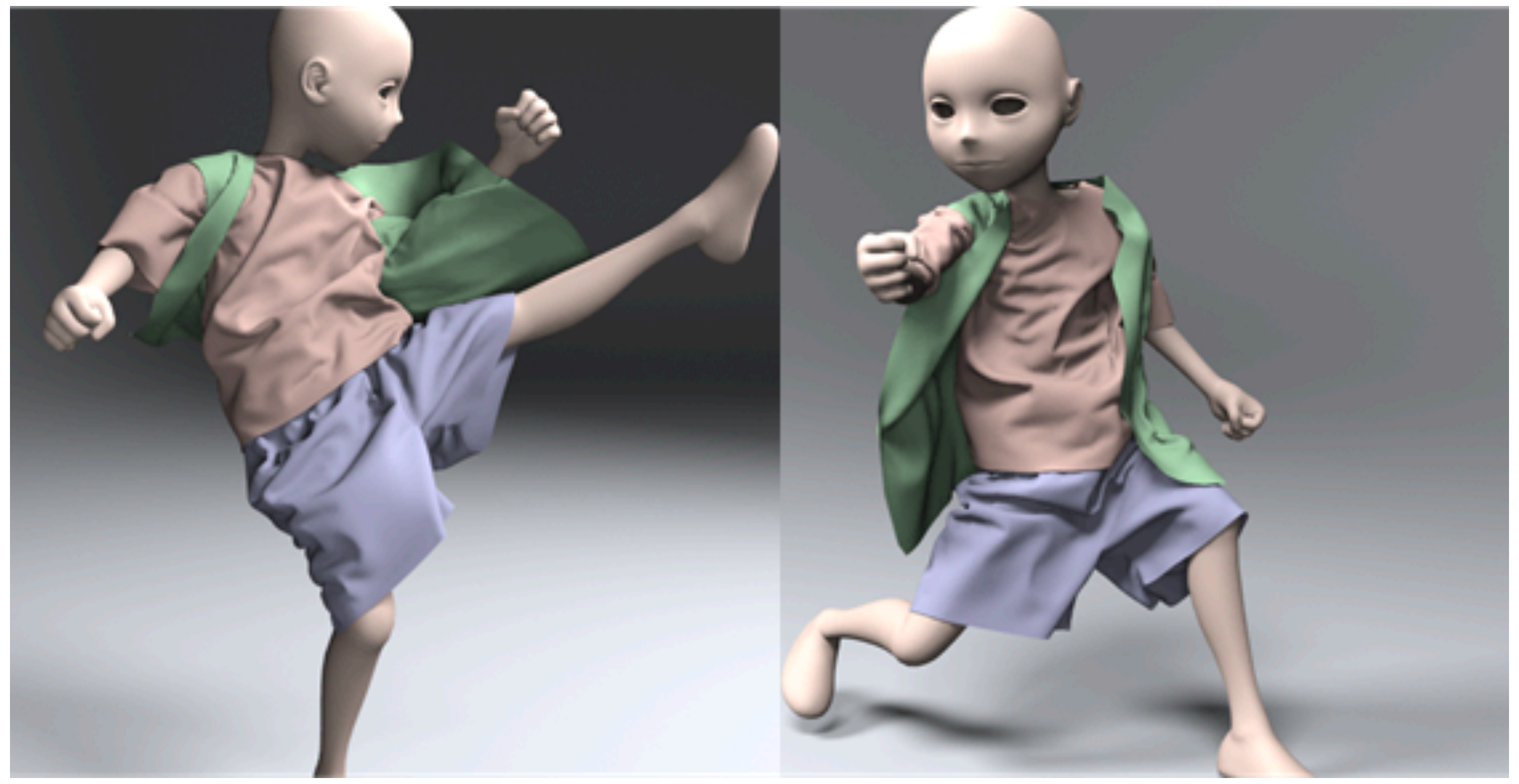
Melting - [Stomakhin et al. 14]

Motivation - accelerating performance

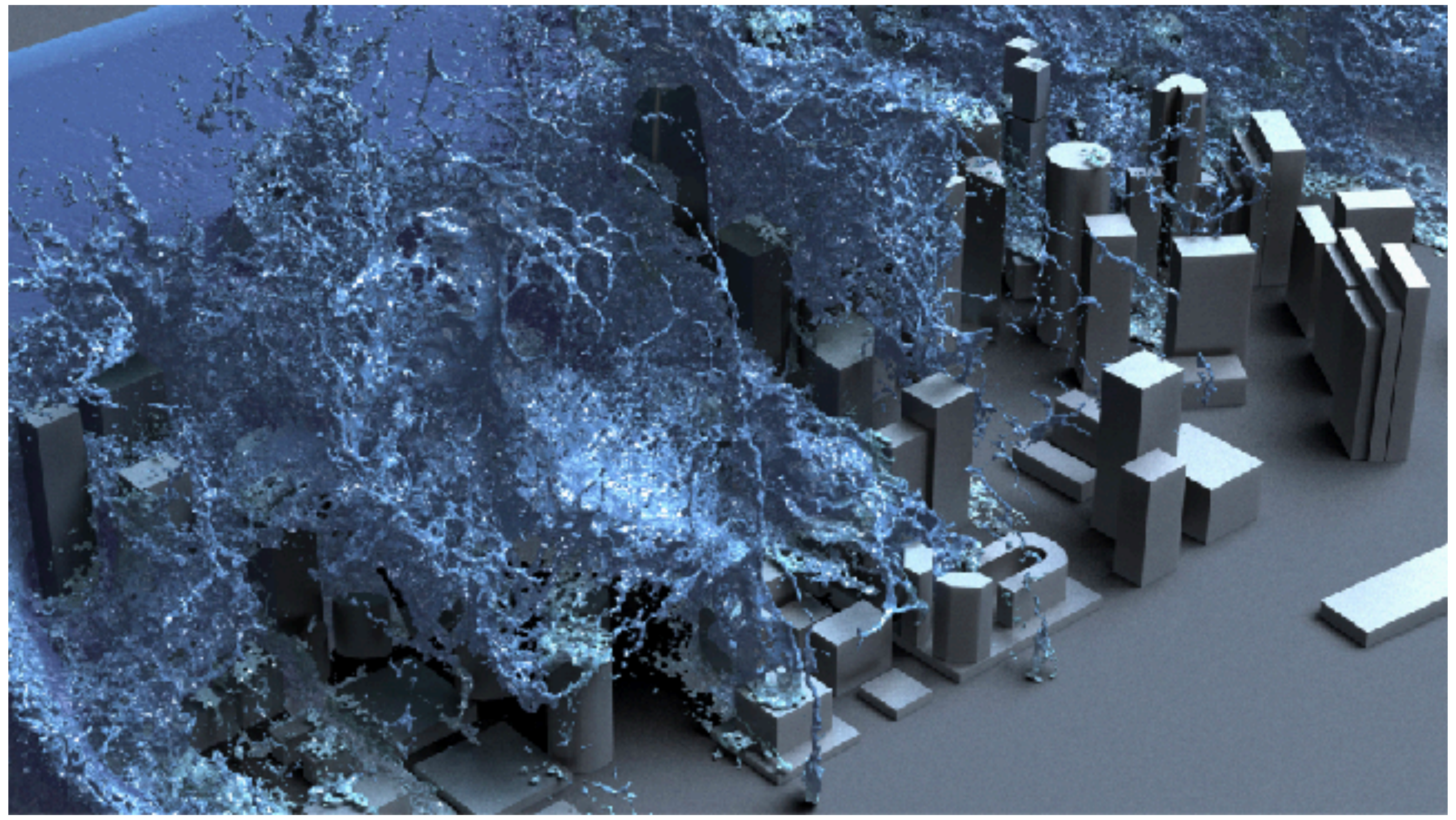


OpenVDB - [Museth et al. 13]

PhysGrid - [Milne et al. 16]

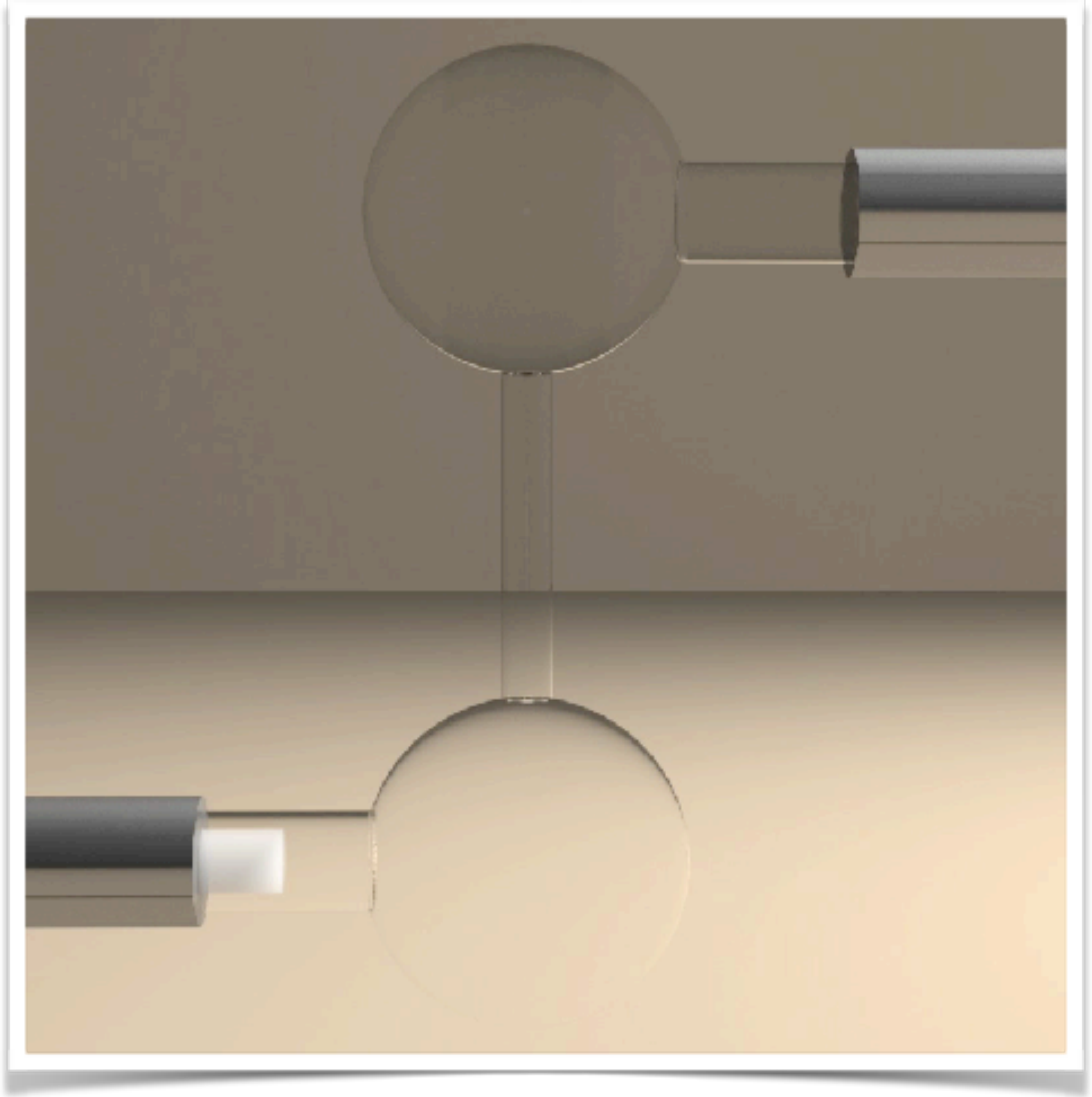


Cloth - [Tang et al. 16]



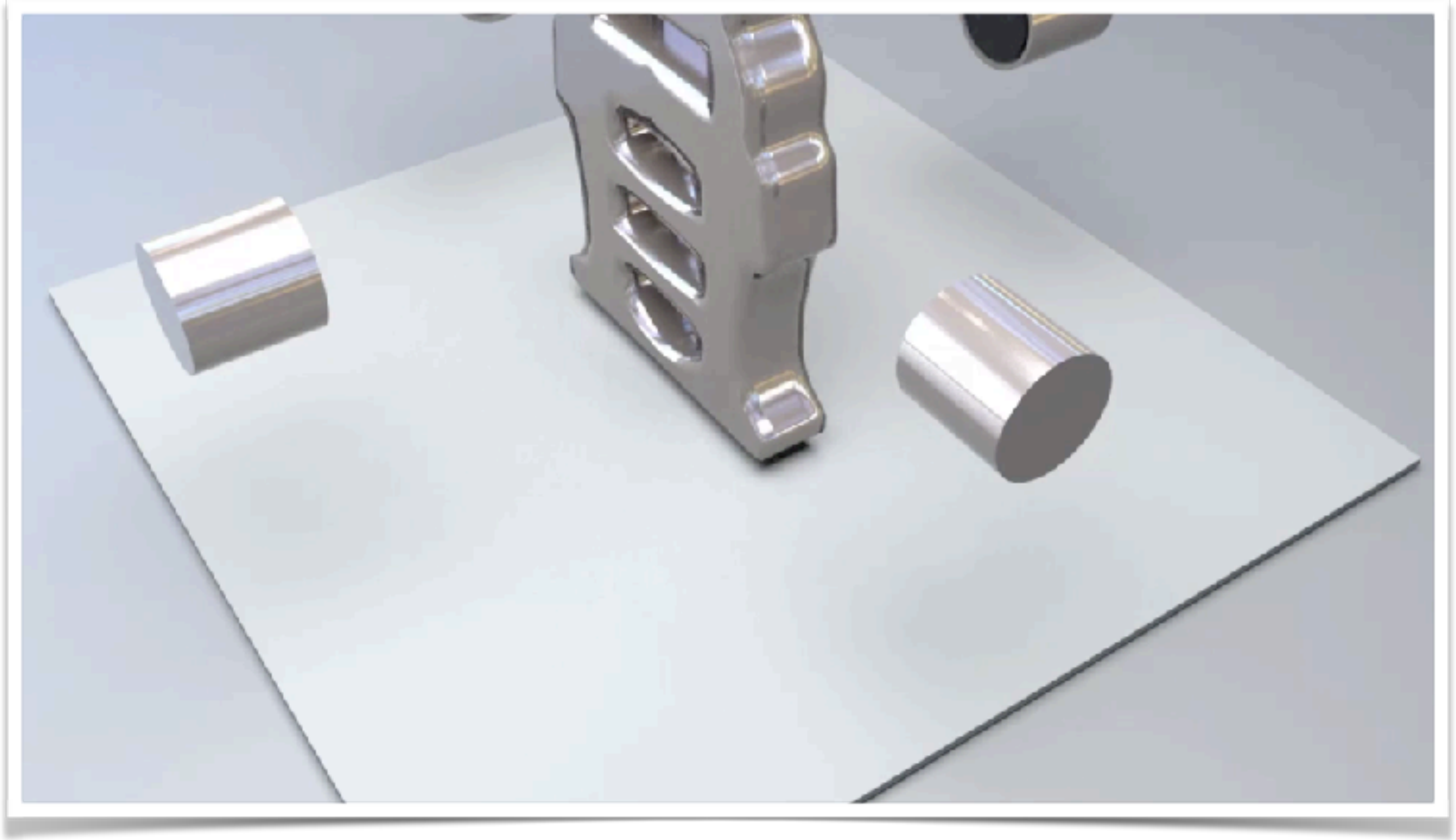
Fluid - [Wu et al. 18]

Feature breadth vs. optimal performance

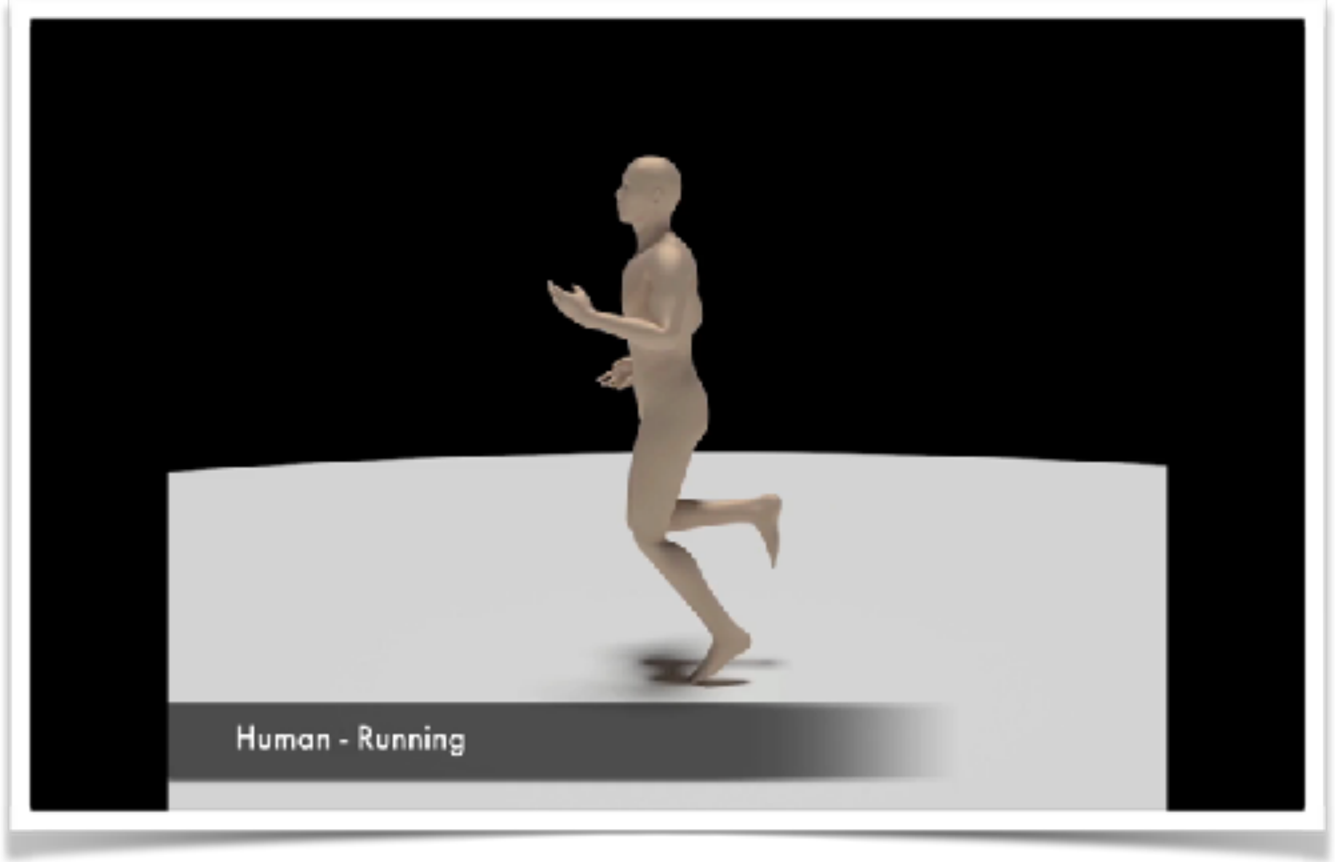


Sparse paged grid

Material point method

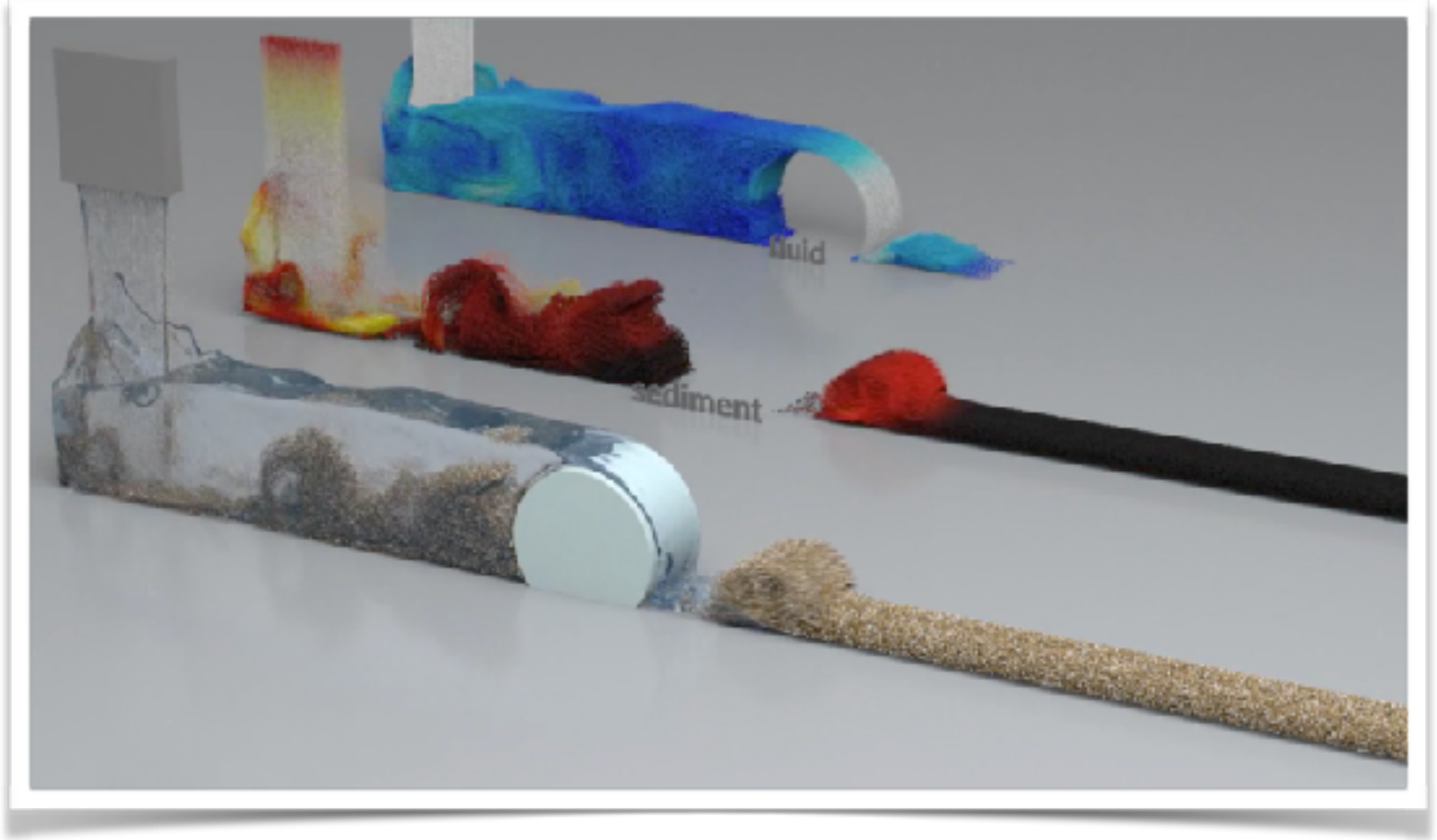


Adaptivity

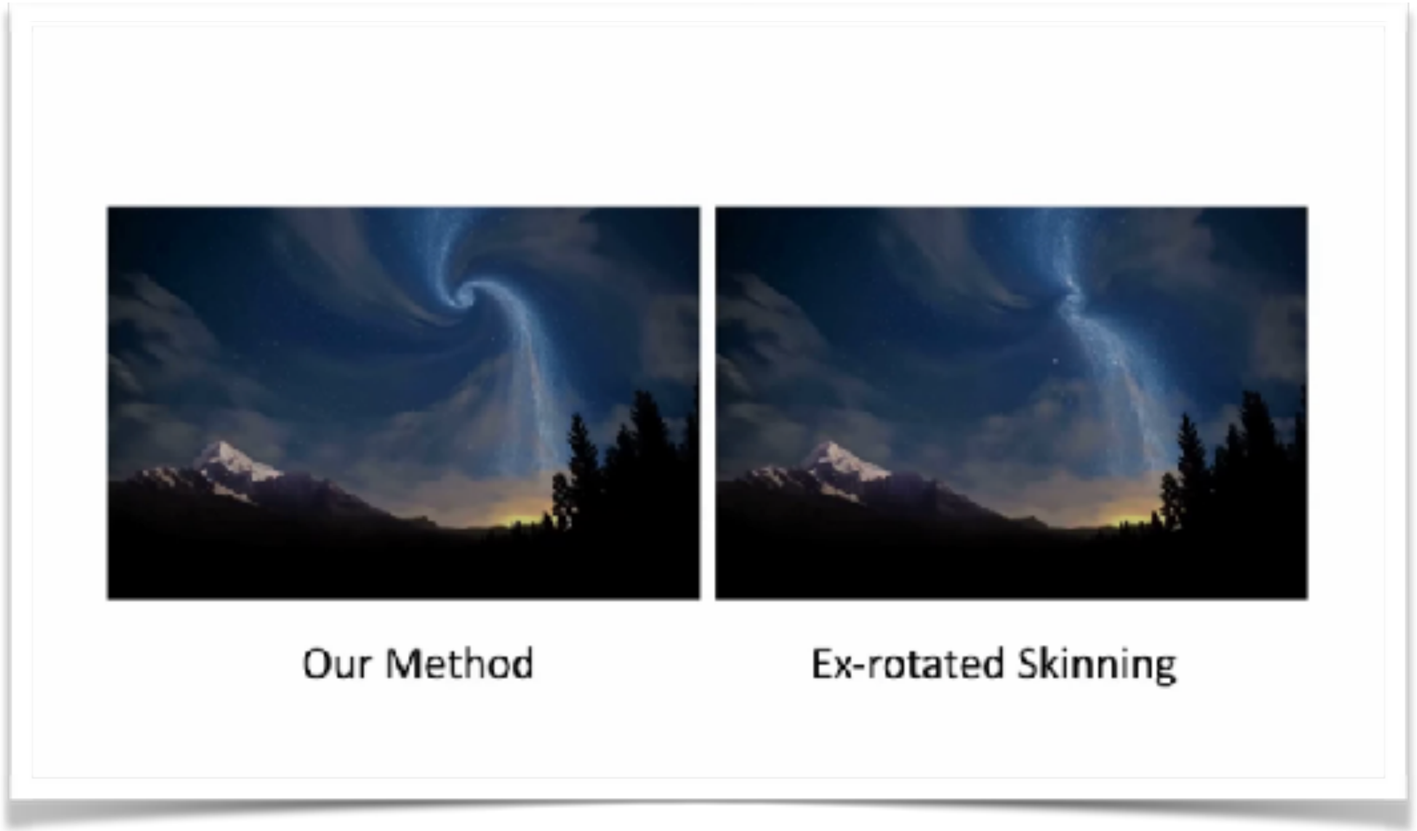


SCA 14

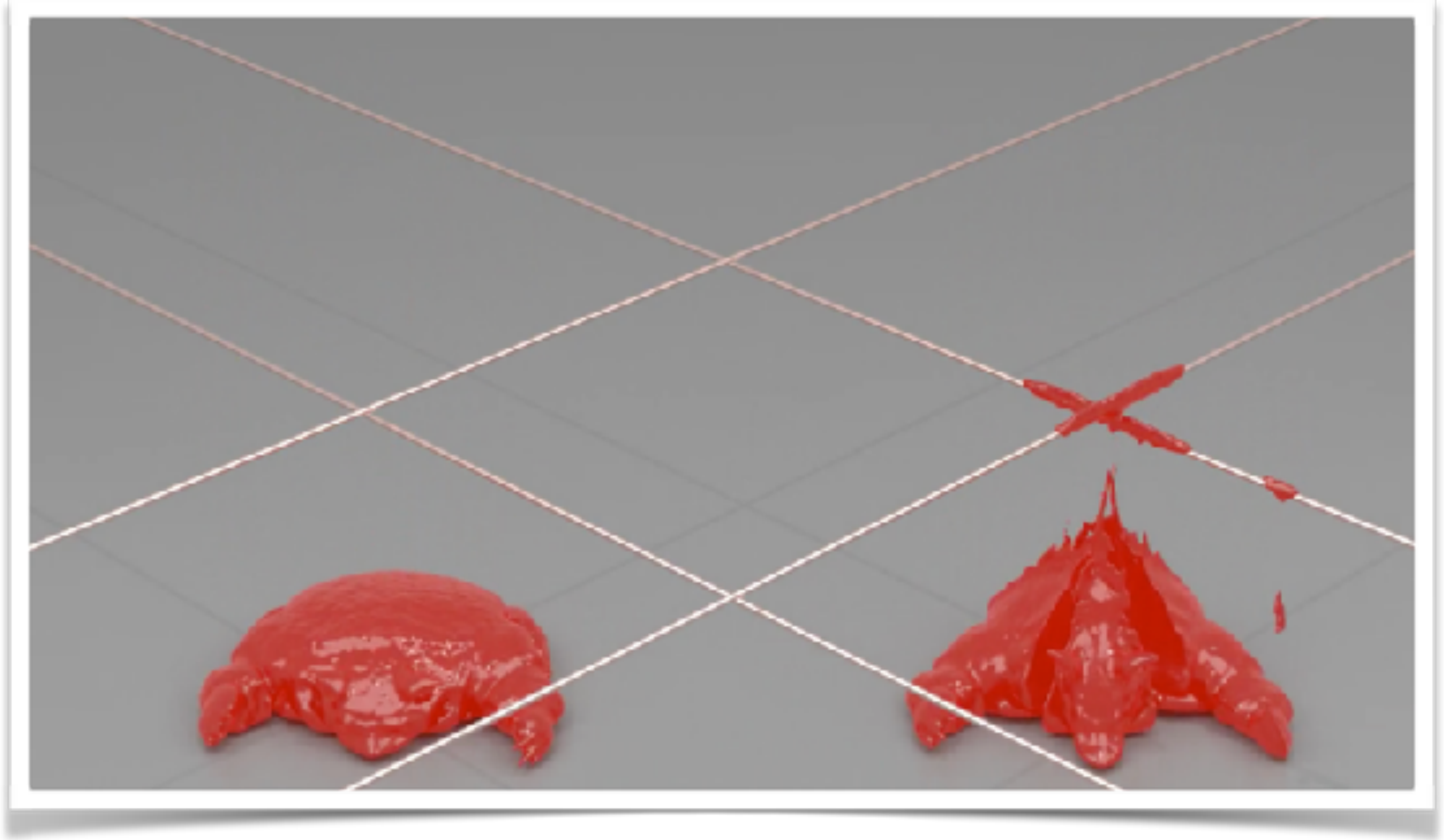
SIGGRAPH 17



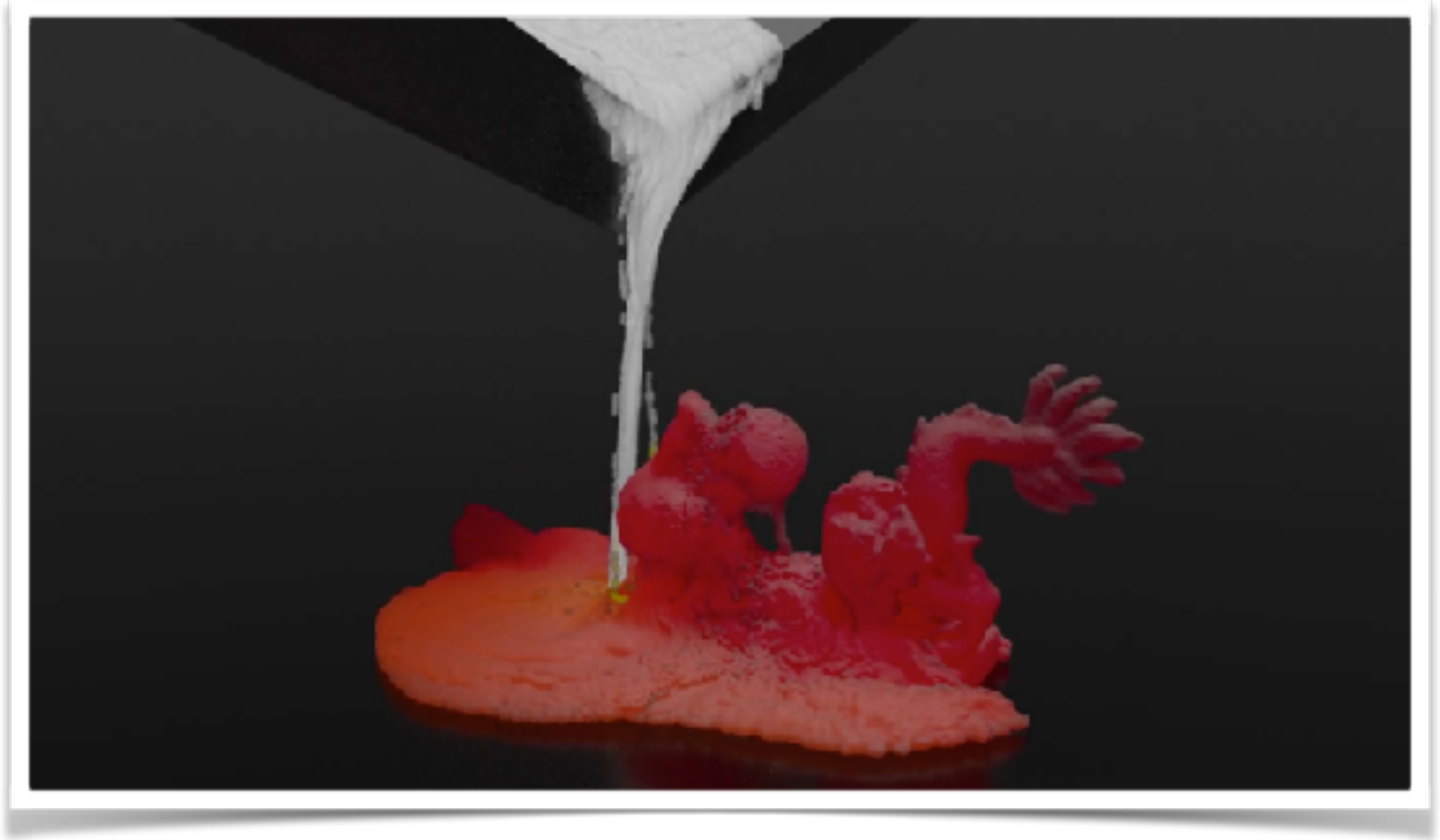
SIGGRAPH 18



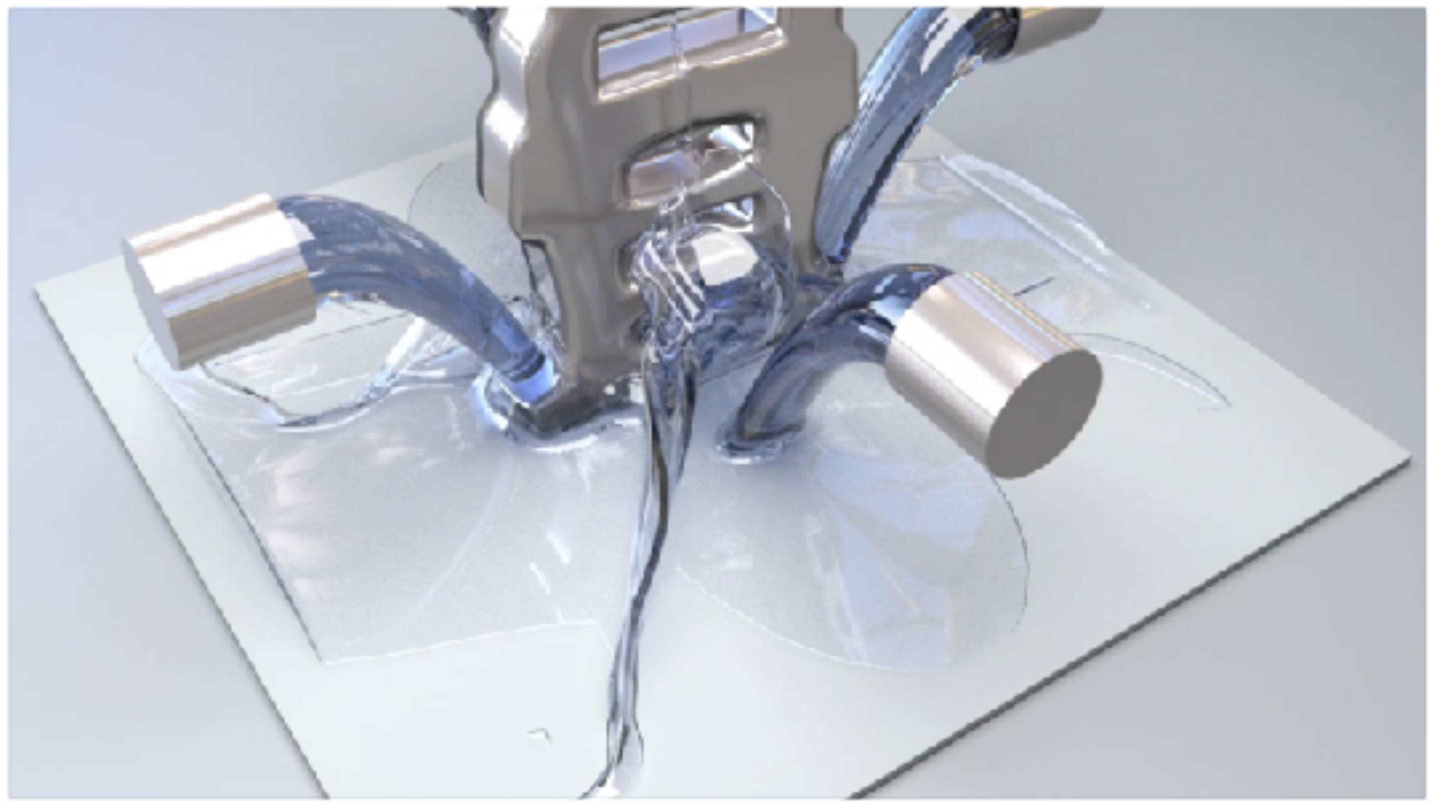
Eurographics 16



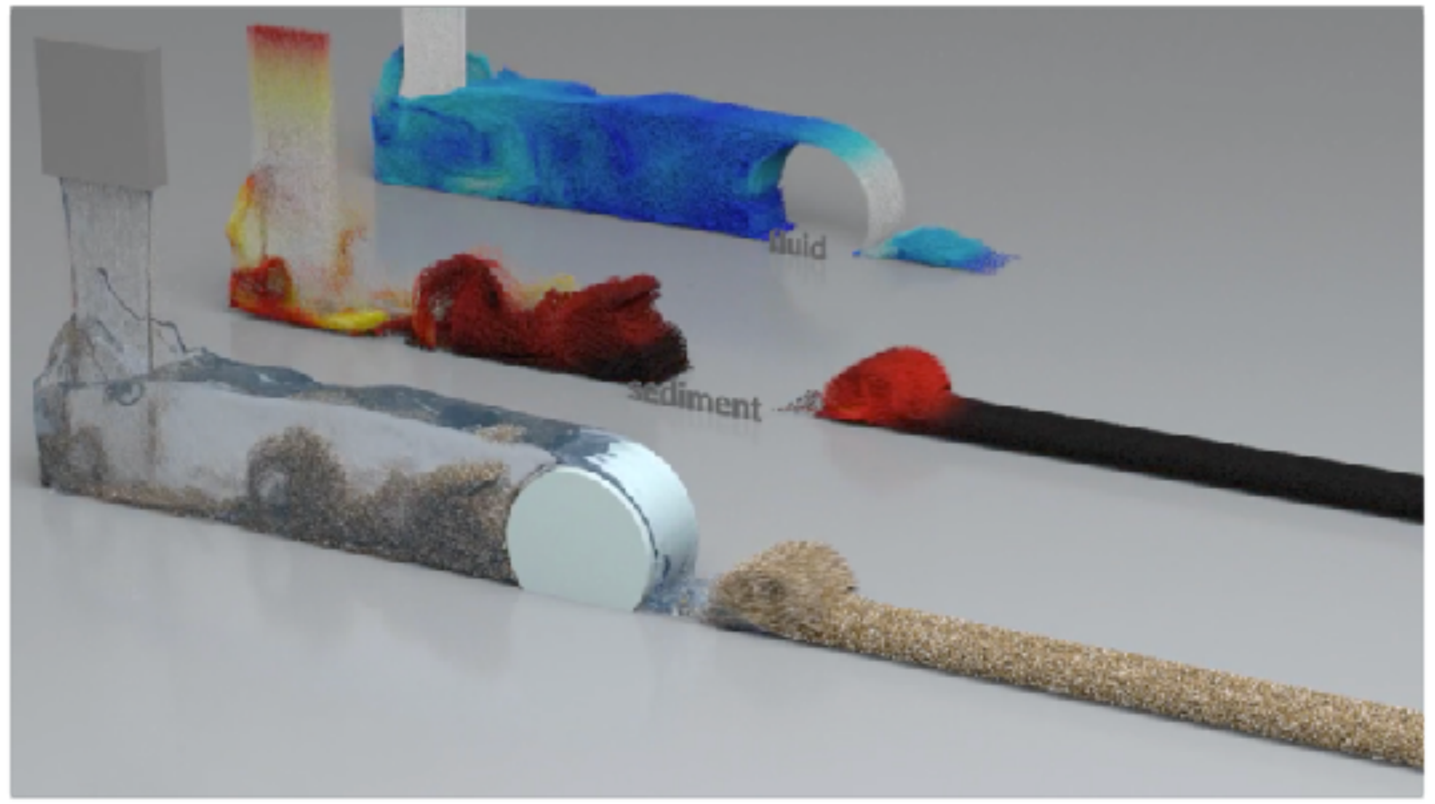
SIGGRAPH Asia 17



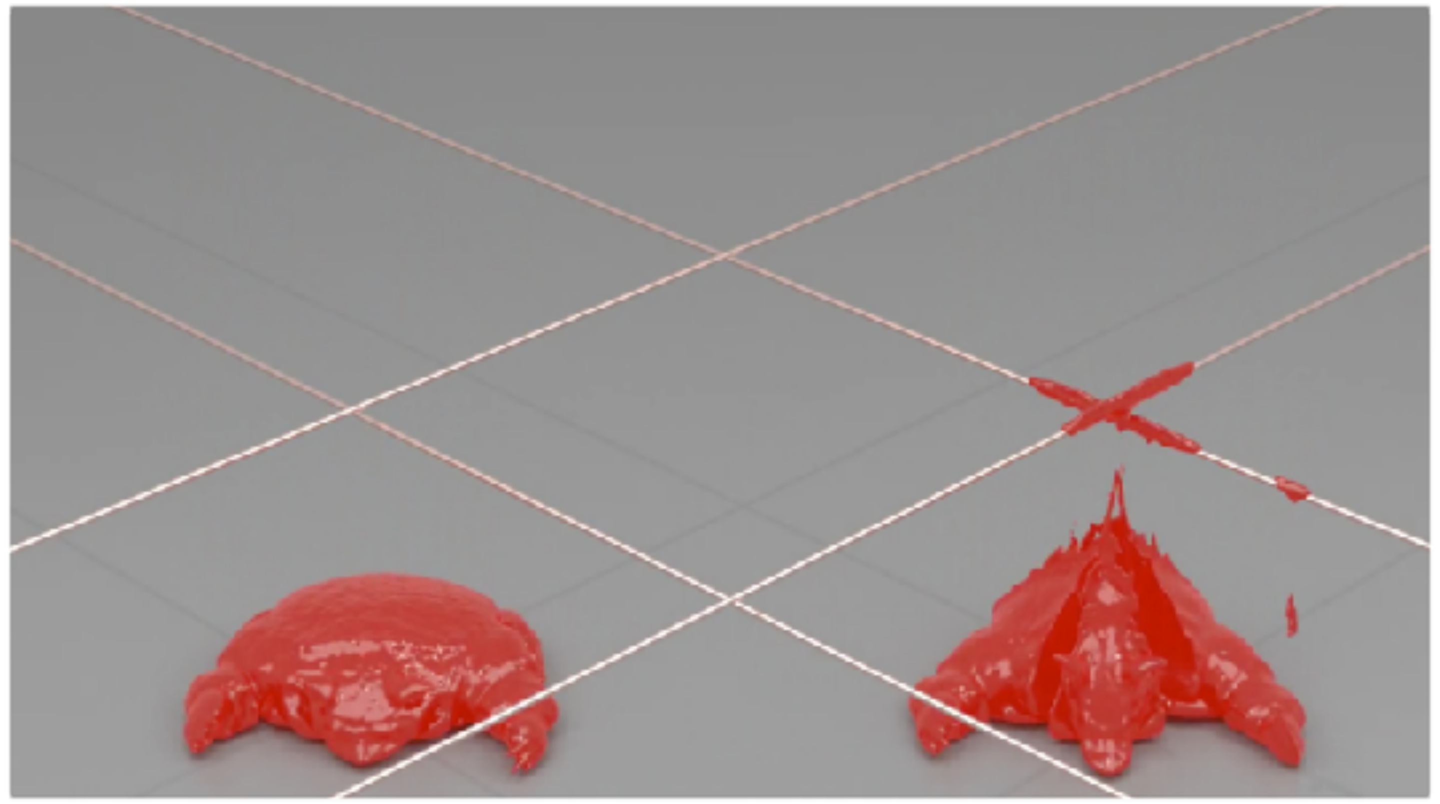
SIGGRAPH Asia 18
(under review)



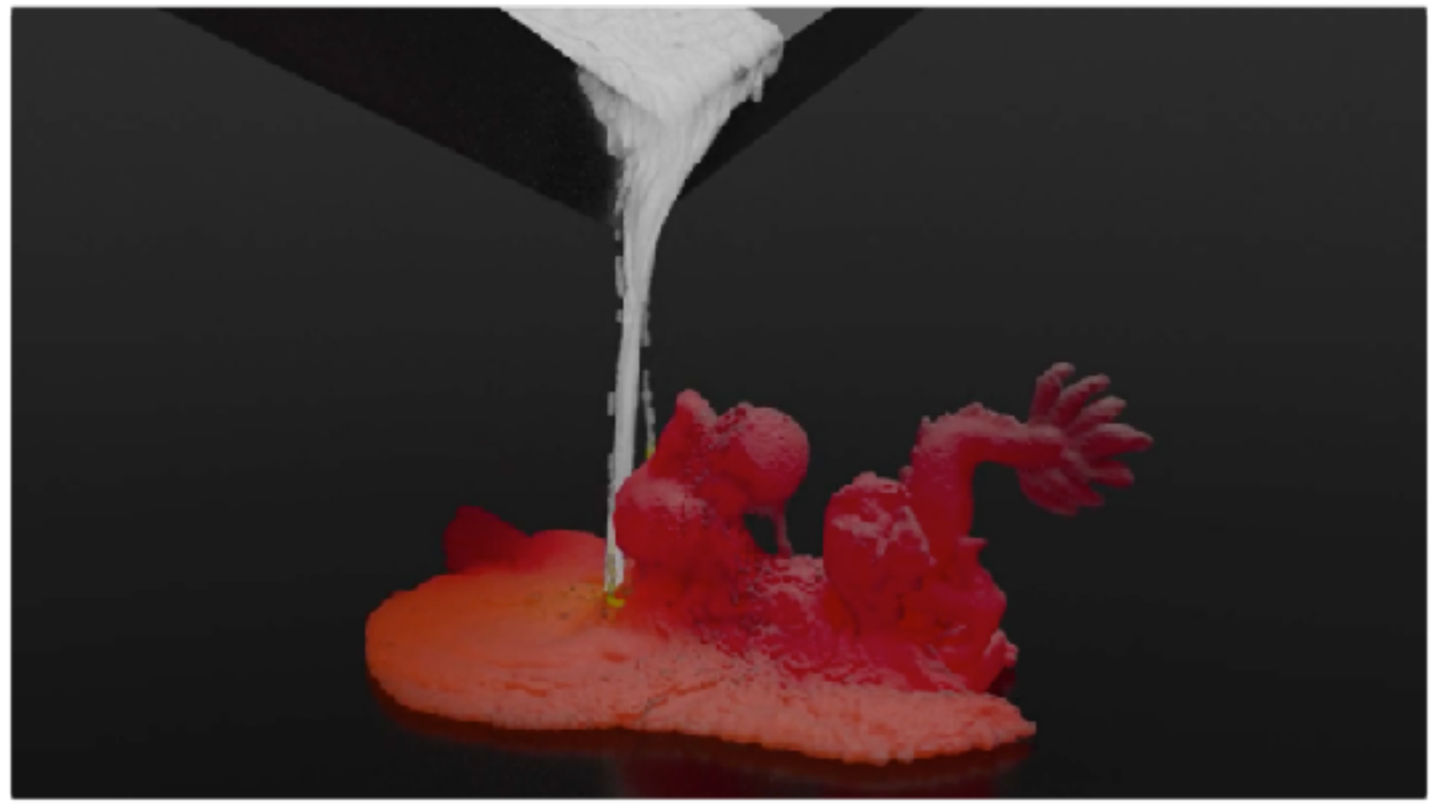
SPGrid / Adaptivity



SPGrid / MPM



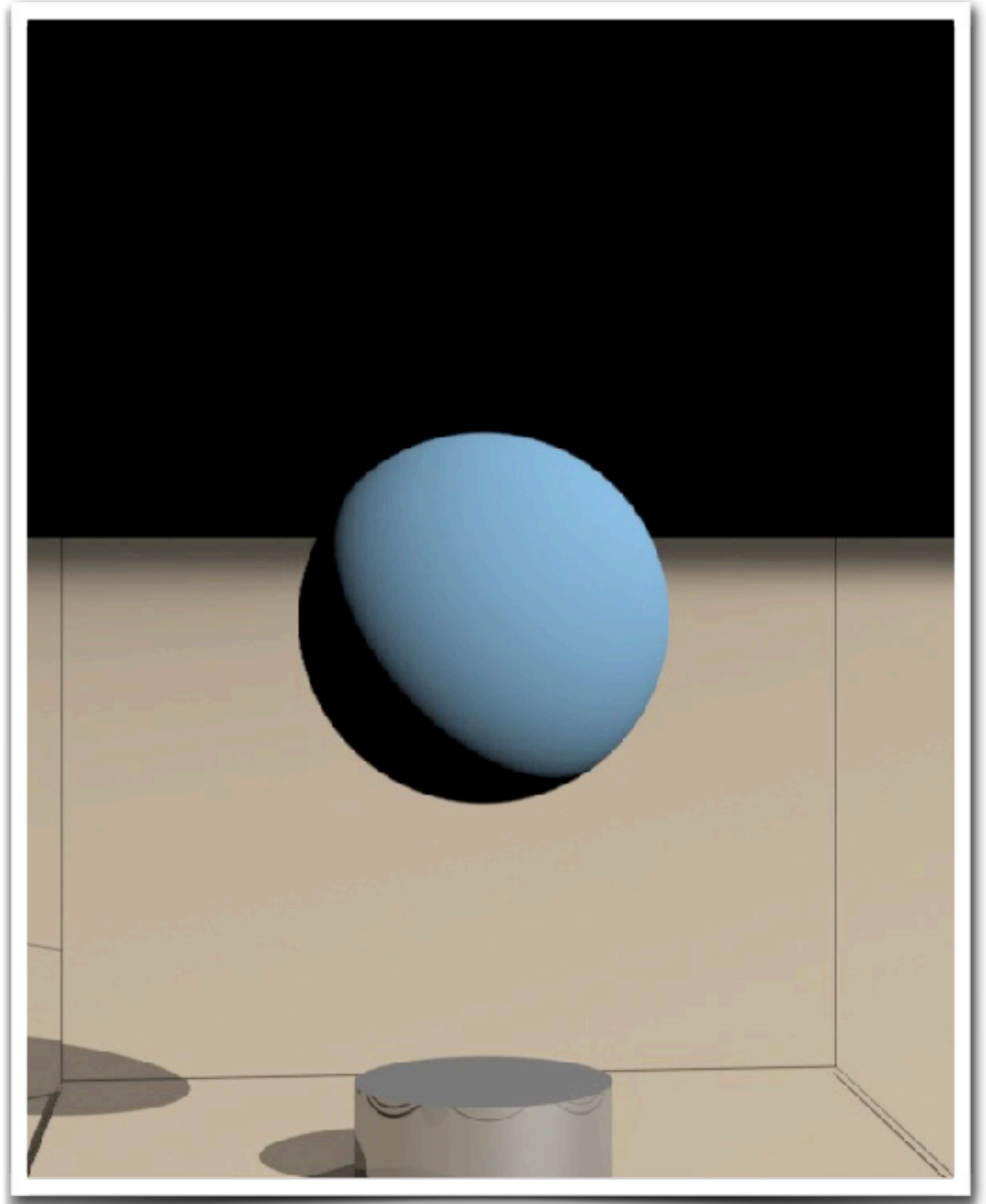
SPGrid / MPM / Adaptivity



SPGrid / MPM / GPU

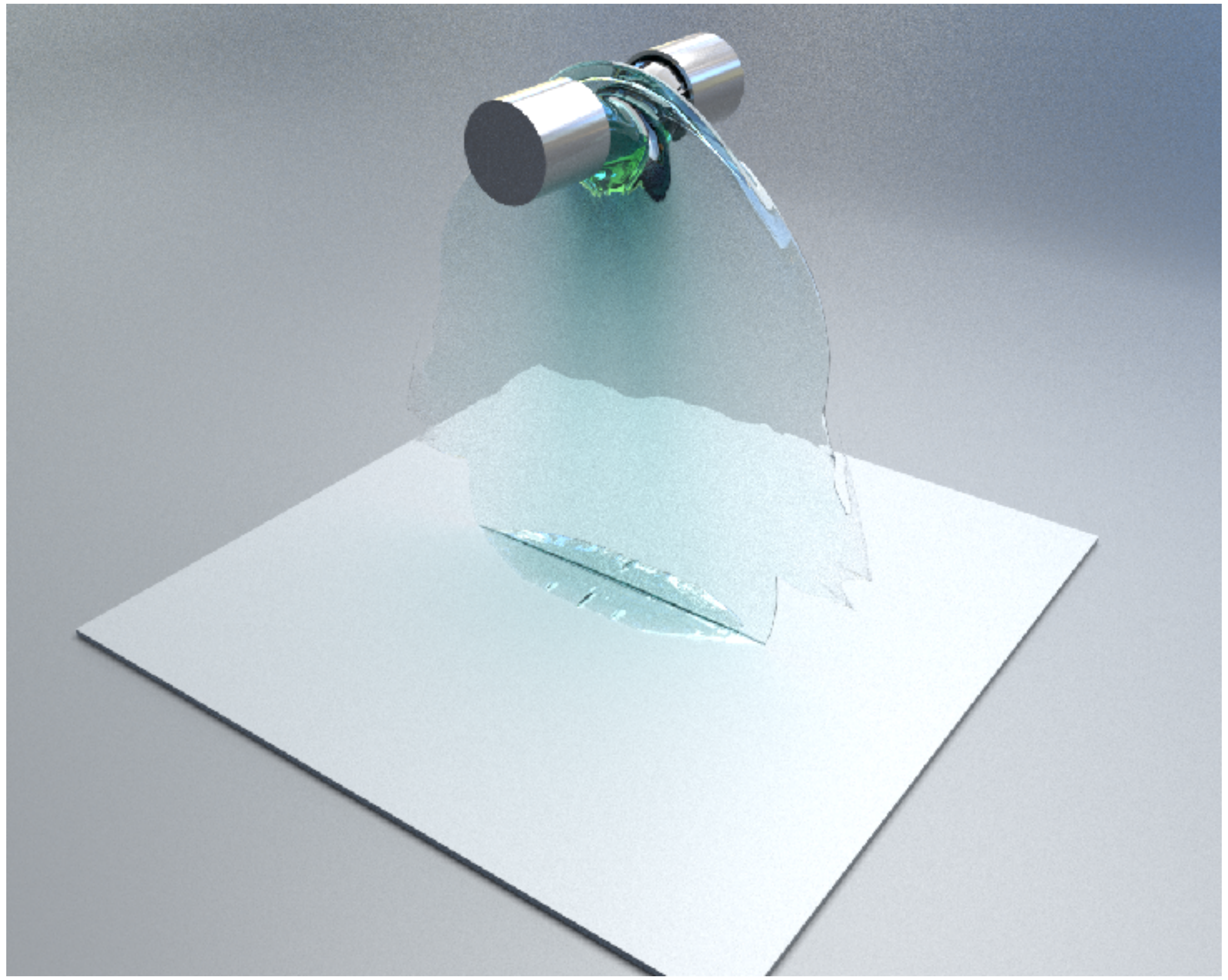
Sparse paged grid (SPGrid)

135M voxels
23GB

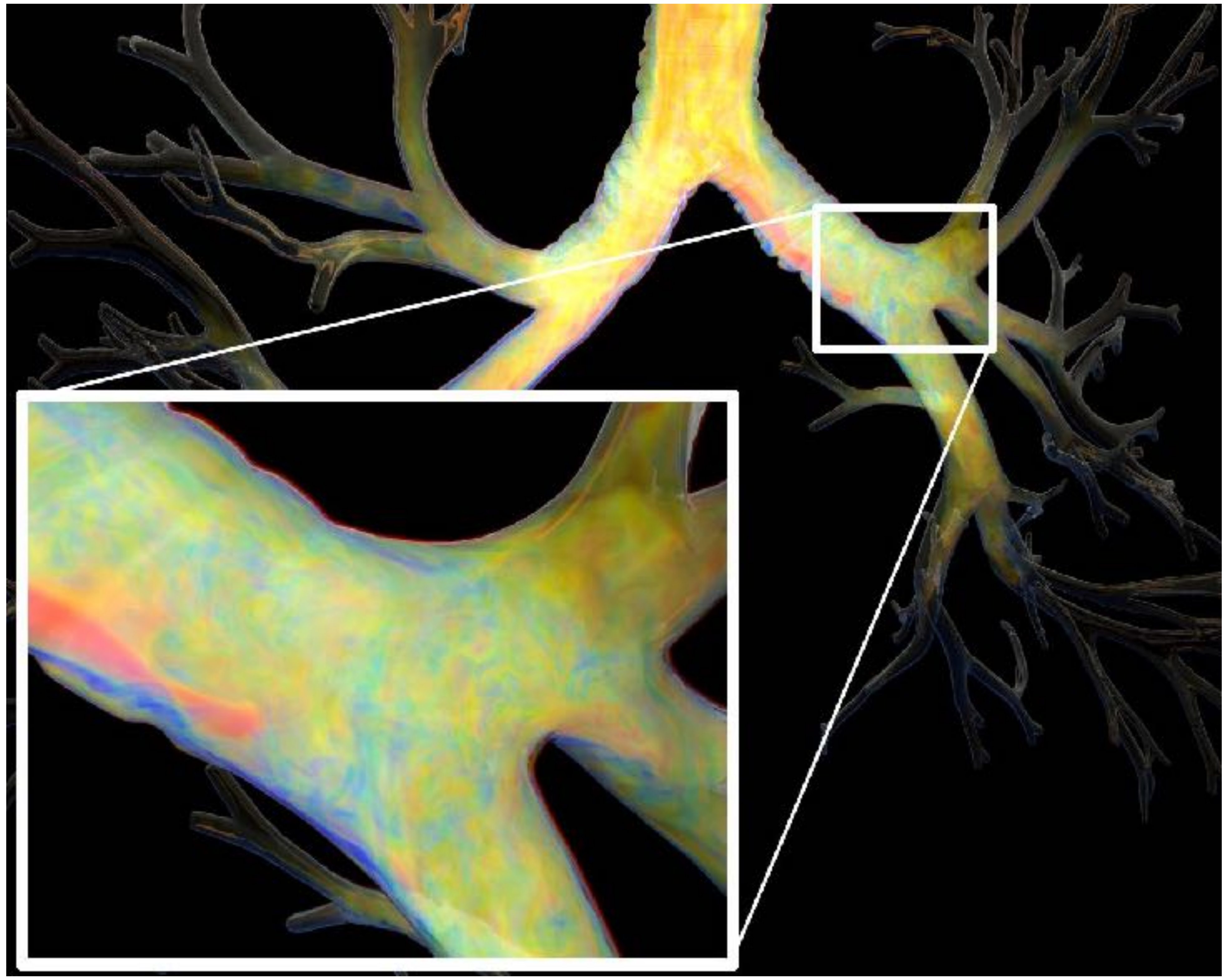


Setaluri et al. 14

Sparsity

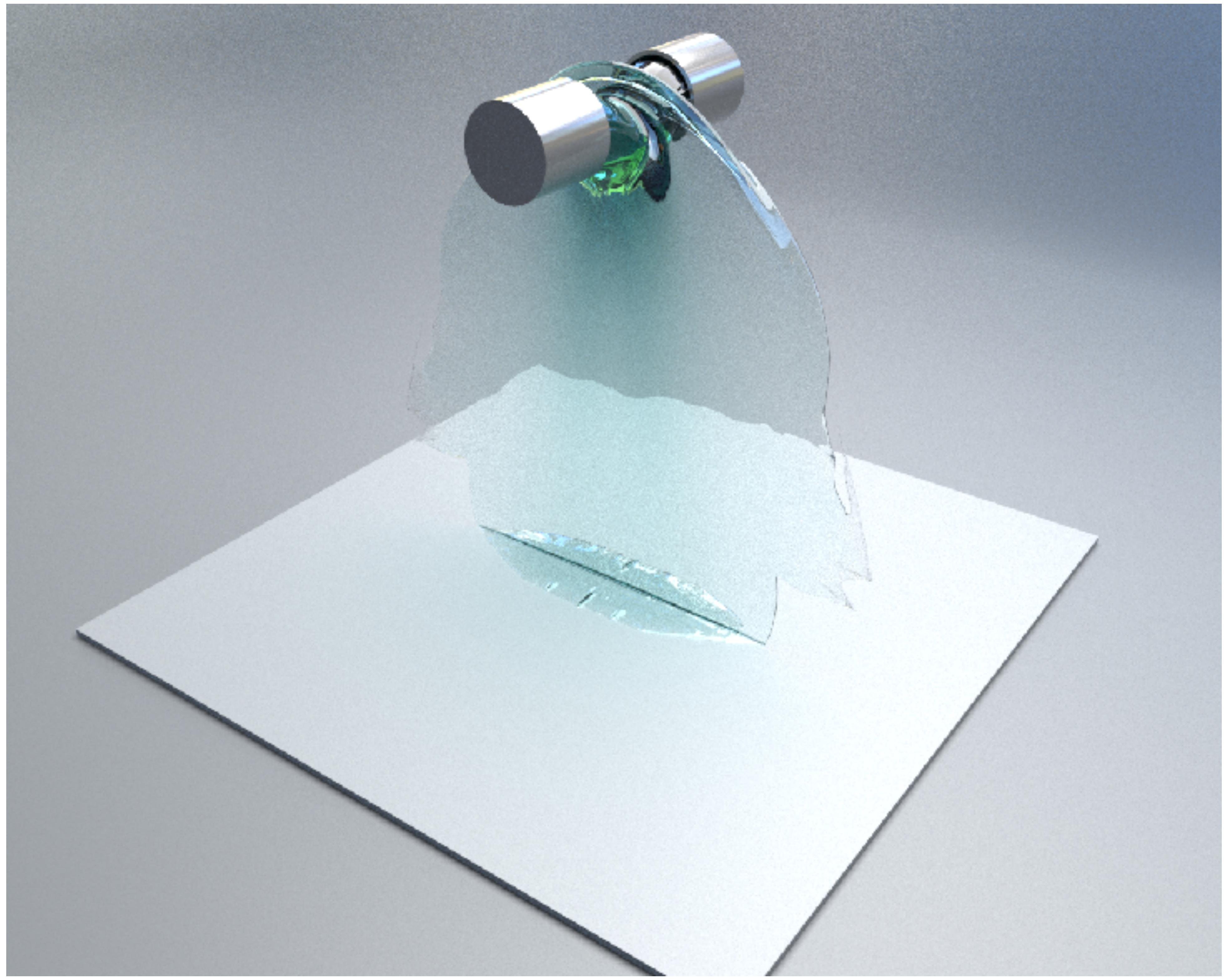


Liu et al. 16



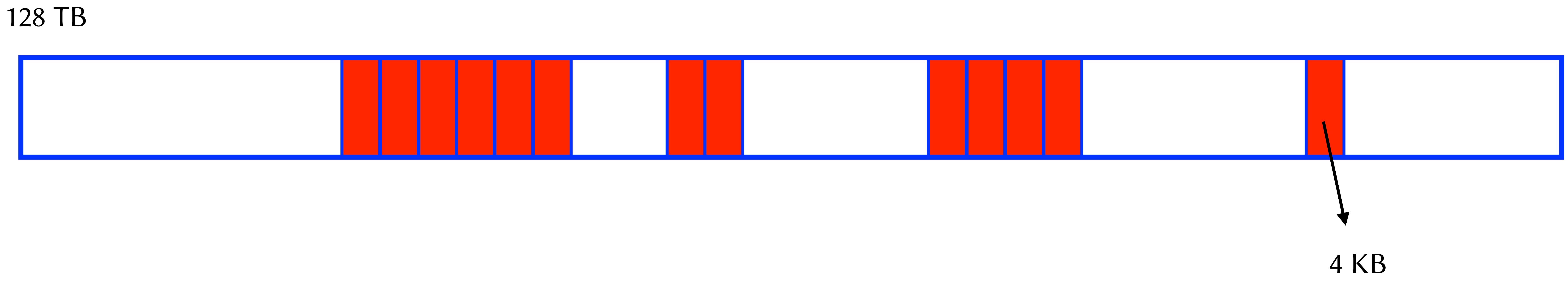
Resolution: $8192 \times 8192 \times 4096$

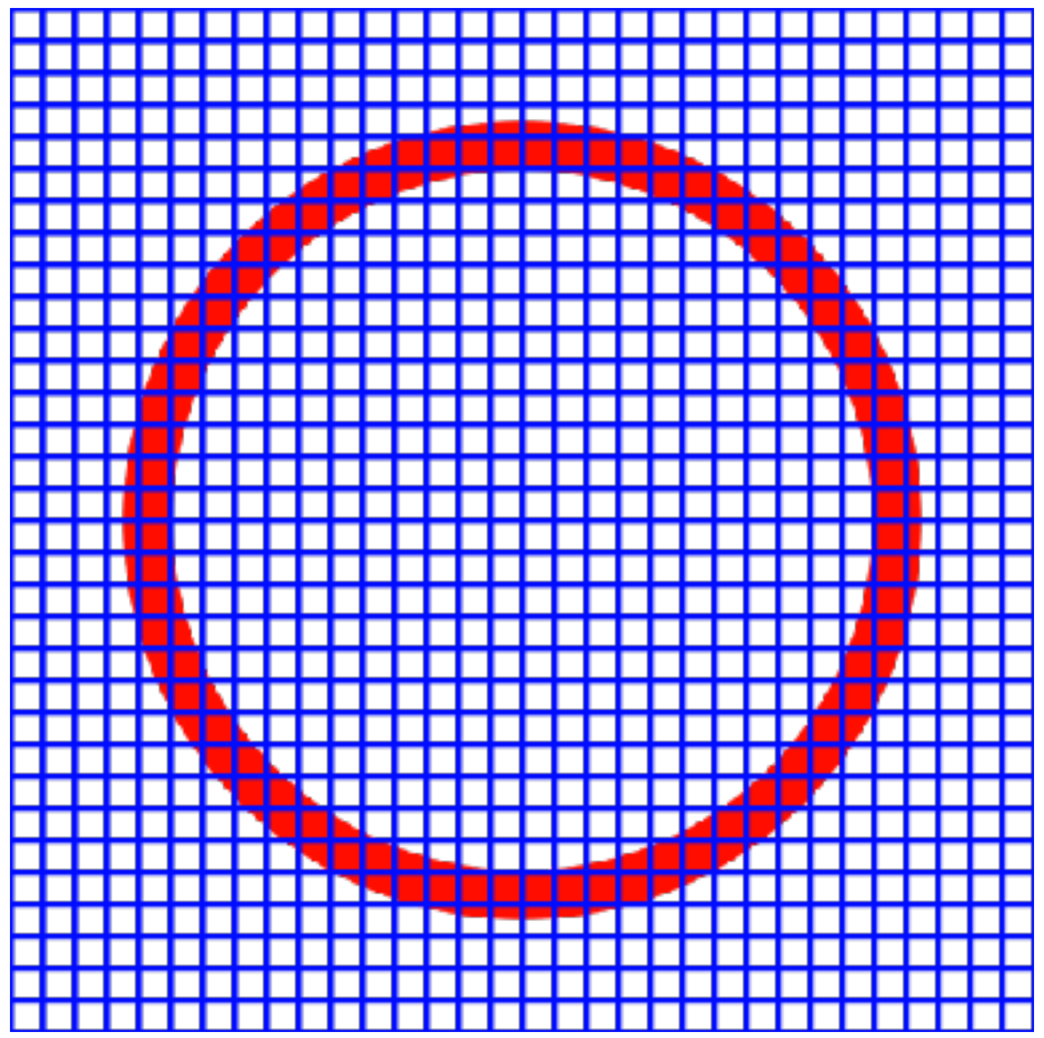
Sparsity

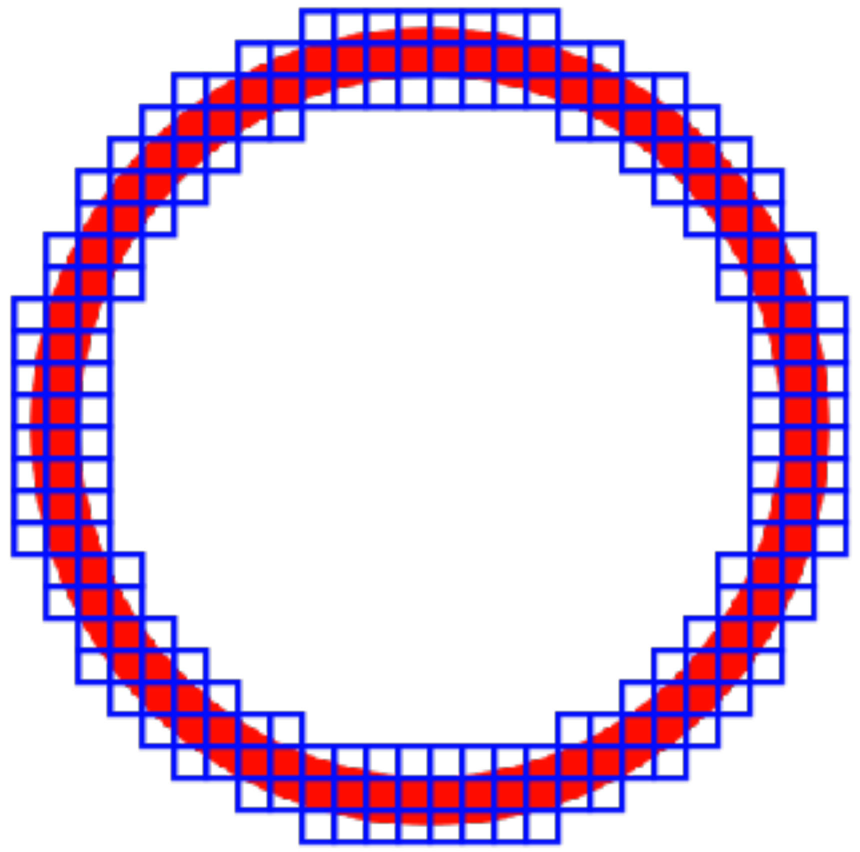


SPGrid sparsity ⟷ Virtual memory

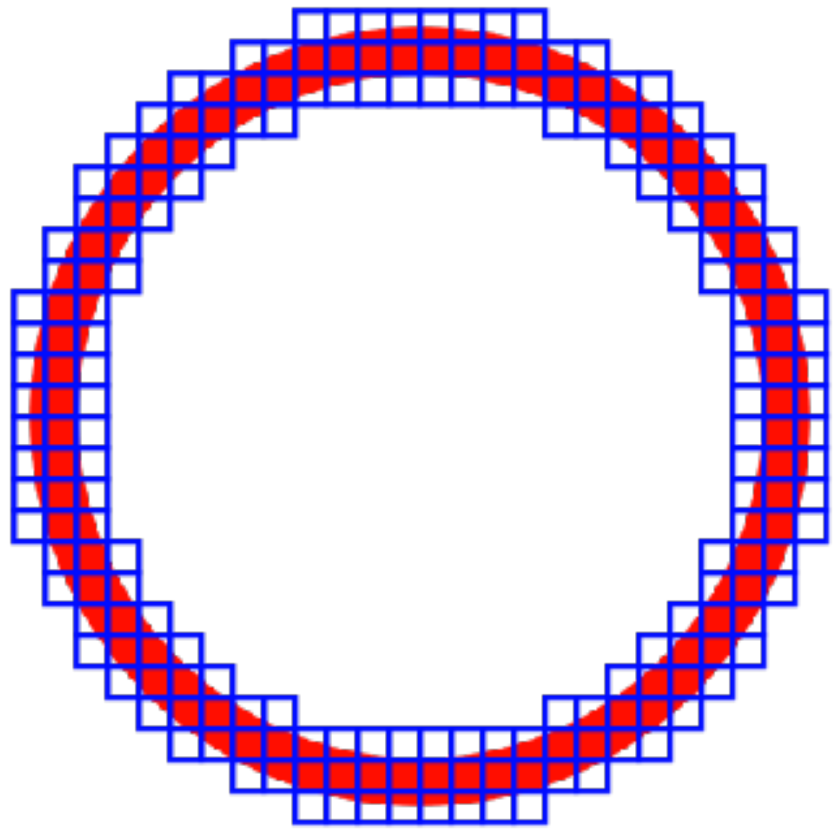
Virtual memory



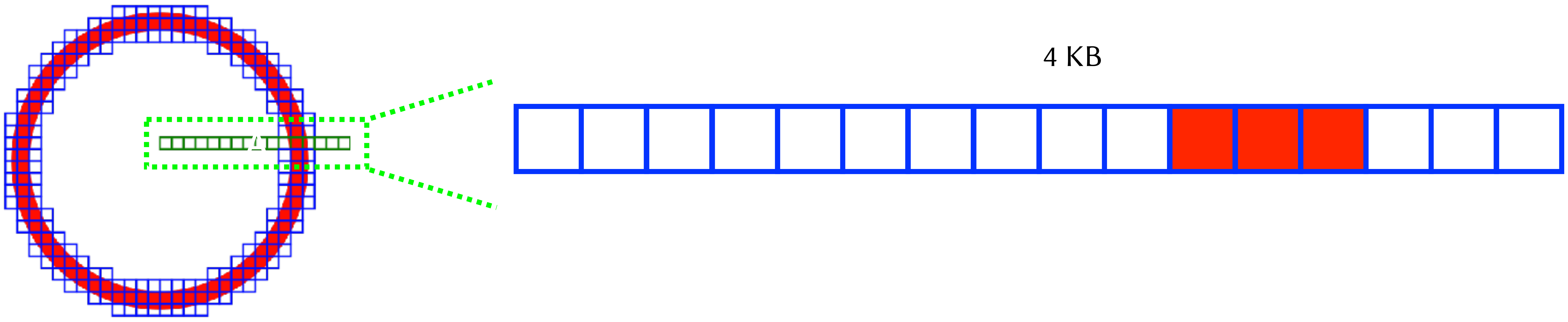


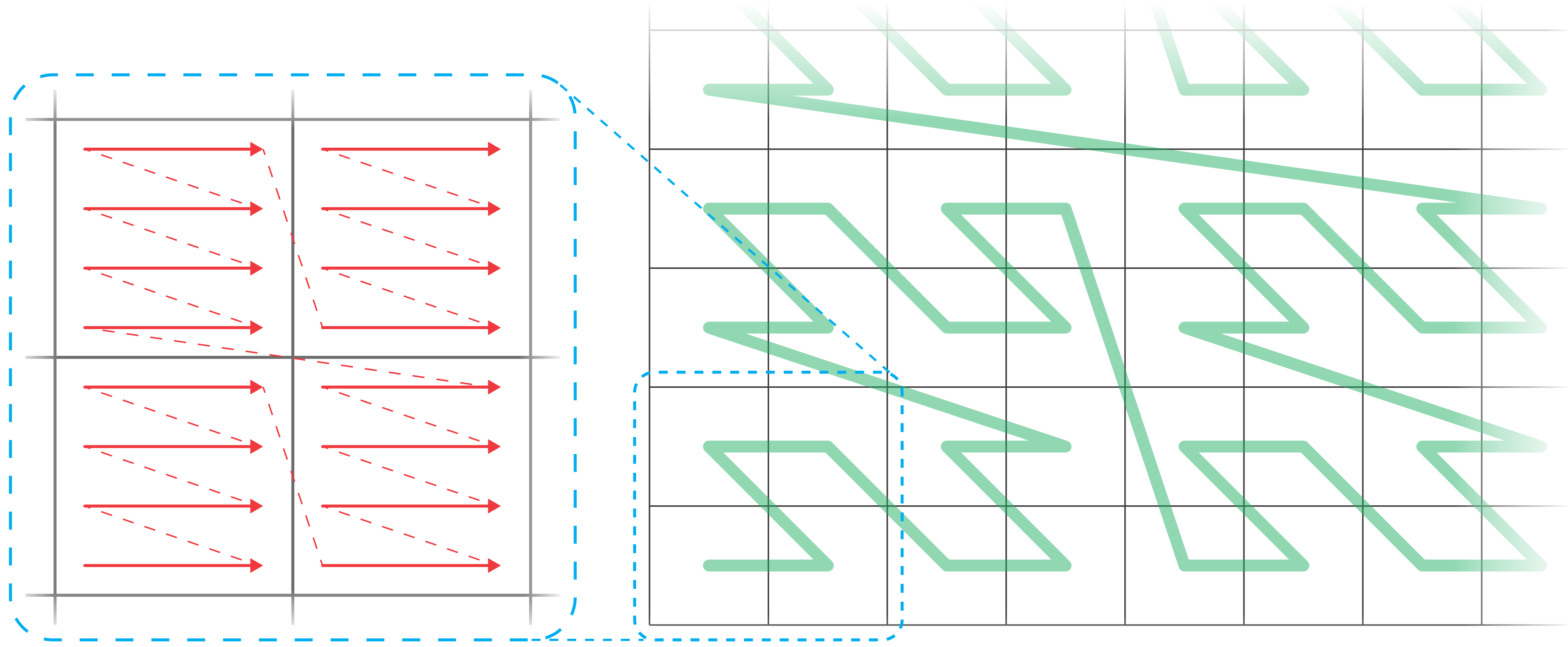


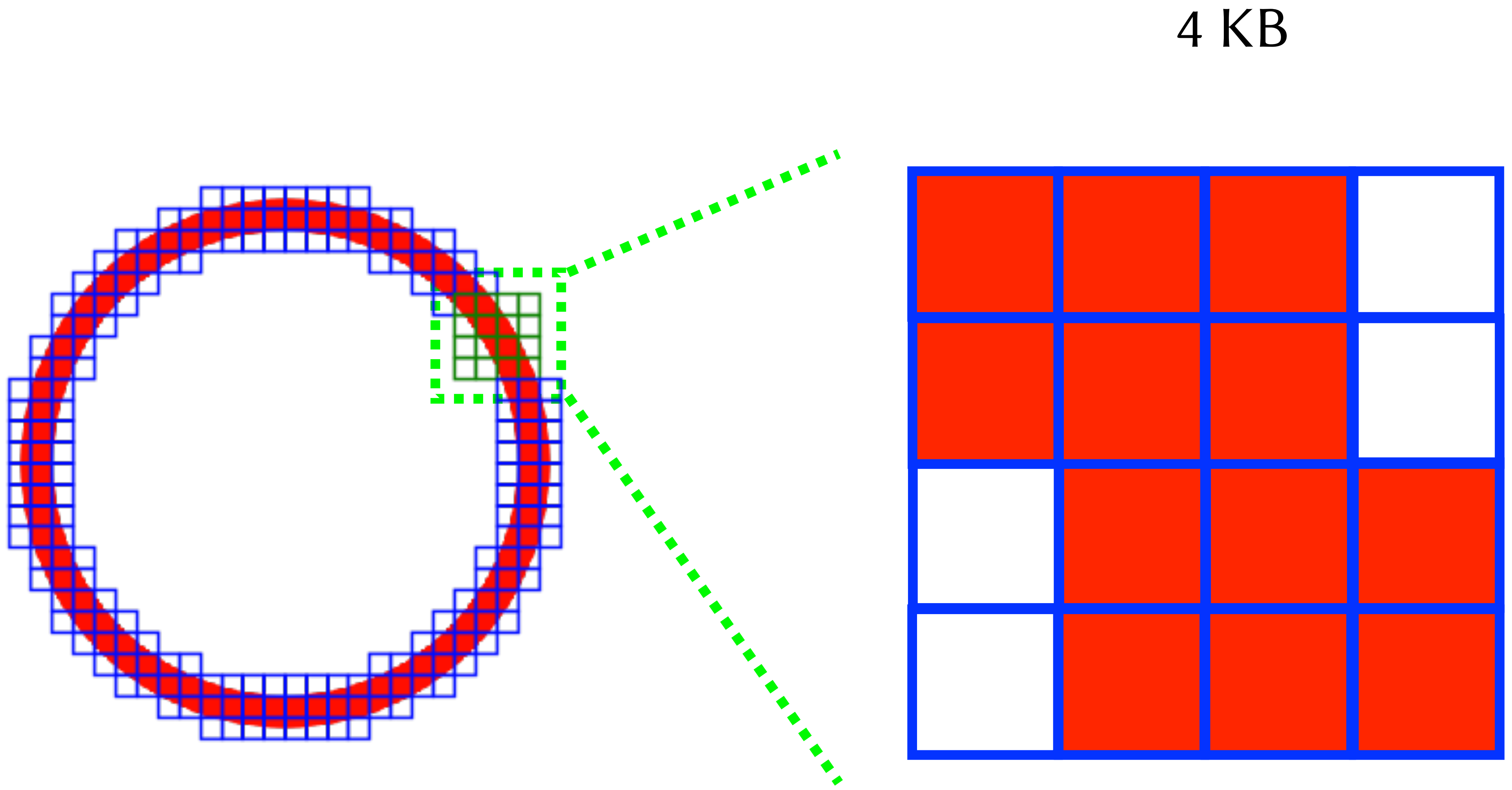
Cell ordering



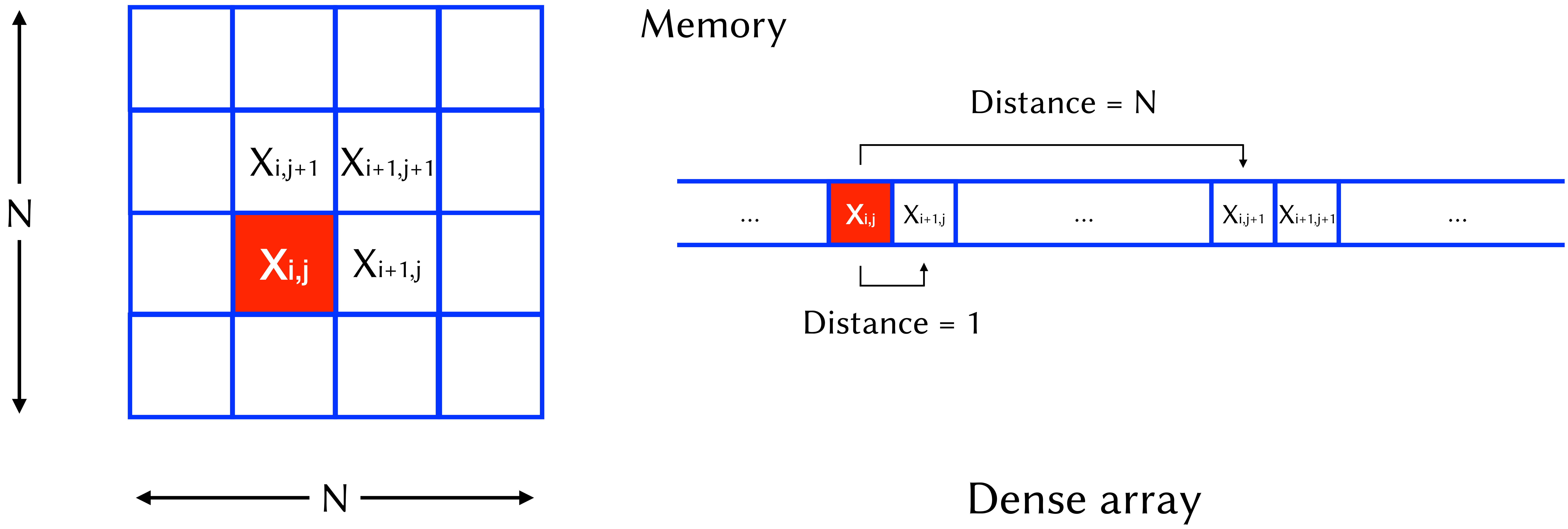
Cell ordering



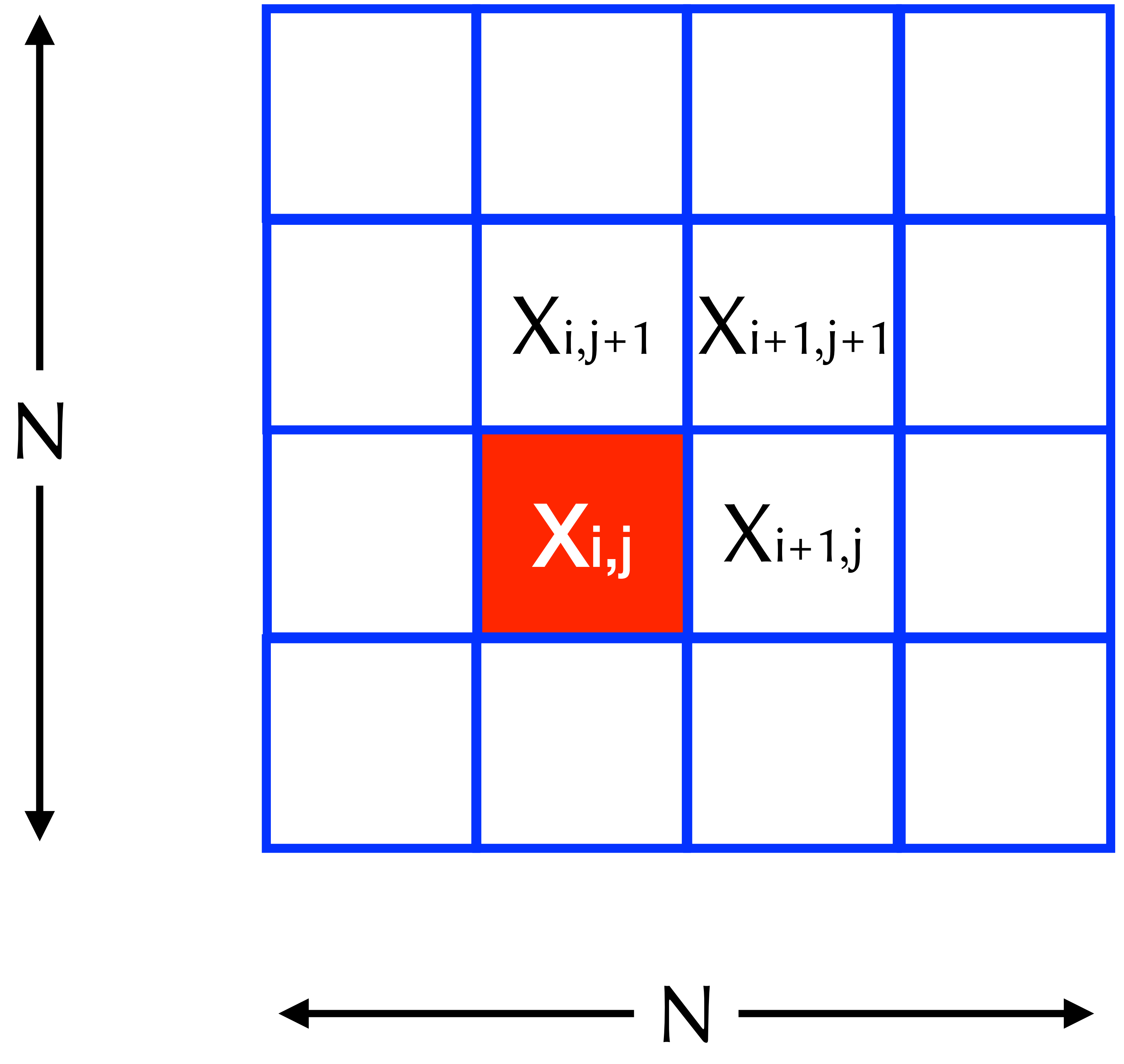


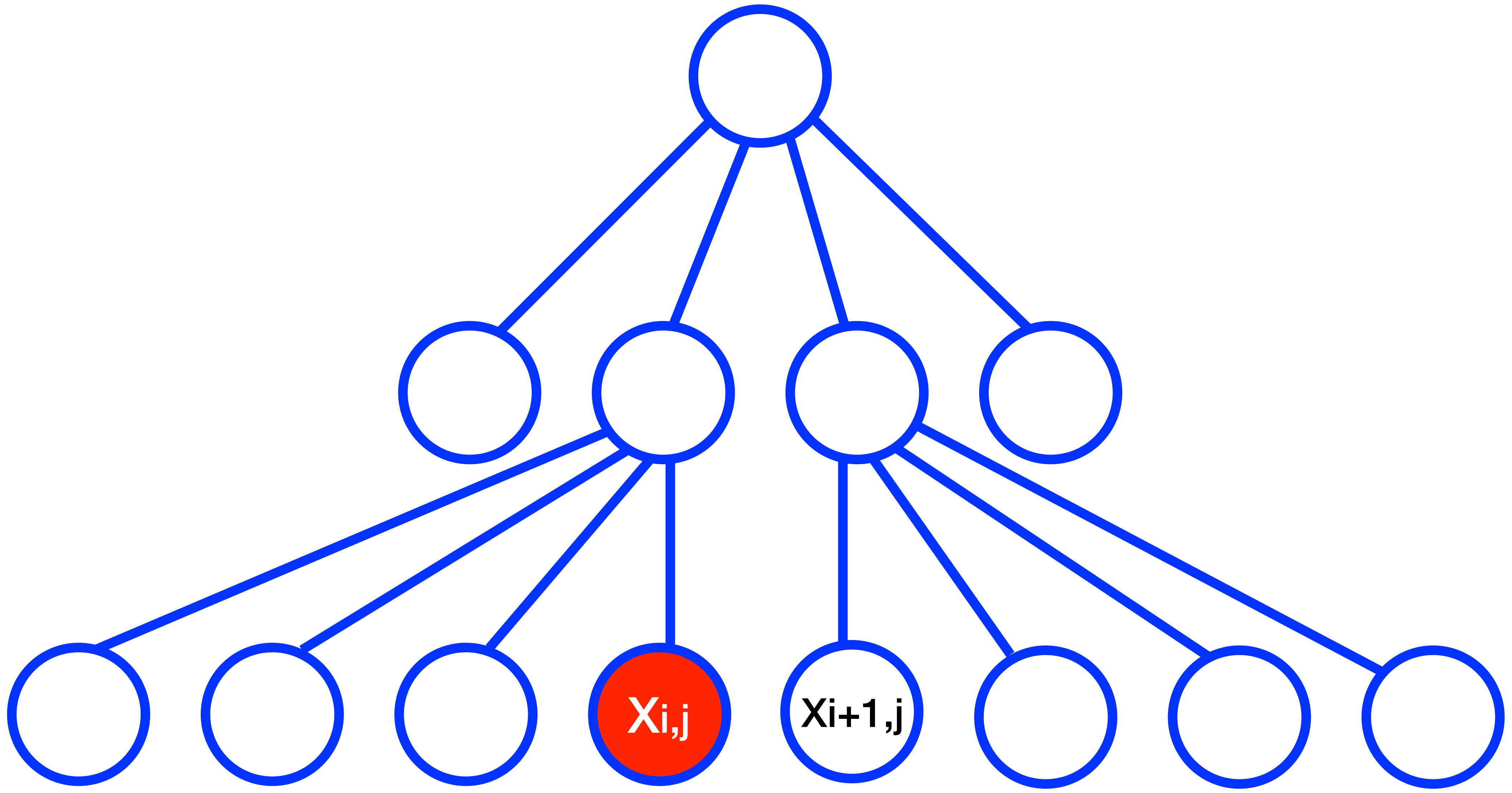


Stencil operations



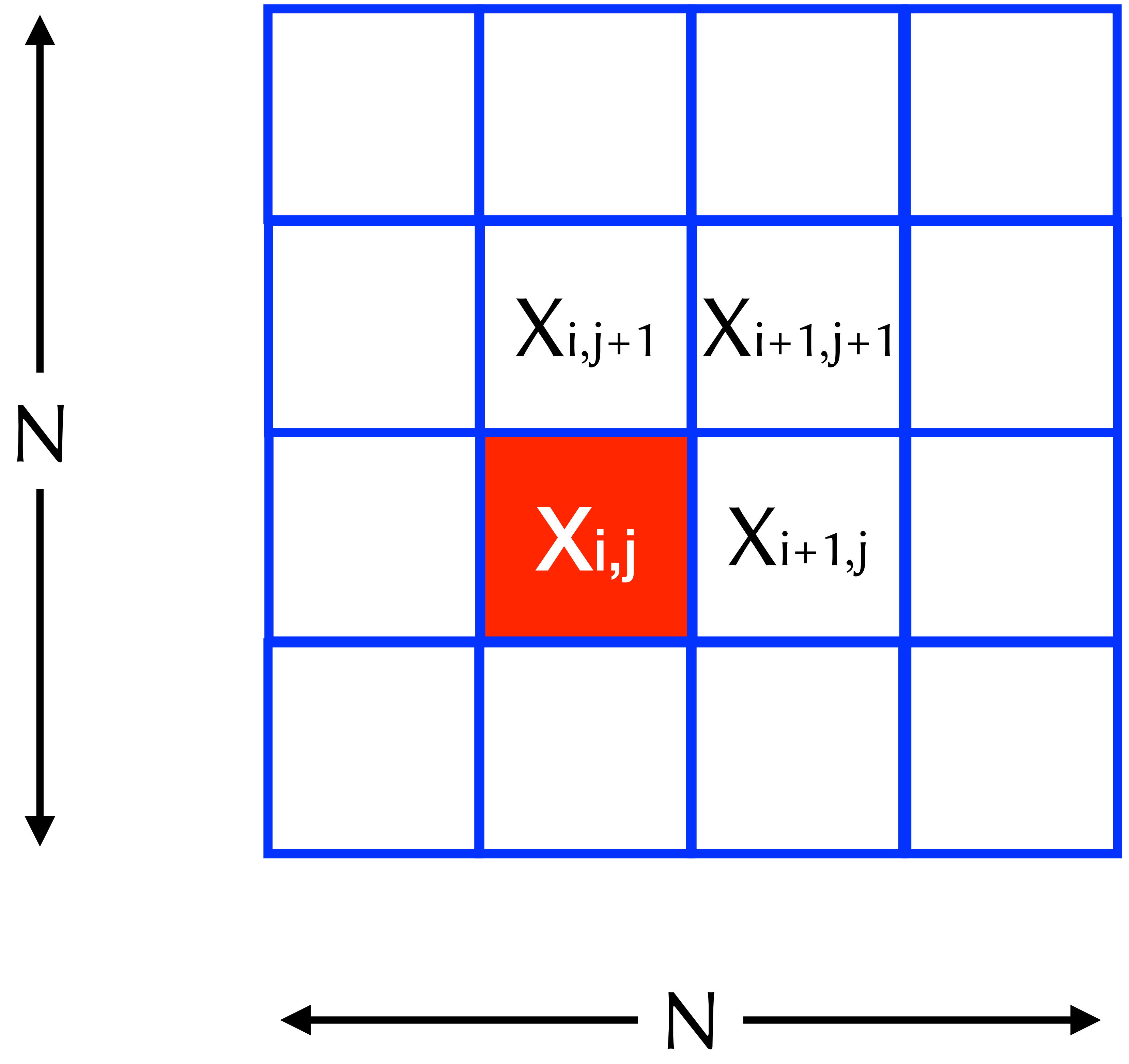
Stencil operations

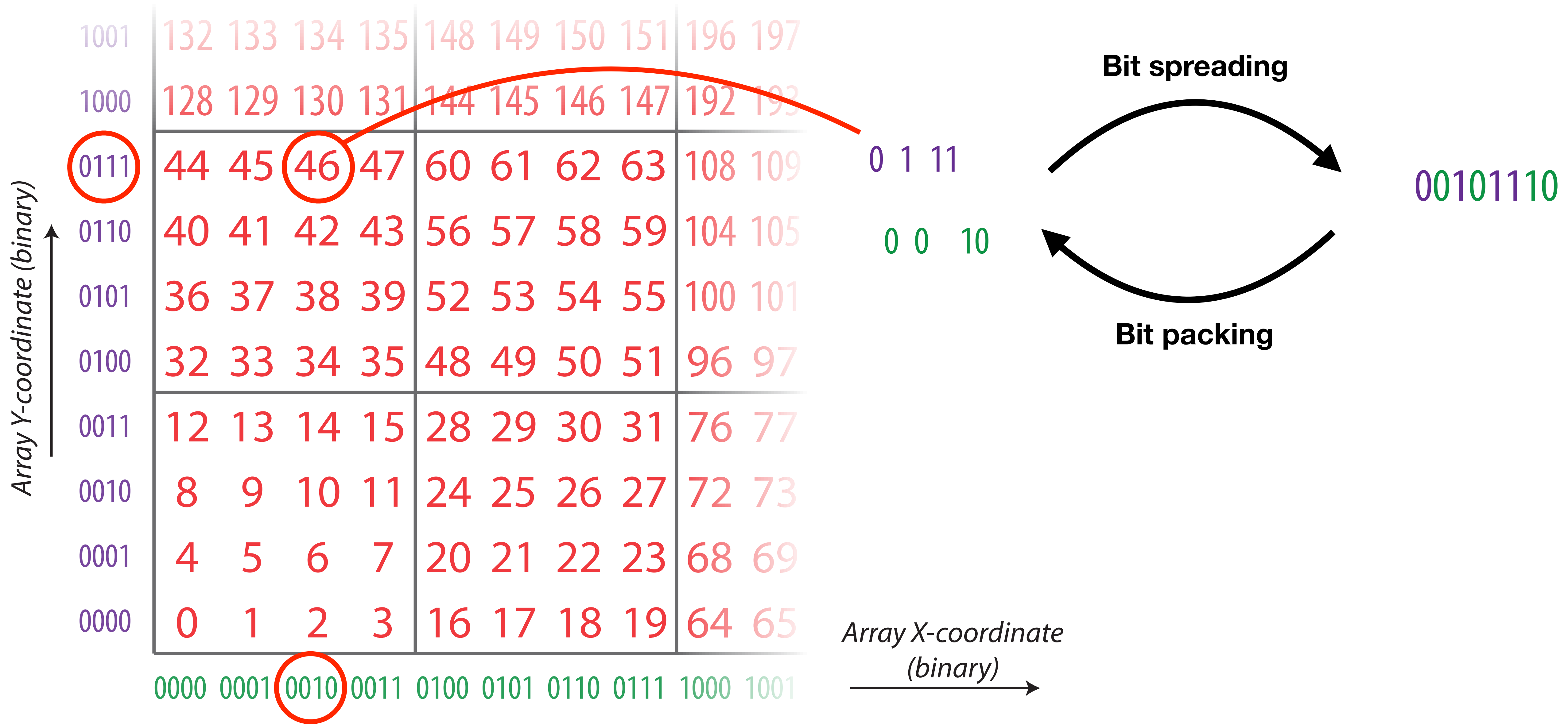




Octree

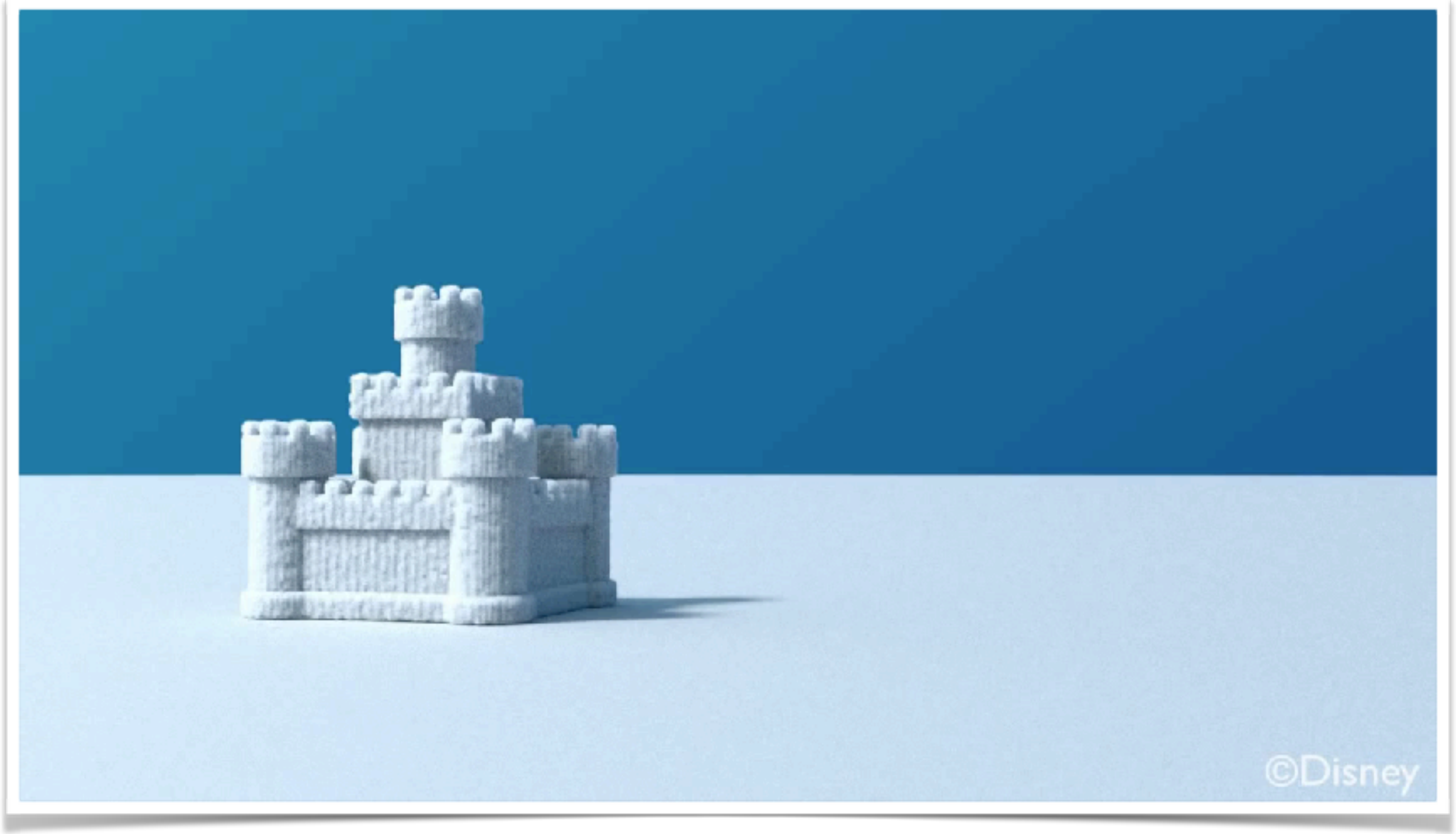
Stencil operations





SPGrid

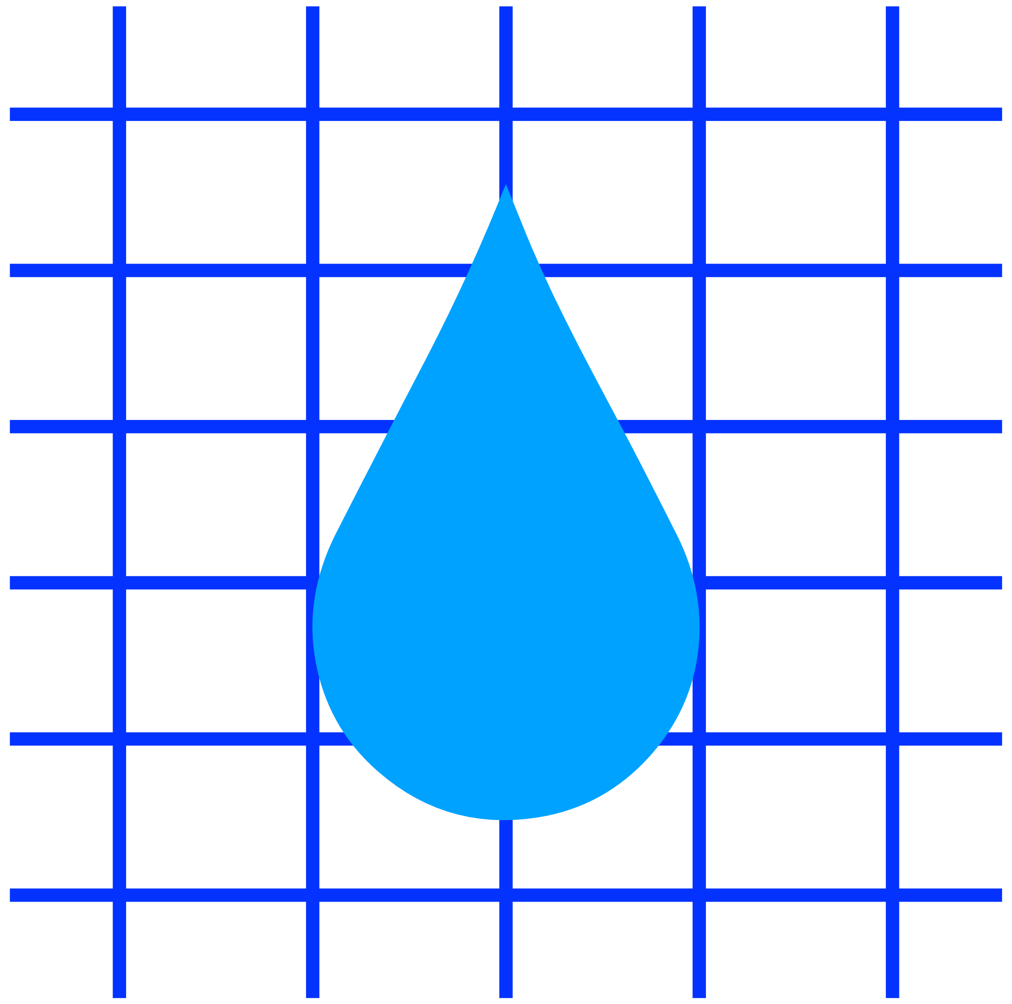
Material point method (MPM)



Stomakhin et al. 13

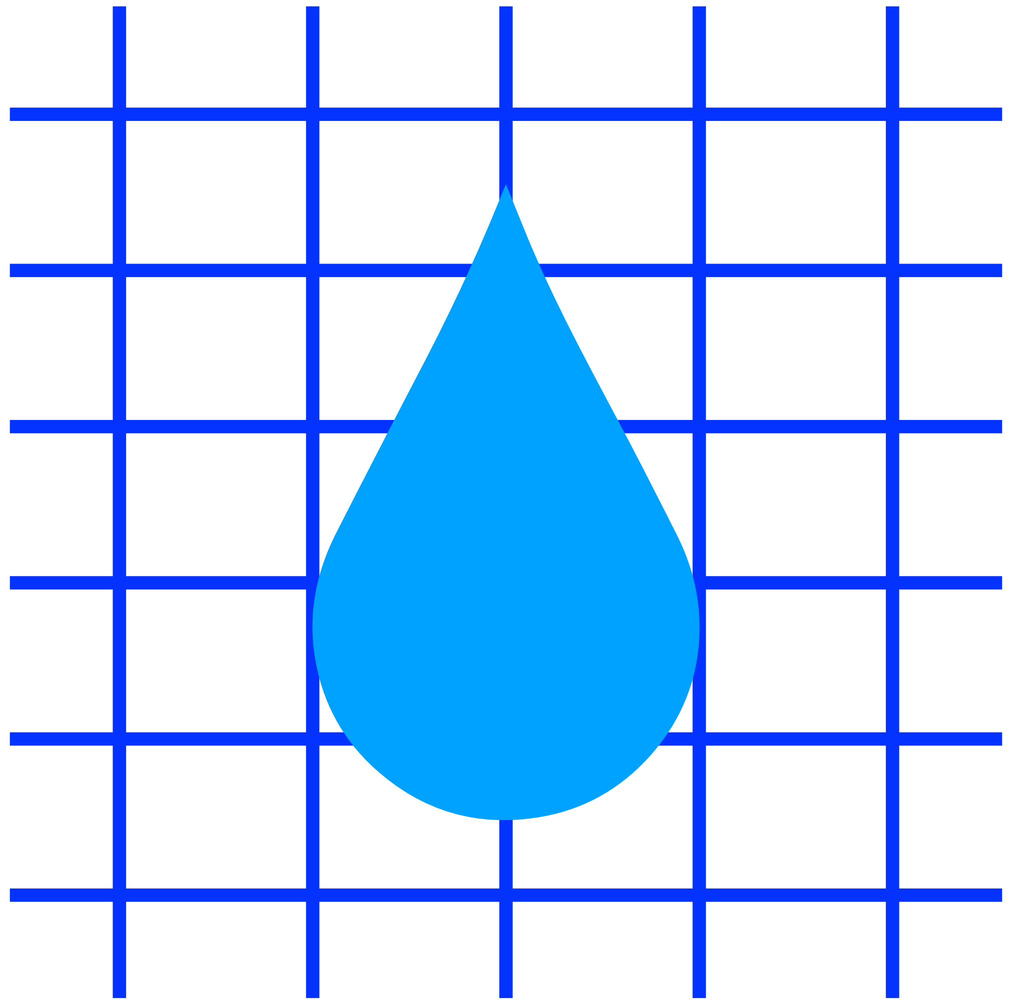
Gao et al. 18

Discretization schemes

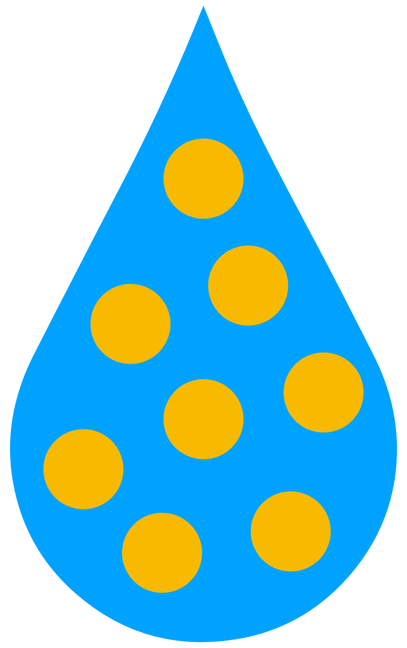


Grid

Discretization schemes

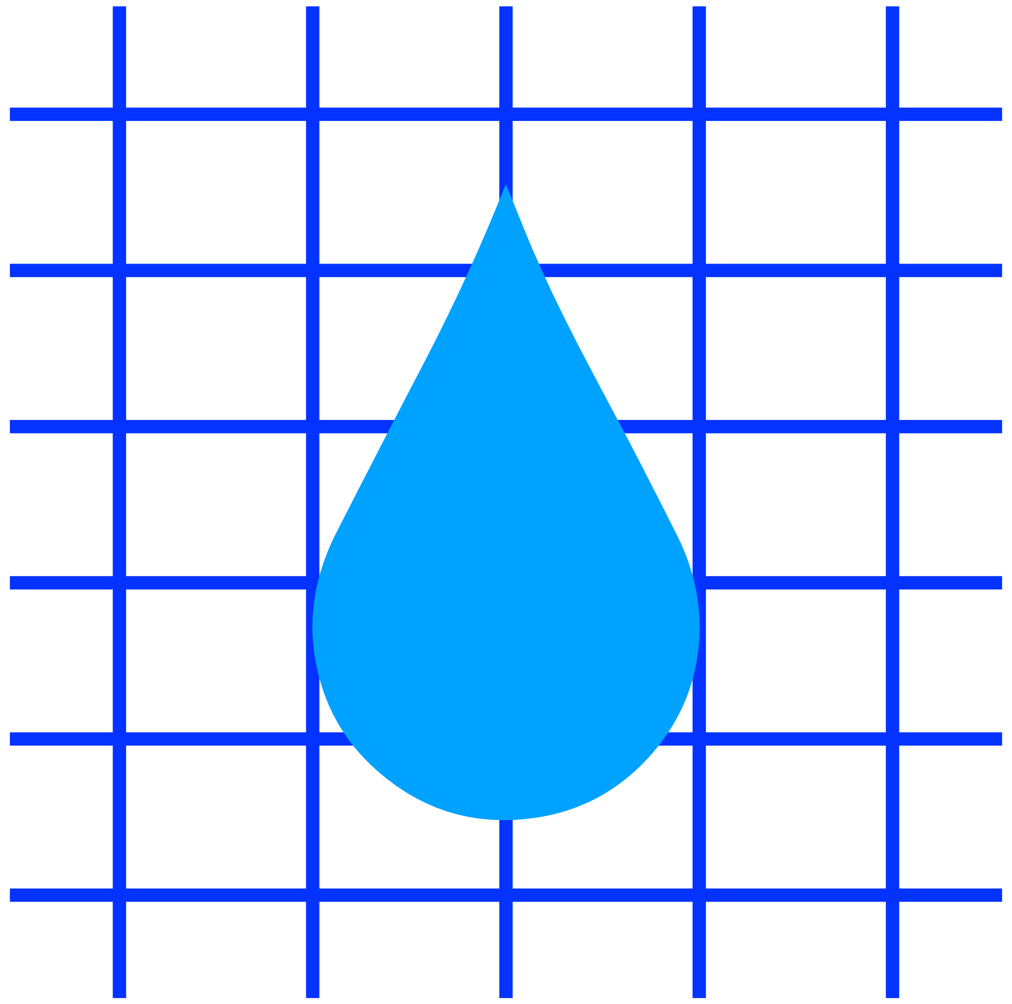


Grid

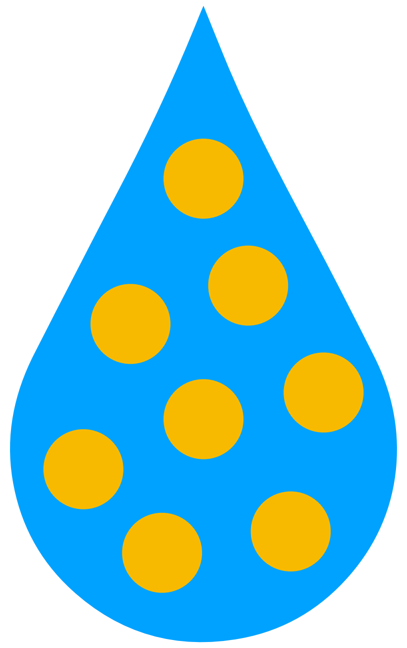


Particle

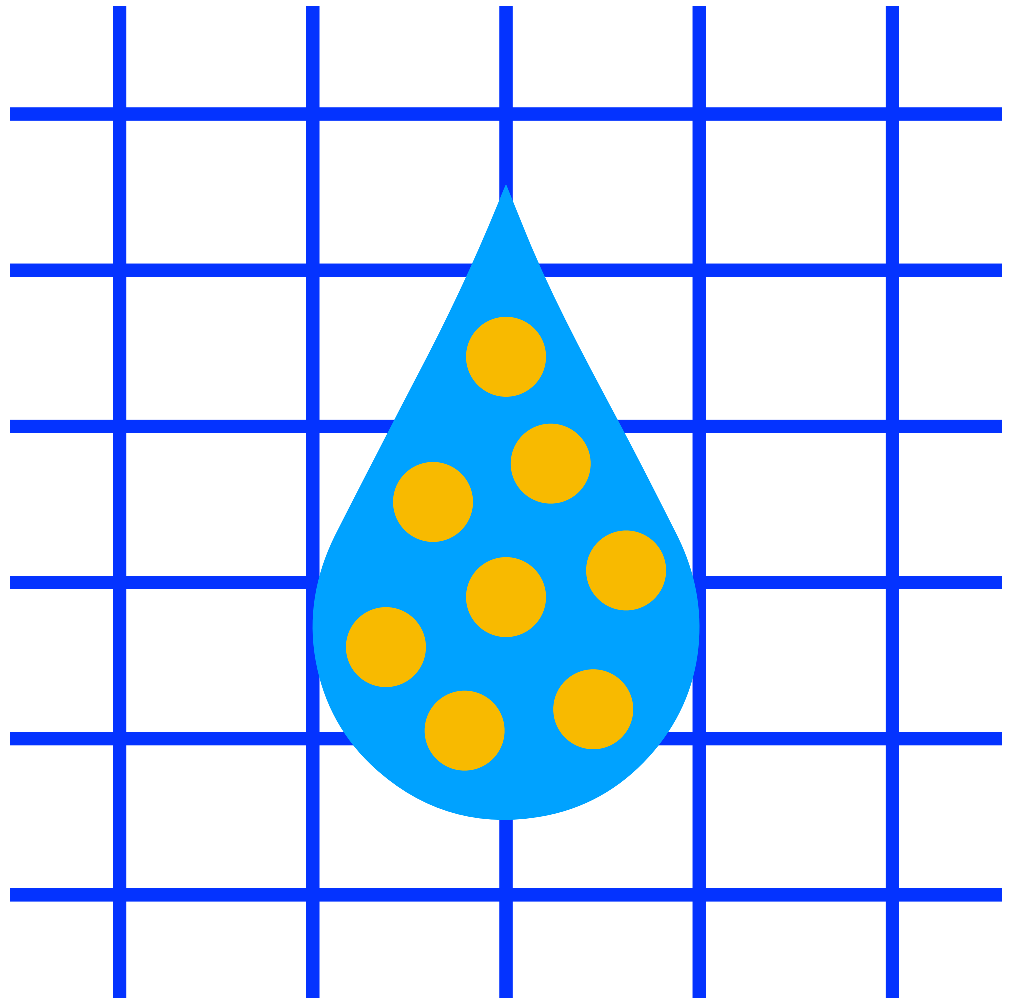
Discretization schemes



Grid

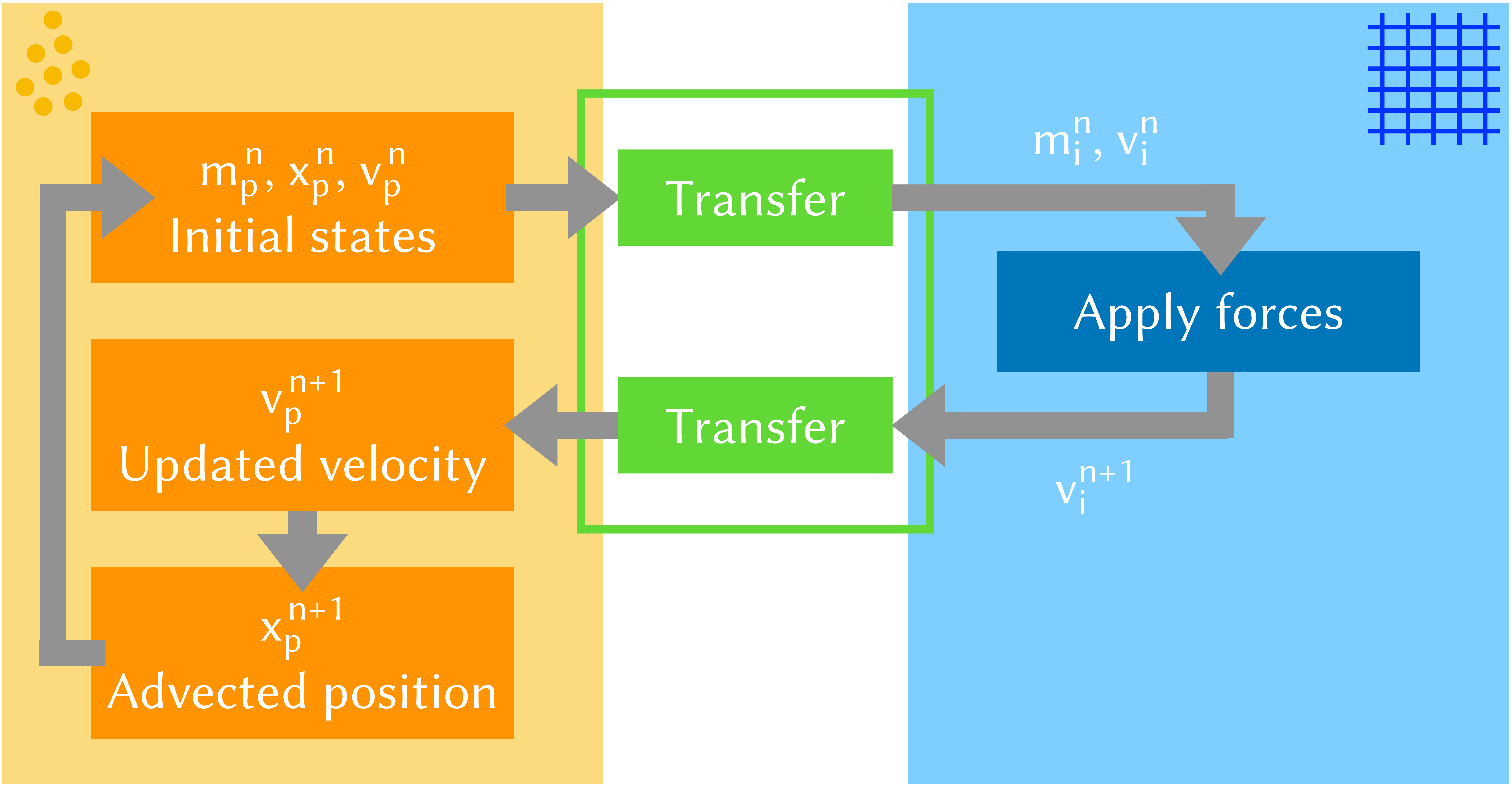


Particle



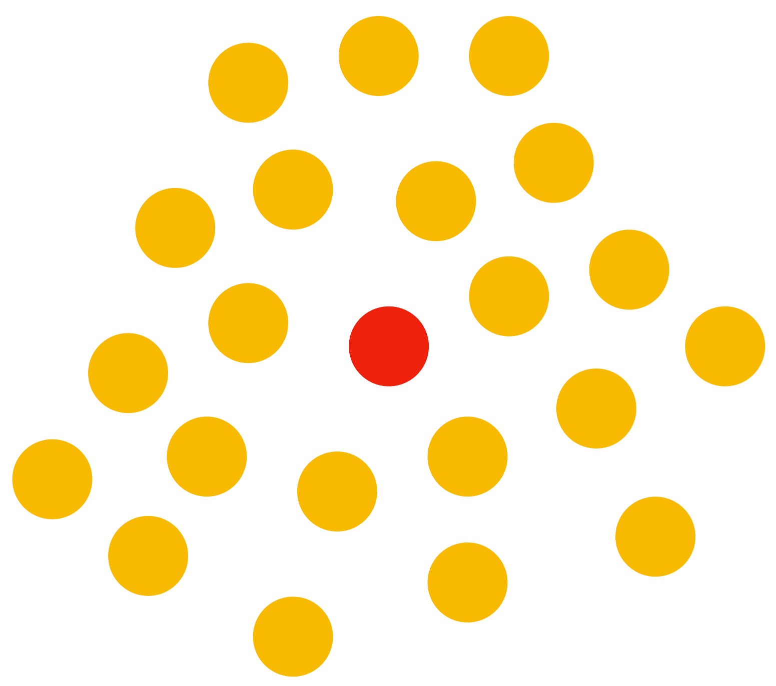
Hybrid

Data flow

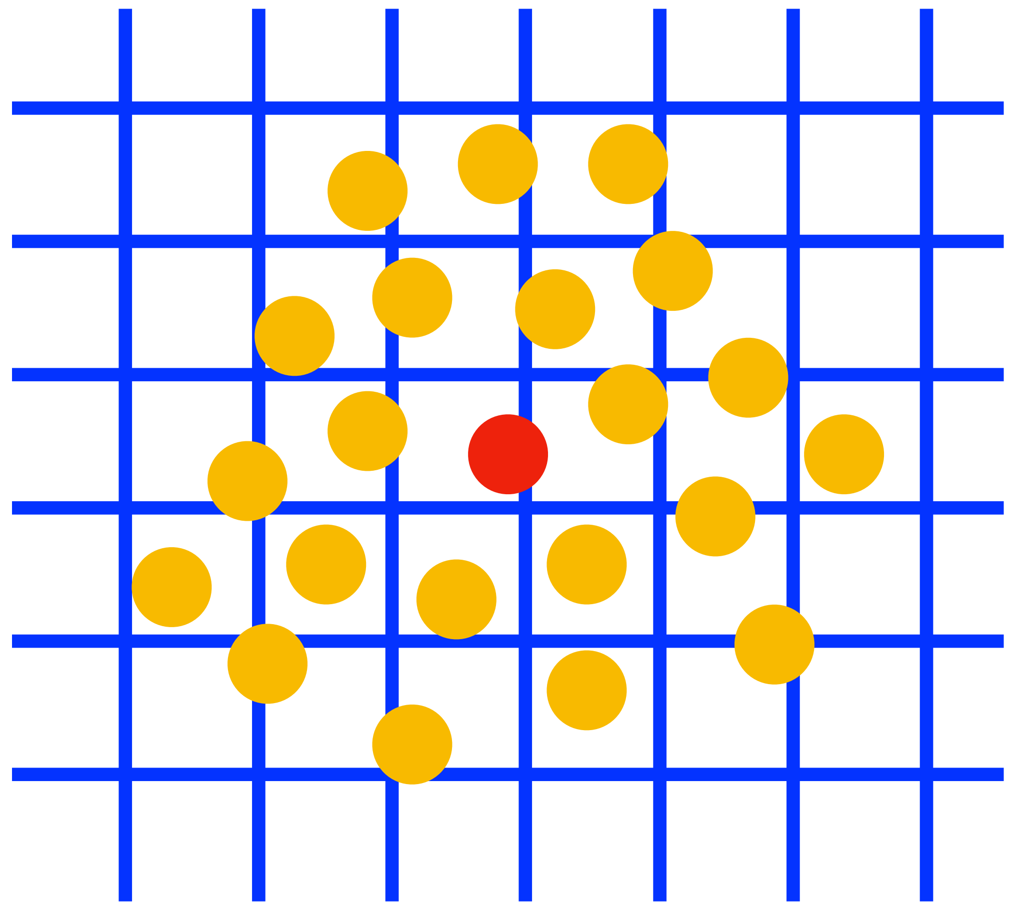


m_p^n, x_p^n, v_p^n
Initial states
Transfer
m_i^n, v_i^n
Apply forces
v_p^{n+1}
Updated velocity
Transfer
v_i^{n+1}
x_p^{n+1}
Advected position

Particle communication

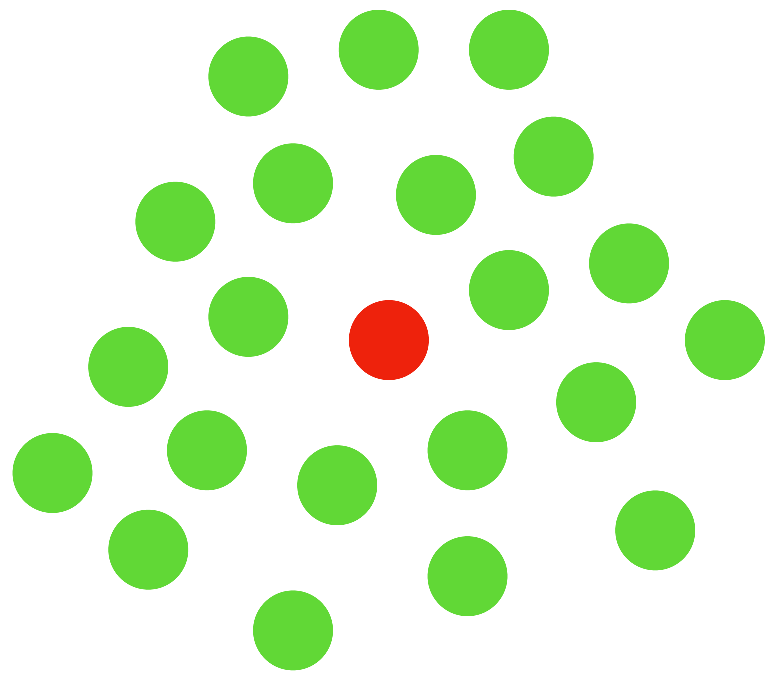


SPH

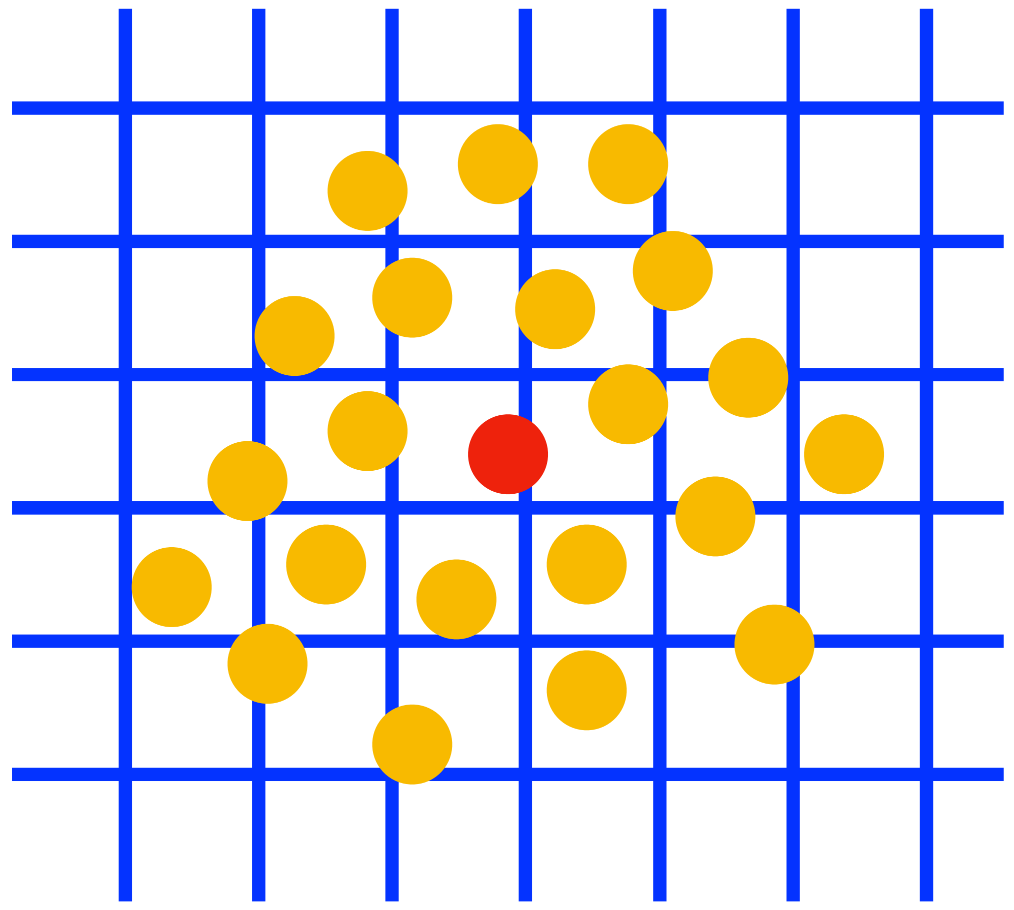


MPM

Particle communication

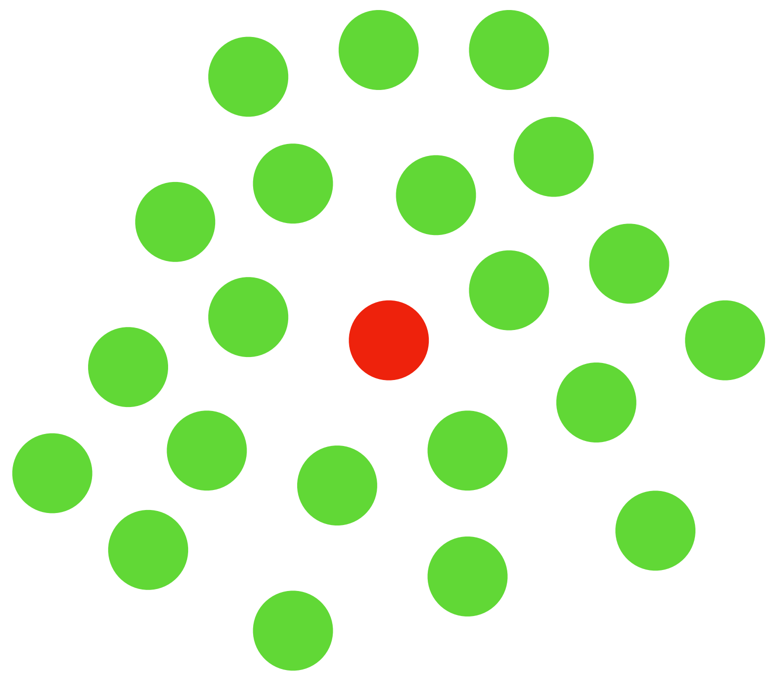


SPH

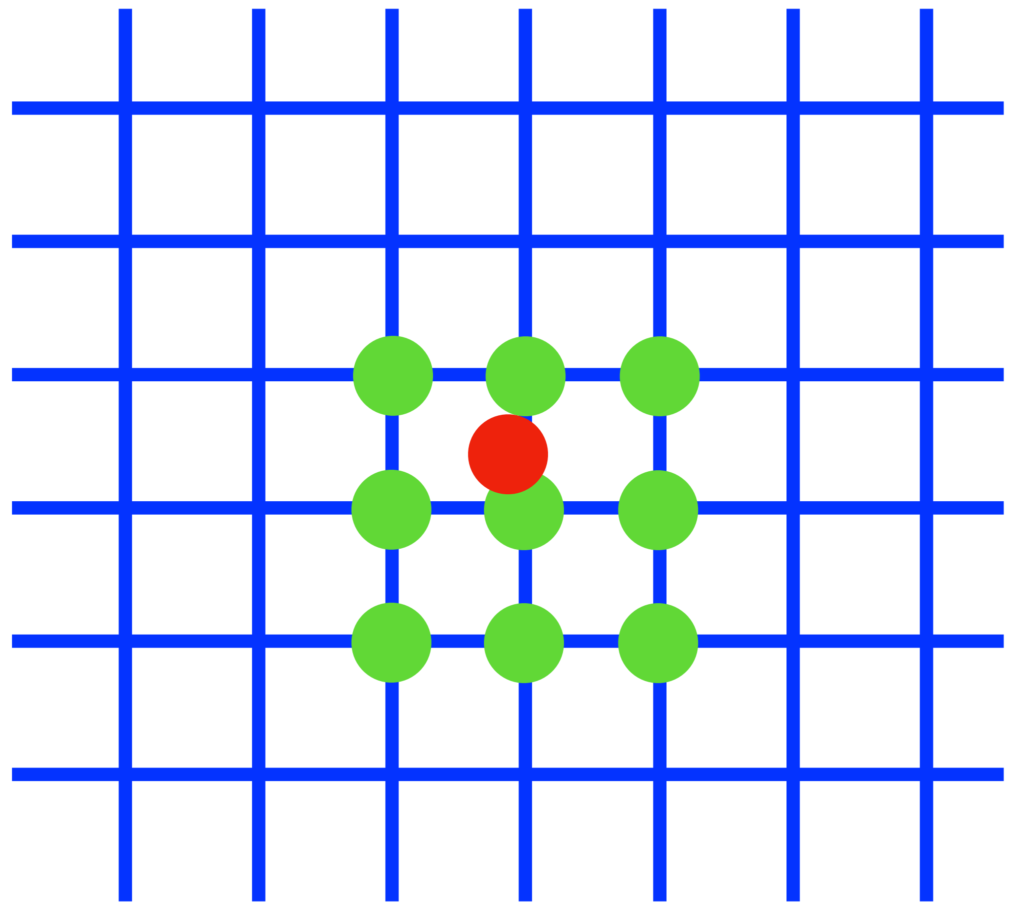


MPM

Particle communication

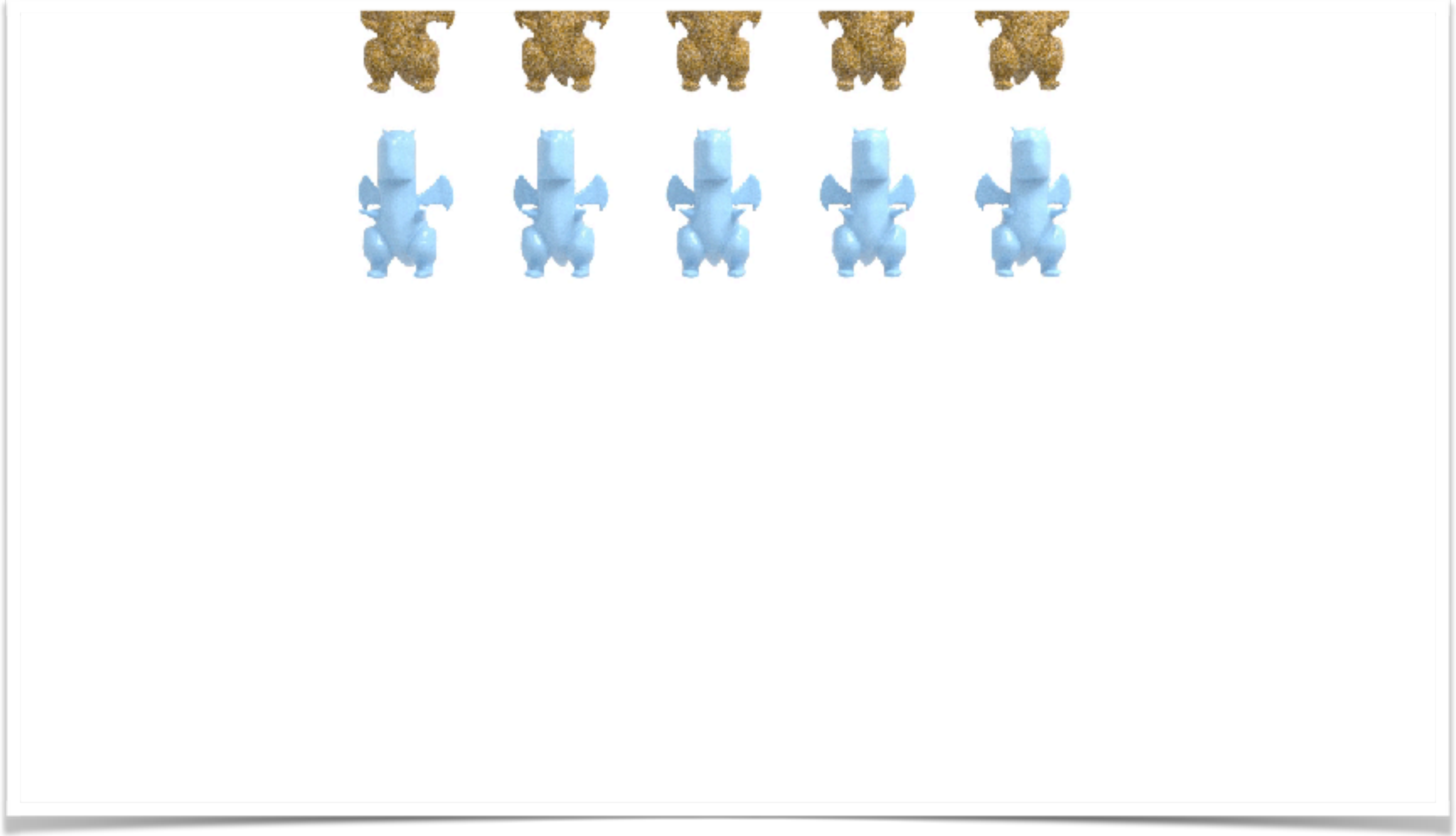


SPH

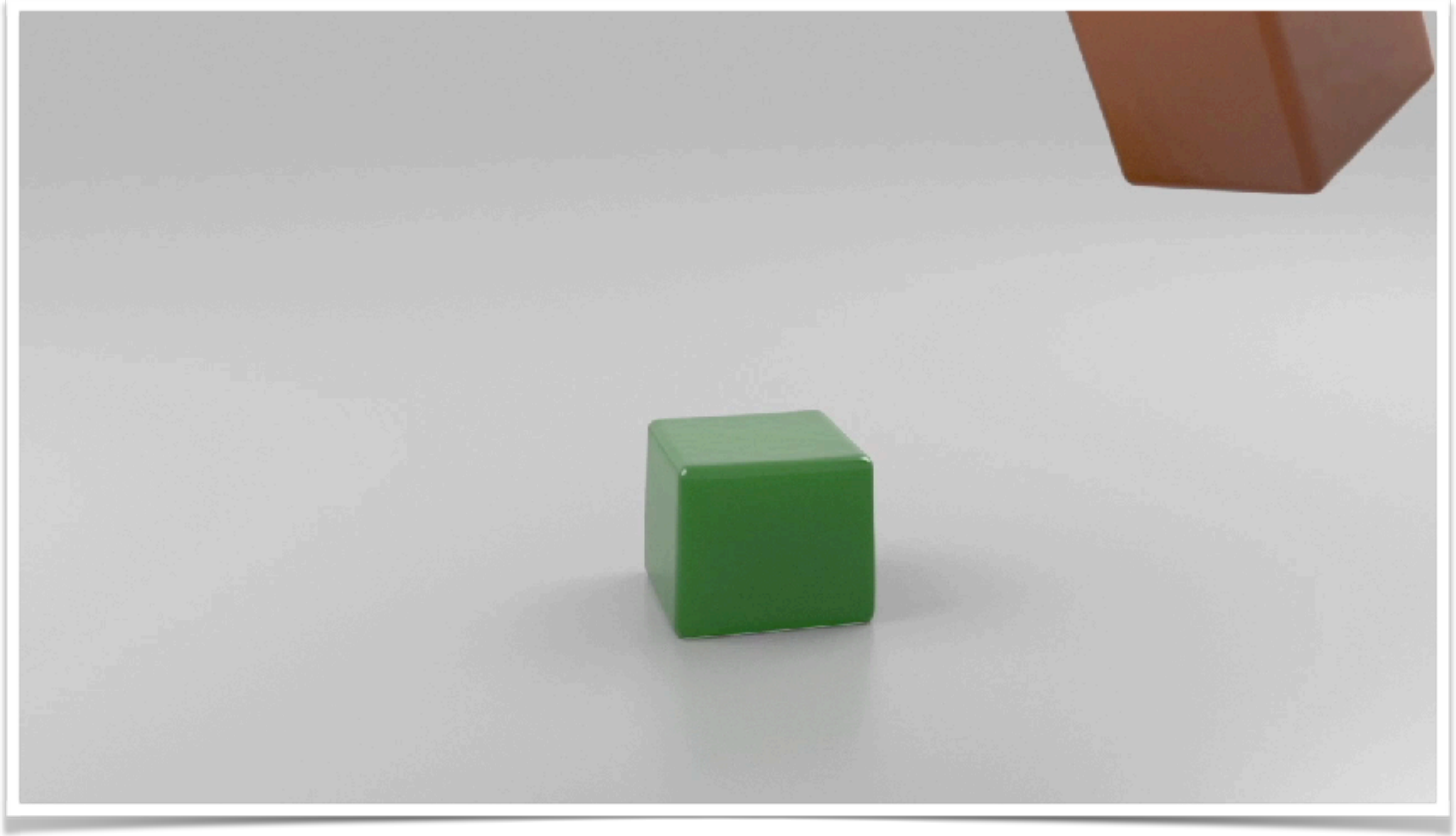


MPM

Sparsity



Gao et al. 18



Gao et al. 17

Power Diagrams and Sparse Paged Grids for High Resolution Adaptive Liquids

Power Diagrams and Sparse Paged Grids for High Resolution Adaptive Liquids

MRIDUL AANJANEYA[†*], Rutgers University
MING GAO[†] and HAIXIANG LIU, University of Wisconsin - Madison
CHRISTOPHER BATTY, University of Waterloo
EFTYCHIOS SIFAKIS, University of Wisconsin - Madison

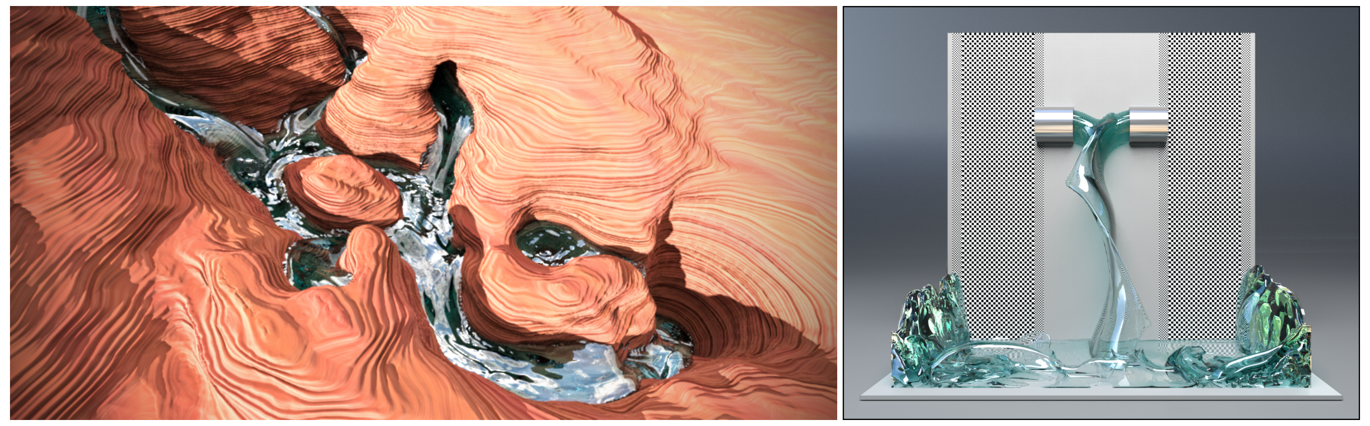


Fig. 1. (Left) Water filling a river bed surrounded by a canyon, with effective resolution $512^2 \times 1024$. Three refinement levels are used, based on proximity to the terrain. (Right) Sources inject water into a container and collide to form a thin sheet, with effective resolution 512^3. Adaptivity pattern shown on background.

We present an efficient and scalable octree-inspired fluid simulation framework with the flexibility to leverage adaptivity in any part of the computational domain, even when resolution transitions reach the free surface. Our methodology ensures symmetry, definiteness and second order accuracy of the discrete Poisson operator, and eliminates numerical and visual artifacts of prior octree schemes. This is achieved by adapting the operators acting on the octree's simulation variables to reflect the structure and connectivity of a *power diagram*, which recovers primal-dual mesh orthogonality and eliminates problematic T-junction configurations. We show how such operators can be efficiently implemented using a pyramid of sparsely populated uniform grids, enhancing the regularity of operations and facilitating parallelization. A novel scheme is proposed for encoding the topology of the power diagram in the neighborhood of each octree cell, allowing us to locally reconstruct it on the fly via a lookup table, rather than resorting to costly explicit meshing. The pressure Poisson equation is solved via a highly efficient, matrix-free multigrid preconditioner for Conjugate Gradient, adapted to the power diagram discretization. We use another sparsely populated uniform grid for high resolution interface tracking with a narrow band level set representation. Using the recently introduced SPGrid data structure, sparse uniform grids in both the power diagram discretization and our narrow band level set can be compactly stored and efficiently updated via streaming operations. Additionally, we present enhancements to adaptive level set advection, velocity extrapolation, and the fast marching method for redistancing. Our overall framework gracefully accommodates the task of dynamically adapting the octree topology during simulation. We demonstrate end-to-end simulations of complex adaptive flows in irregularly shaped domains, with tens of millions of degrees of freedom.

CCS Concepts: • **Computing methodologies → Physical simulation**;

Additional Key Words and Phrases: Power diagrams, Octrees, Adaptivity

ACM Reference format:
Mridul Aanjaneya, Ming Gao, Haixiang Liu, Christopher Batty, and Eftychios Sifakis. 2017. Power Diagrams and Sparse Paged Grids for High Resolution Adaptive Liquids. *ACM Trans. Graph.* 36, 4, Article 140 (July 2017), 12 pages.
DOI: http://dx.doi.org/10.1145/3072959.3073625

1 INTRODUCTION

Liquids exhibit complex and detailed motion across a vast range of scales, from tiny ripples to huge waves; this fact motivates the desire for liquid simulation tools that can handle ever increasing levels of resolution. While a key avenue towards this goal is the development of more efficient numerical methods on regular uniform grids that conserve mass with large time steps [Chentanez and Müller 2012; Lentine et al. 2011, 2012] and allow for fast pressure projection [Ando et al. 2015; Dick et al. 2016; Lentine et al. 2010; Liu

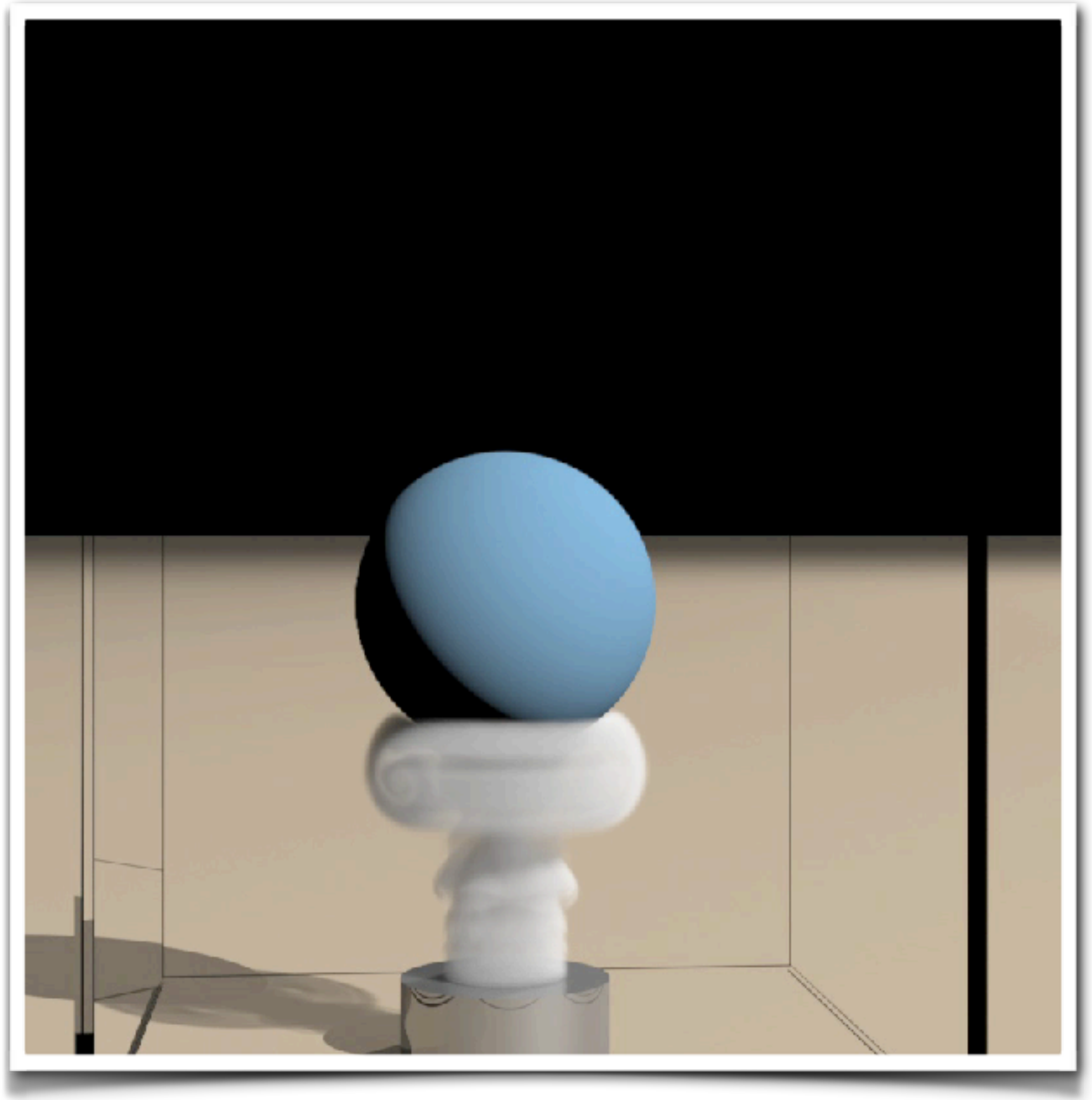
[†] M. Aanjaneya and M. Gao are joint first authors.
[*] M. Aanjaneya was with the University of Wisconsin - Madison during this work.
This work was supported in part by National Science Foundation grants IIS-1253598, CCF-1423064, CCF-1533885 and by the Natural Sciences and Engineering Research Council of Canada under grant RGPIN-04360-2014. The authors are grateful to Nathan Mitchell for his indispensable help with modeling and rendering of examples. C. Batty would like to thank Ted Ying for carrying out preliminary explorations on quadtrees.

DOI: http://dx.doi.org/10.1145/3072959.3073625

M. Aanjaneya*, **M. Gao* (joint first authors)**, H. Liu, C. Batty, E. Safaris
ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), 2017

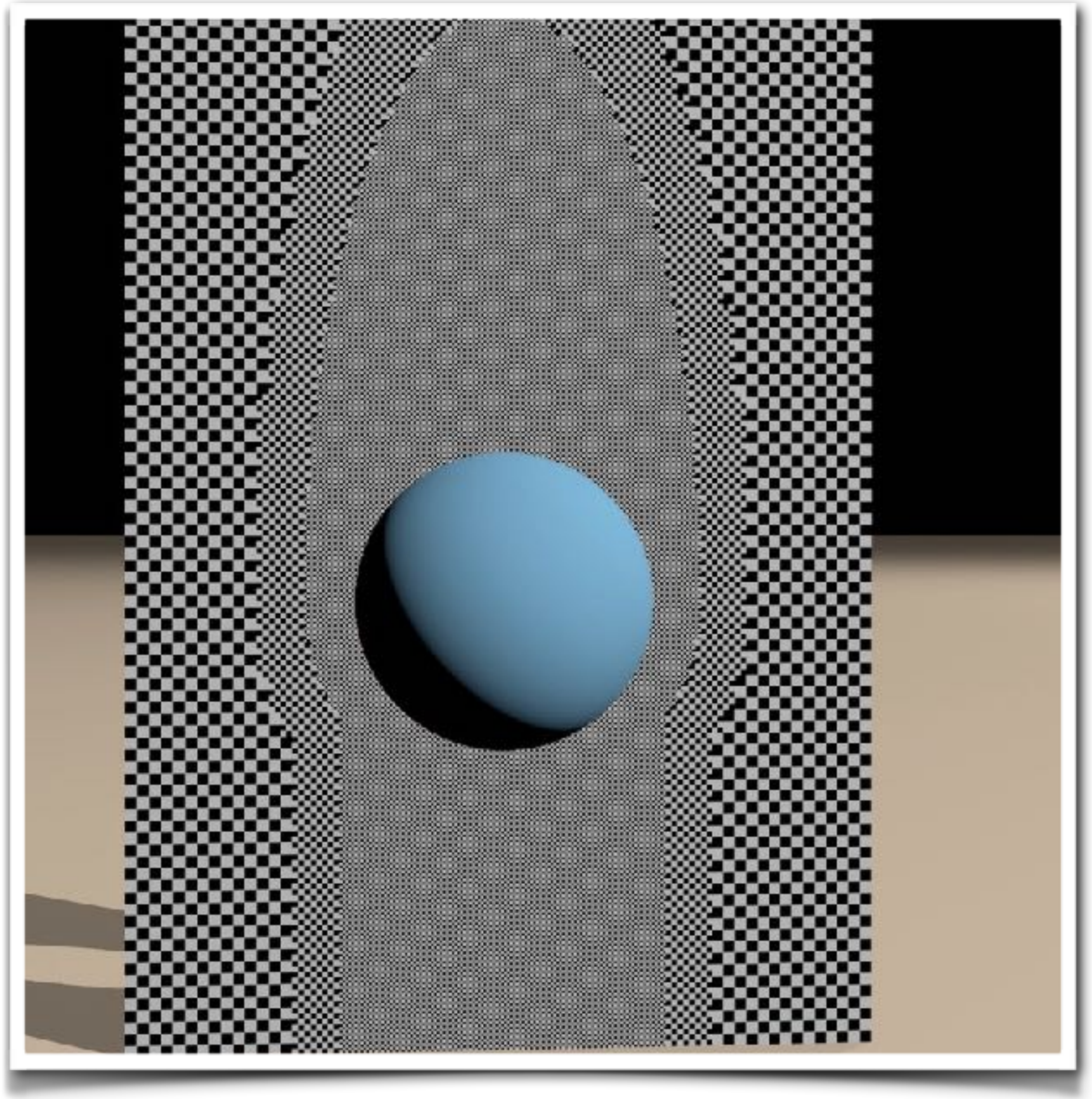
30 JULY – 3 AUGUST Los Angeles
SIGGRAPH 2017

Previous work

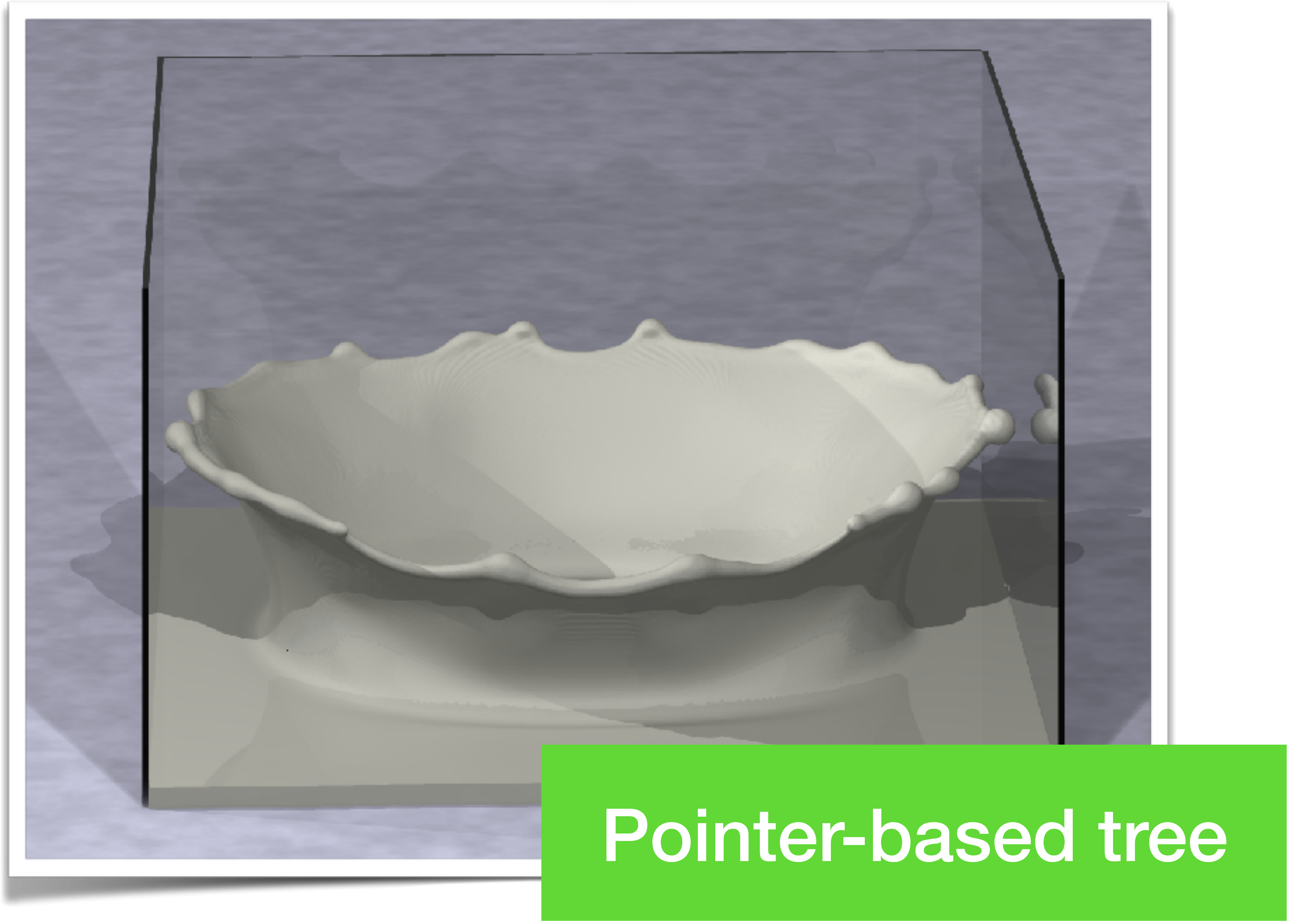


Setaluri et al. 14

Previous work

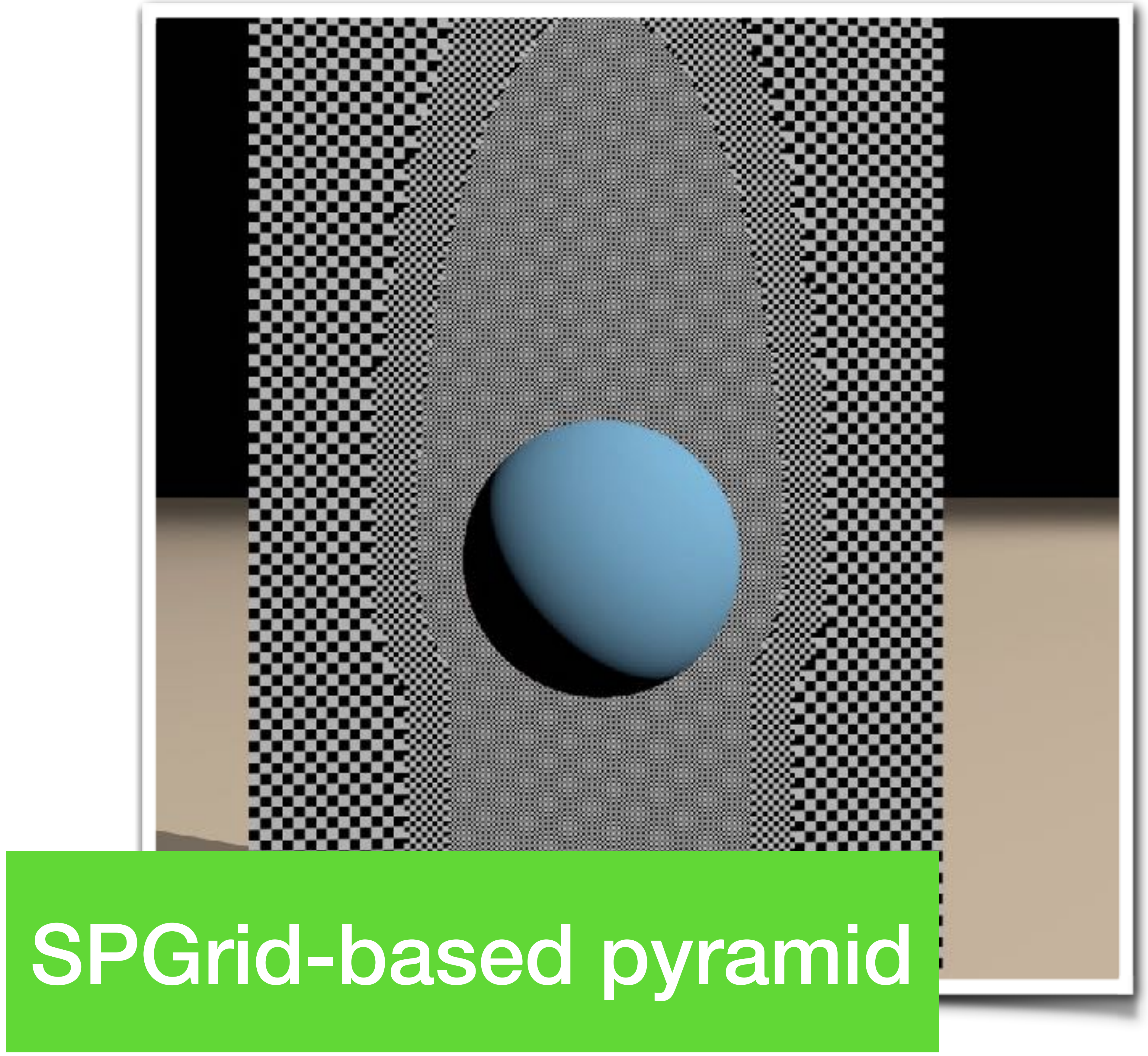


Setaluri et al. 14

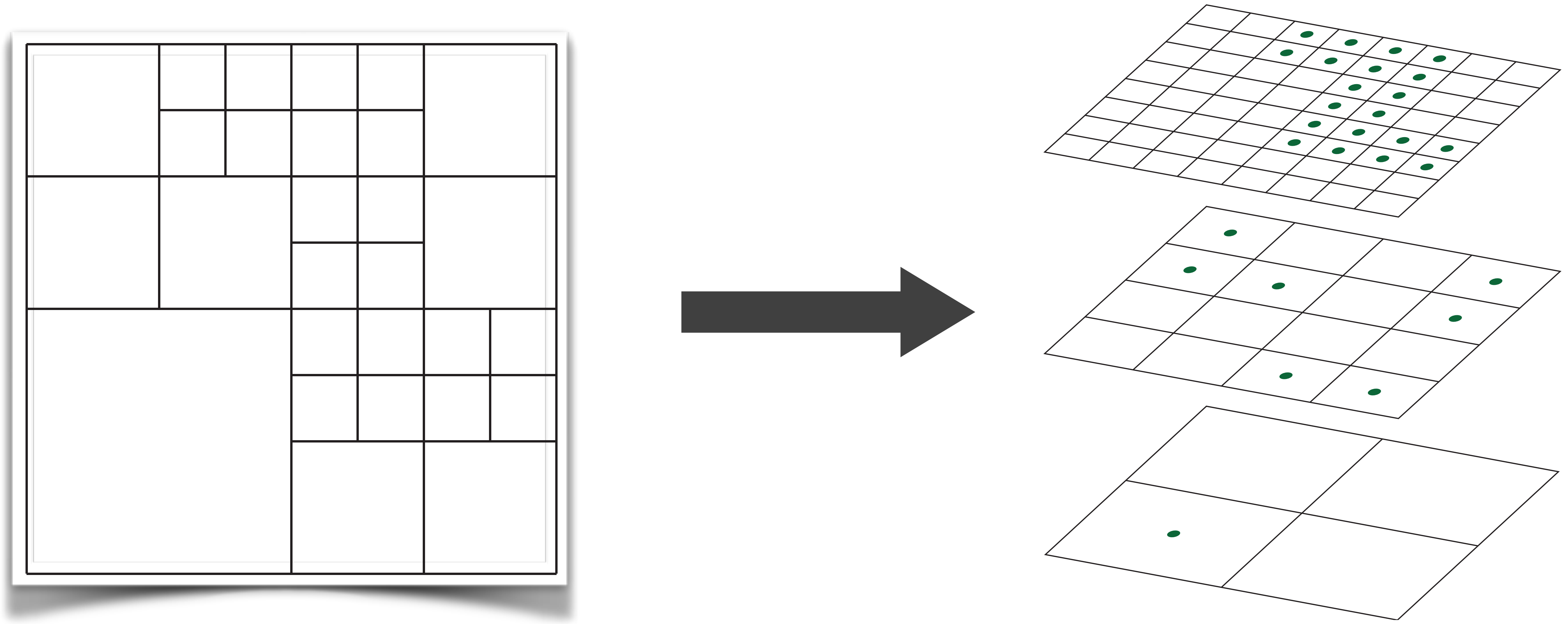


Losasso et al. 04

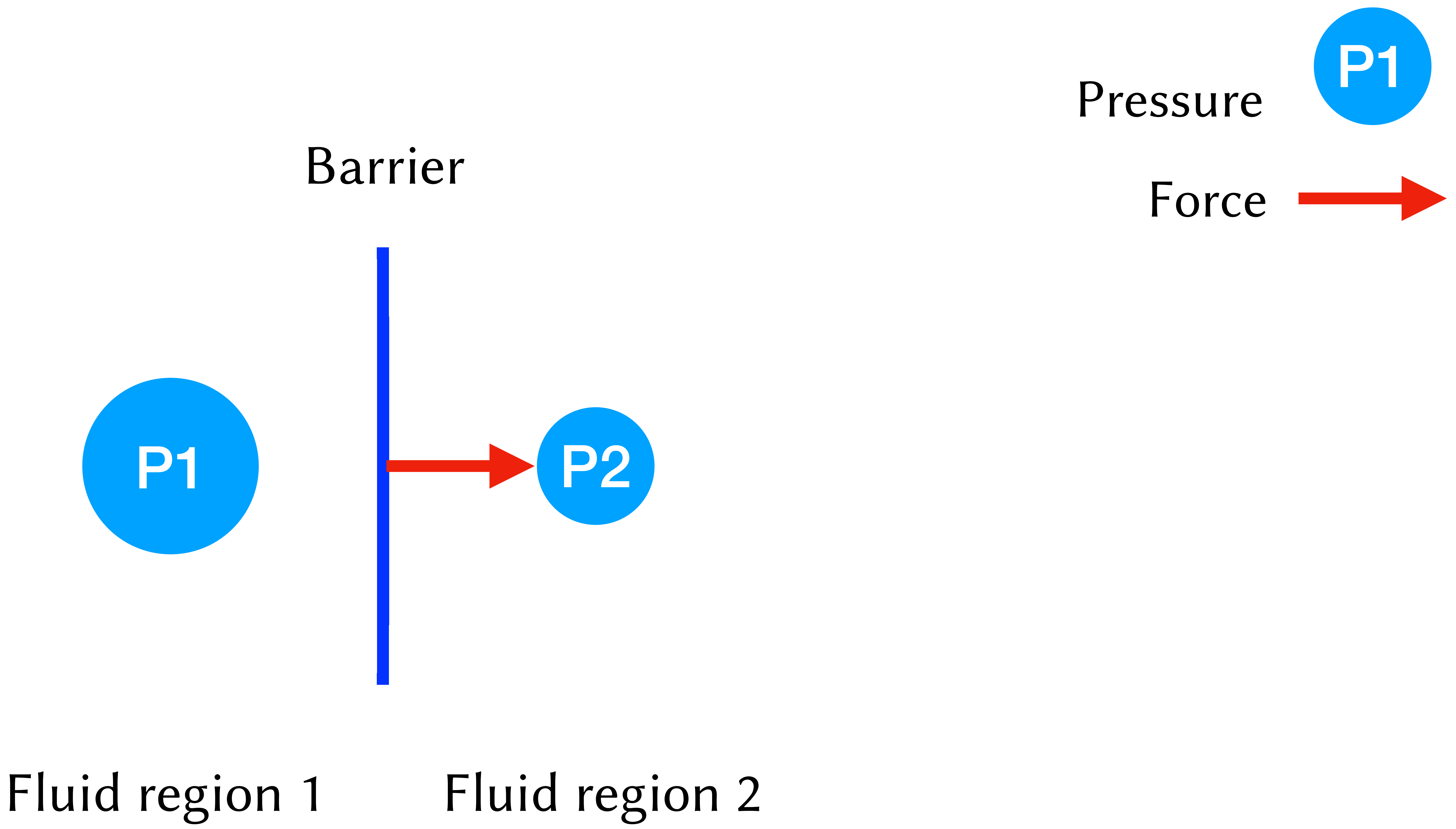
Previous work



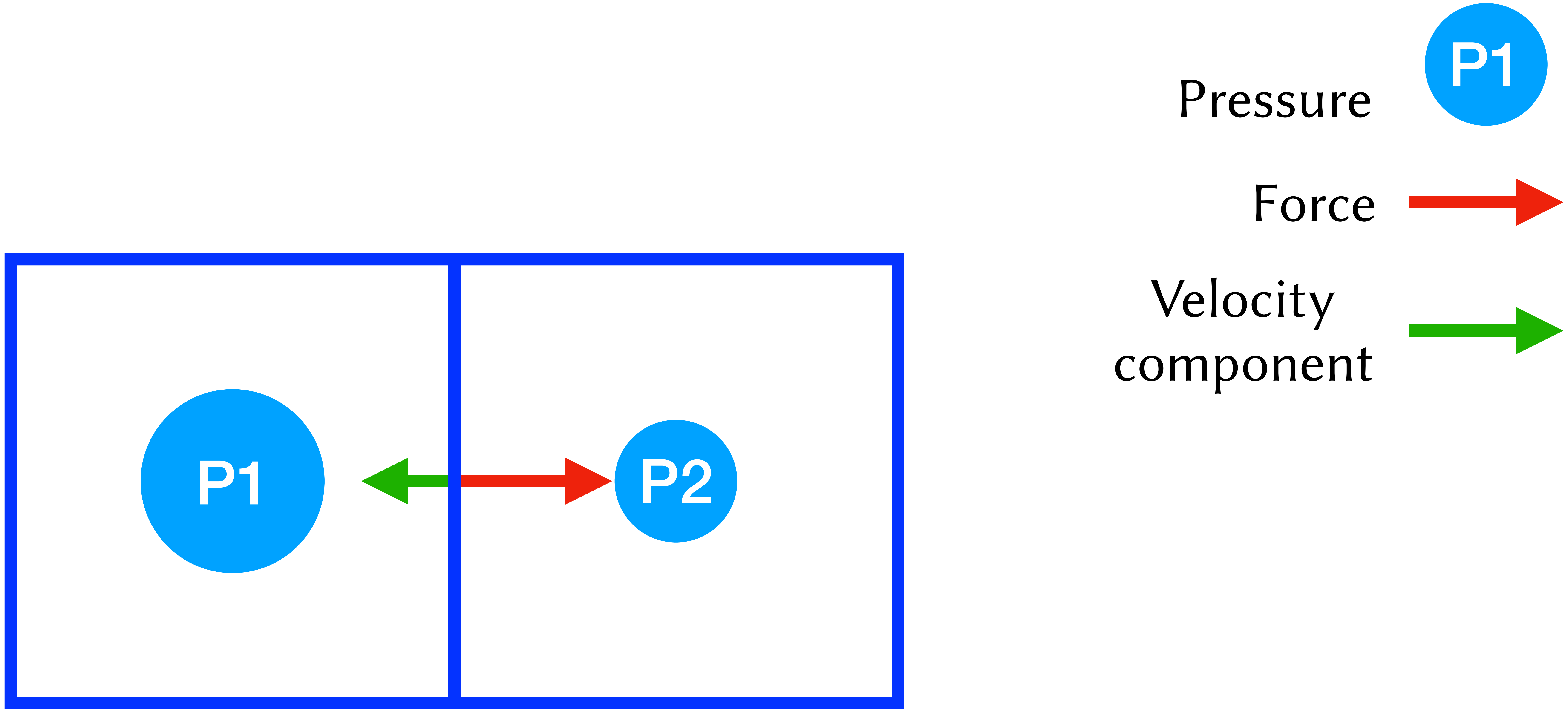
Setaluri et al. 14



Loss of orthogonality

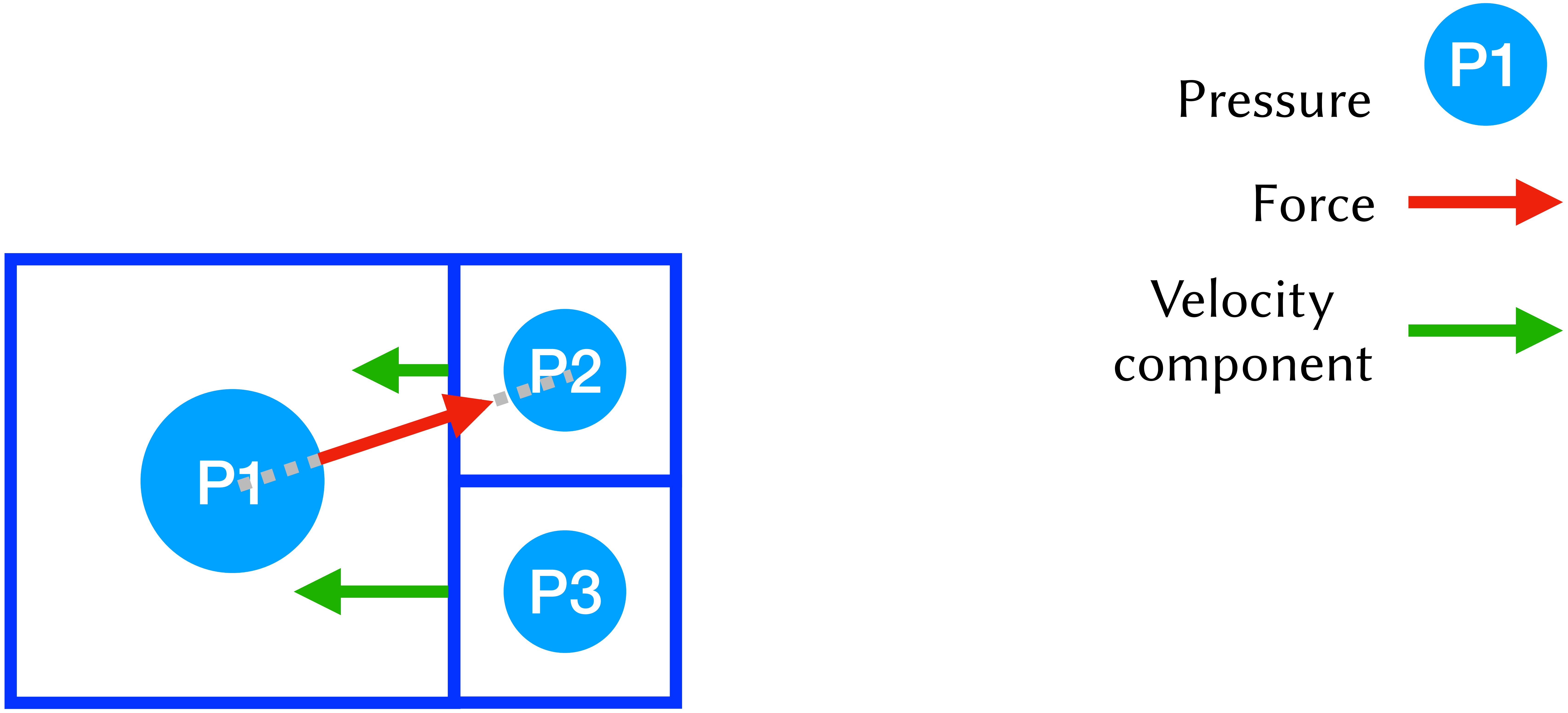


Loss of orthogonality

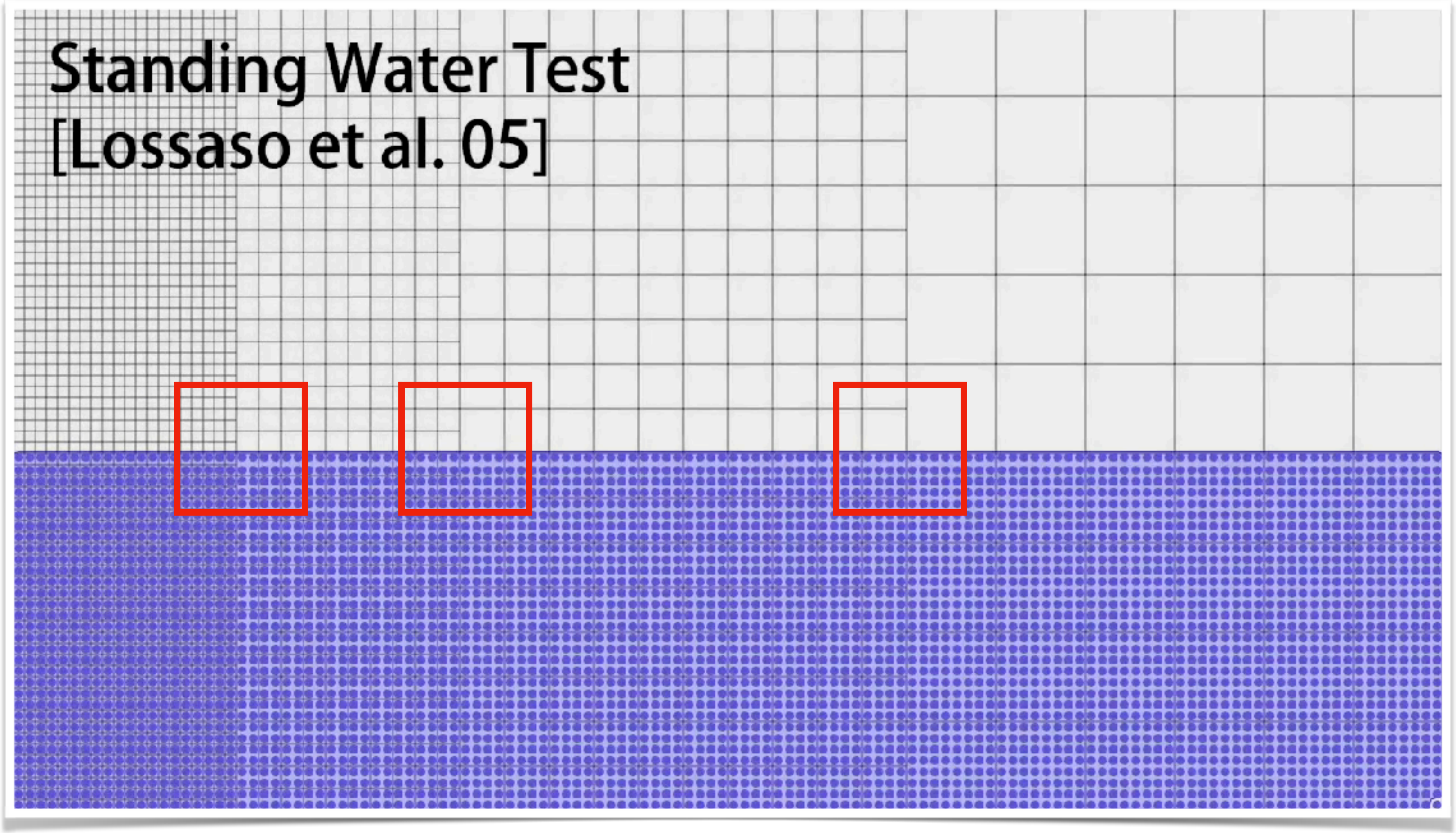


Staggered grid

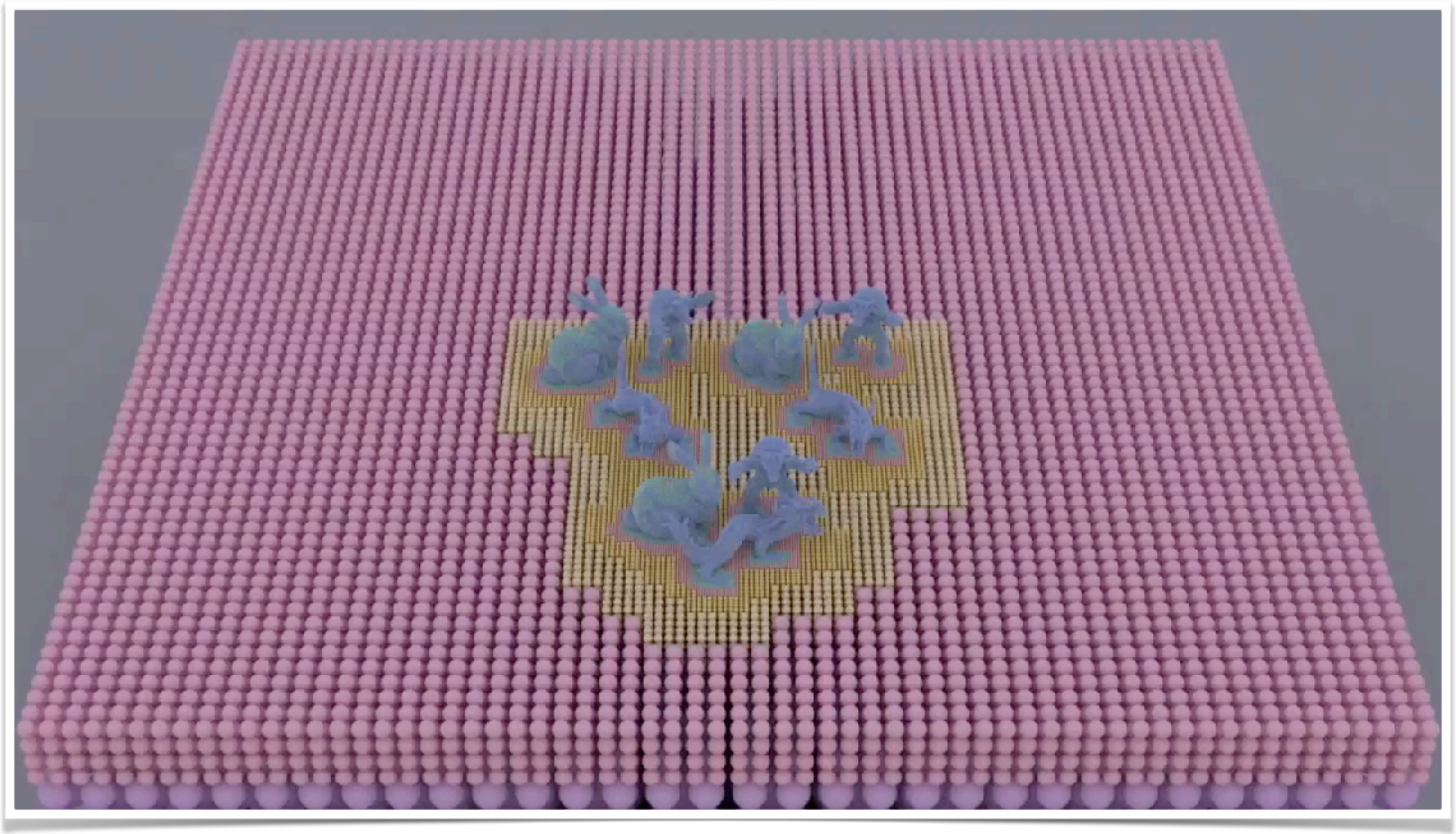
Loss of orthogonality



Spurious motions

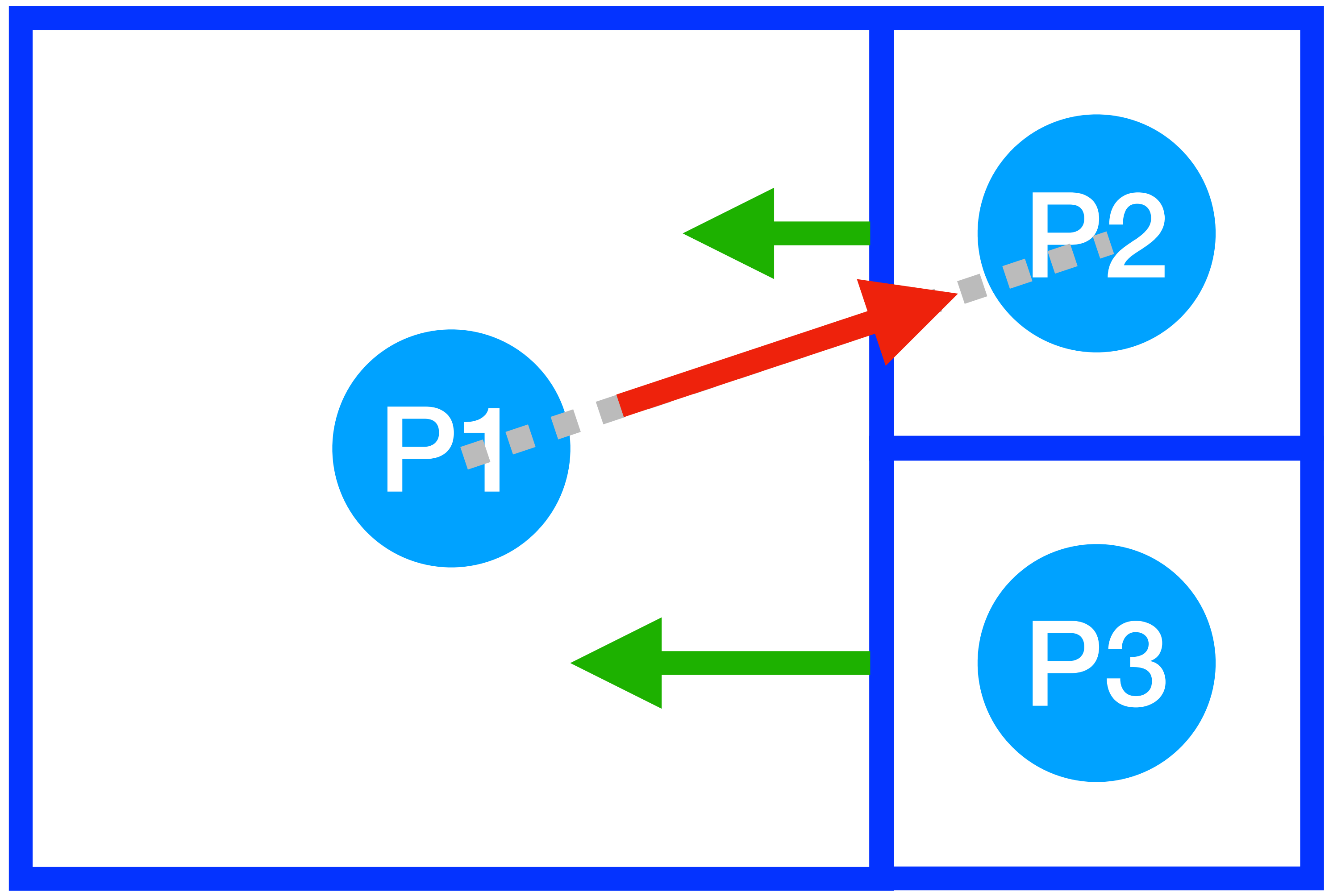


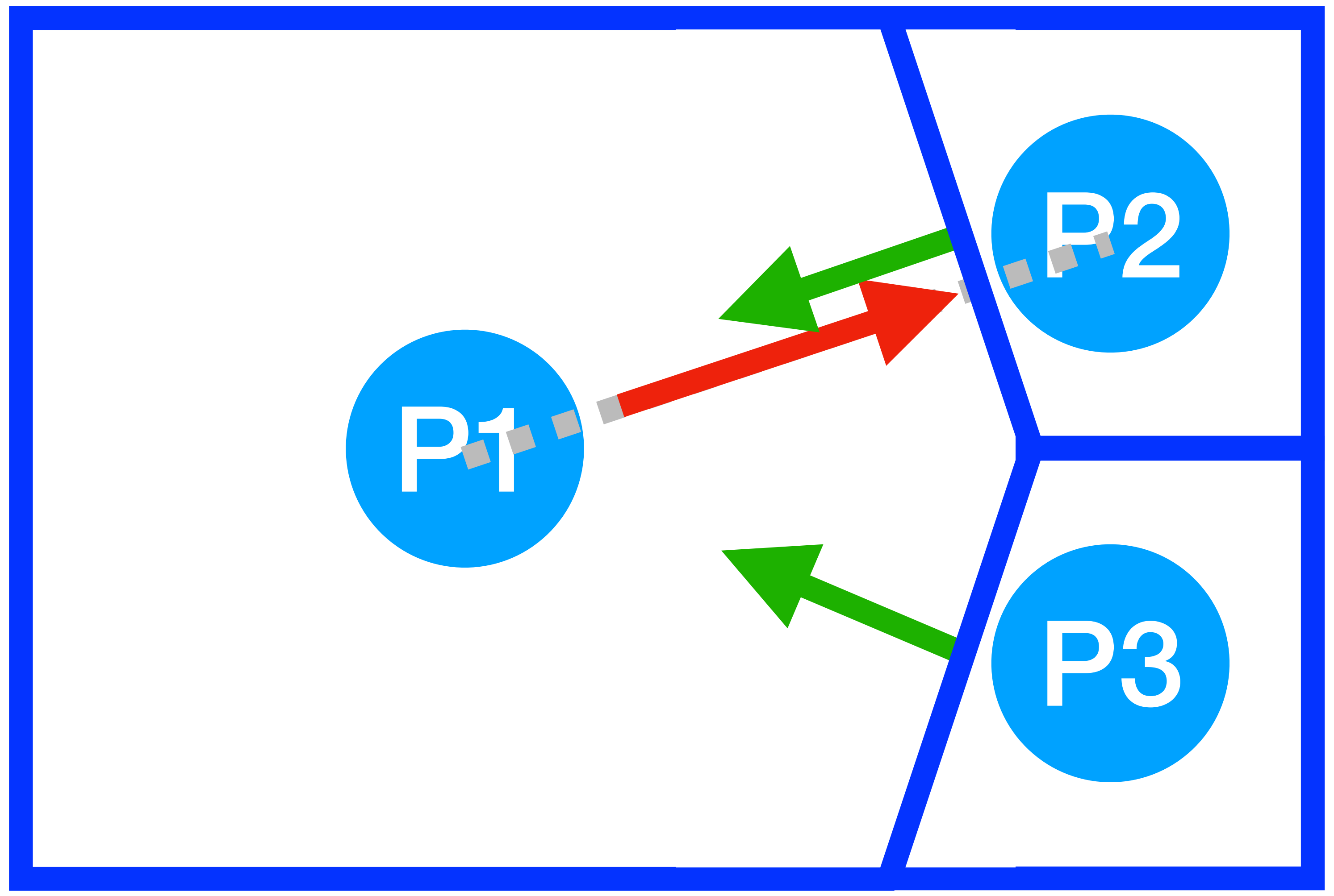
Unstructured mesh

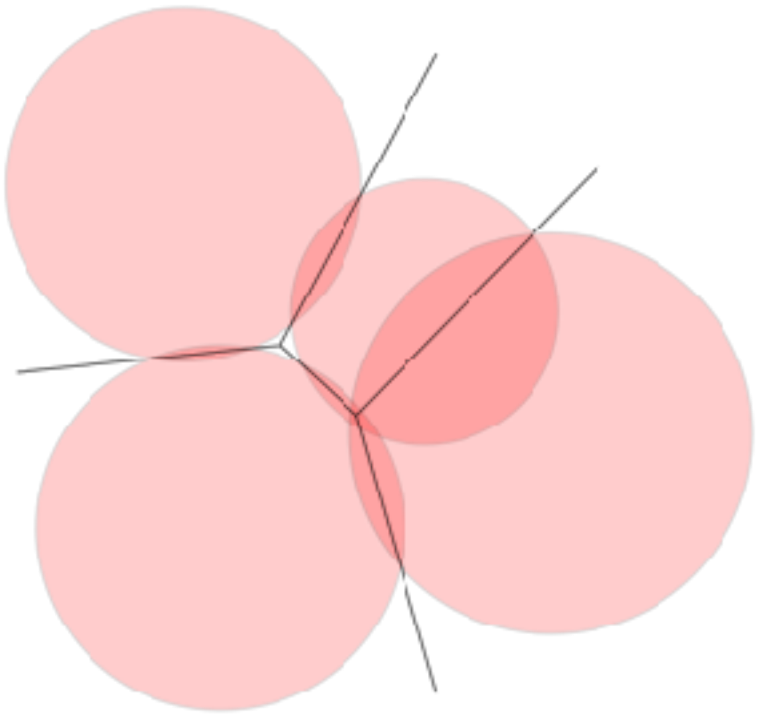


Ando et al. 13

Our solution: power diagram



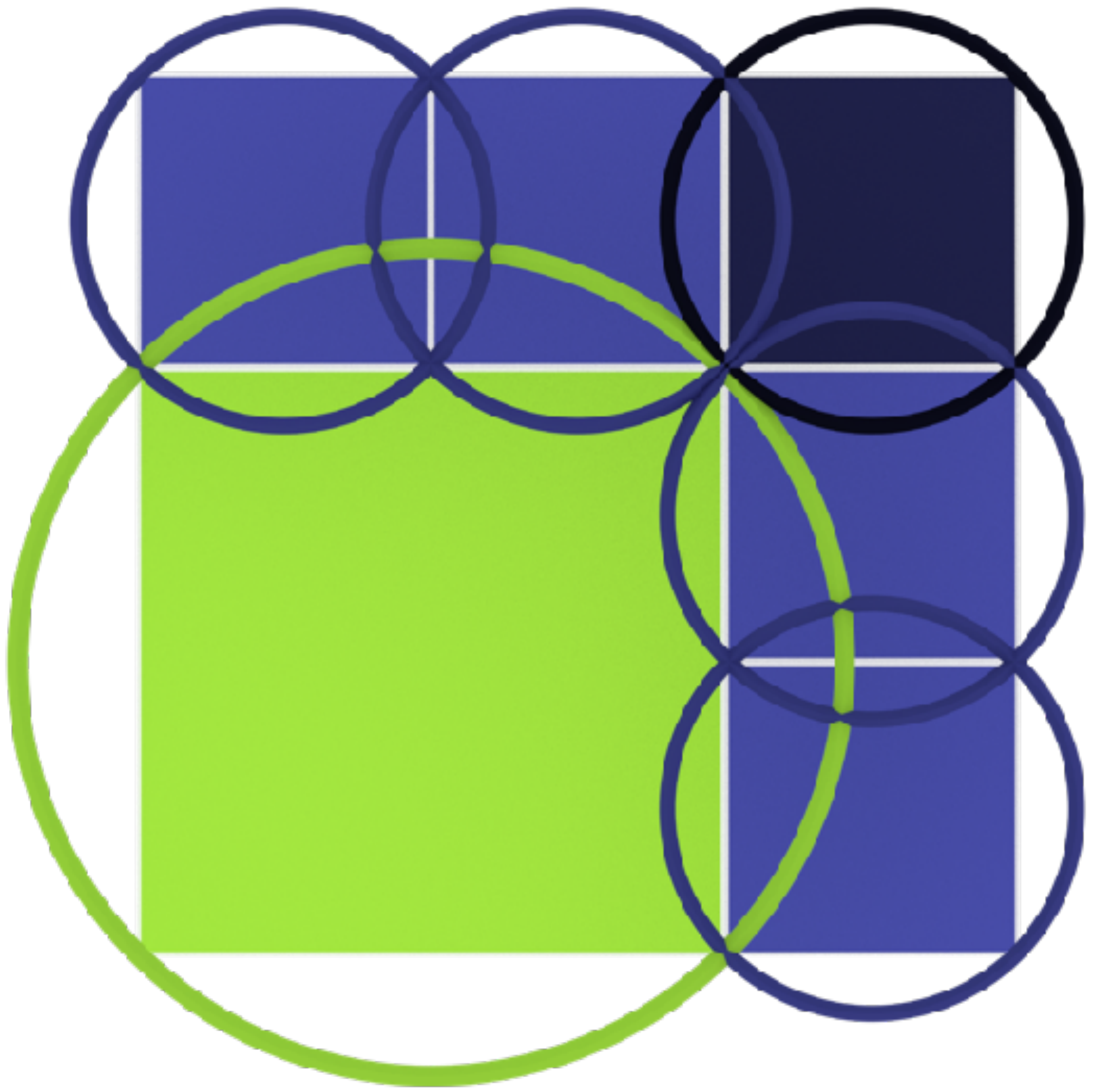




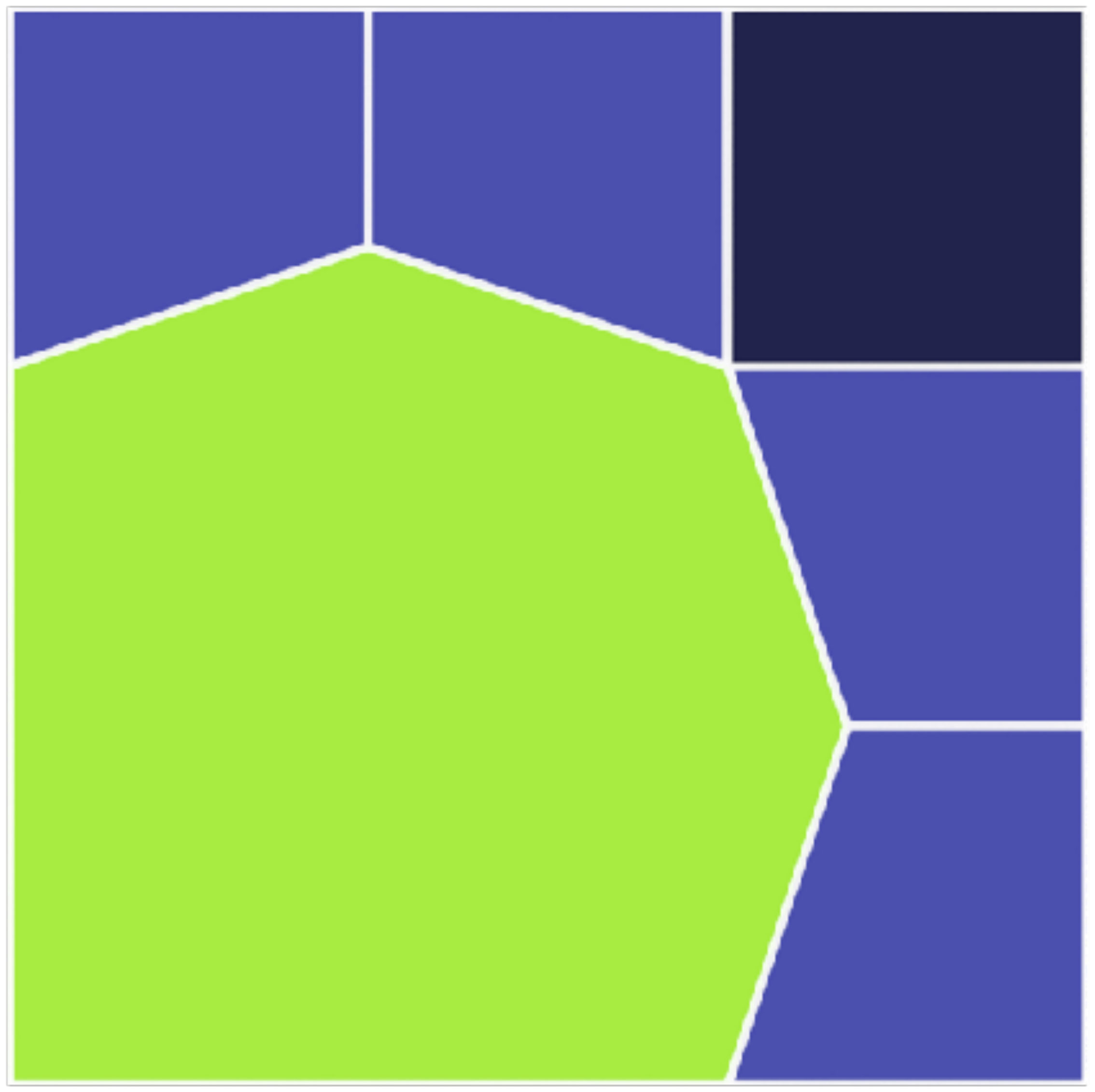
Wiki

Comparison

Original topology

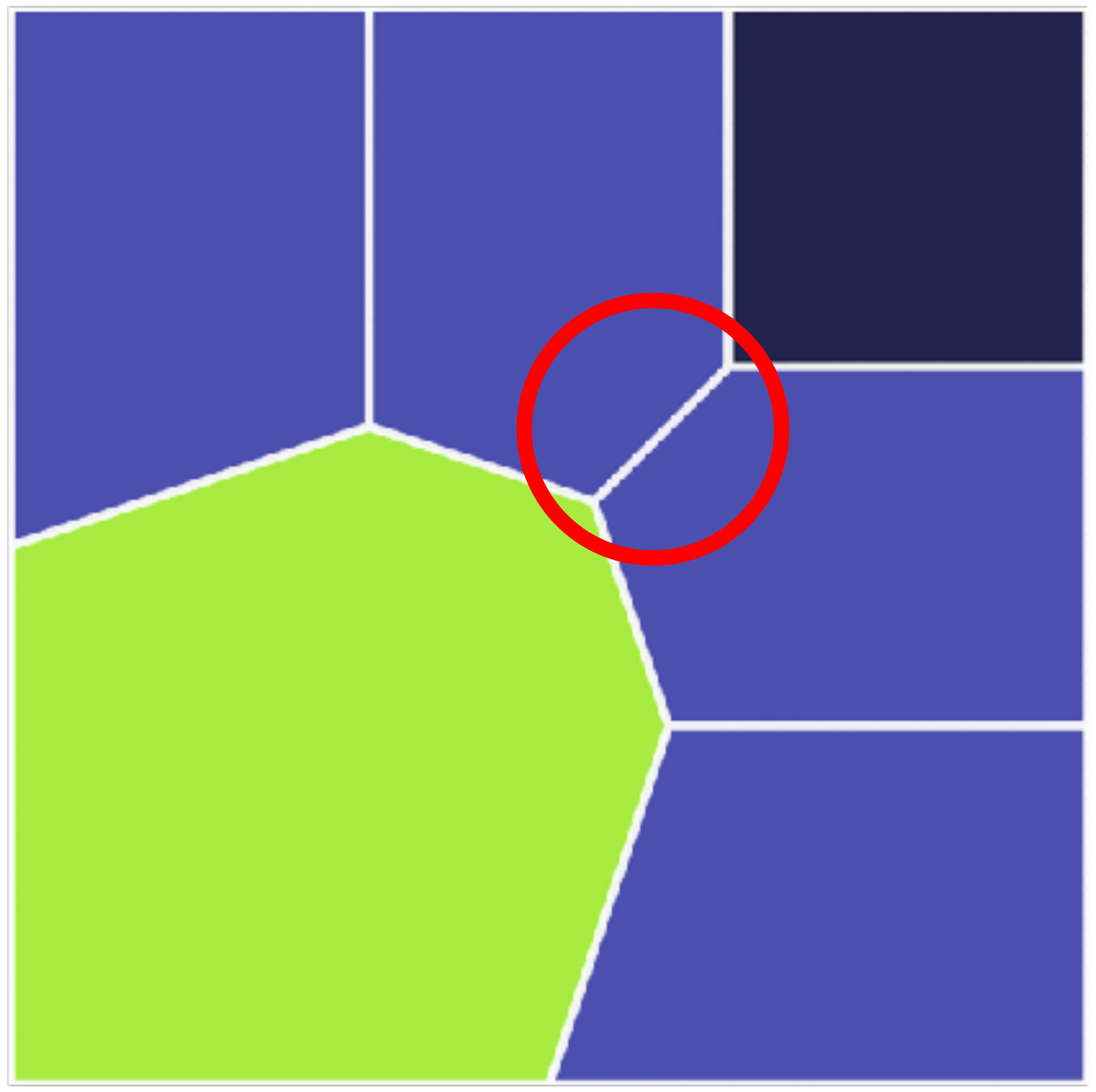


Power diagram



$d^2 - r^2$

Voronoi diagram



d^2

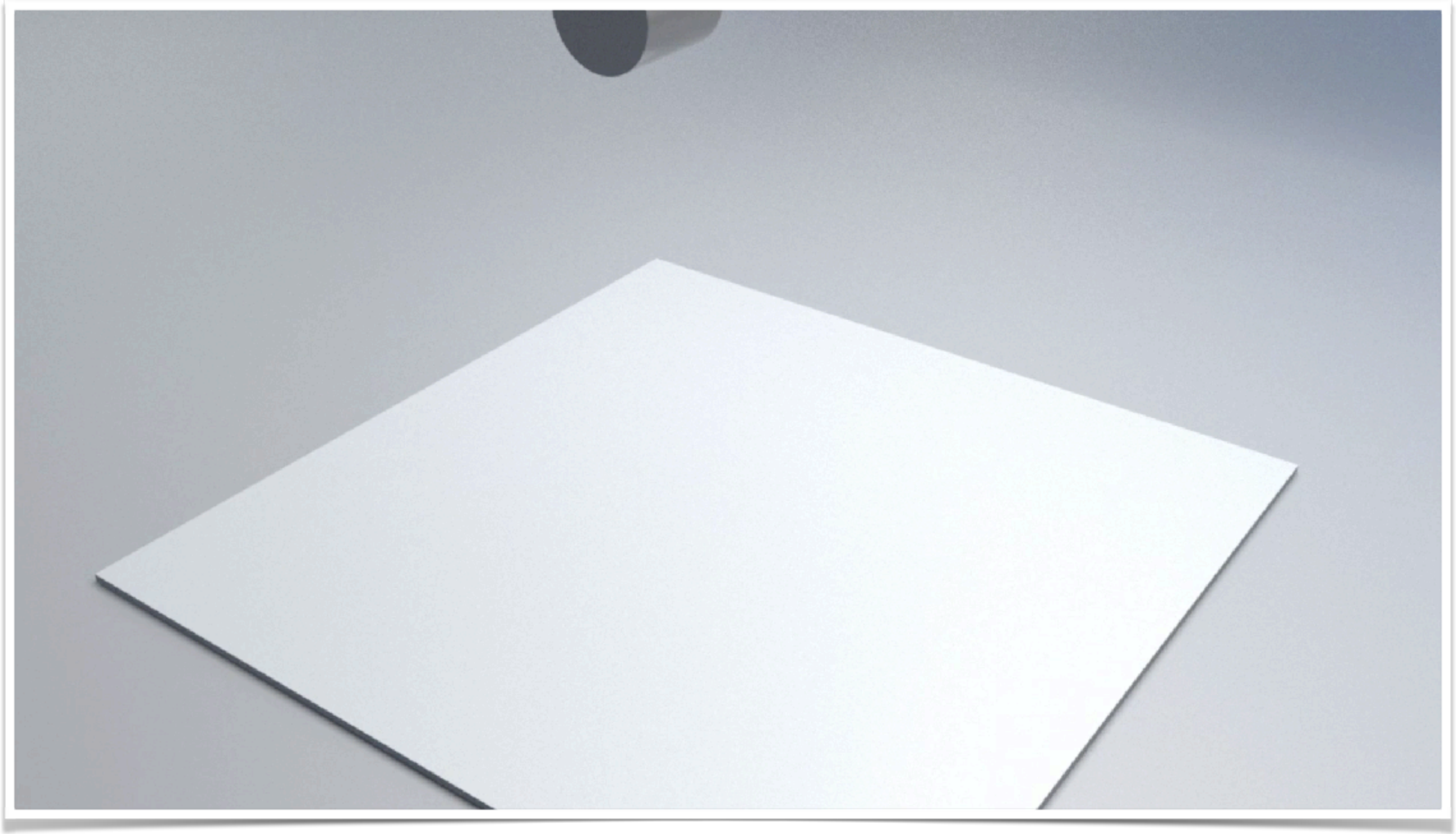
Opportunities

- Power diagram ensures orthogonality
- Can still use octree for storage
- Accelerate via SPGrid

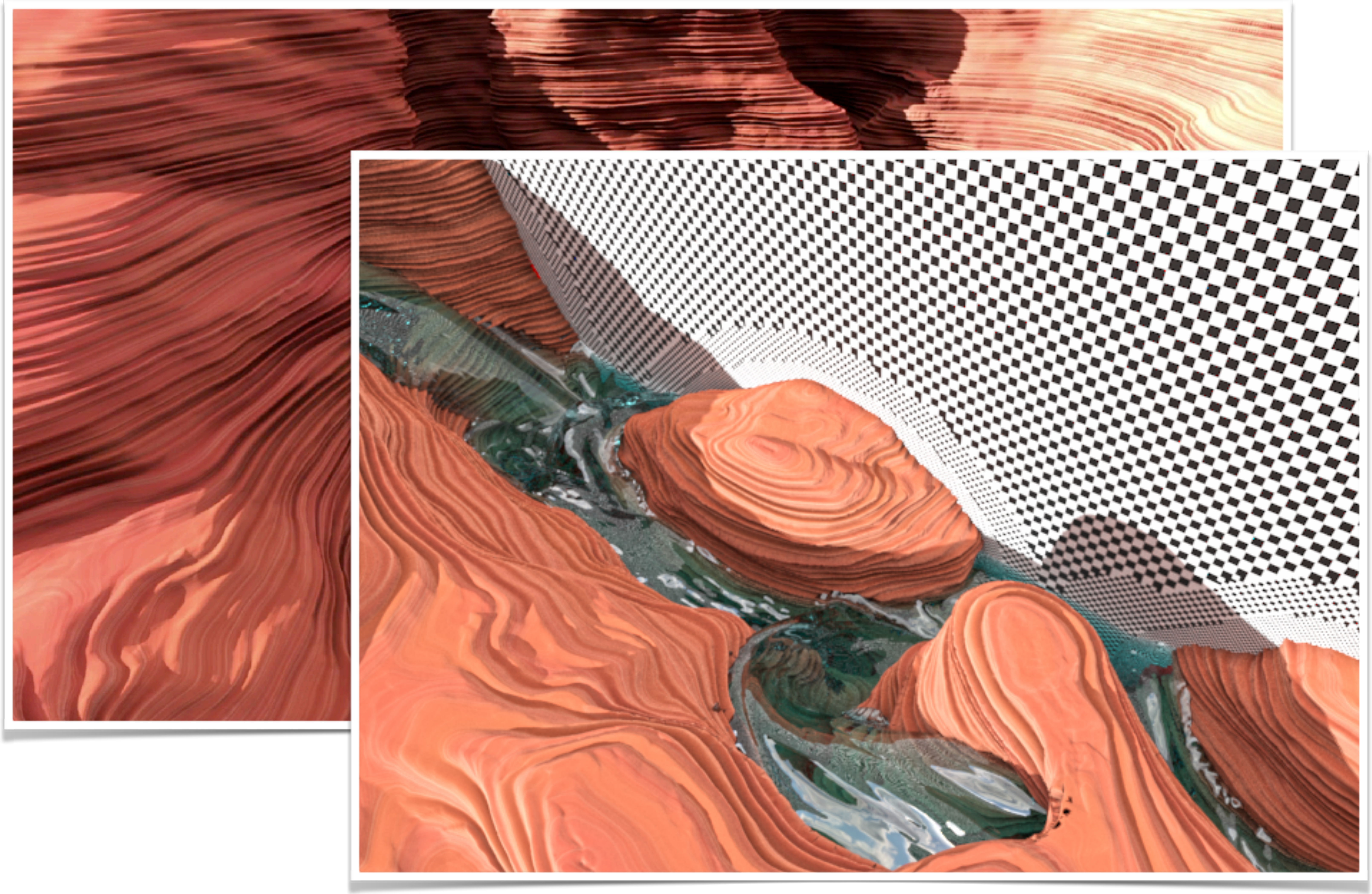
Further technicalities (see thesis)

- Minor topological complications (3D)
- Velocity interpolation
- Retrieval of Poisson equation stencils

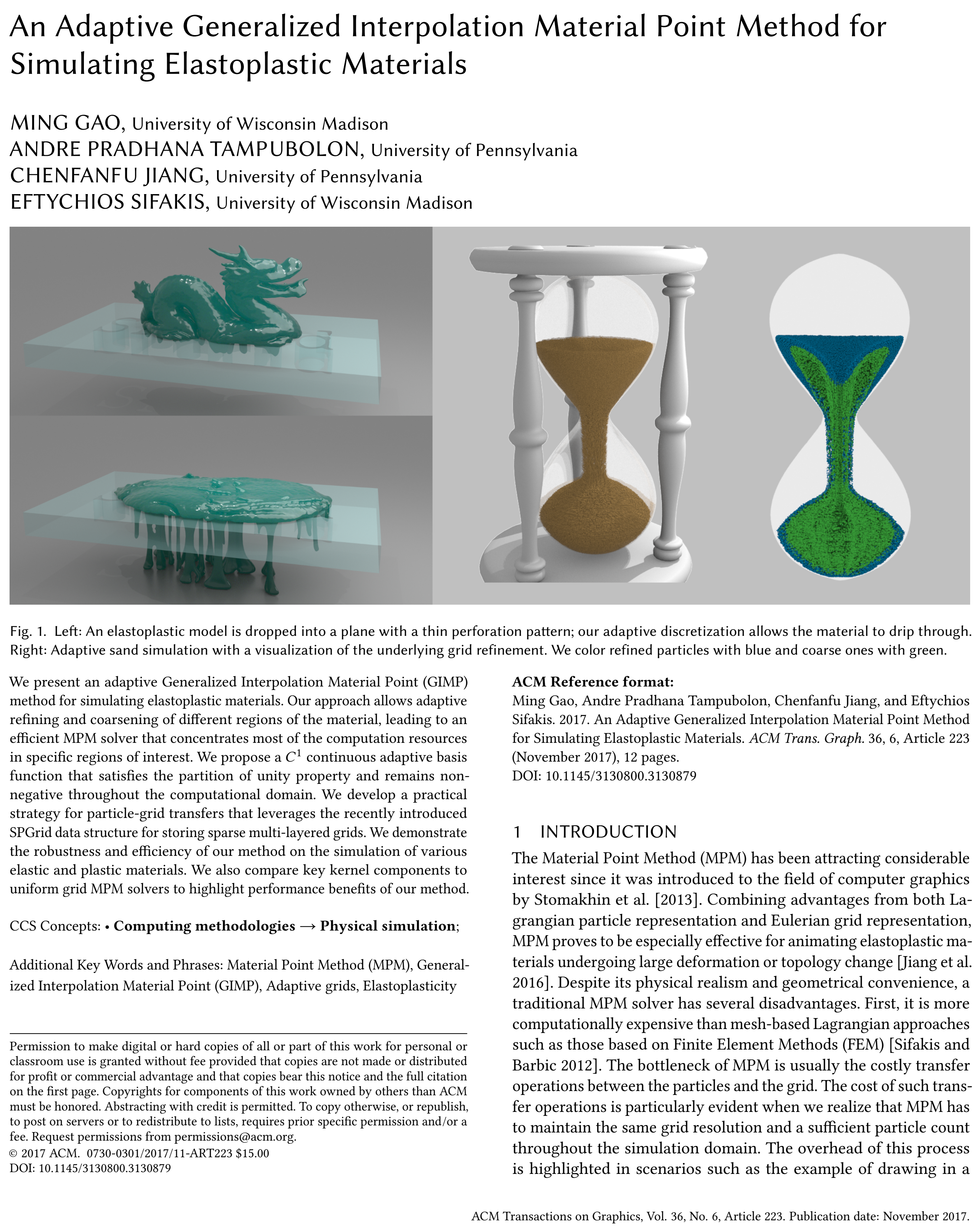
Results



Results



An Adaptive Generalized Interpolation Material Point Method for Simulating Elastoplastic Materials



An Adaptive Generalized Interpolation Material Point Method for Simulating Elastoplastic Materials

MING GAO, University of Wisconsin Madison
ANDRE PRADHANA TAMPUBOLON, University of Pennsylvania
CHENFANFU JIANG, University of Pennsylvania
EFTYCHIOS SIFAKIS, University of Wisconsin Madison

Fig. 1. Left: An elastoplastic model is dropped into a plane with a thin perforation pattern; our adaptive discretization allows the material to drip through. Right: Adaptive sand simulation with a visualization of the underlying grid refinement. We color refined particles with blue and coarse ones with green.

We present an adaptive Generalized Interpolation Material Point (GIMP) method for simulating elastoplastic materials. Our approach allows adaptive refining and coarsening of different regions of the material, leading to an efficient MPM solver that concentrates most of the computation resources in specific regions of interest. We propose a C^1 continuous adaptive basis function that satisfies the partition of unity property and remains non-negative throughout the computational domain. We develop a practical strategy for particle-grid transfers that leverages the recently introduced SPGrid data structure for storing sparse multi-layered grids. We demonstrate the robustness and efficiency of our method on the simulation of various elastic and plastic materials. We also compare key kernel components to uniform grid MPM solvers to highlight performance benefits of our method.

CCS Concepts: • **Computing methodologies → Physical simulation**;

Additional Key Words and Phrases: Material Point Method (MPM), Generalized Interpolation Material Point (GIMP), Adaptive grids, Elastoplasticity

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2017 ACM. 0730-0301/2017/11-ART223 $15.00
DOI: 10.1145/3130800.3130879

ACM Reference format:
Ming Gao, Andre Pradhana Tampubolon, Chenfanfu Jiang, and Eftychios Sifakis. 2017. An Adaptive Generalized Interpolation Material Point Method for Simulating Elastoplastic Materials. *ACM Trans. Graph.* 36, 6, Article 223 (November 2017), 12 pages.
DOI: 10.1145/3130800.3130879

1 INTRODUCTION

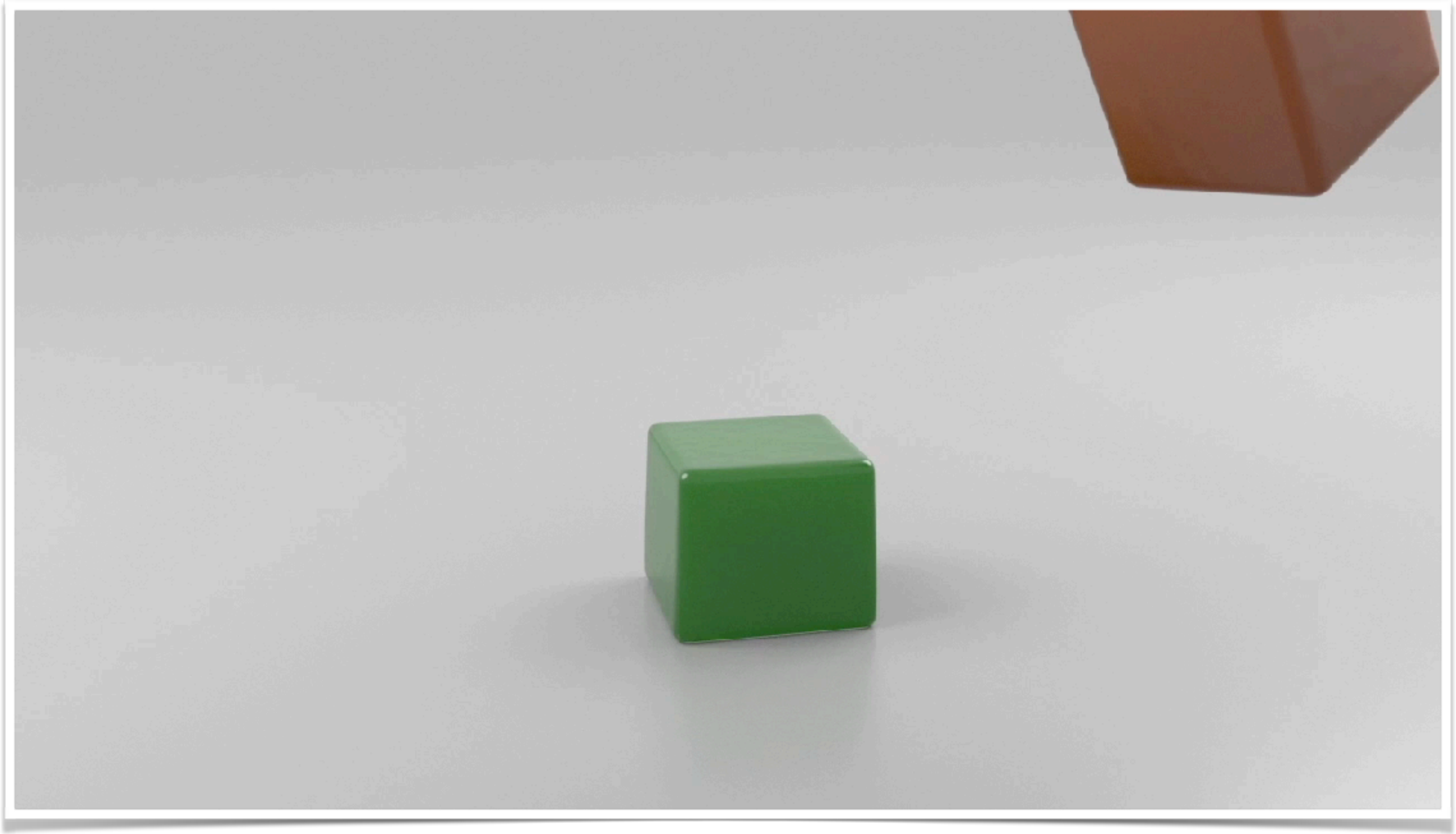
The Material Point Method (MPM) has been attracting considerable interest since it was introduced to the field of computer graphics by Stomakhin et al. [2013]. Combining advantages from both Lagrangian particle representation and Eulerian grid representation, MPM proves to be especially effective for animating elastoplastic materials undergoing large deformation or topology change [Jiang et al. 2016]. Despite its physical realism and geometrical convenience, a traditional MPM solver has several disadvantages. First, it is more computationally expensive than mesh-based Lagrangian approaches such as those based on Finite Element Methods (FEM) [Sifakis and Barbic 2012]. The bottleneck of MPM is usually the costly transfer operations between the particles and the grid. The cost of such transfer operations is particularly evident when we realize that MPM has to maintain the same grid resolution and a sufficient particle count throughout the simulation domain. The overhead of this process is highlighted in scenarios such as the example of drawing in a

ACM Transactions on Graphics, Vol. 36, No. 6, Article 223. Publication date: November 2017.

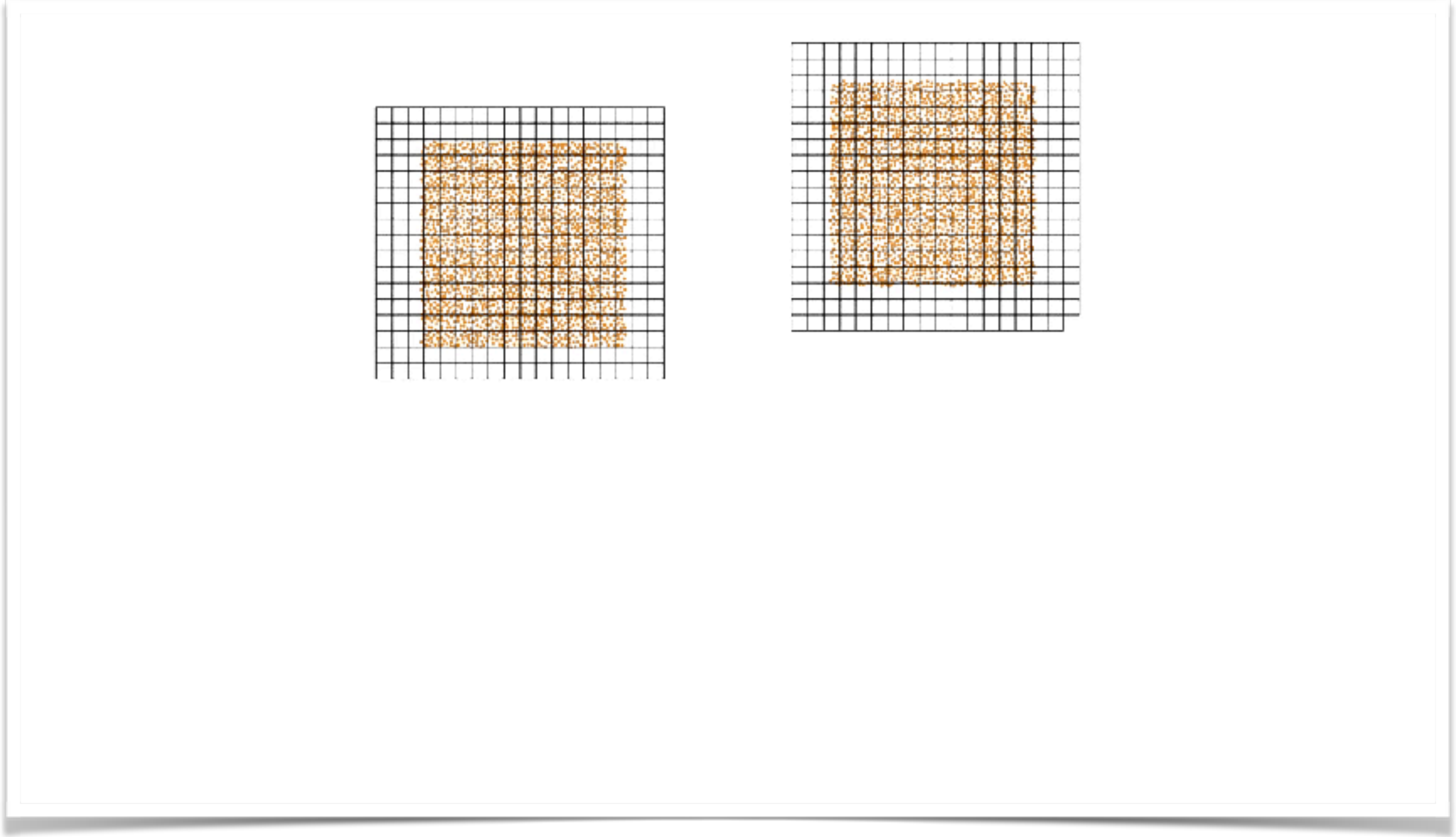
M. Gao, A. Tampubolon, C. Jiang, E. Sifakis
ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia), 2017

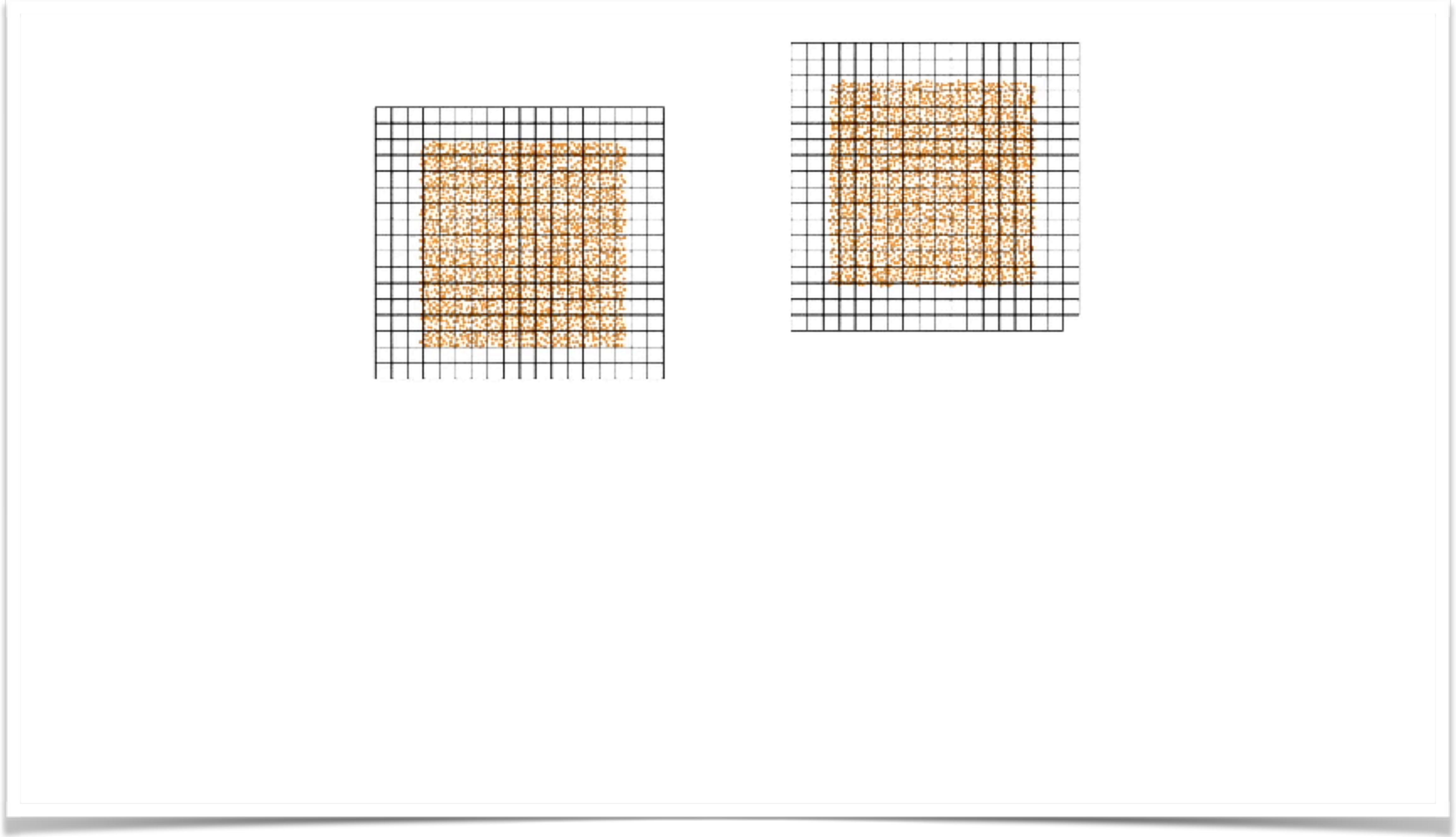
SIGGRAPH ASIA 2017 BANGKOK

Motivation

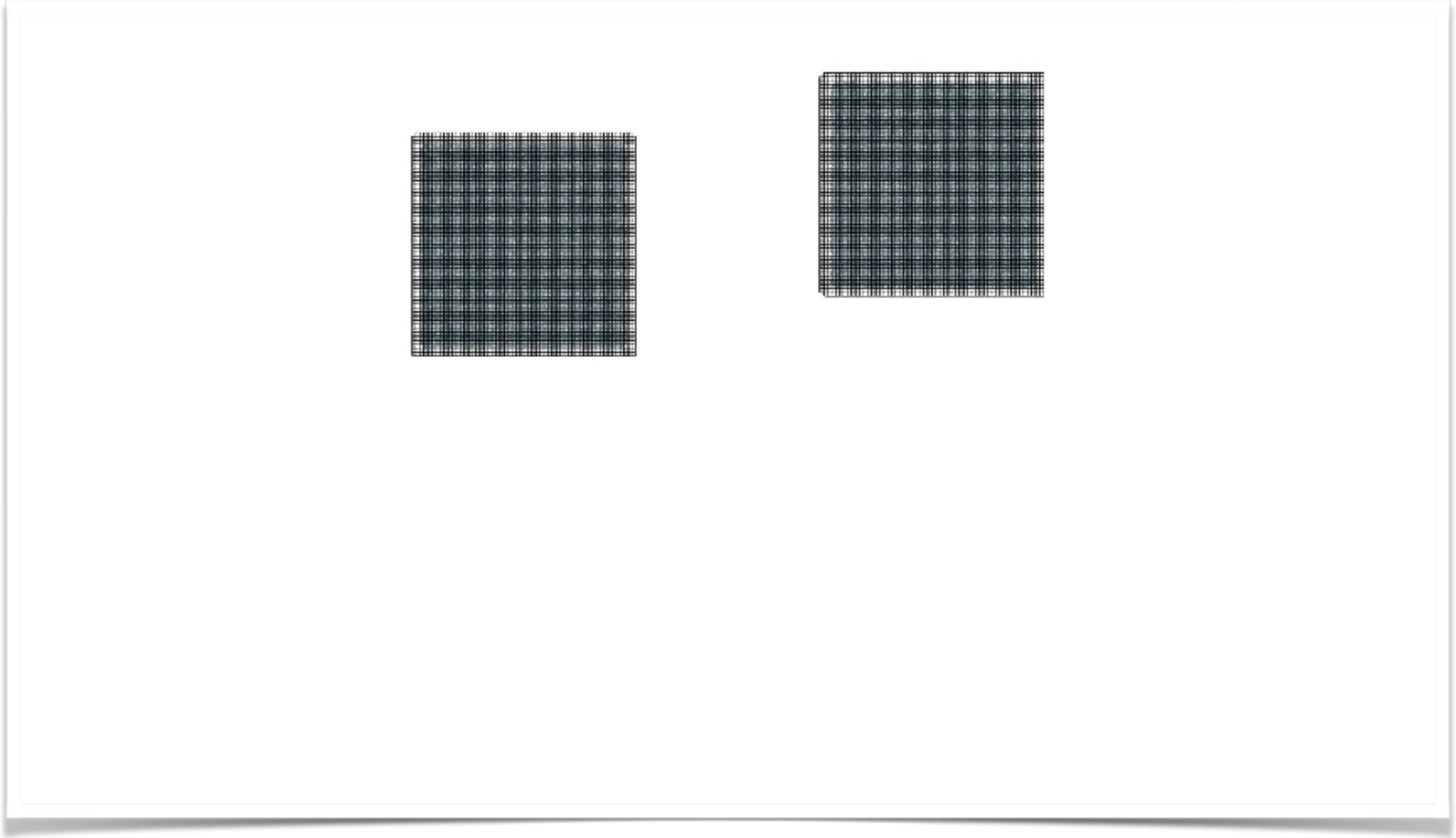


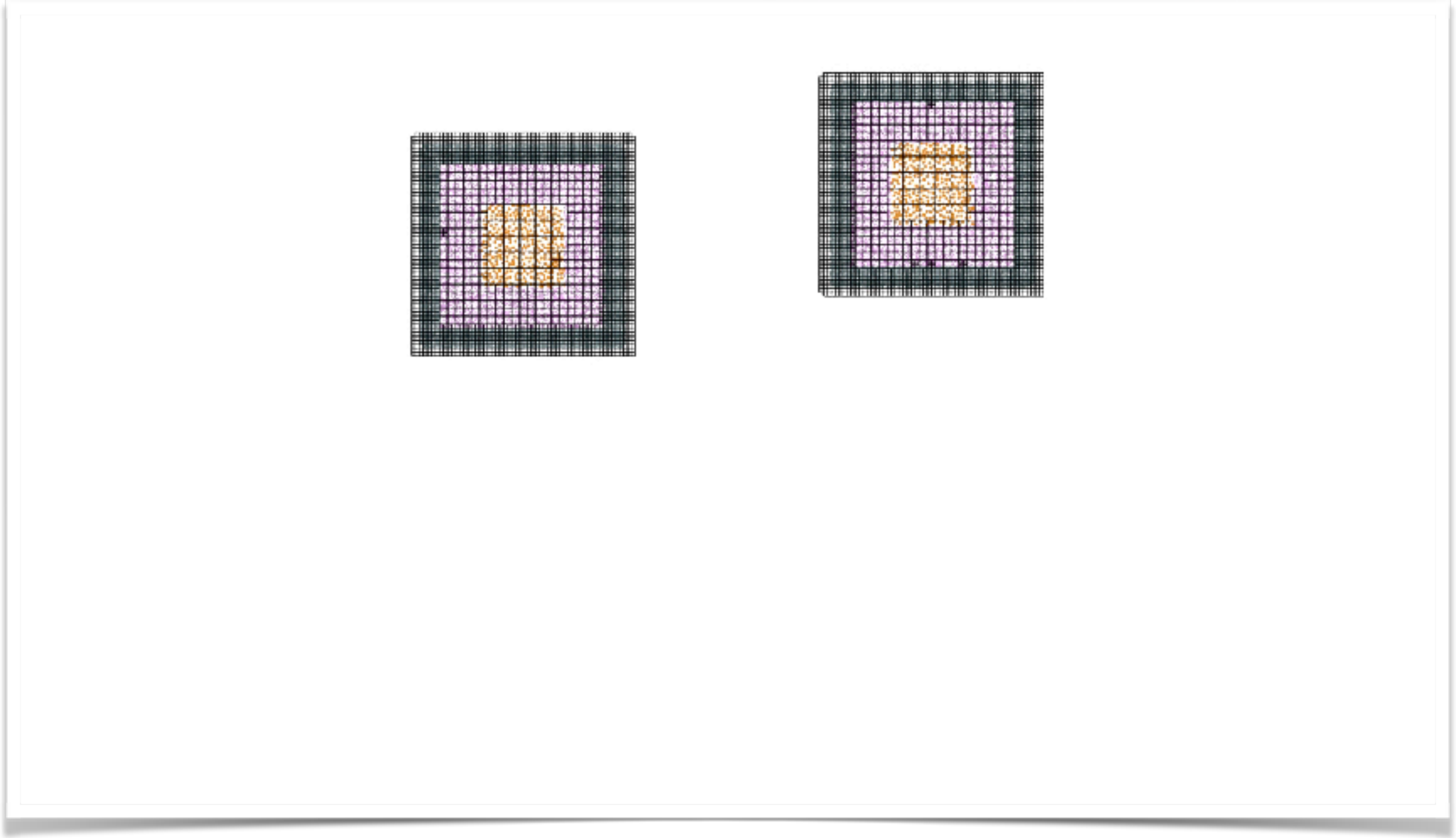
Motivation



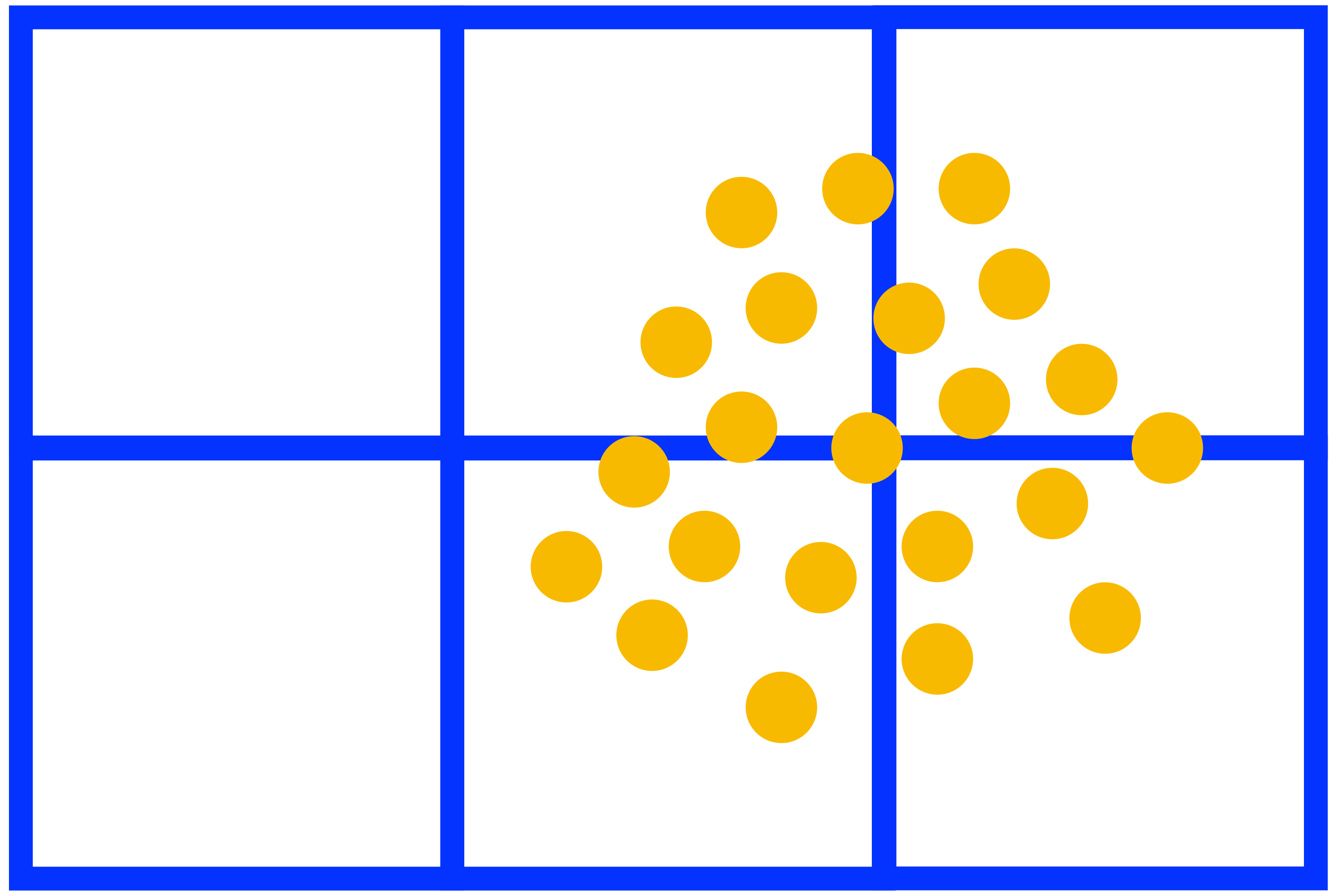


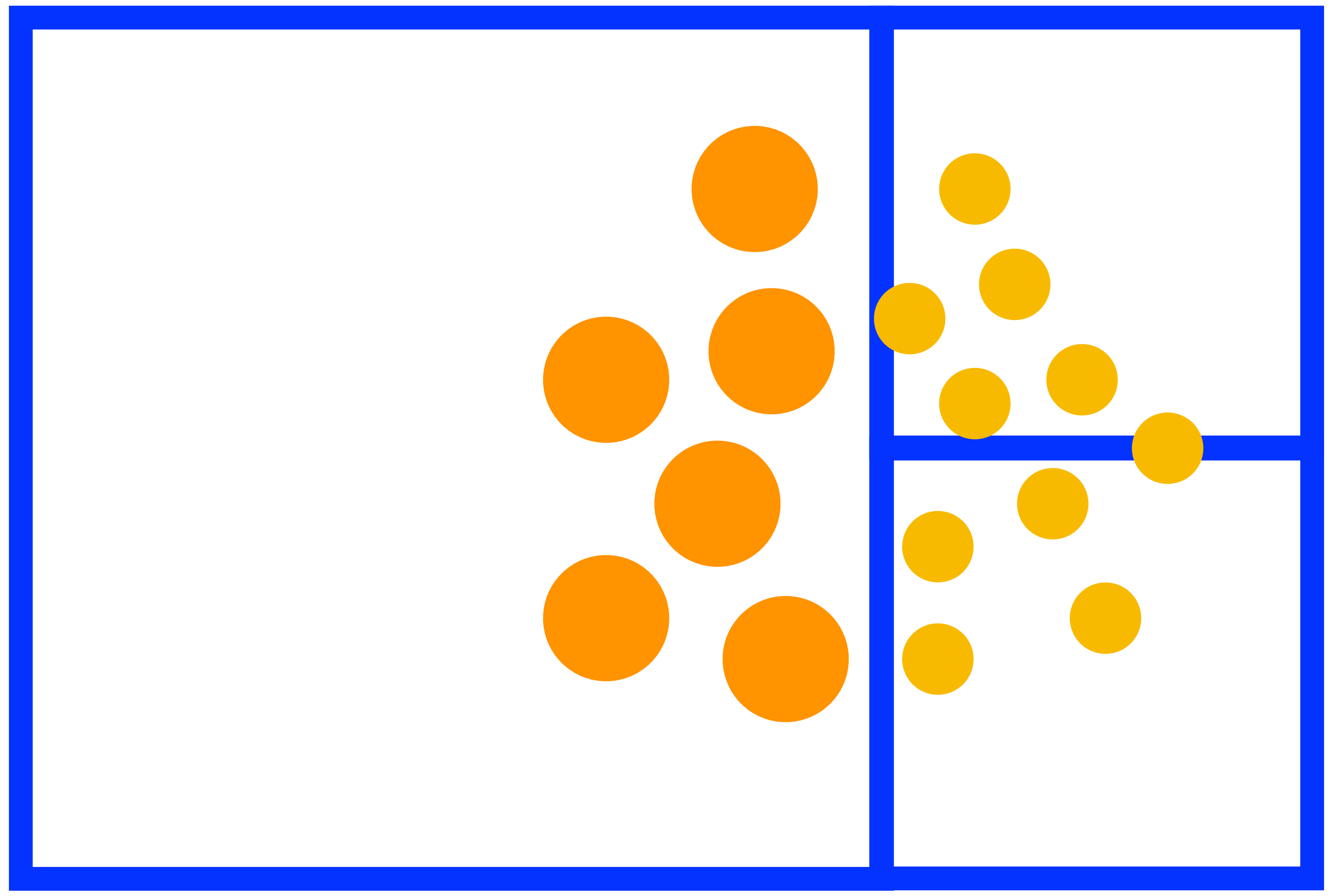
Motivation



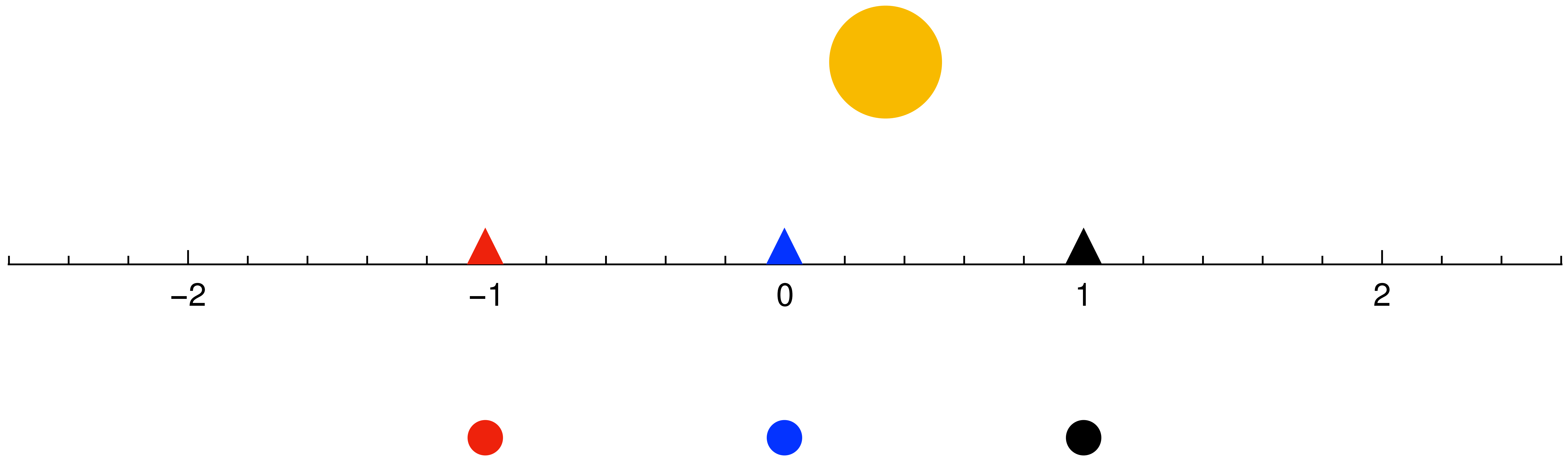


MPM adaptivity

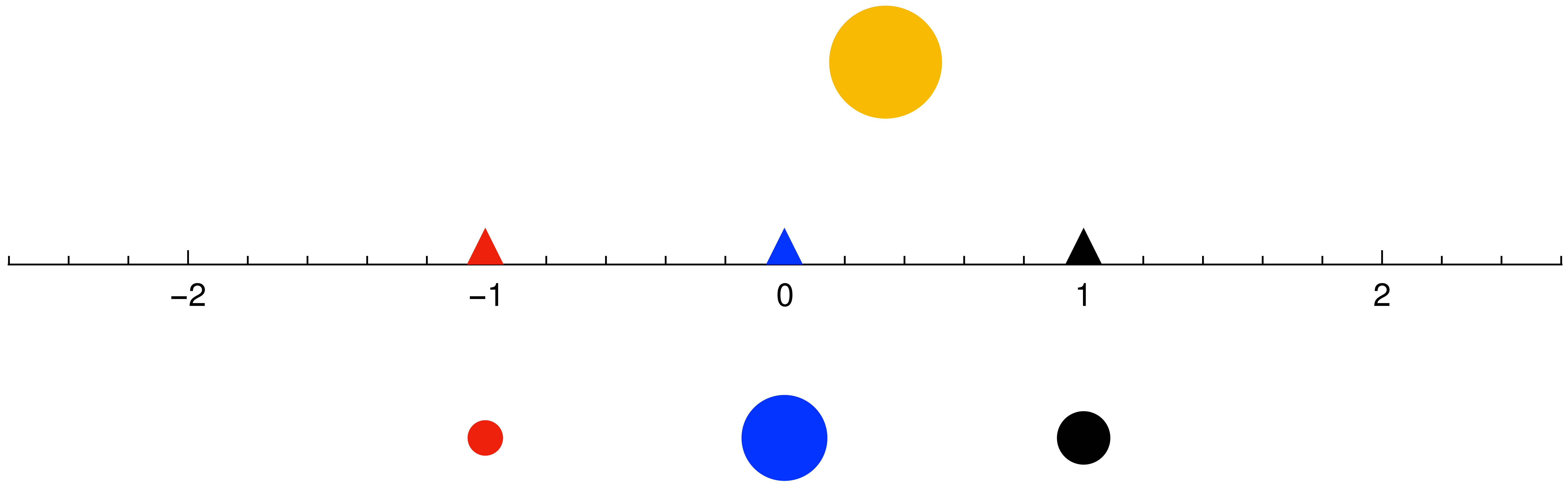




Transfer of mass



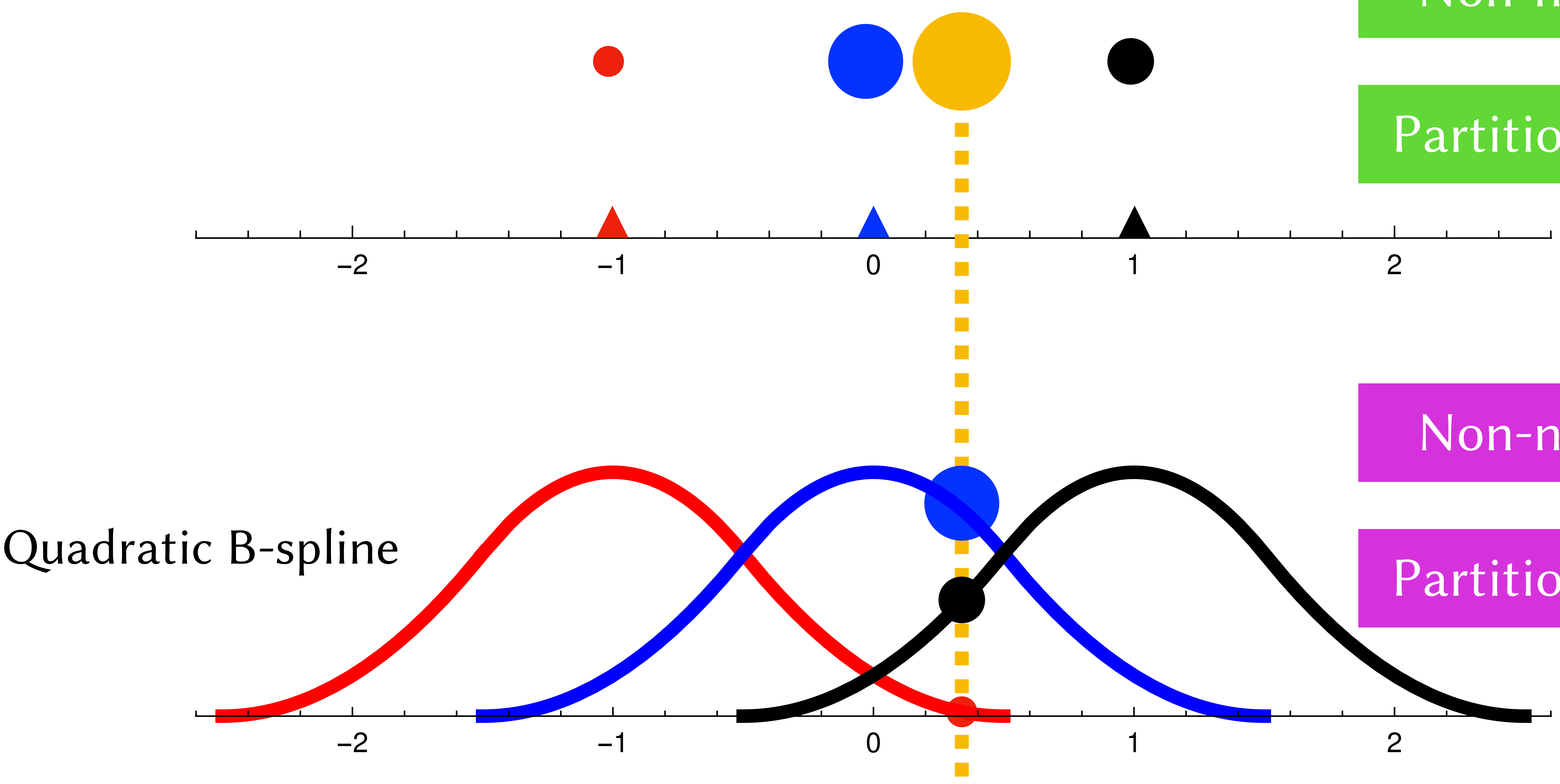
Transfer of mass



Non-negativity

Partition of unity

Transfer of mass



Non-negativity

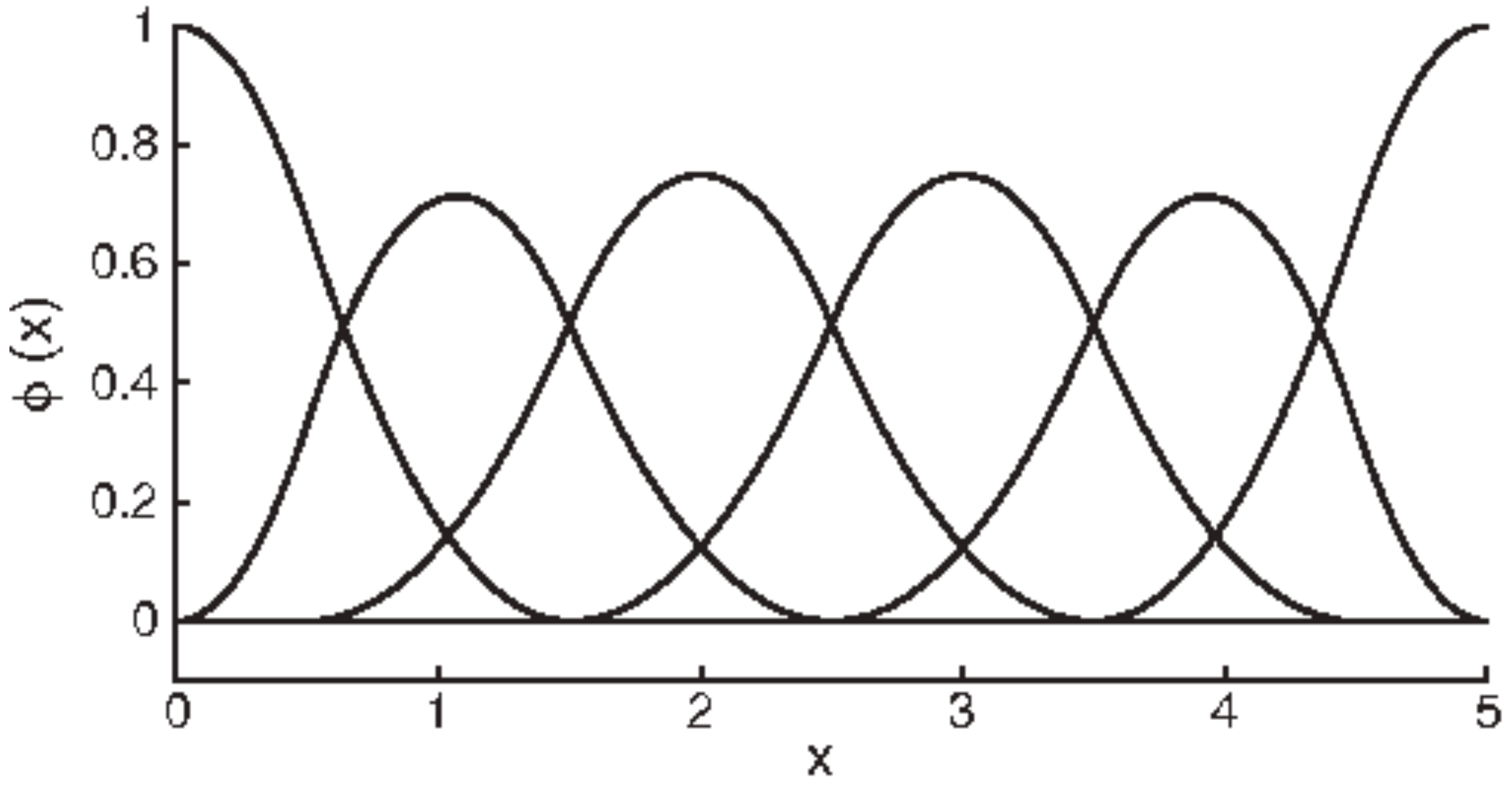
Partition of unity

Non-negativity

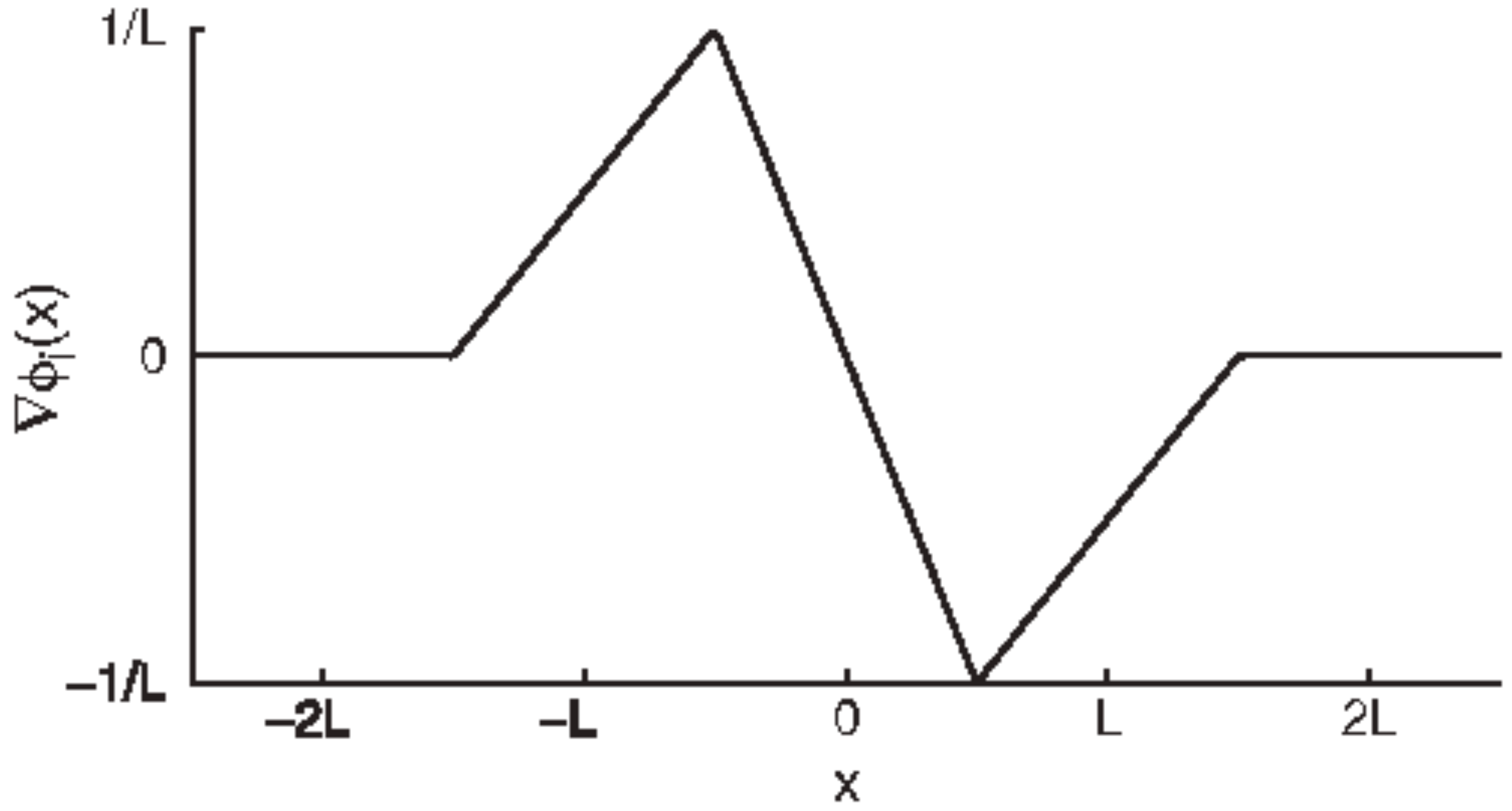
Partition of unity

C^1 continuity

Weight / mass



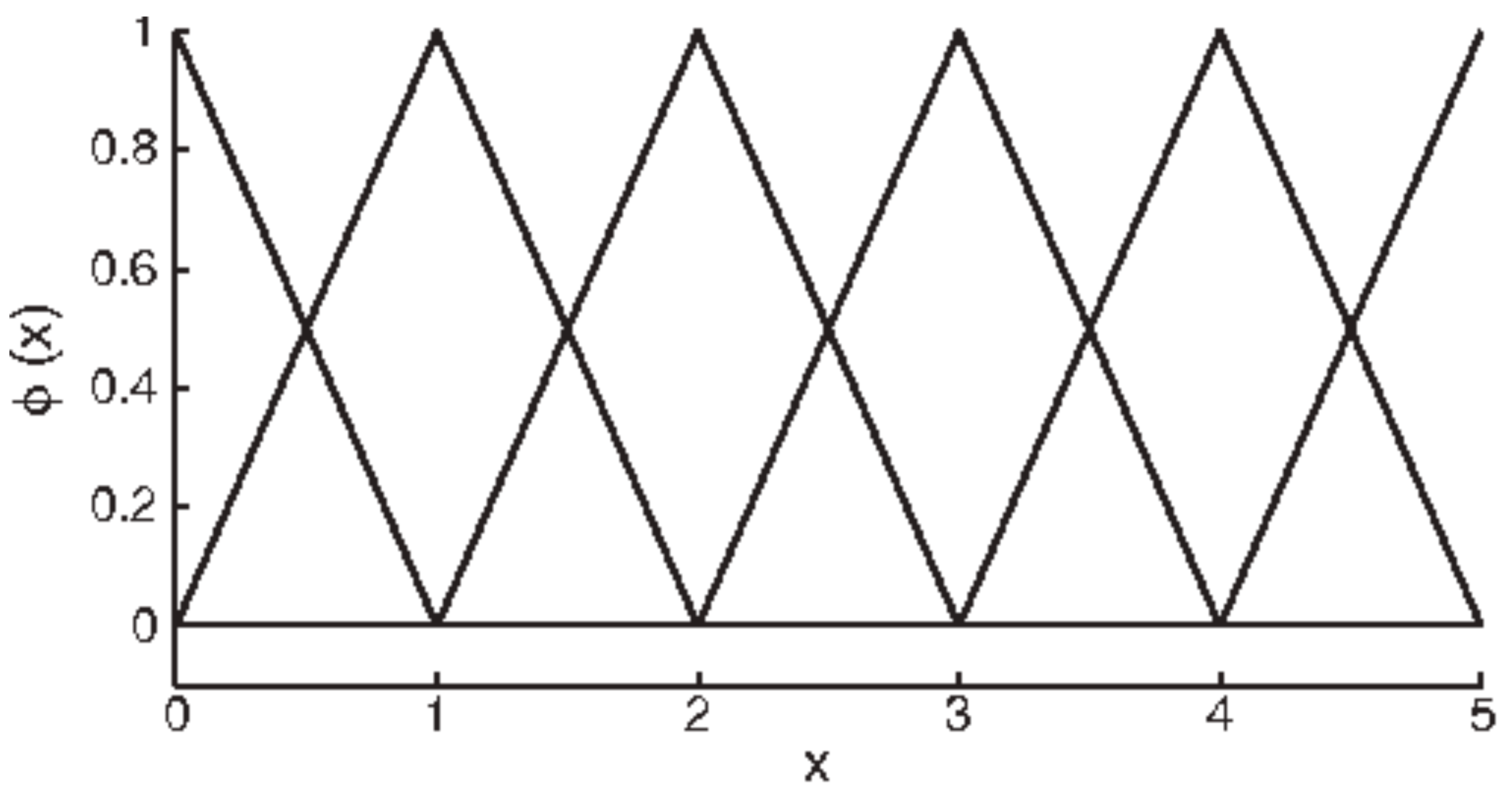
Weight gradient / force



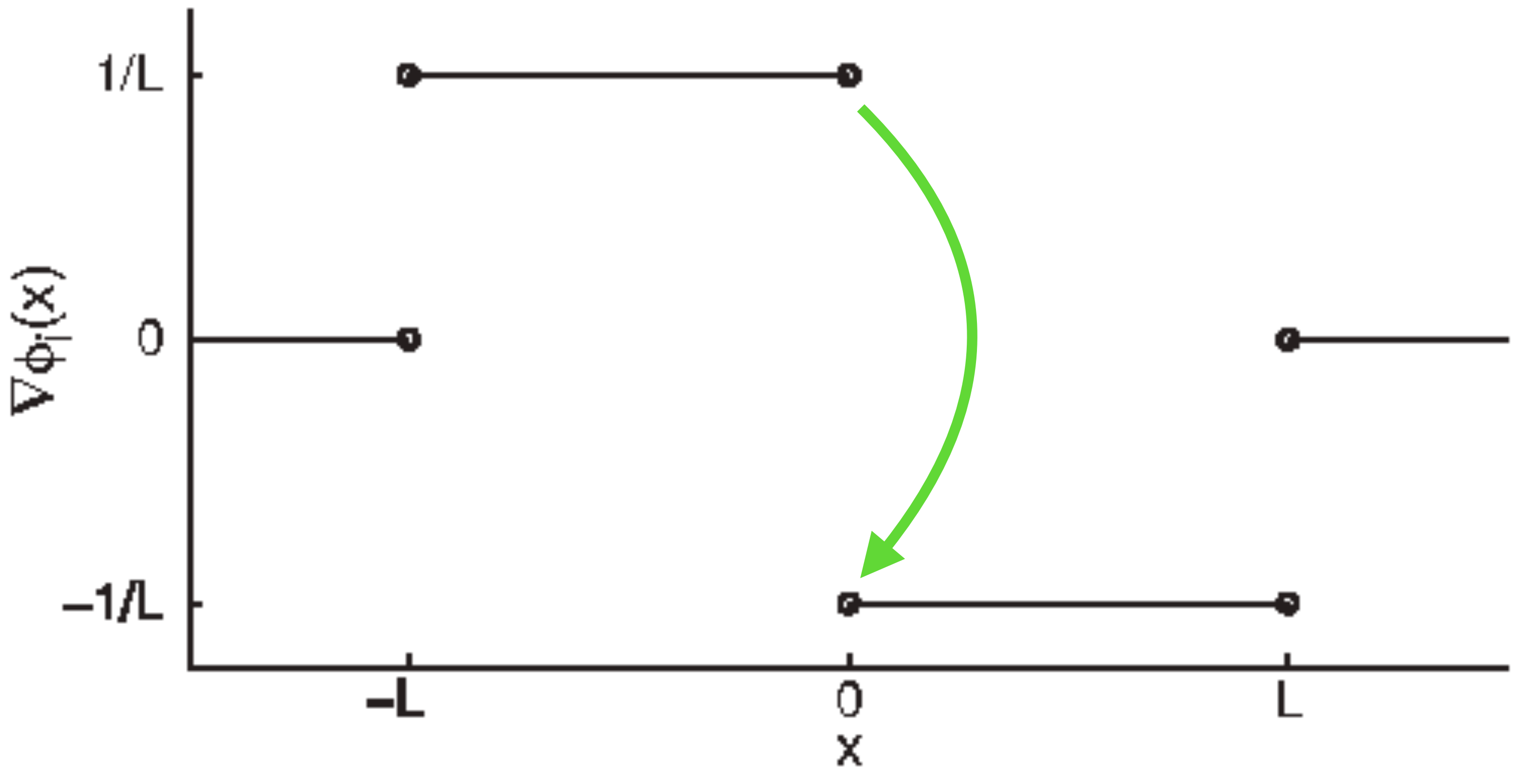
Steffen et al. 08

C^0 continuity

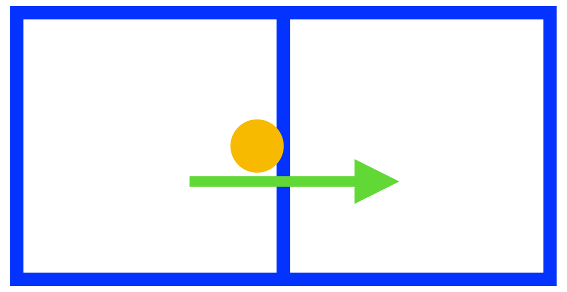
Weight / mass



Weight gradient / force

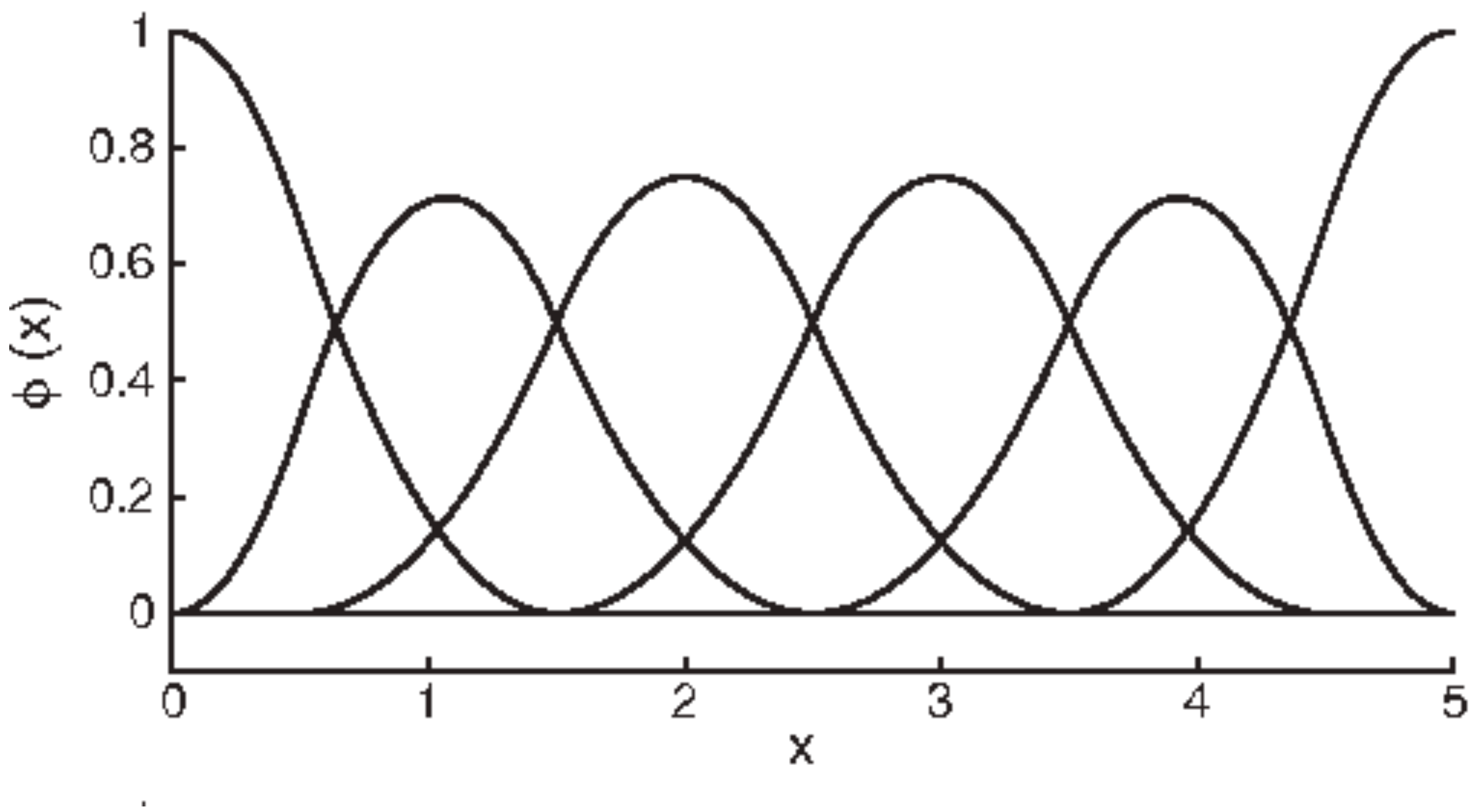


Steffen et al. 08

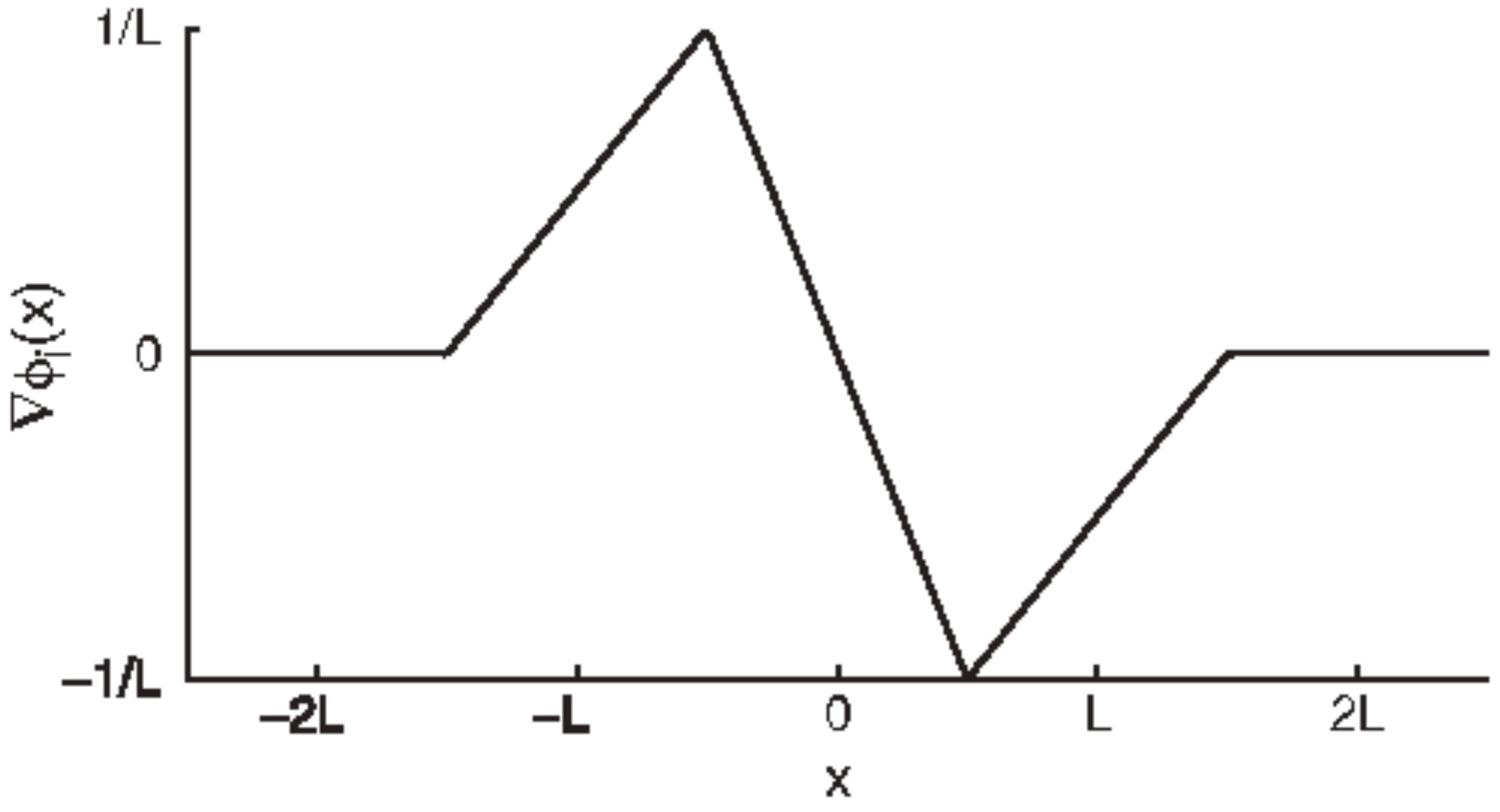


C^1 continuity in octree ?

Weight / mass



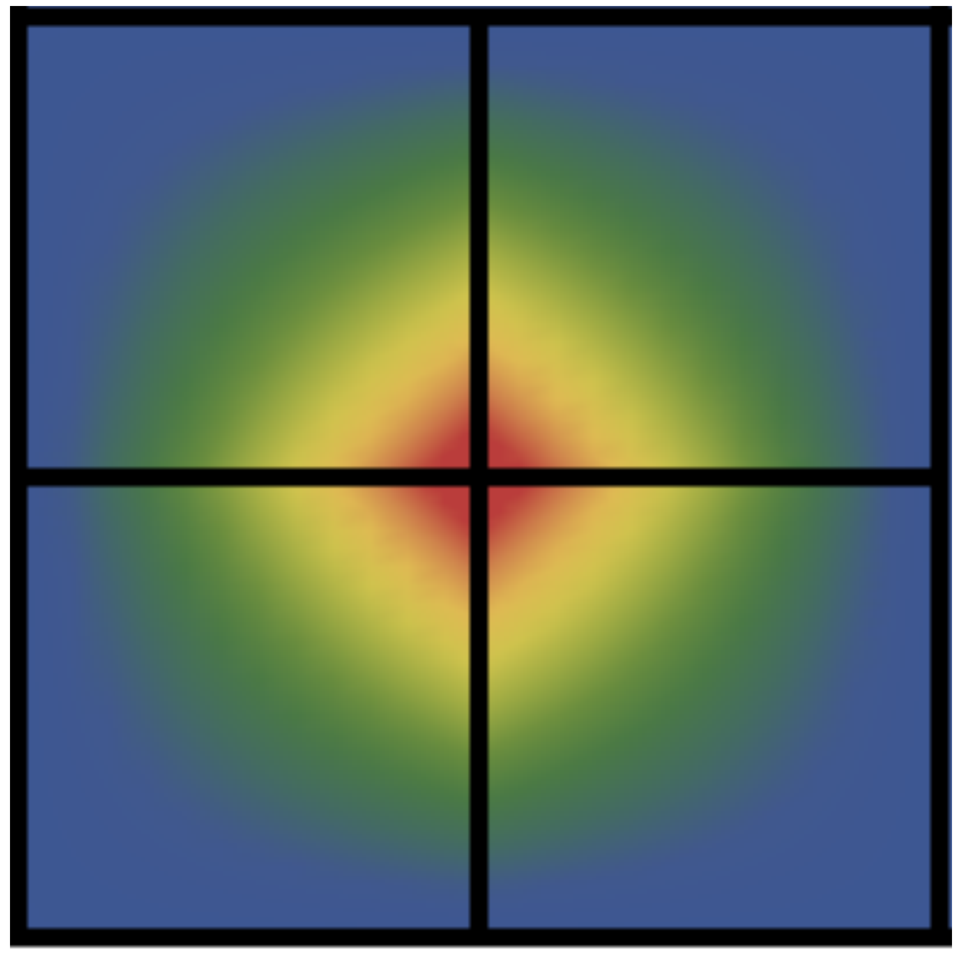
Weight gradient / force



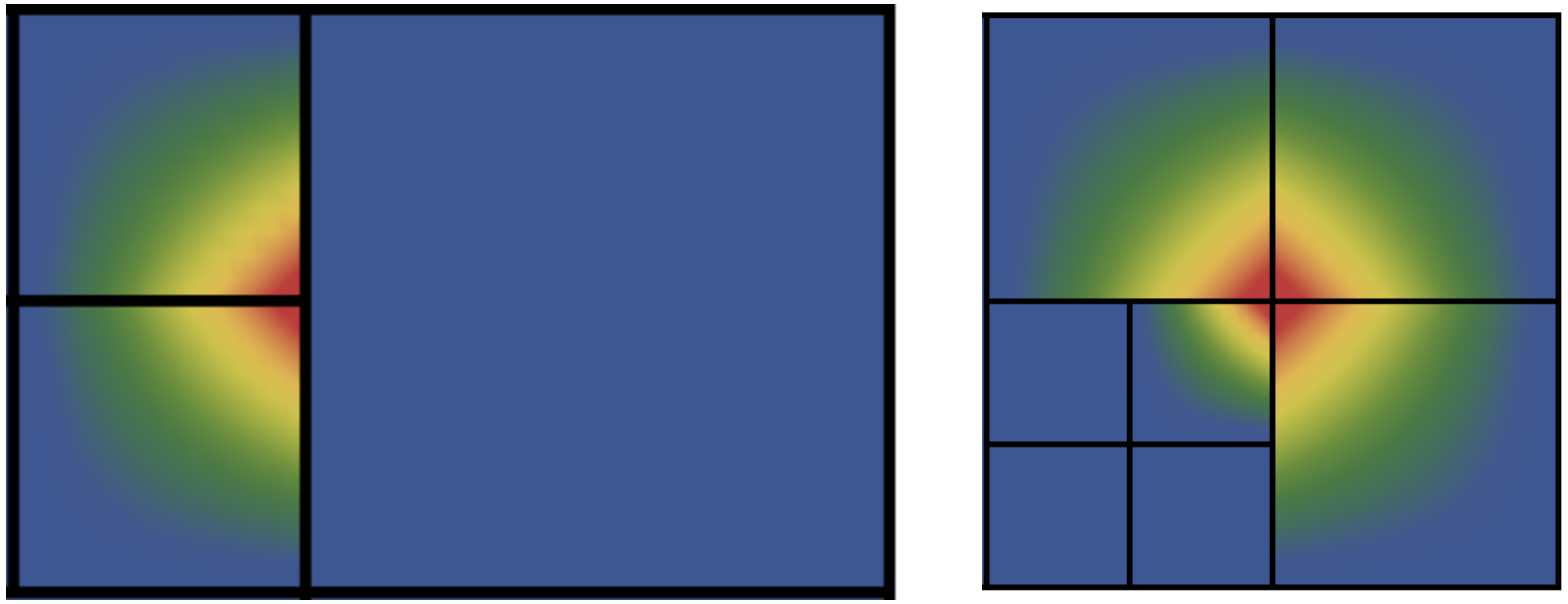
Steffen et al. 08

C^0 → C^1

C^0 from uniform to quadtree



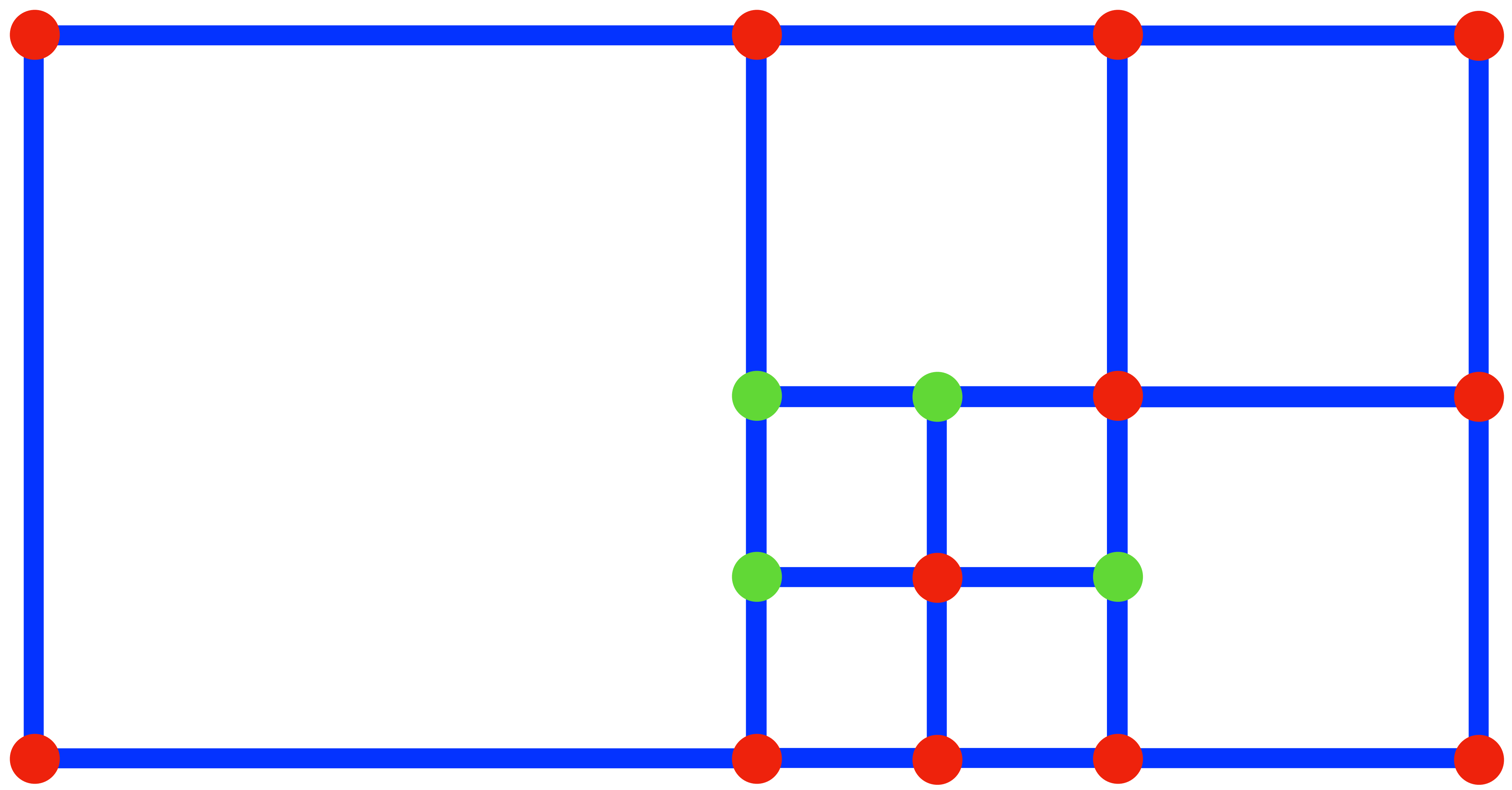
Uniform



Quadtree

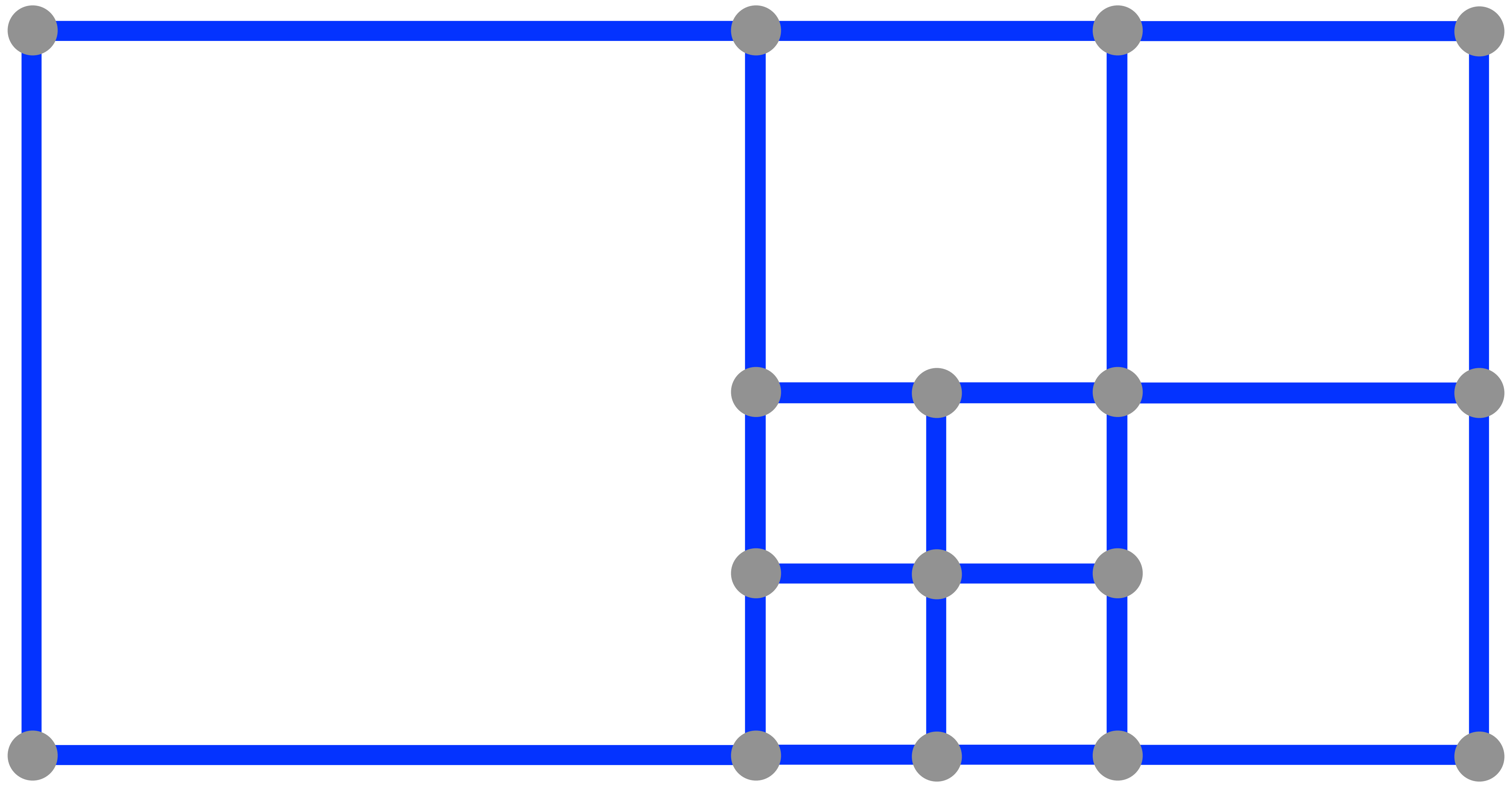
Embedding T-junctions

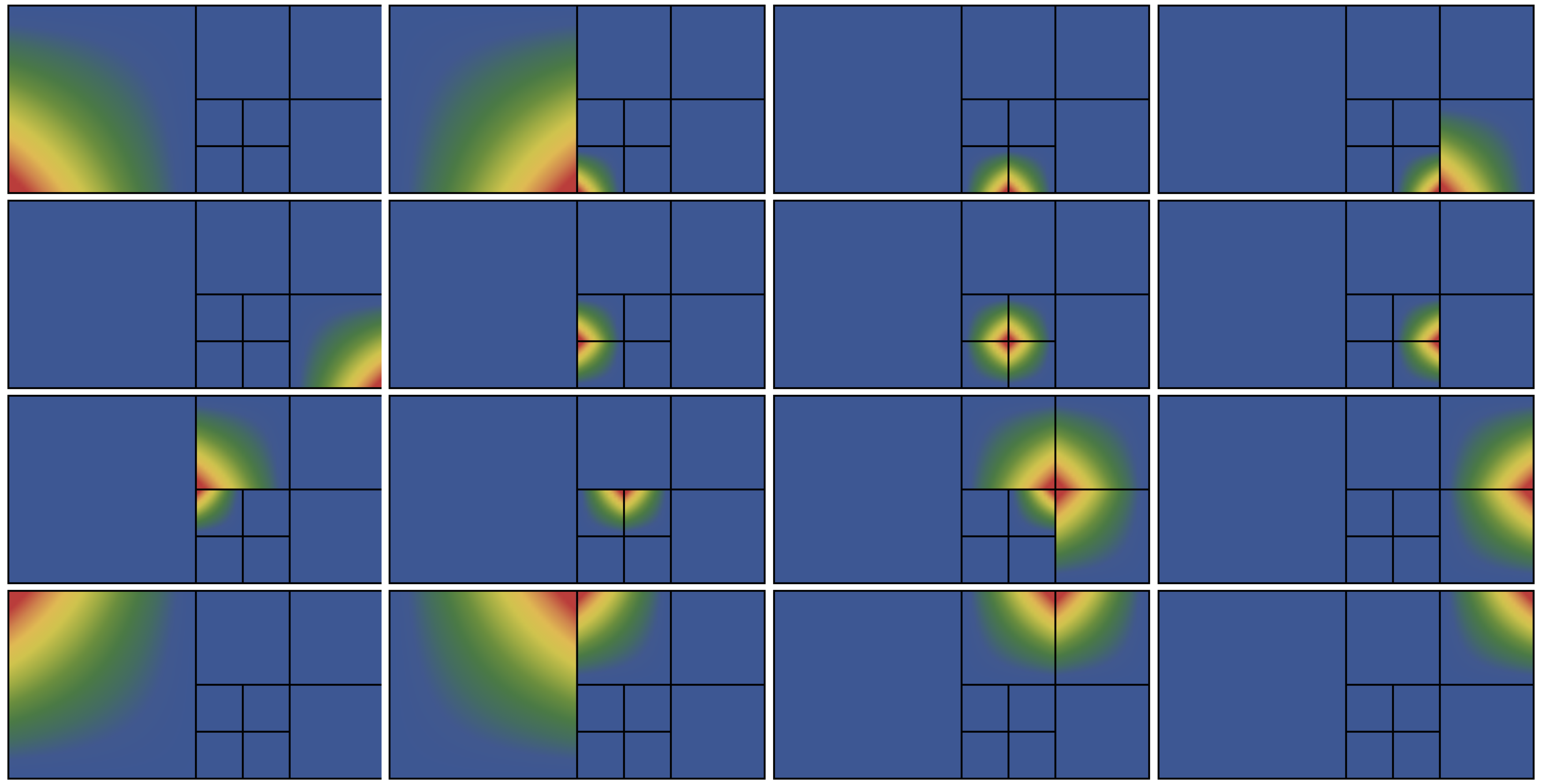
- DOF node
- Embedded node / T-junction node



Step1 - set all nodes free

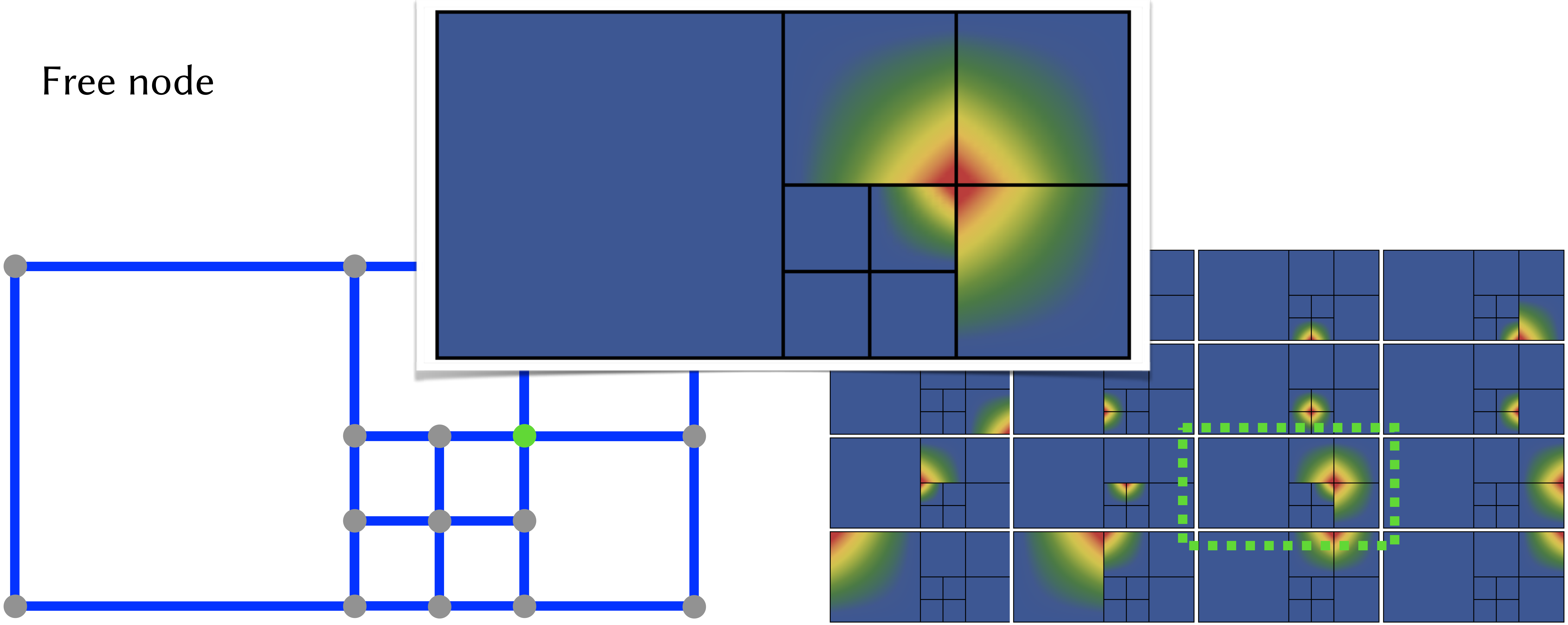
● Free node





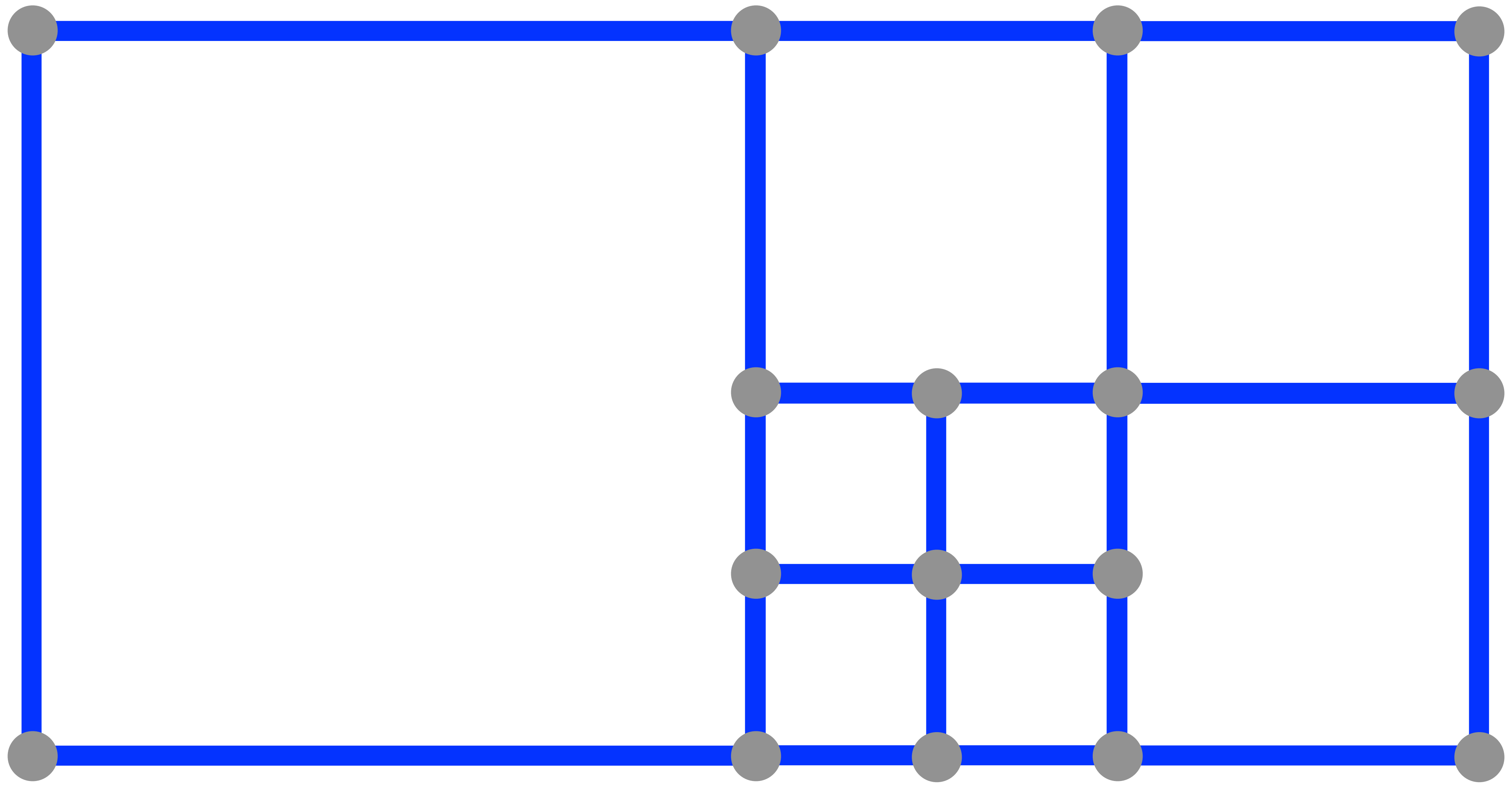
Step1 - set all nodes free

● Free node



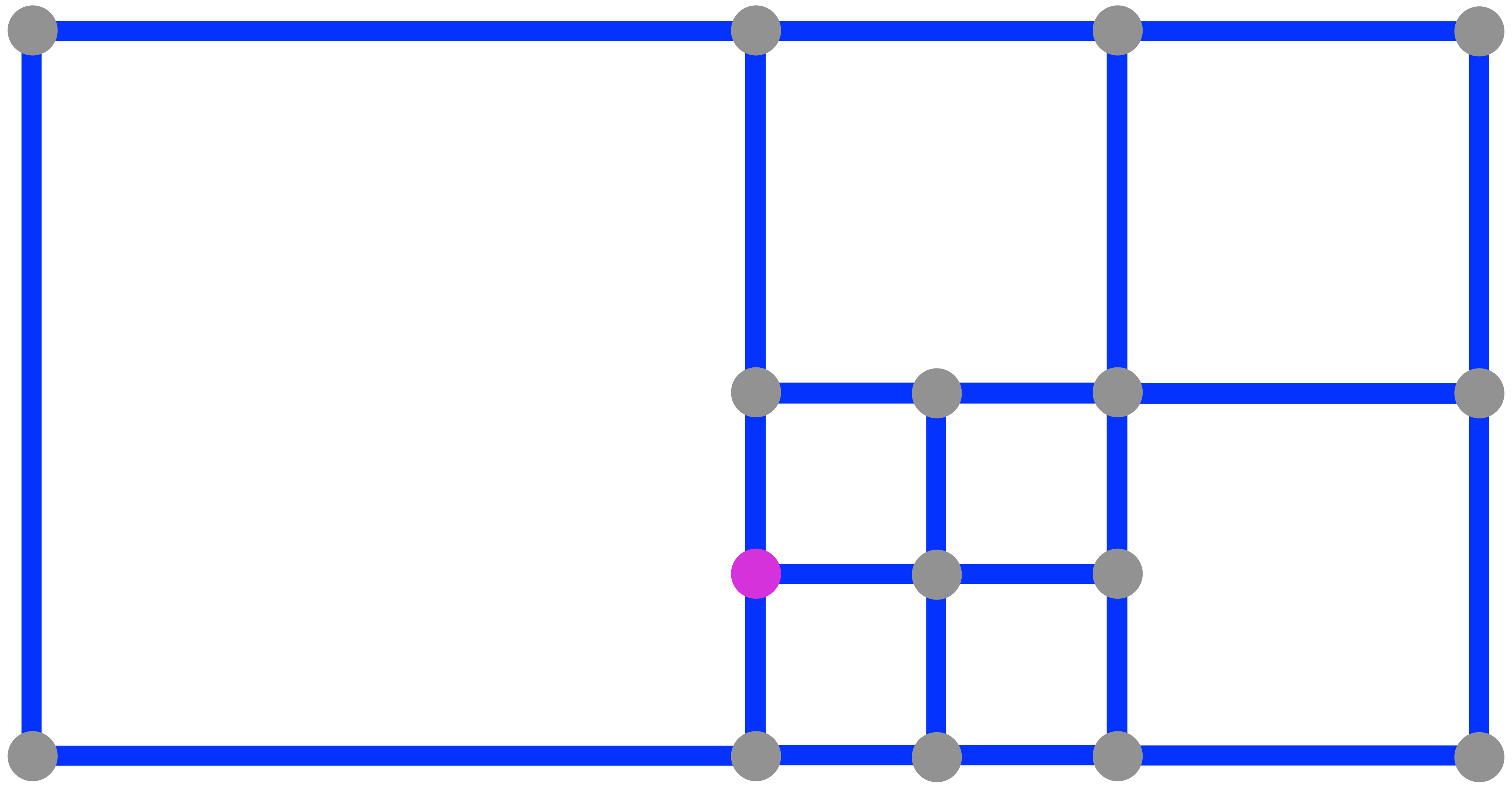
Step 2 - constrain T-junctions

- Free node



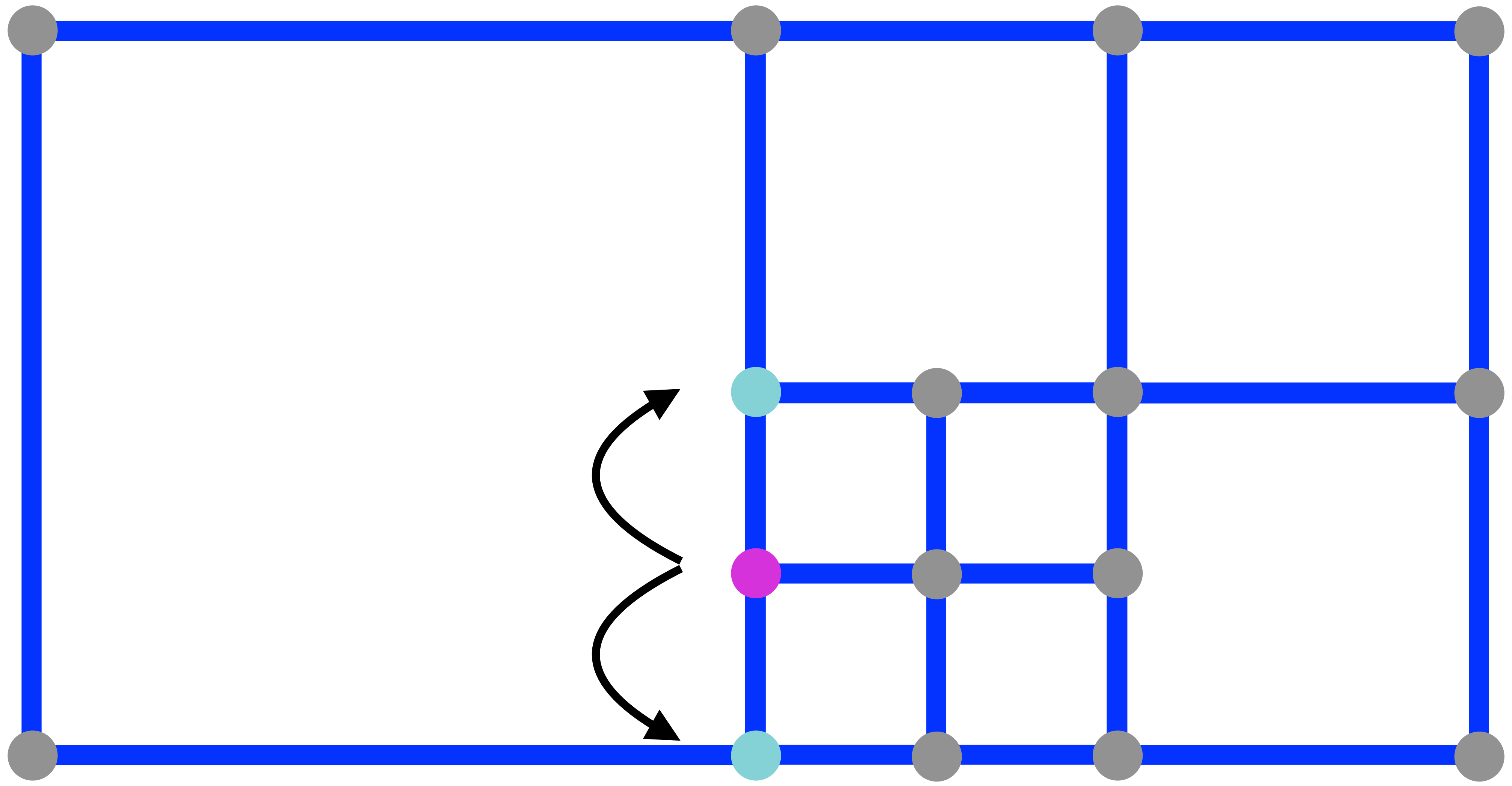
Step 2 - constrain T-junctions

- Free node
- T-junction node

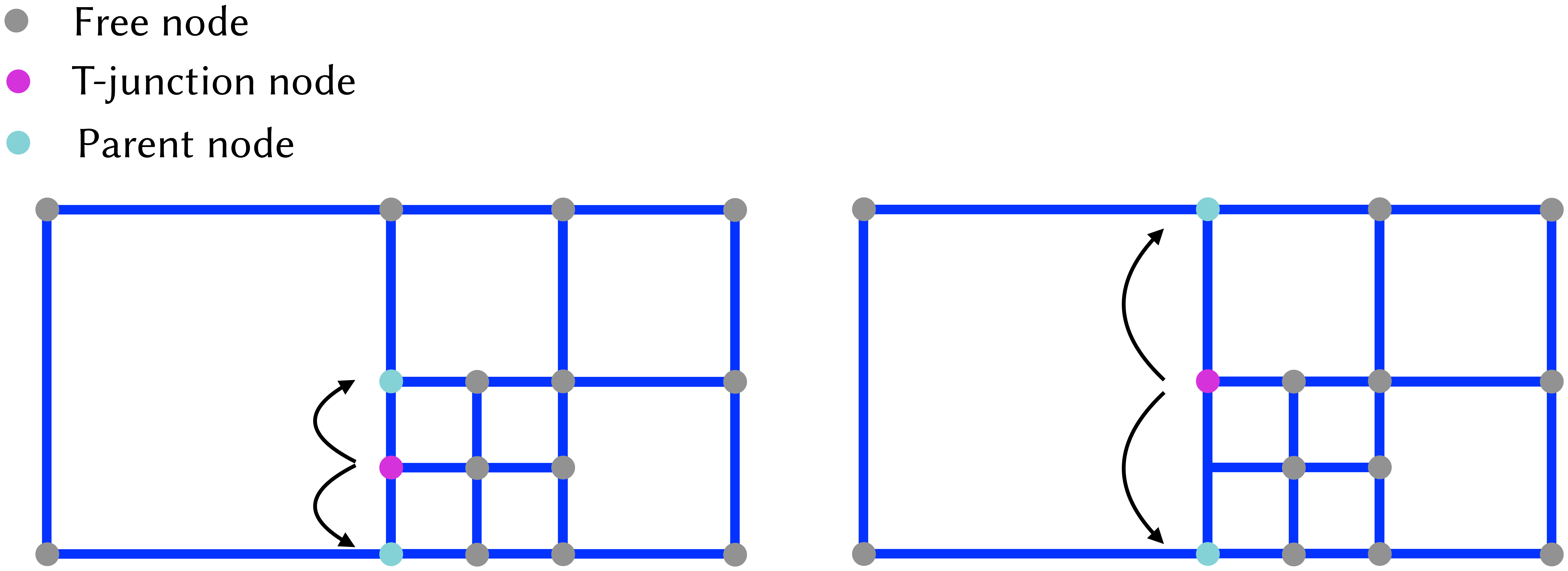


Step 2 - constrain T-junctions

- Free node
- T-junction node
- Parent node

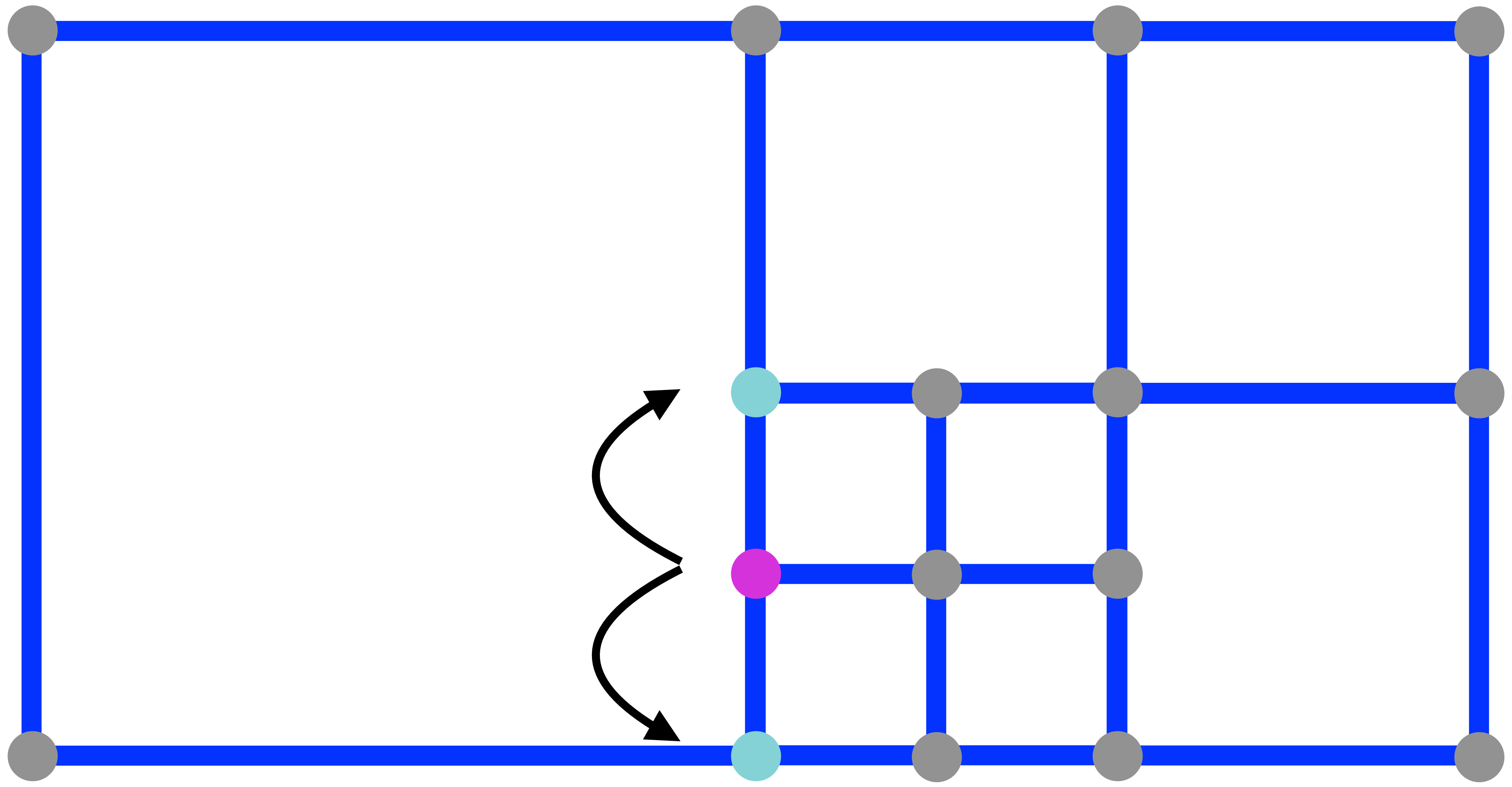


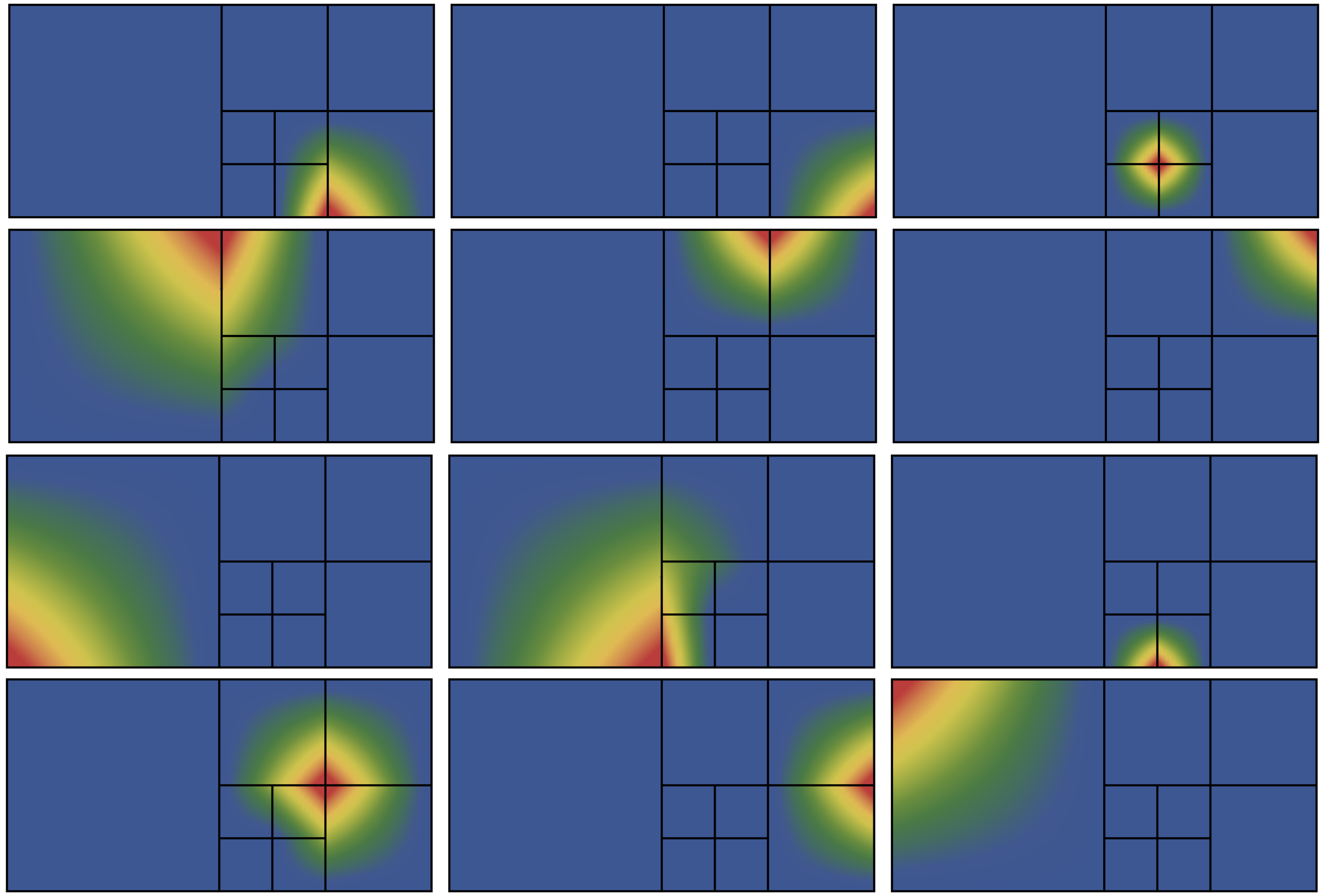
Step 2 - constrain T-junctions



Step 2 - constrain T-junctions

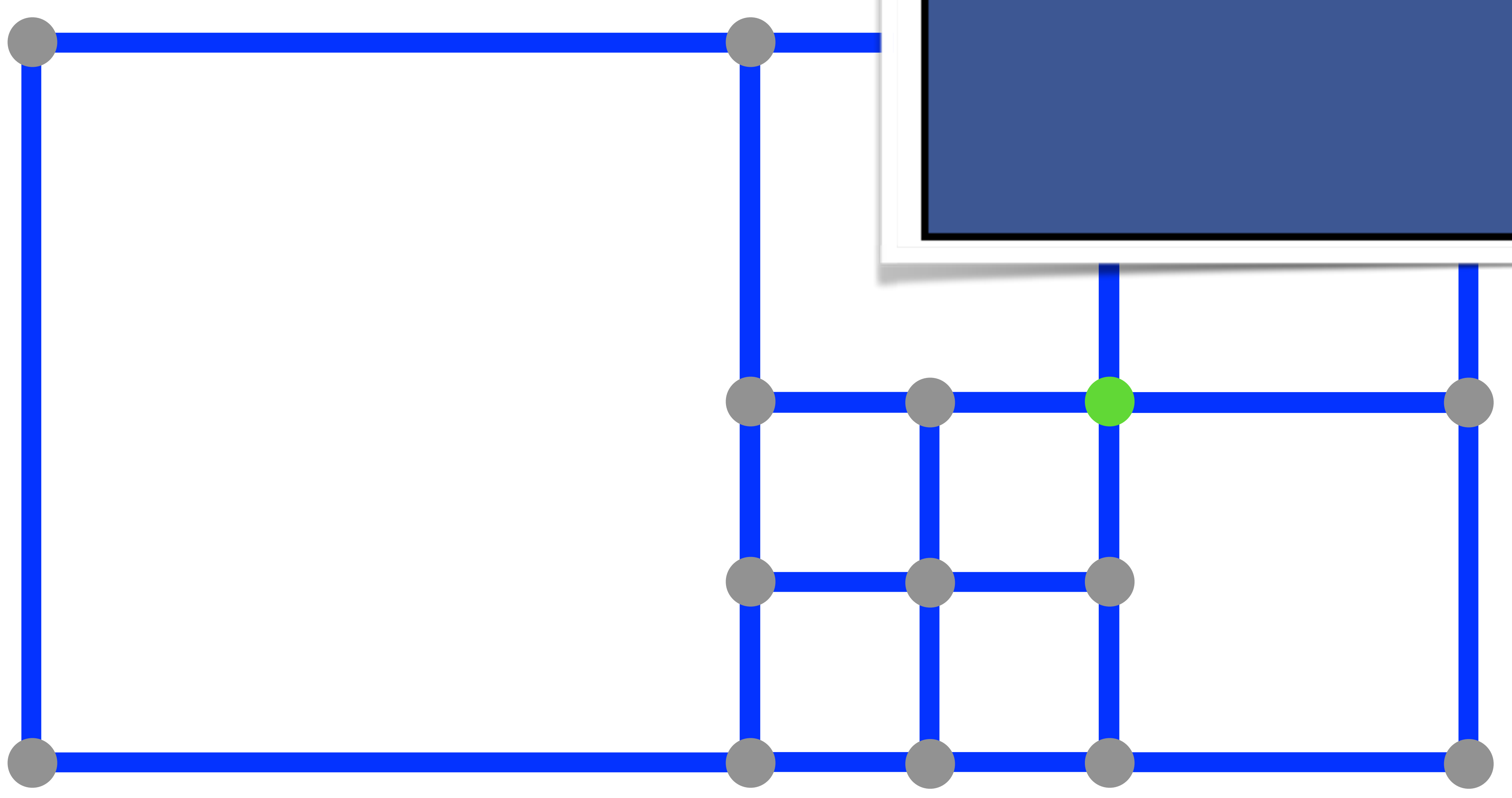
- Free node
- T-junction node
- Parent node

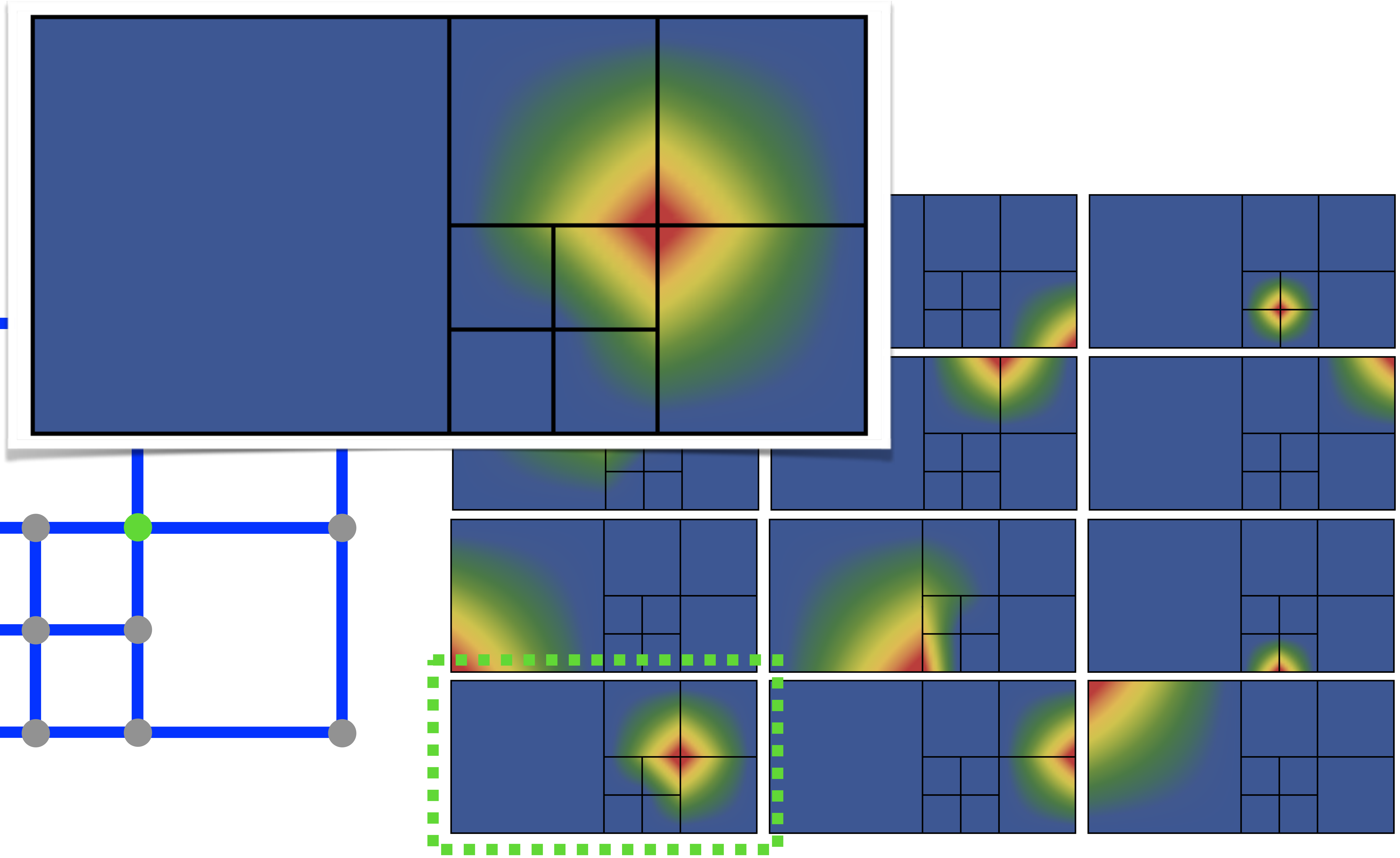




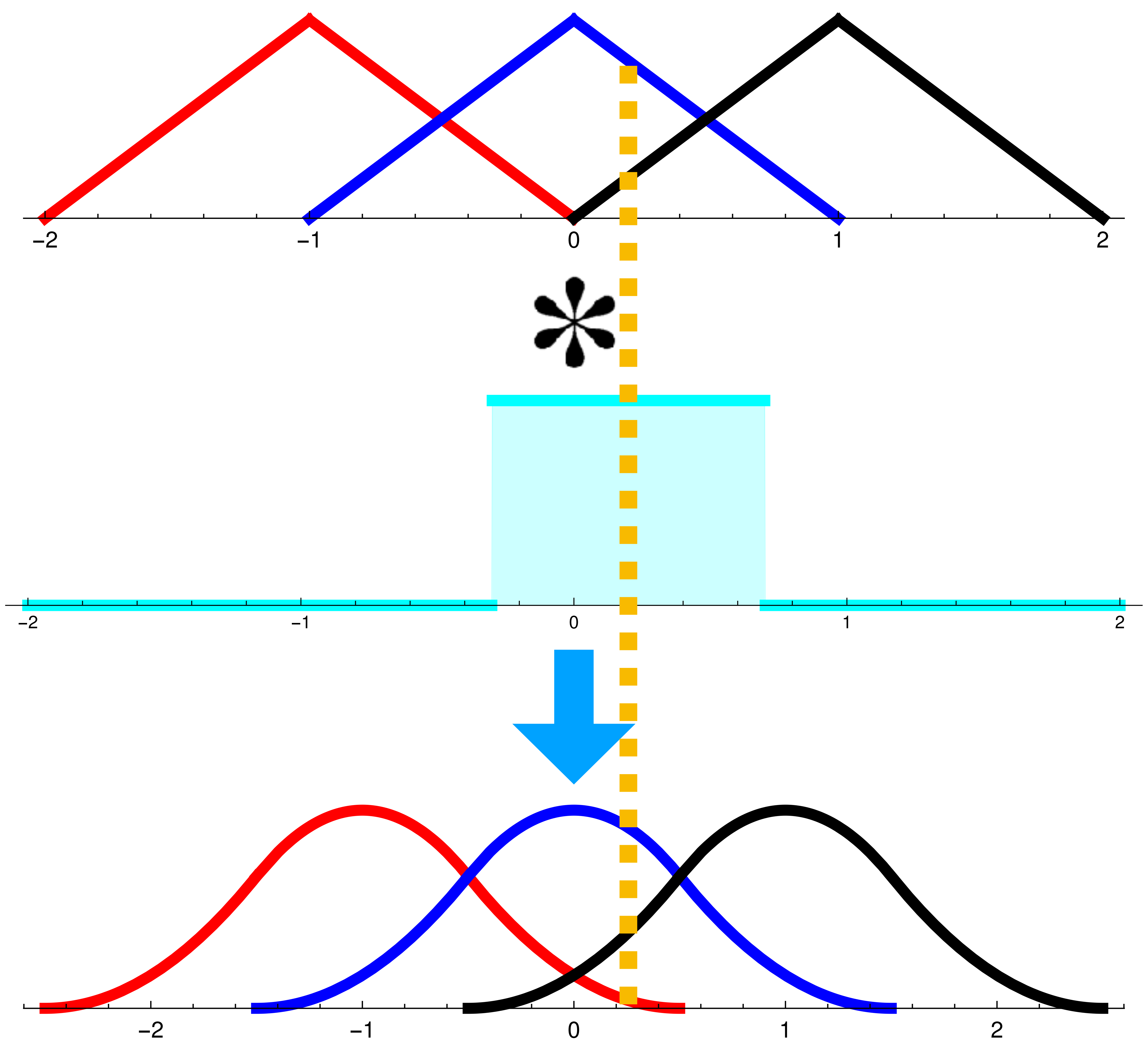
Step 2 - constrain T-junctions

- Free node
- T-junction node
- Parent node





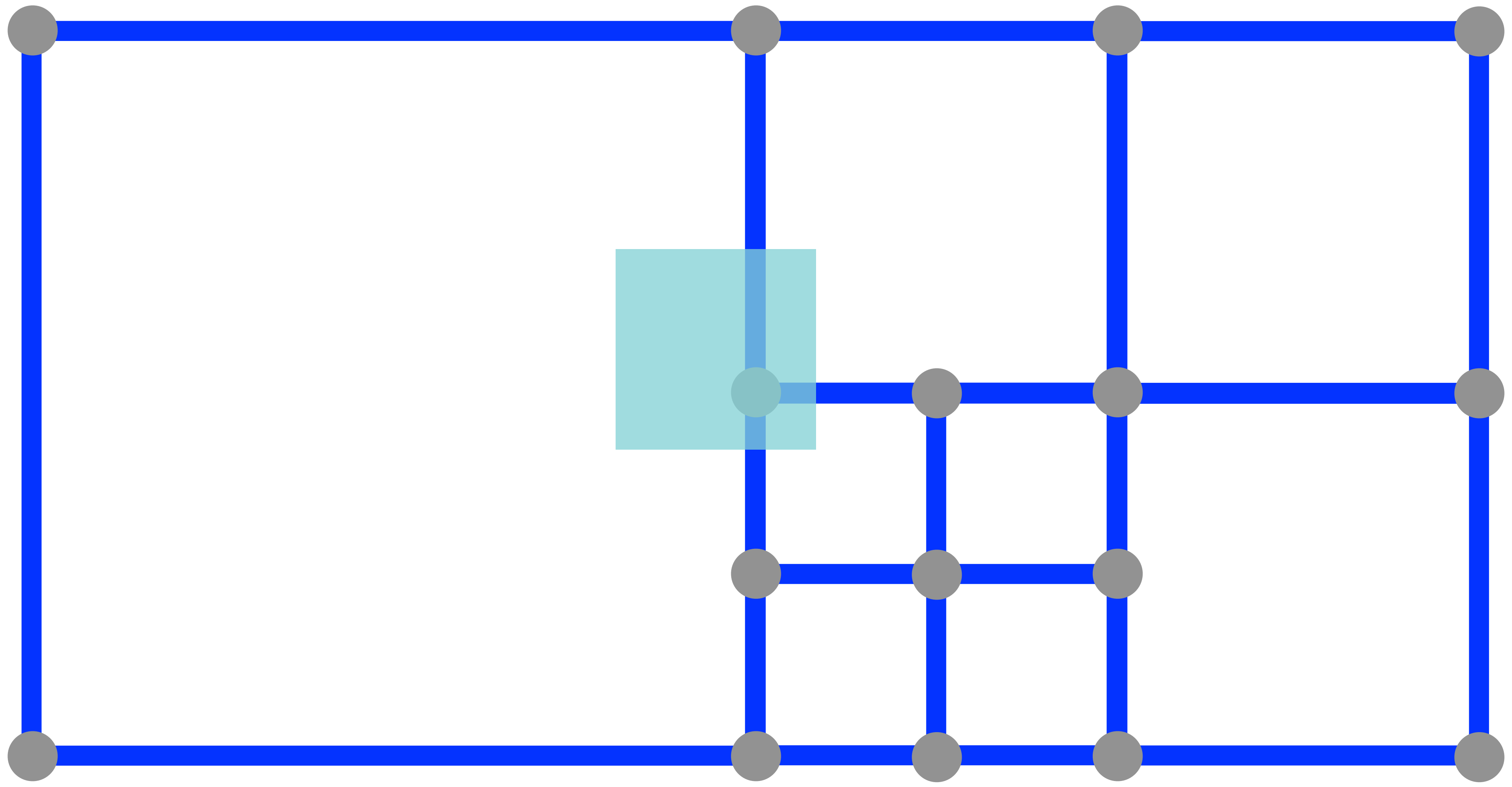
Step 3 - upgrade to C^1 continuity



GIMP

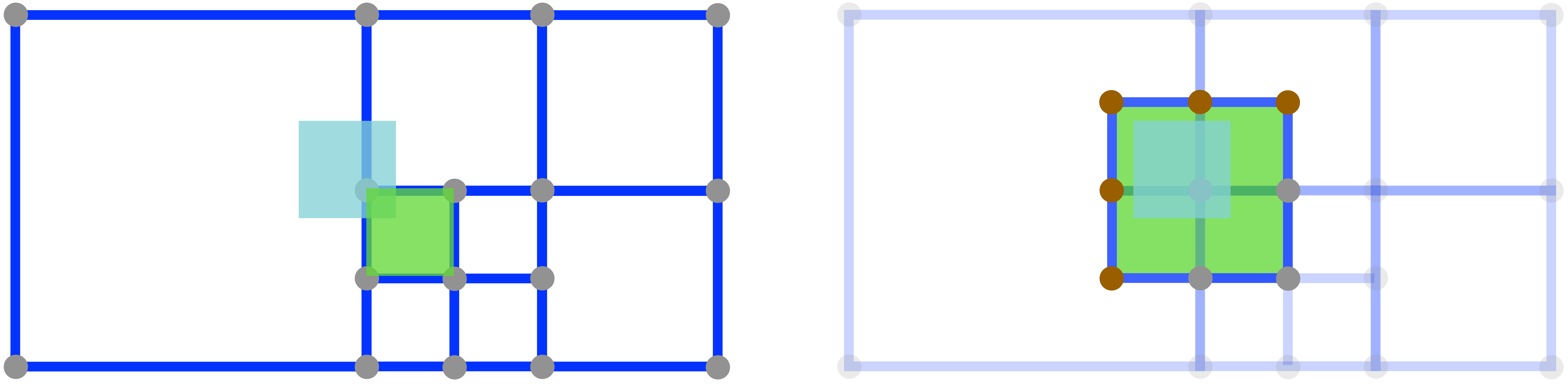
Step 3 - upgrade to C^1 continuity

- Free node



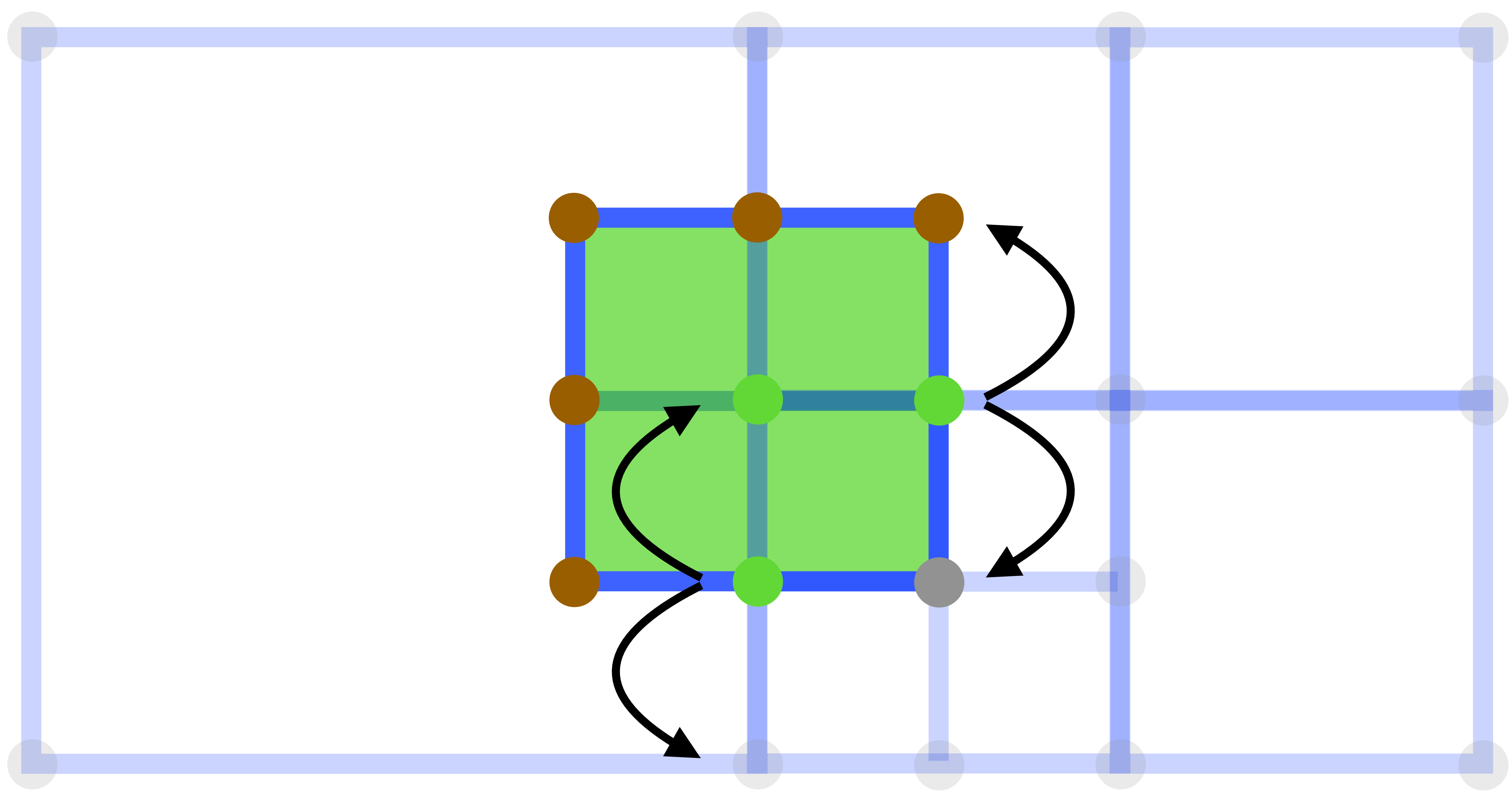
Parallelism optimization

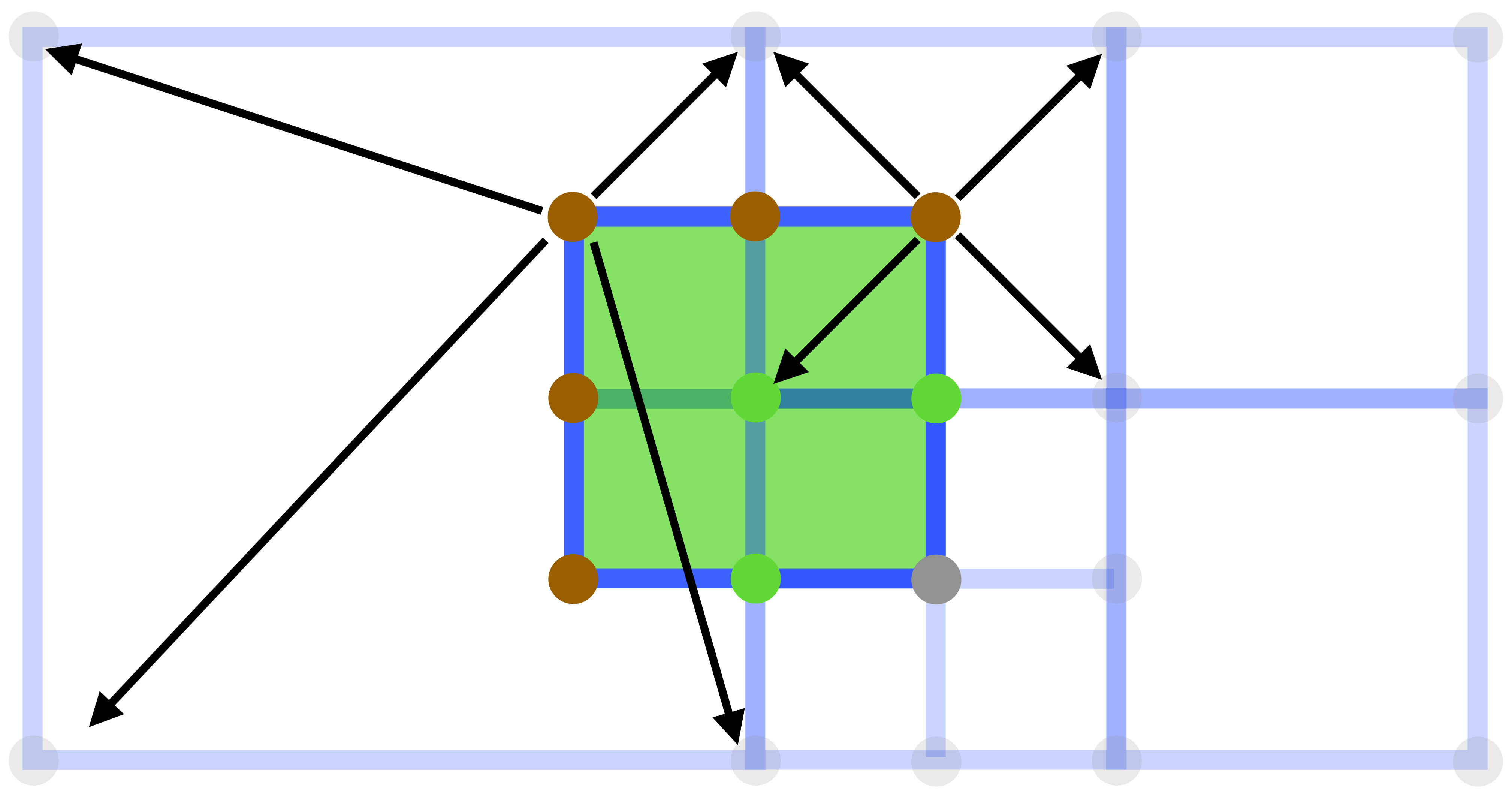
- Free node
- Ghost node



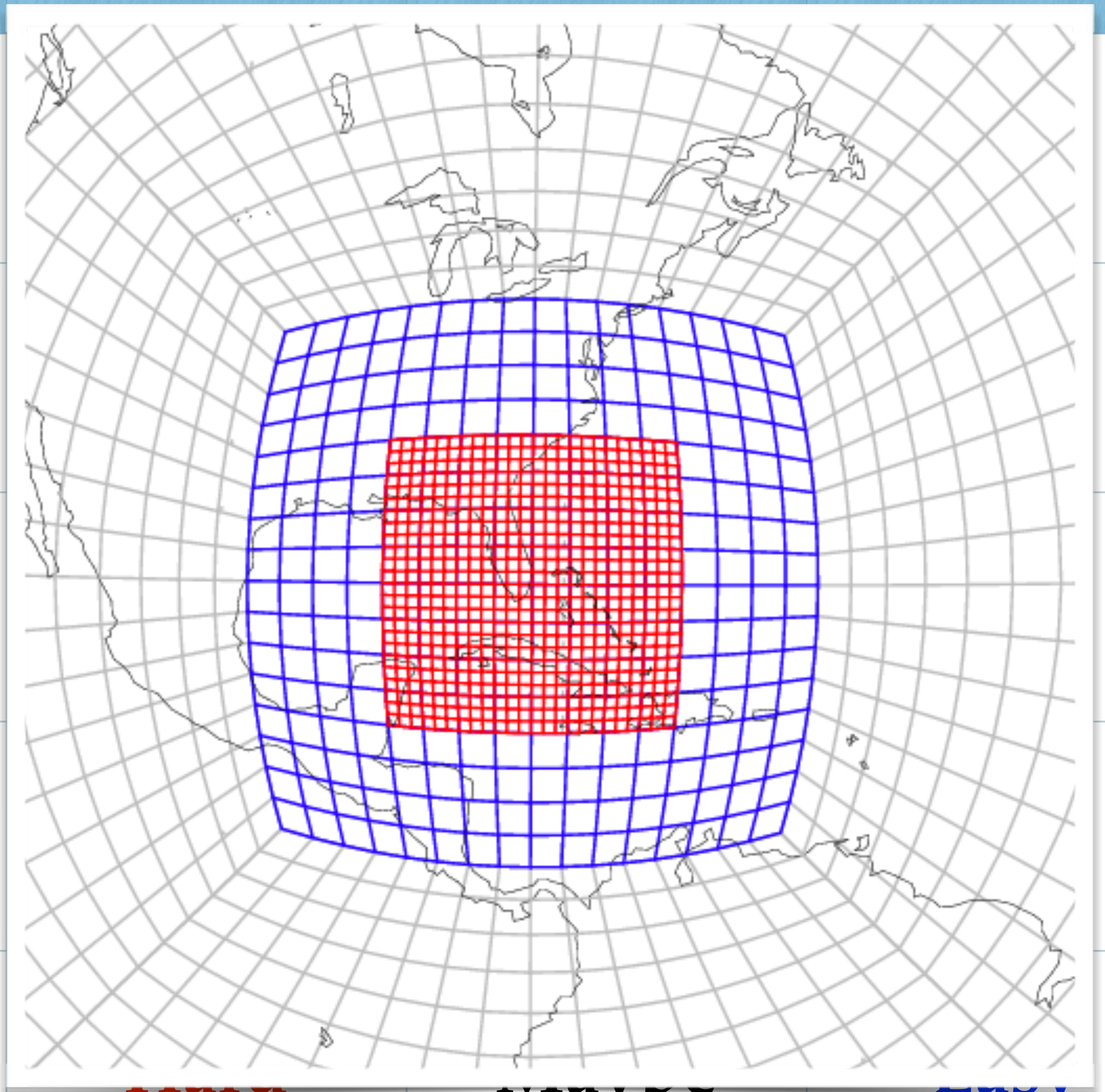
Parallelism optimization

- Free node
- Ghost node
- T-junction node



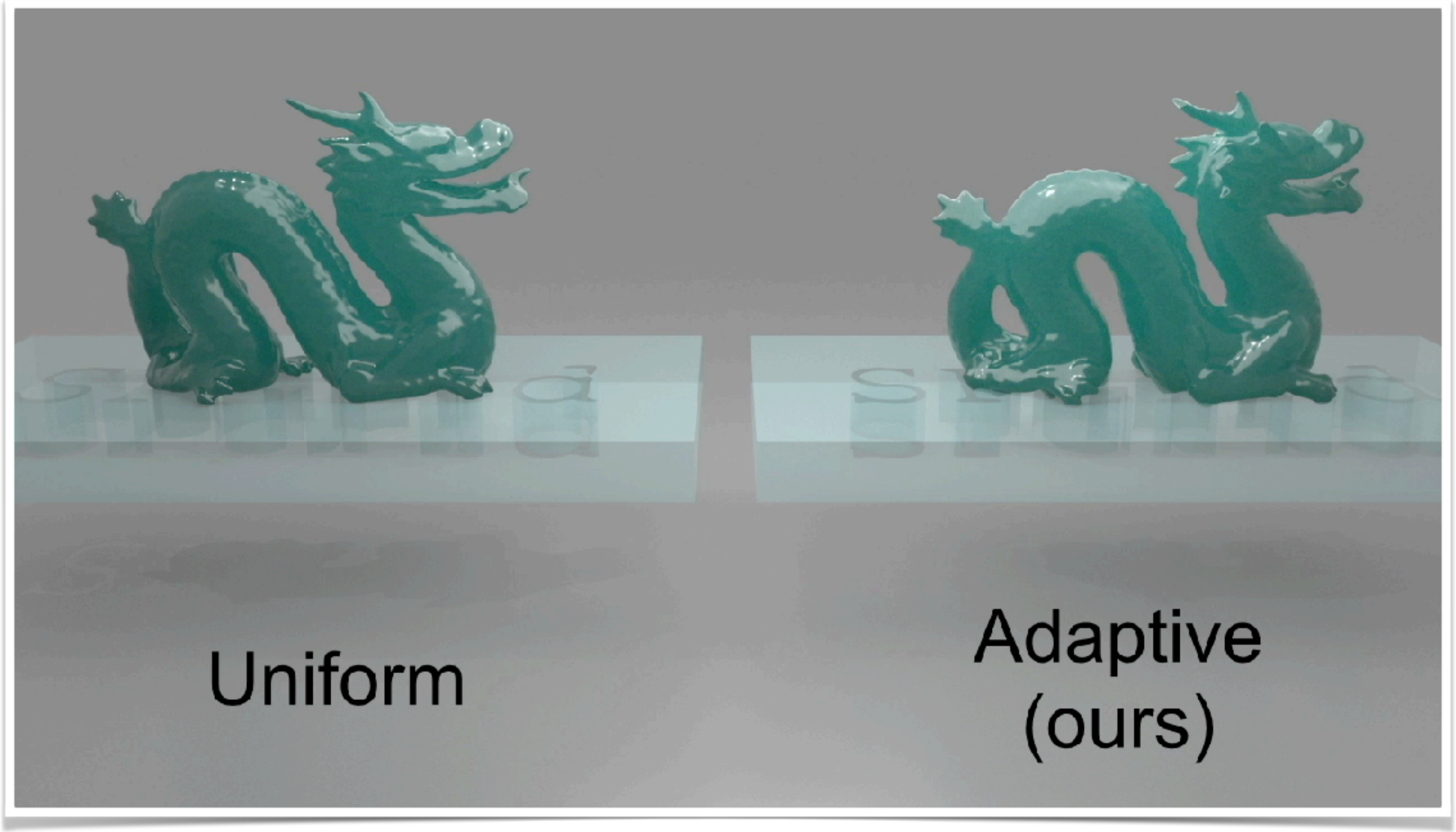


	Tan 02	Ma 05	Lian 14	Lian 15	Our
C1 continuity	No	Yes			
Partition of unity	Yes	Yes			
Non-negativity	Yes	Yes			
Arbitrary octree	Yes	No			
Ease of parallelism	Hard	Maybe			

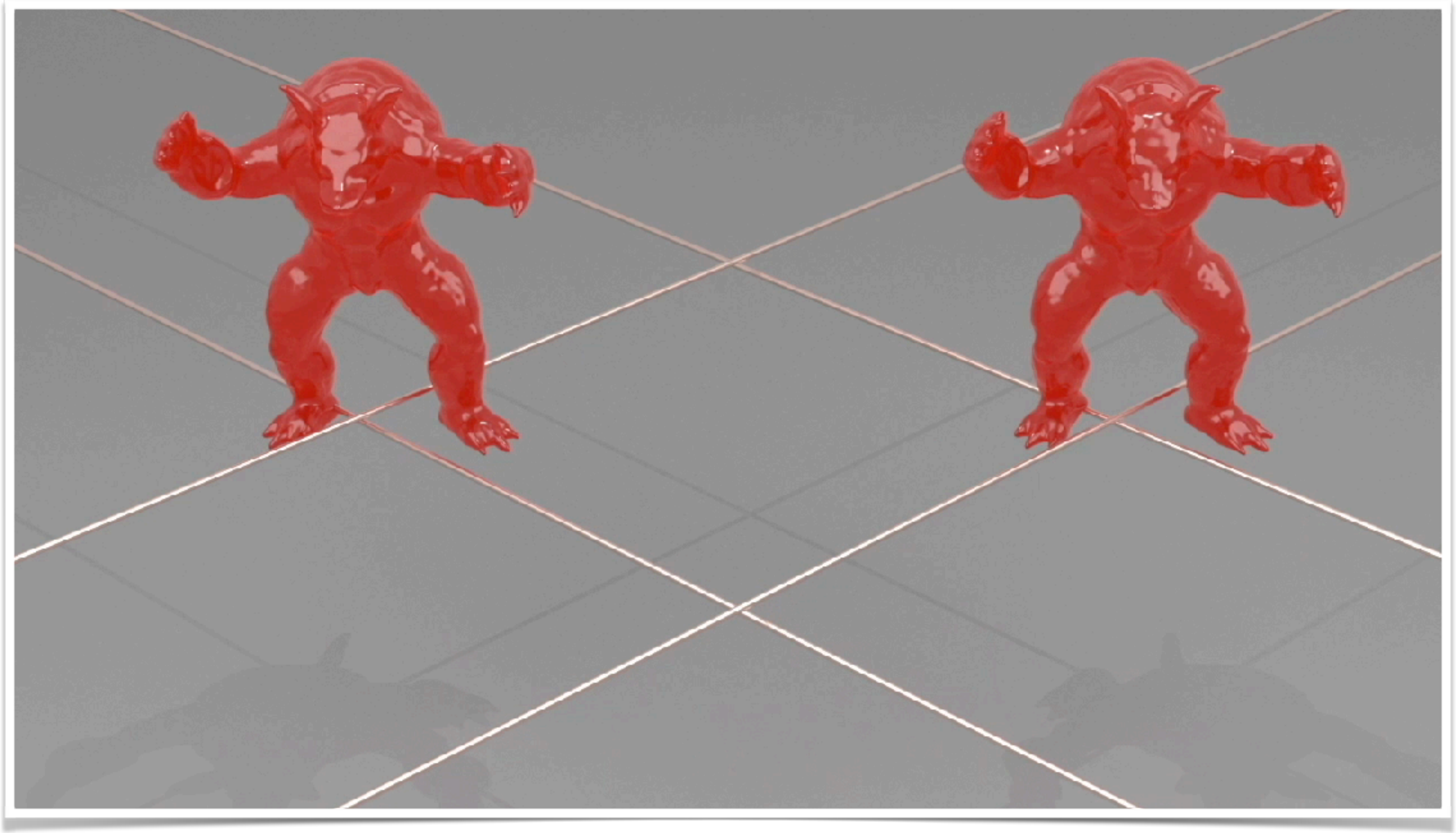


H. Tan and J. A. Nairn. 2002. **Hierarchical, adaptive, material point method for dynamic energy release rate calculations**.
J. Ma, H. Lu, B. Wang, S. Roy, R. Hornung, A. Wissink, and R. Komanduri. 2005. **Multiscale simulations using generalized interpolation material point (GIMP) method and SAMRAI parallel processing.** Comp Model Eng & Sci 8, 2 (2005), 135–152.
Y. Lian, X. Zhang, F. Zhang, and X. Cui. 2014. **Tied interface grid material point method for problems with localized extreme deformation.** Int J Imp Eng 70 (2014), 50–61.
Y.P. Lian, P.F. Yang, X. Zhang, F. Zhang, Y. Liu, and P. Huang. 2015. **A mesh-grading material point method and its parallelization for problems with localized extreme deformation.** Comp Meth App Mech Eng 289 (2015), 291 – 315.

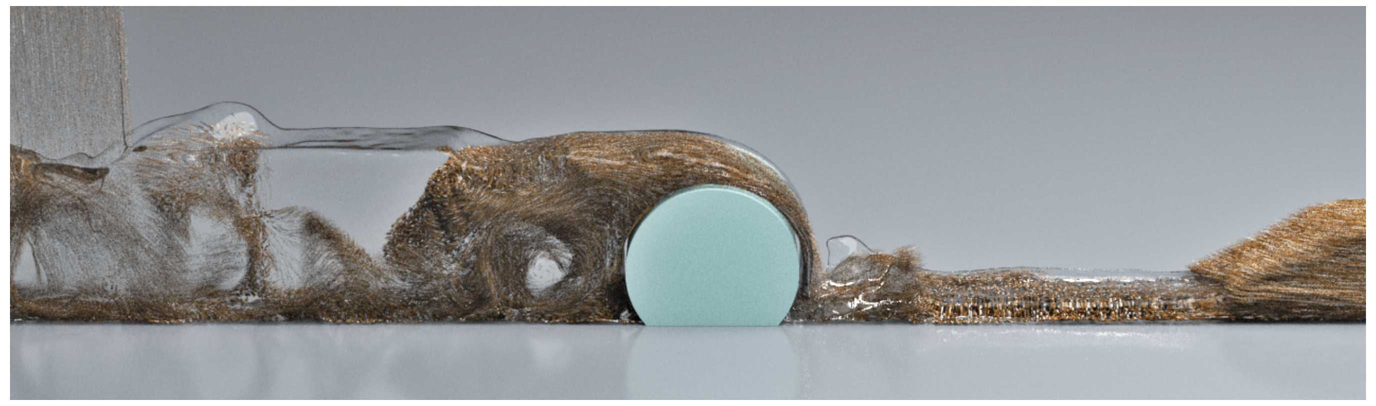
Results



Results



Animating Fluid Sediment Mixture in Particle-Laden Flows



Animating Fluid Sediment Mixture in Particle-Laden Flows

MING GAO, University of Wisconsin, Madison
ANDRE PRADHANA, University of Pennsylvania
XUCHEN HAN, University of California, Los Angeles
QI GUO, University of California, Los Angeles
GRANT KOT, Phosphorus
EFTYCHIOS SIFAKIS, University of Wisconsin, Madison
CHENFANFU JIANG, University of Pennsylvania

Fig. 1. **Sediment transport:** Our method can animate intricate two-way coupled particle-laden flows such as sediment transport in liquid.

In this paper, we present a mixed explicit and semi-implicit Material Point Method for simulating particle-laden flows. We develop a Multigrid Preconditioned fluid solver for the Locally Averaged Navier Stokes equation. This is discretized purely on a semi-staggered standard MPM grid. Sedimentation is modeled with the Drucker-Prager elastoplasticity flow rule, enhanced by a novel particle density estimation method for converting particles between representations of either continuum or discrete points. Fluid and sediment are two-way coupled through a momentum exchange force that can be easily resolved with two MPM background grids. We present various results to demonstrate the efficacy of our method.

CCS Concepts: • **Computing methodologies** → **Physical simulation**;

Additional Key Words and Phrases: Material Point Method (MPM), particle-fluid interaction, multiphase, sedimentation, sediment transport

ACM Reference format:
Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. 2018. Animating Fluid Sediment Mixture in Particle-Laden Flows. *ACM Trans. Graph.* 37, 4, Article 1 (August 2018), 11 pages.
DOI: 10.1145/3197517.3201309

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 ACM. 0730-0301/2018/8-ART1 $15.00
DOI: 10.1145/3197517.3201309

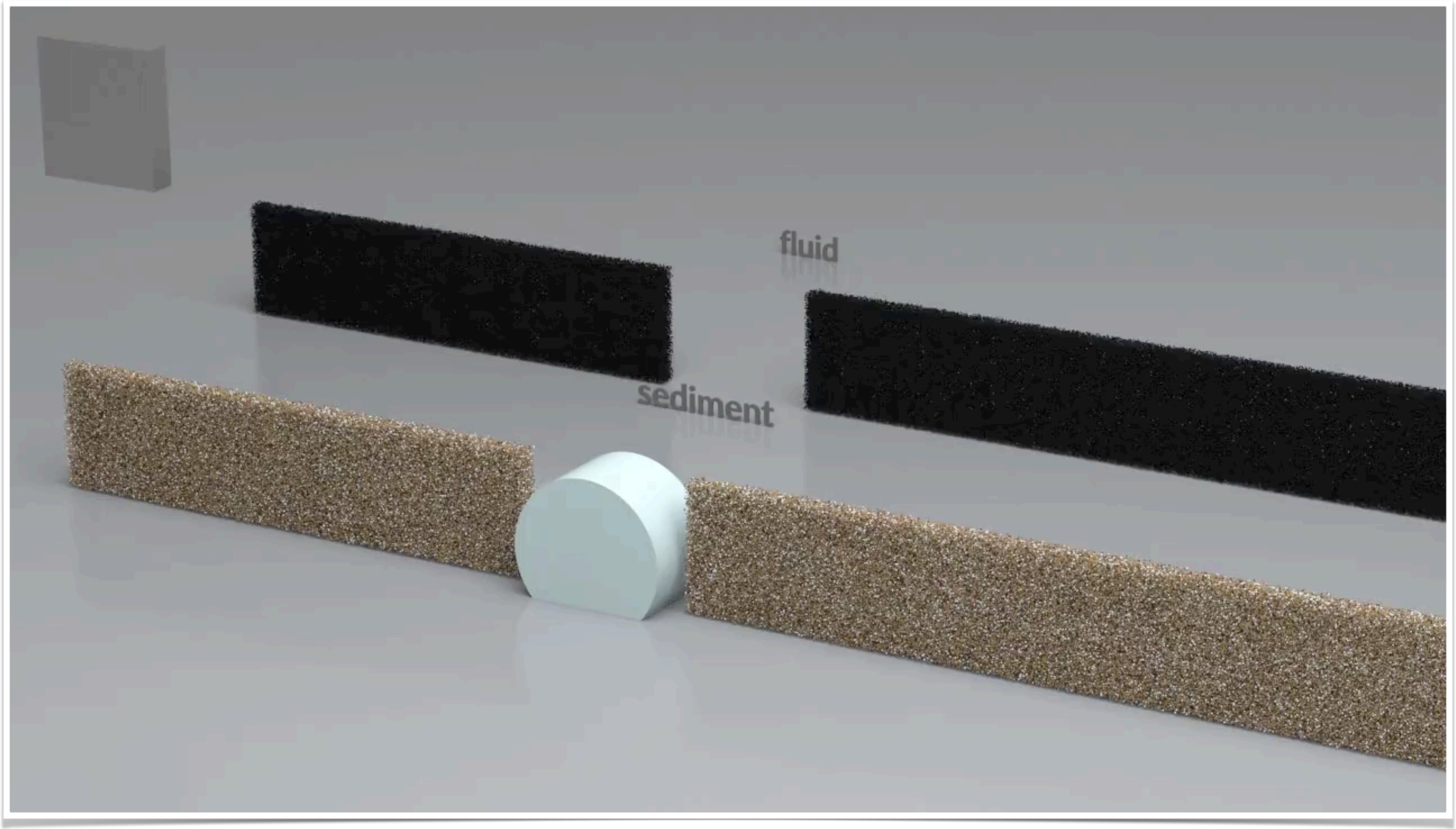
1 INTRODUCTION

Recently, multi-phase multi-material simulations are increasingly gaining attention from computer graphics researchers. Simulating various phases or materials in a unified framework is particularly favored. Existing work includes coupled Lagrangian particle simulation with Position Based Dynamics (PBD) [Macklin et al. 2014], water-gas mixtures [Nielsen and Østerby 2013] with an Eulerian method, solid-fluid phase-change [Stomakhin et al. 2014] and porous granular media [Pradhana-Tampubolon et al. 2017] with Material Point Method (MPM), as well as interactive solids and fluids based on the mixture model with Smoothed Particle Hydrodynamics (SPH) [Yan et al. 2016].

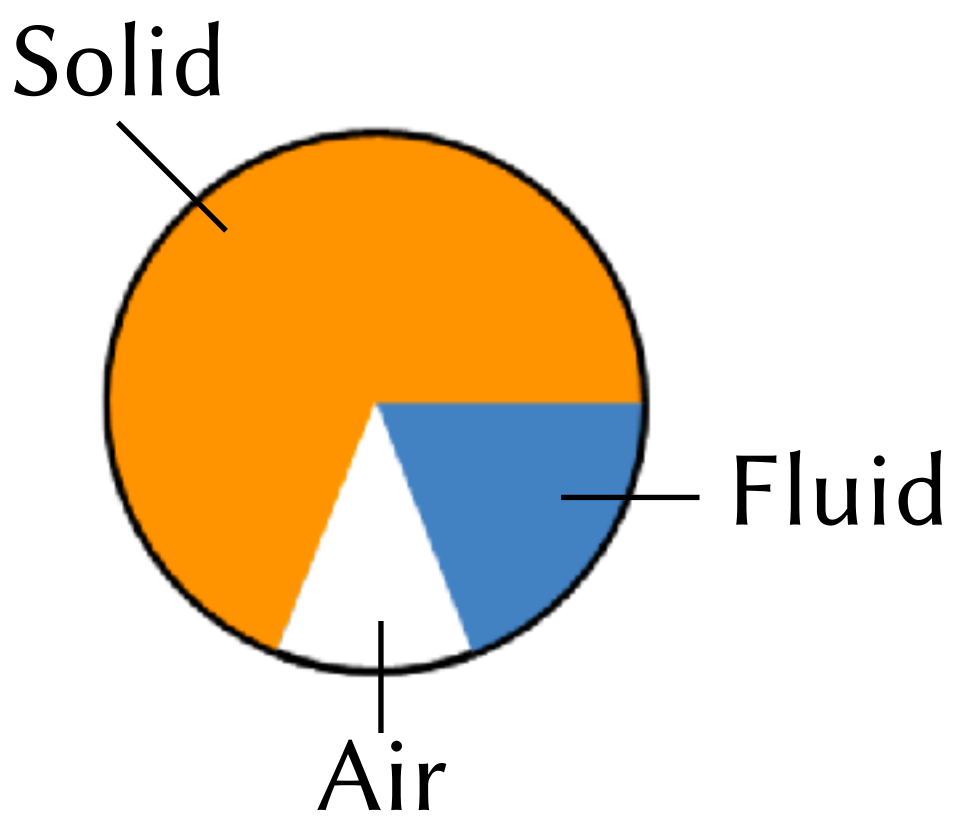
Most of the existing approaches are based on *continuum* mixture theory [Manninen et al. 1996]. The continuum assumption for each material phase is essential for simulations of macroscopic porous media (e.g., landslides and liquid blending). However, it may fail to capture the correct behavior of particle-laden flows where the solid phase is on a relatively small scale. Note that particle-laden sediment flow is ubiquitous in natural systems. Typical examples include sediment transport, sedimentation, volcano eruption, dune migration by erosion with ripples, and dust storms. The significance of understanding and simulating these phenomena is also recognized in many engineering applications, such as granular material fluidization [van der Hoef et al. 2006] and coastal erosion prediction [Sun and Xiao 2016a].

ACM Transactions on Graphics, Vol. 37, No. 4, Article 1. Publication date: August 2018.

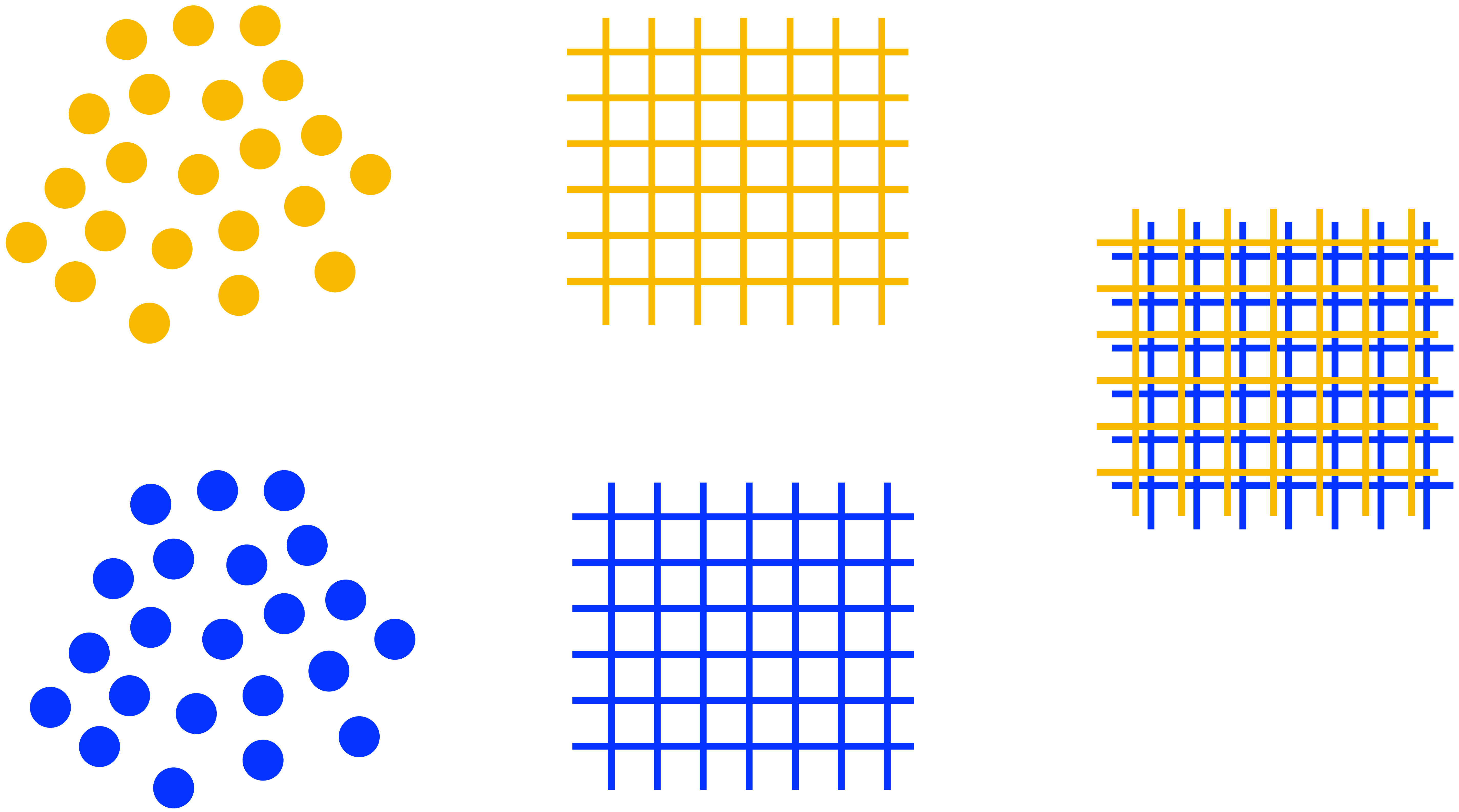
M. Gao, A. Tampubolon, X. Han, Q. Guo, G. Kot, E. Sifakis, C. Jiang
ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), 2018


fluid
sediment

Approach: mixture in particles vs. grid



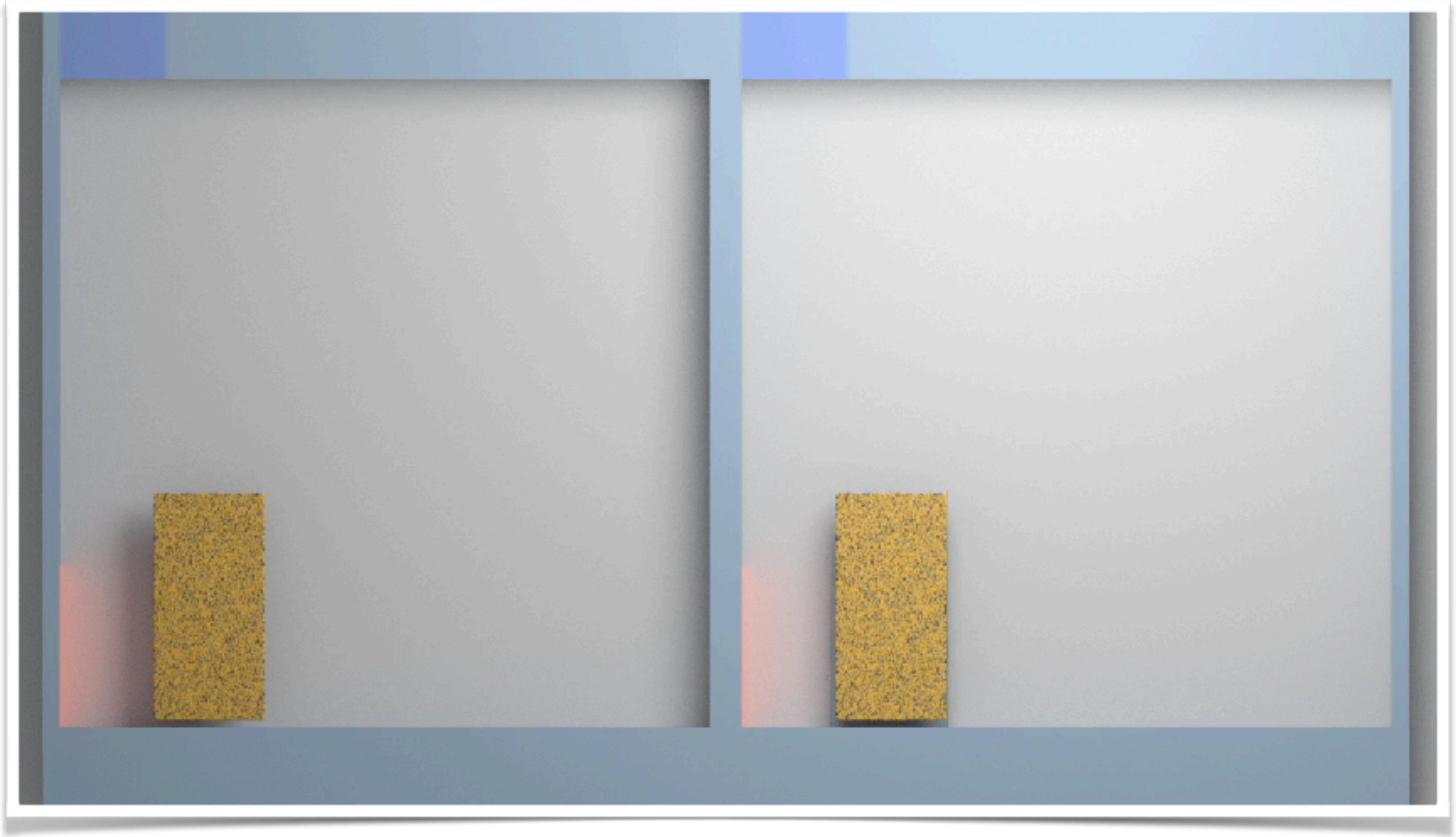
Colom et al. 15



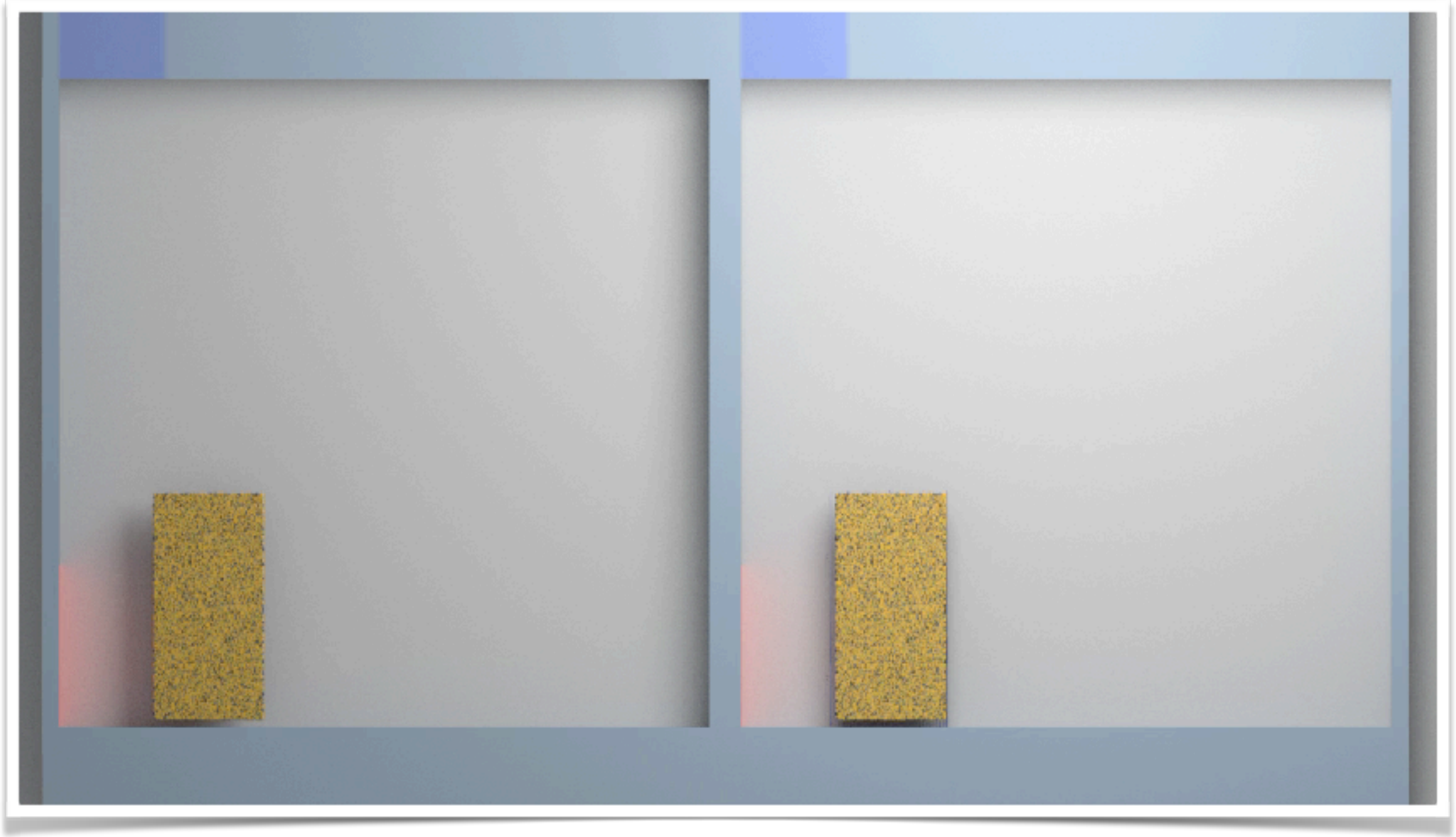
3 phases 1 point

2 phases 2 point

Approach: one-way coupling vs. two-way

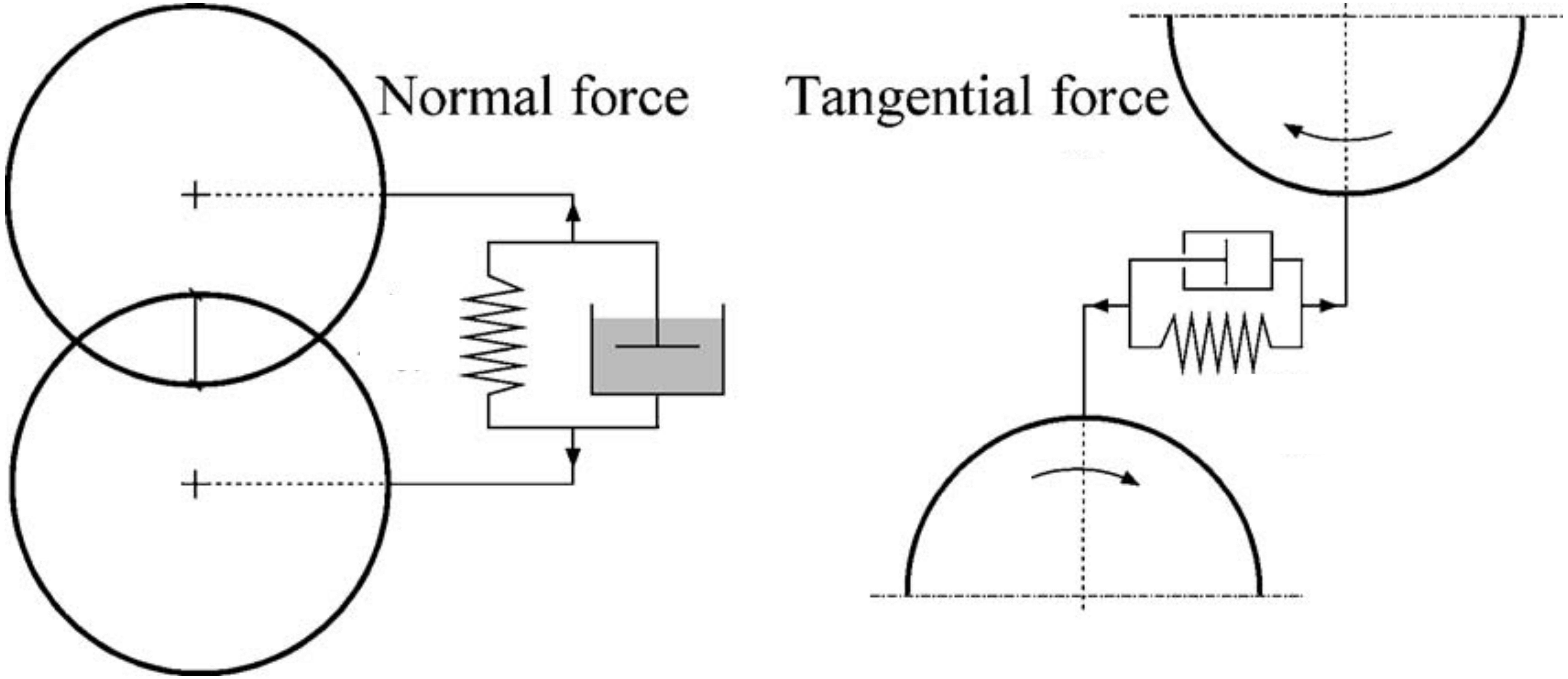


One way coupling



Two way coupling

Approach: DEM vs. MPM

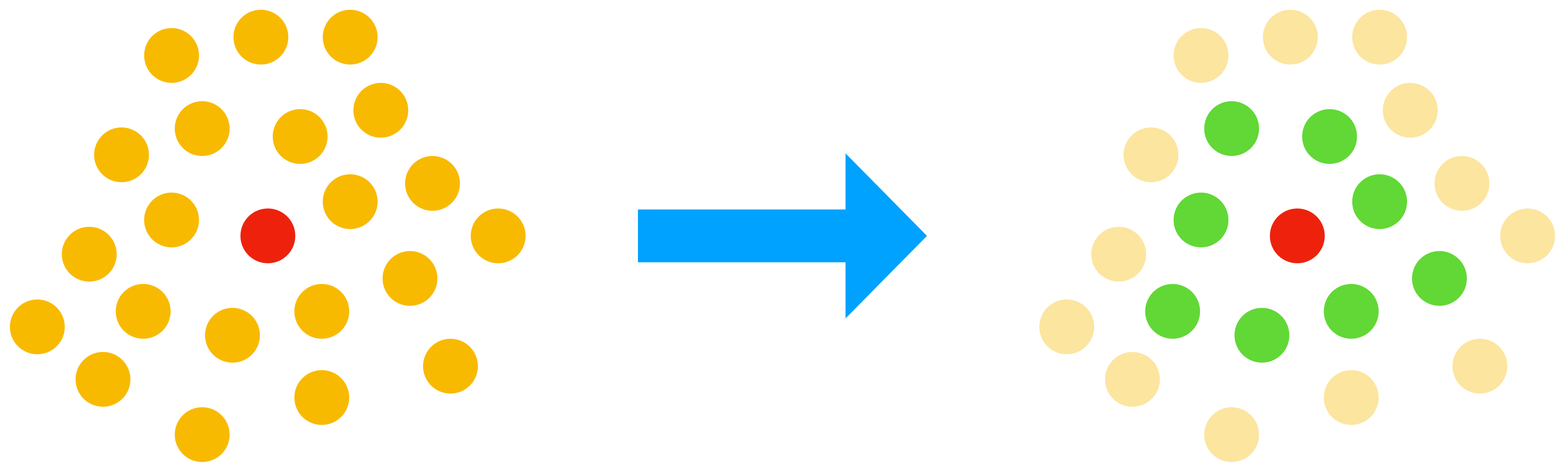


Krugger-Emden et al. 05

Translational interaction

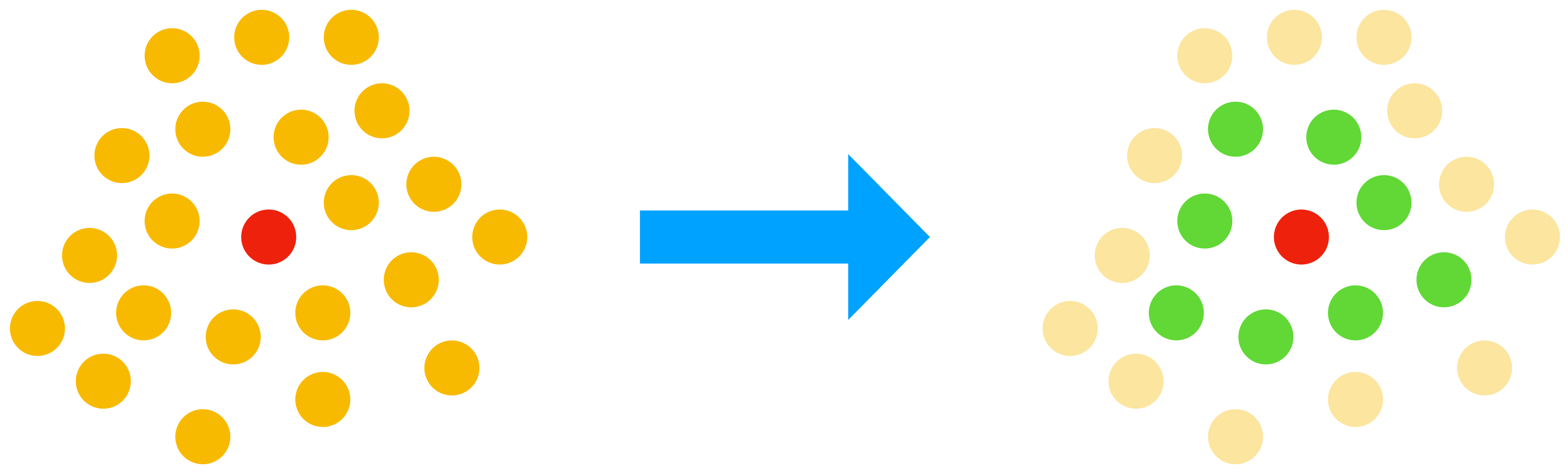
Rotational interaction

Approach: DEM vs. MPM

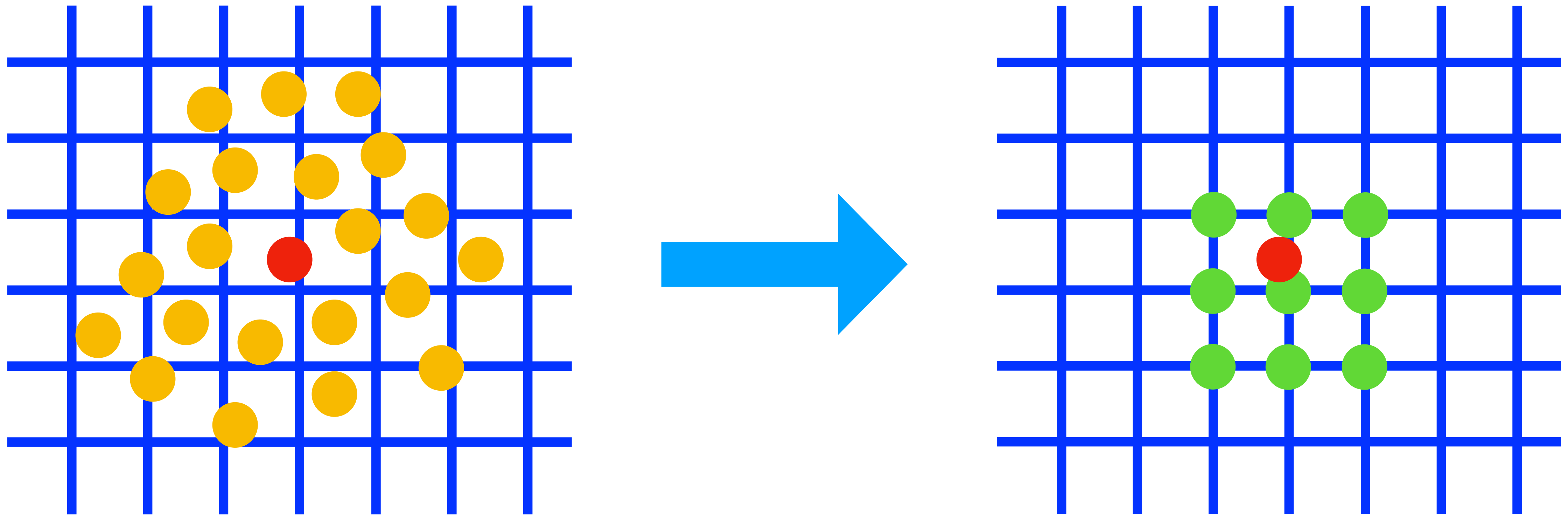


DEM - discrete view

Approach: DEM vs. MPM

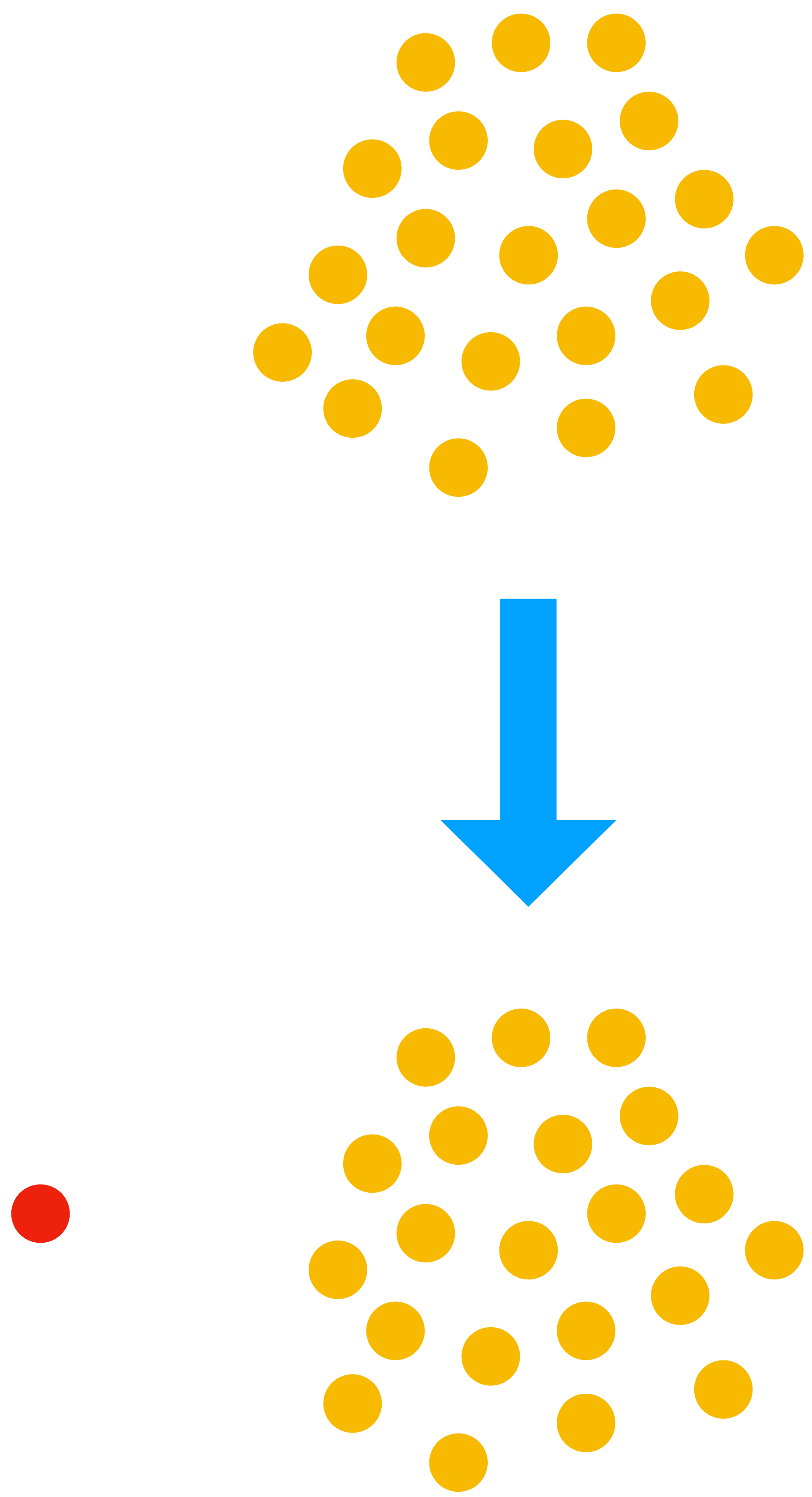


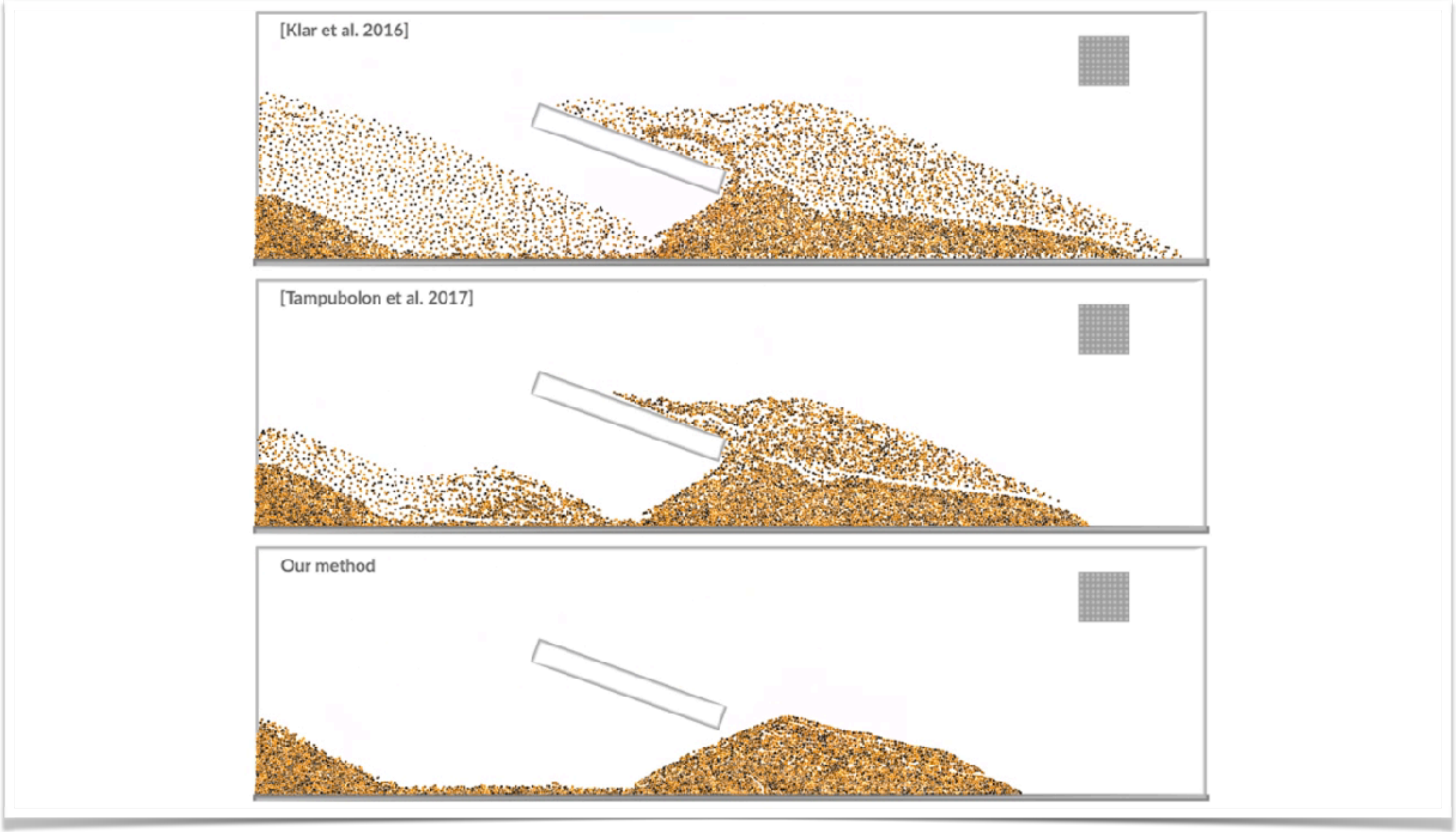
DEM - discrete view



MPM - continuum view

Challenge of MPM: handle discrete particles

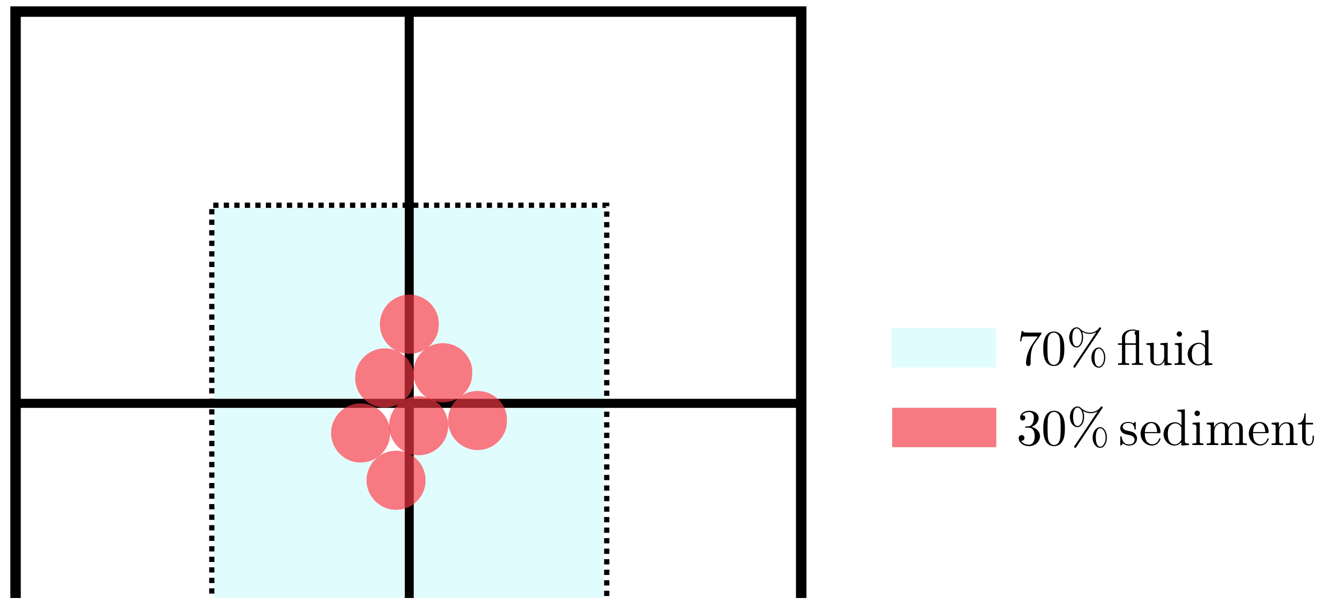




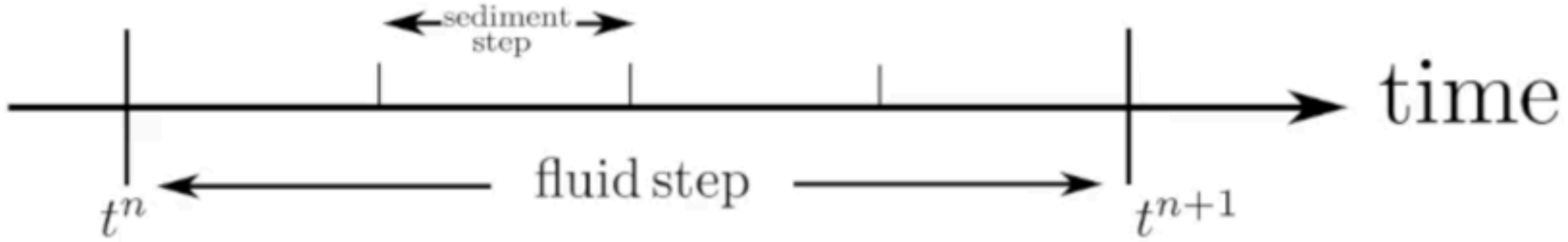
Volume gain problem

Challenges

- Mixture theory

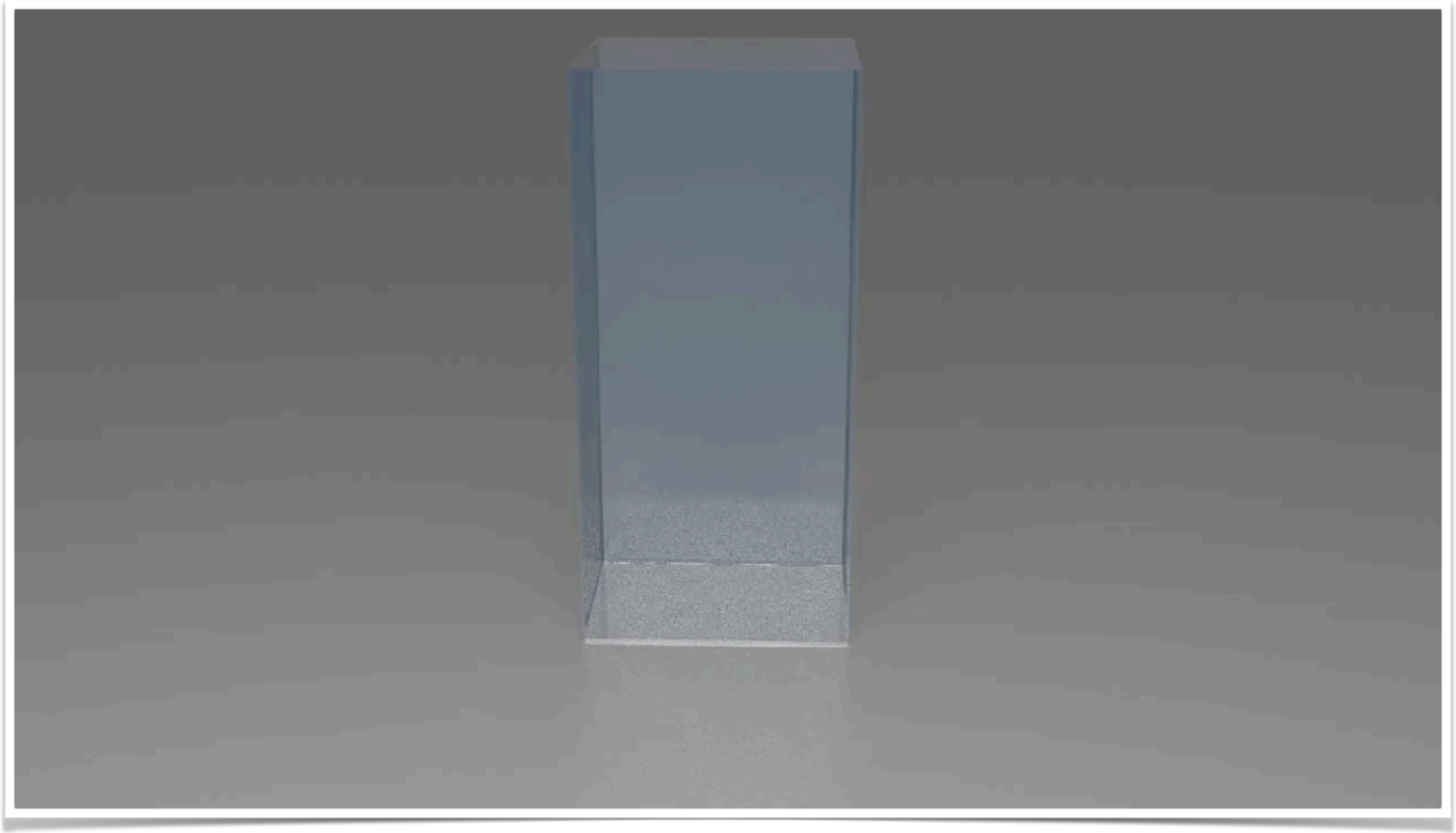


- Sub-stepping

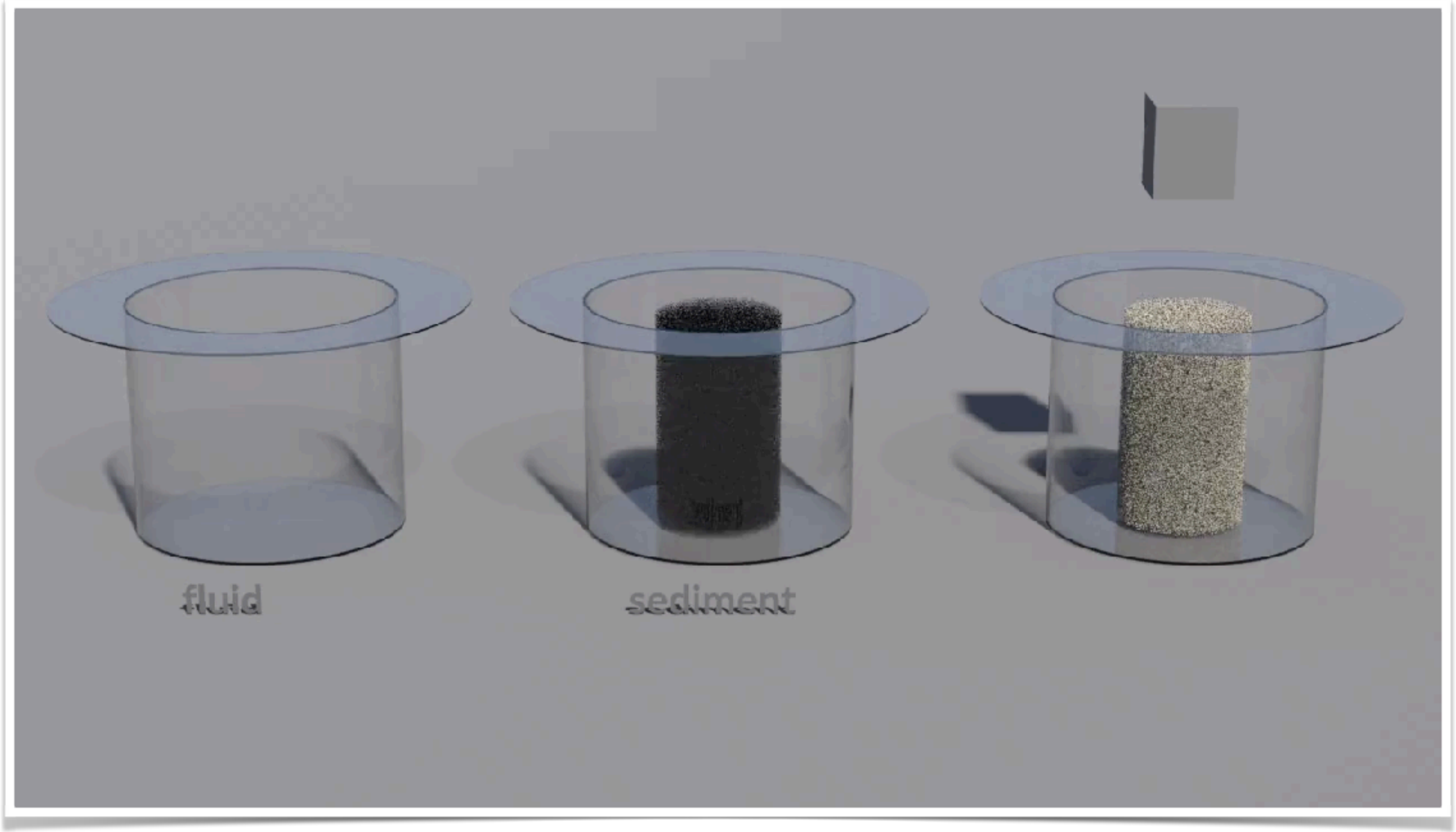


- Momentum conservation

Results



Results



GPU Optimization of Material Point Method

GPU Optimization of Material Point Methods

Paper ID: 250

ANONYMOUS AUTHOR(S)

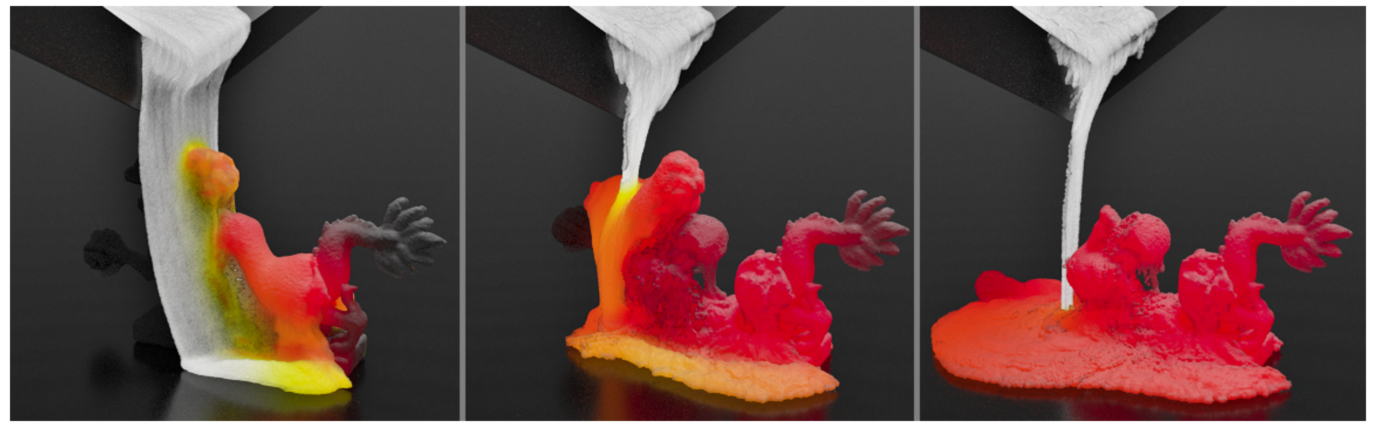


Fig. 1. **How to melt your dragon.** Melting an elastoplastic dragon with 4.2 million particles on a 256^3 grid using our GPU-optimized implicit MPM dynamics and heat solvers on a Nvidia *Quadro* P6000 GPU at an average 10.5 seconds per $48Hz$ frame.

The Material Point Method (MPM) has been shown to facilitate effective simulations of physically complex and topologically challenging materials, with a wealth of emerging applications in computational engineering and visual computing. Borne out of the extreme importance of regularity, MPM is given attractive parallelization opportunities on high-performance modern multiprocessors. Unlike the conceptually simple CPU parallelization, a GPU optimization of MPM that fully leverages computing resources presents challenges that require exploring an extensive design-space for favorable data structures and algorithms. In this paper we introduce methods for addressing the computational challenges of MPM and extending the capabilities of general simulation systems based on MPM, particularly concentrating on GPU optimization. In addition to our open-source high-performance framework, we also perform performance analyses and benchmark experiments to compare against alternative design choices which may superficially appear to be reasonable, but can suffer from suboptimal performance in practice. Our explicit and fully implicit GPU MPM solvers are further equipped with a Moving Least Squares MPM heat solver and a novel sand constitutive model to enable fast simulations of a wide range of materials. We demonstrate that more than an order of magnitude performance improvement can be achieved with our GPU solvers. Practical high-resolution examples with up to ten million particles run in less than one minute per frame.

CCS Concepts: • **Computing methodologies → Physical simulation**;

Additional Key Words and Phrases: Material Point Method (MPM), GPU, SPGrid, GVDB, Hybrid Particle/Grid

DOI: 10.1145/3197517.3201309

ACM Reference format:

Anonymous Author(s). 2018. GPU Optimization of Material Point Methods . *ACM Trans. Graph.* 1, 1, Article 1 (November 2018), 11 pages.

DOI: 10.1145/3197517.3201309

1 INTRODUCTION

The Material Point Method (MPM) is a hybrid Lagrangian/Eulerian computational scheme that has been shown to simulate a large variety of traditionally-challenging materials with visually rich animations in computer graphics. Recent examples of MPM-based methods developed for such materials include simulations of snow [Stomakhin et al. 2013], granular solids [Klár et al. 2016], multi-phase mixtures [Gao et al. 2018; Stomakhin et al. 2014; Tampubolon et al. 2017], cloth [Jiang et al. 2017a] and many others. MPM has been shown to be particularly effective for simulations involving a large number of particles with complex interactions. However, the size and the complexity of these simulations lead to substantial demands on computational resources, thereby limiting the practical use cases of MPM in computer graphics applications.

Using the parallel computation power of today's GPUs is an attractive direction for addressing computational requirements of simulations with MPM. However, the algorithmic composition of an MPM simulation pipeline can pose challenges in fully leveraging compute resources in a GPU implementation. Indeed, MPM simulations include multiple stages with different computational profiles, and the choice of data structures and algorithms used for handling some stages can have cascading effects on the performance of the remaining computation. Thus, discovering how to achieve a performant GPU implementation of MPM involves a software-level design-space exploration for determining the favorable combinations of data structures and algorithms for handling each stage.

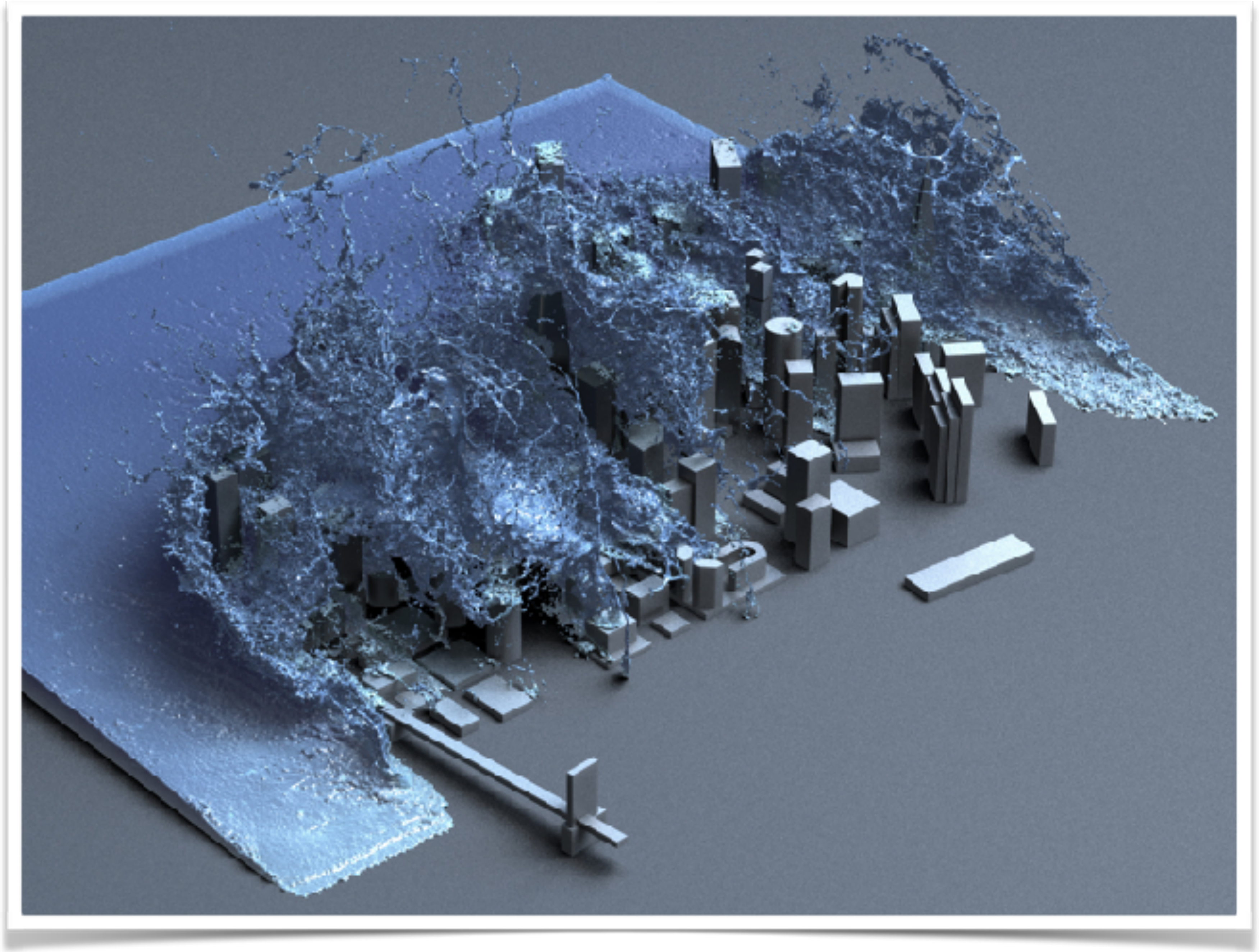
M. Gao*, X. Wang*, K. Wu***(joint first authors)**,
A. Tampubolon, E. Sifakis, C. Yuksel, C. Jiang

SIGGRAPH Asia 2018 (under review)

SIGGRAPH ASIA 2018 TOKYO

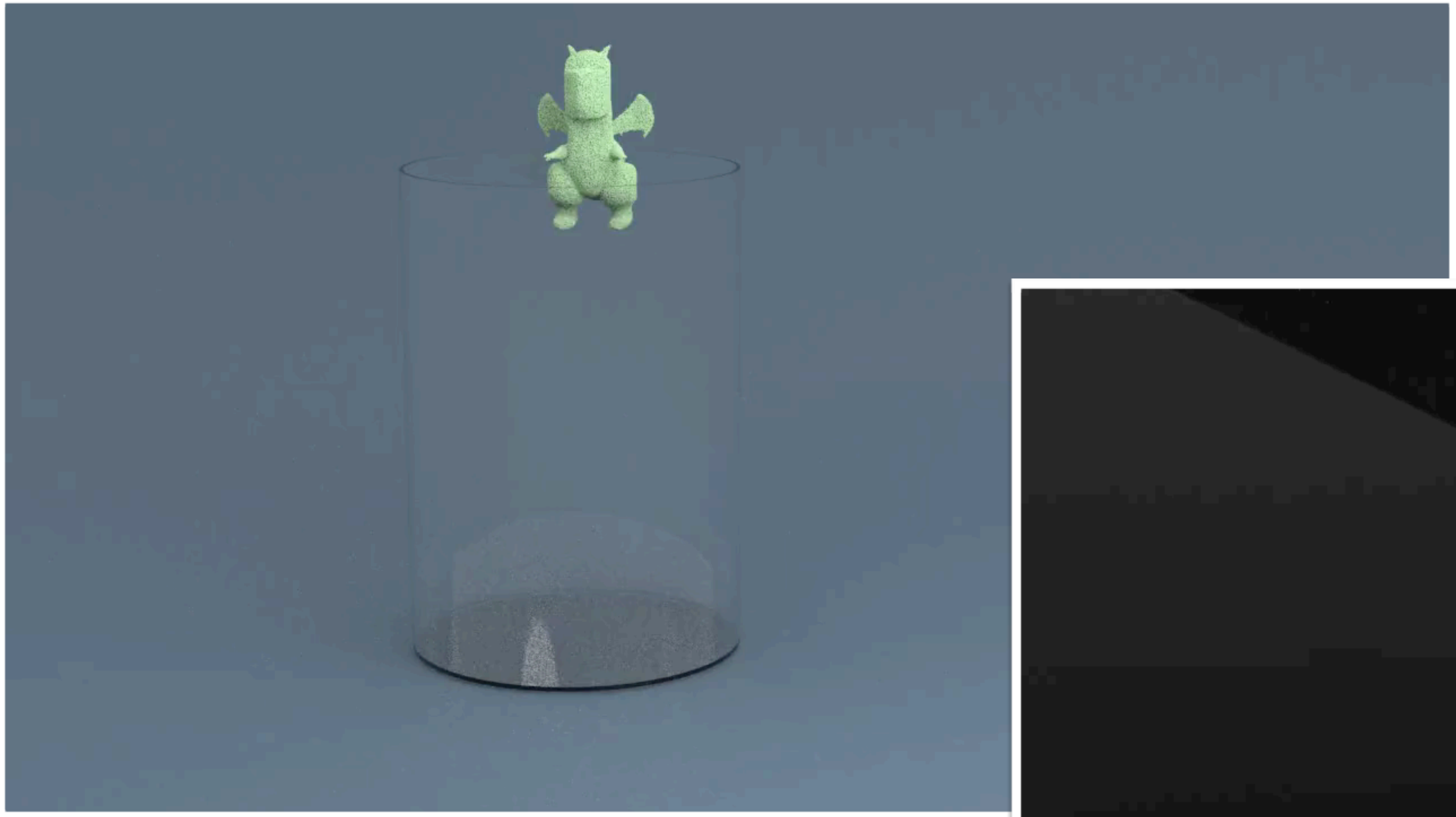
Acceleration for hybrid methods

Klar et al. 17



Wu et al. 18

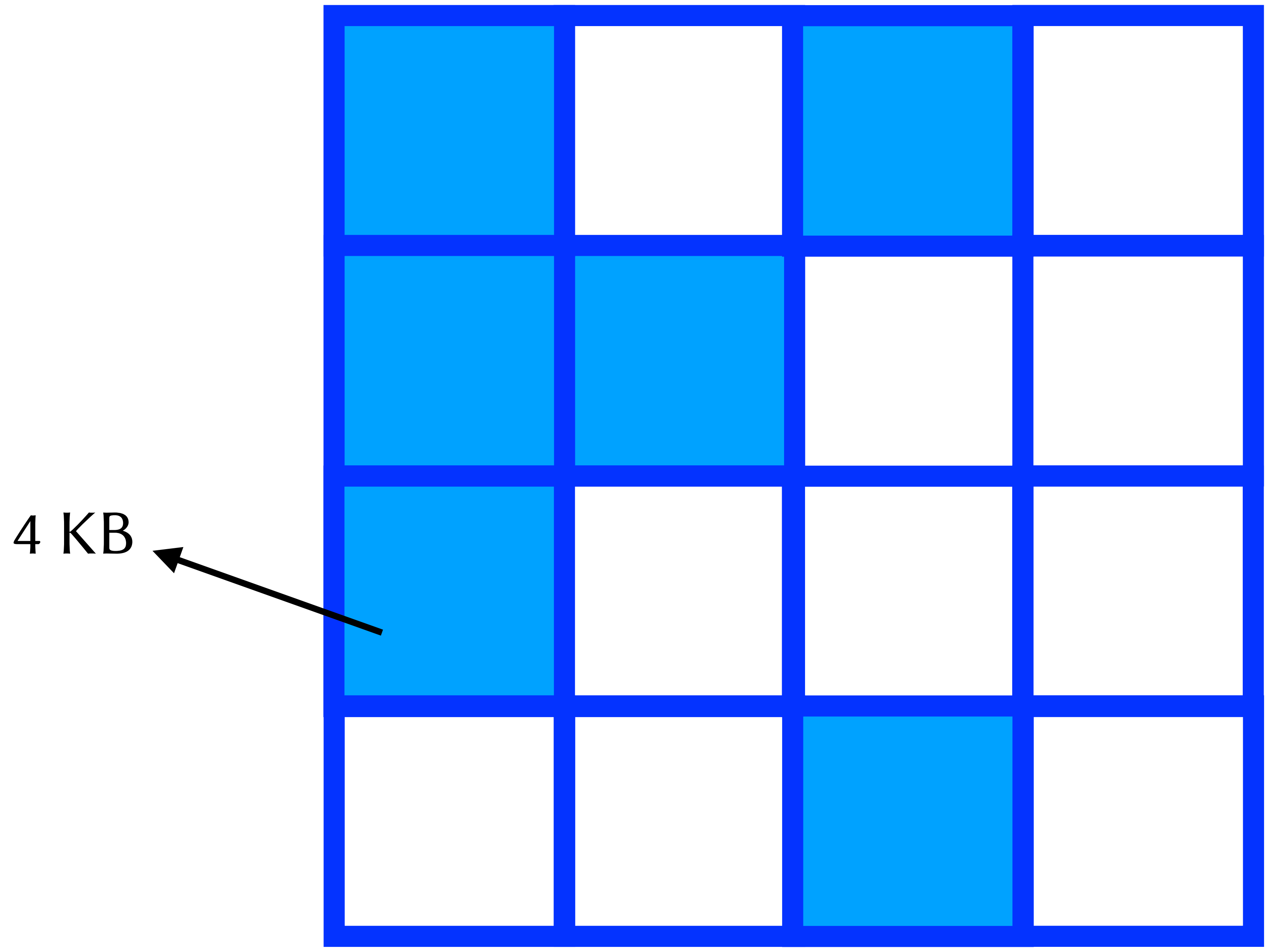
Target benchmarks



Particles: 9.0 M
Grid resolution: 512^3
Simulation: 21.88 secs/frame

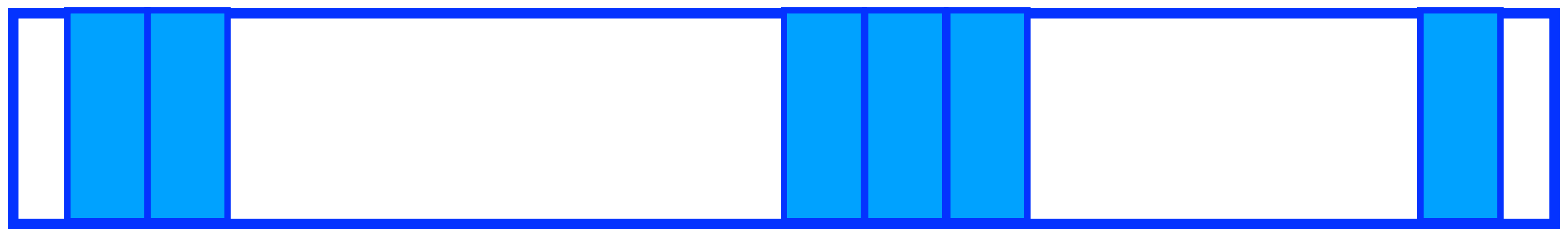
Particles: 4.2 M
Grid resolution: 256^3
Simulation: 10.48 secs/frame

Sparsity - (G)SPGrid

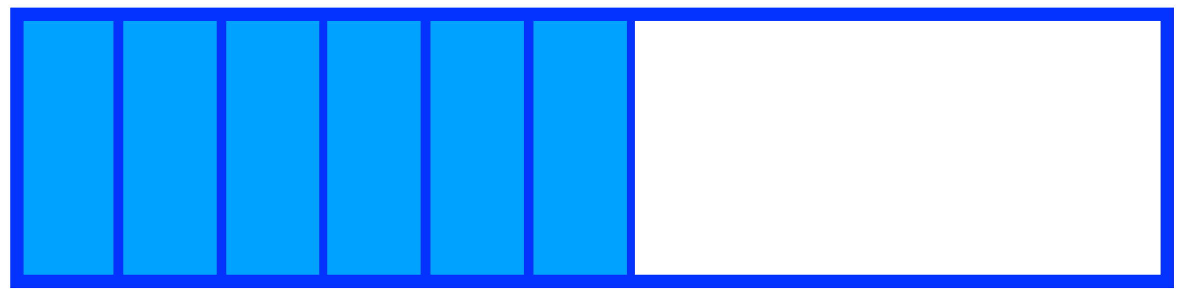


CPU - SPGird GPU - GSPGird

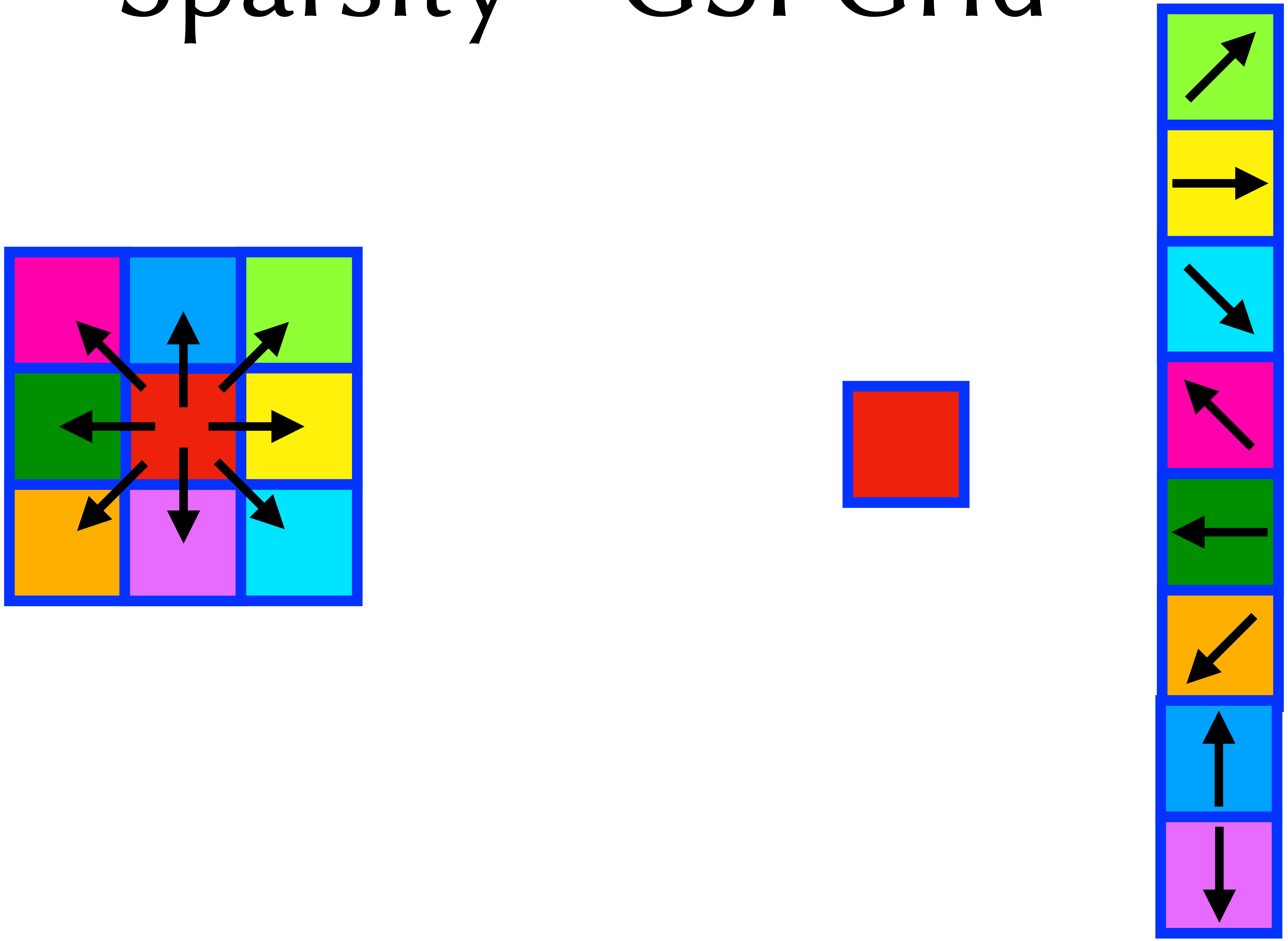
CPU virtual memory



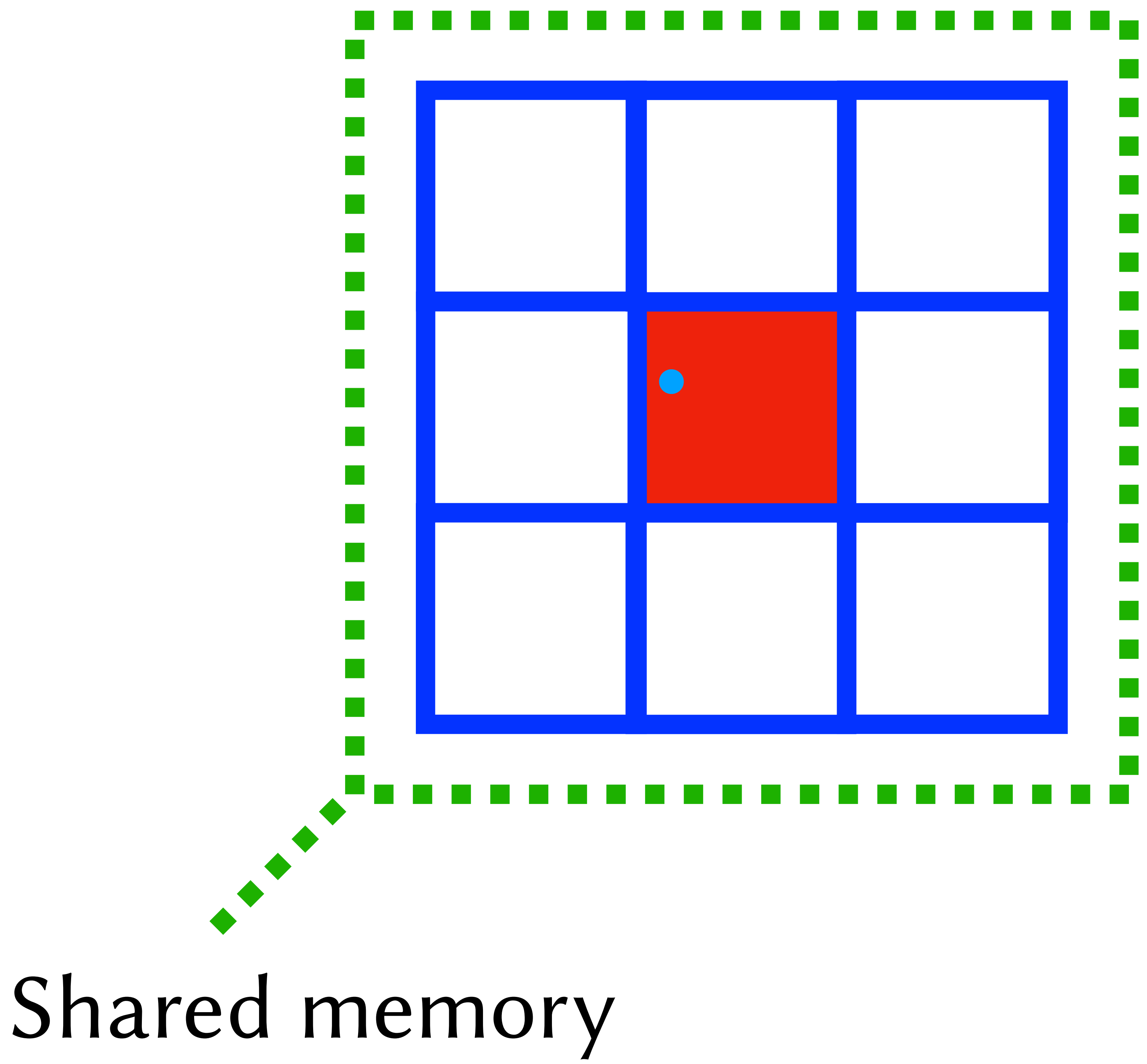
GPU memory

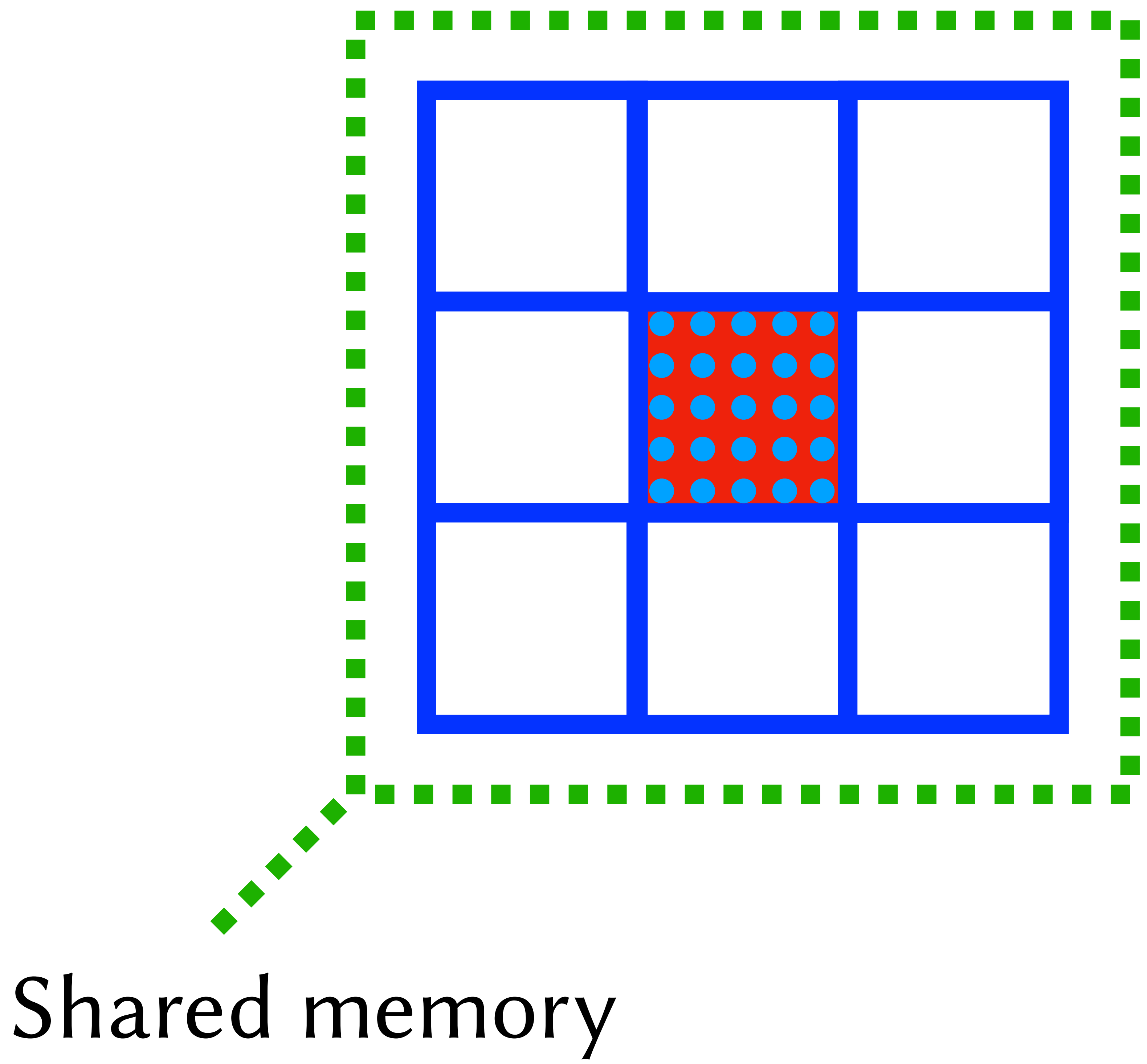


Sparsity - GSPGrid

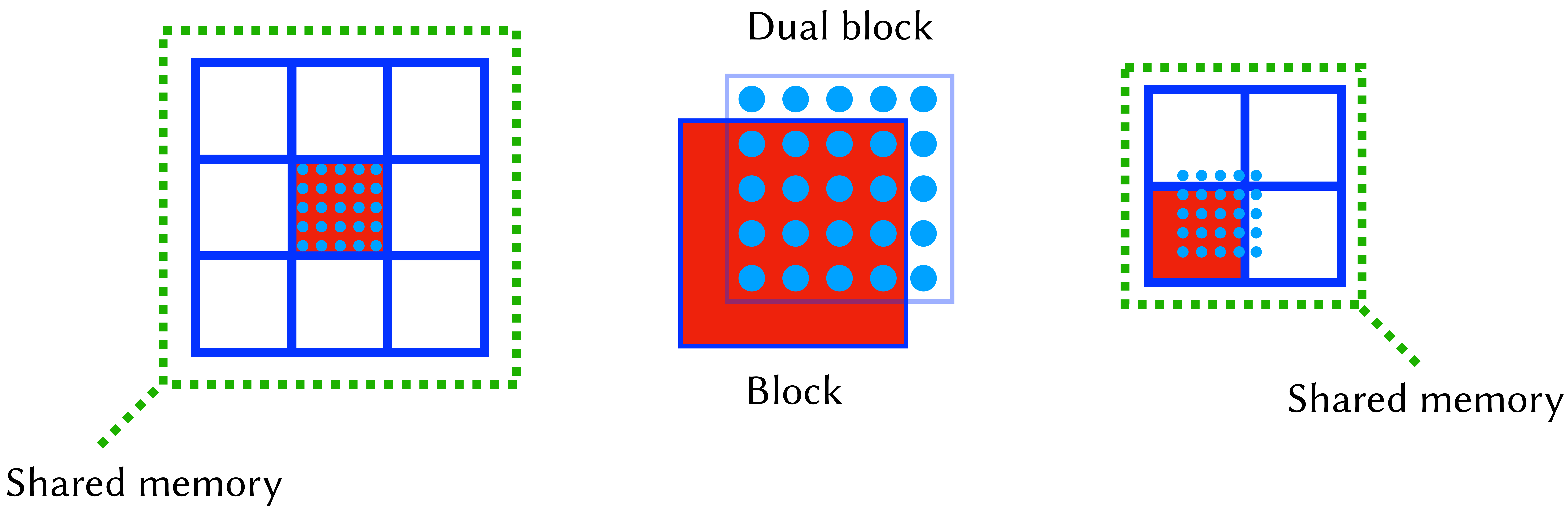


Sparsity - GSPGrid

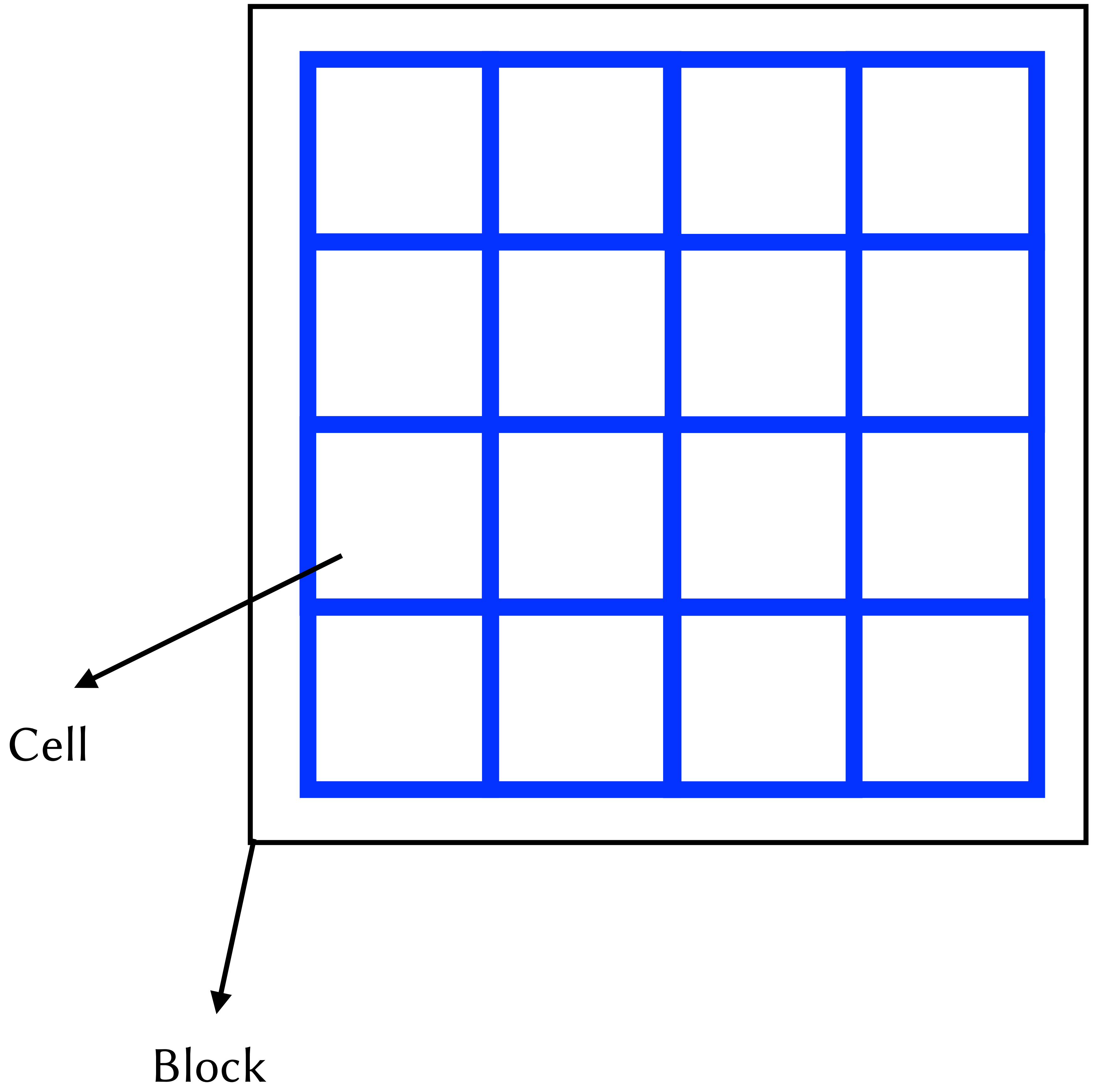




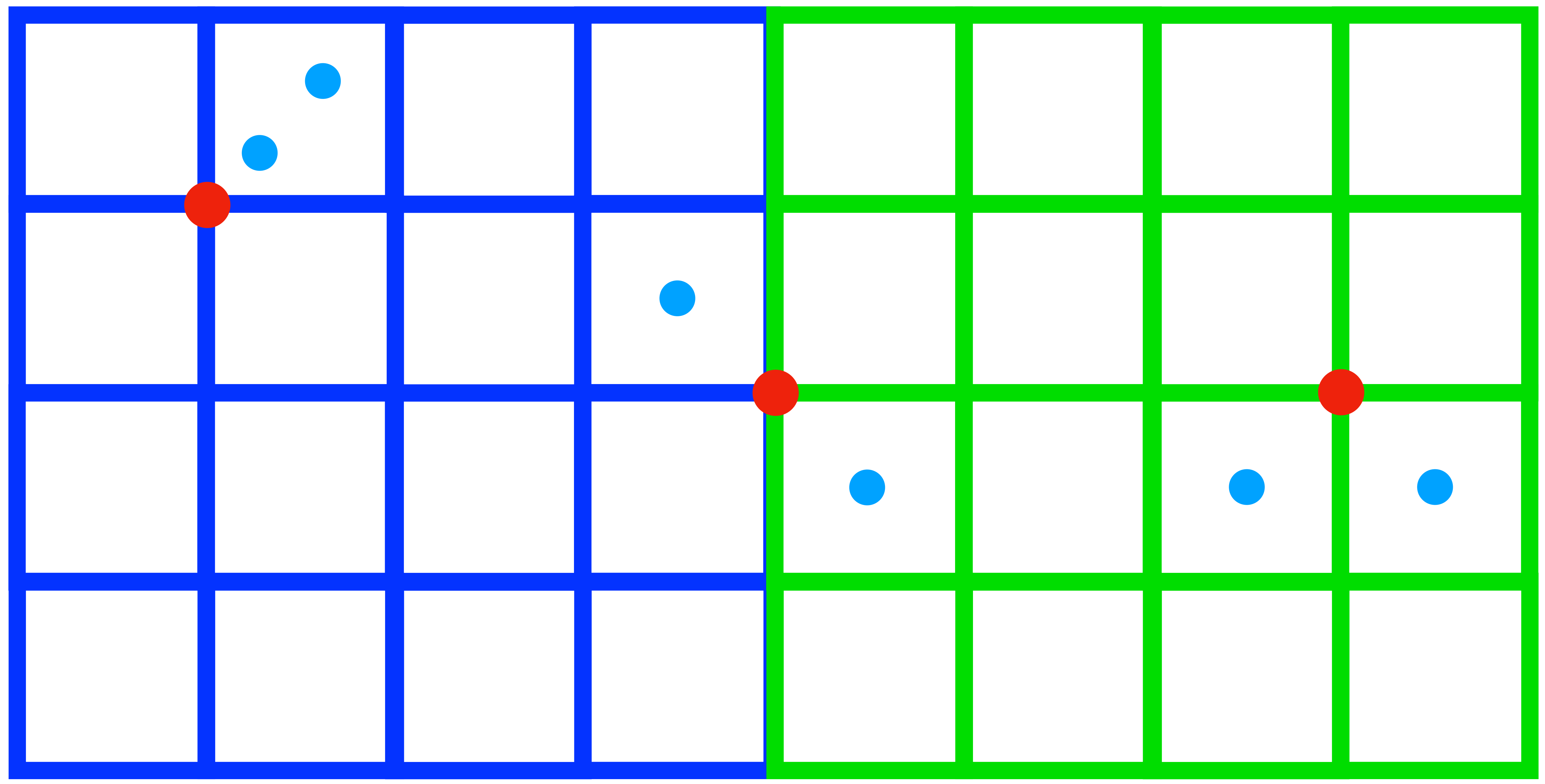
Sparsity - GSPGrid



Write hazards

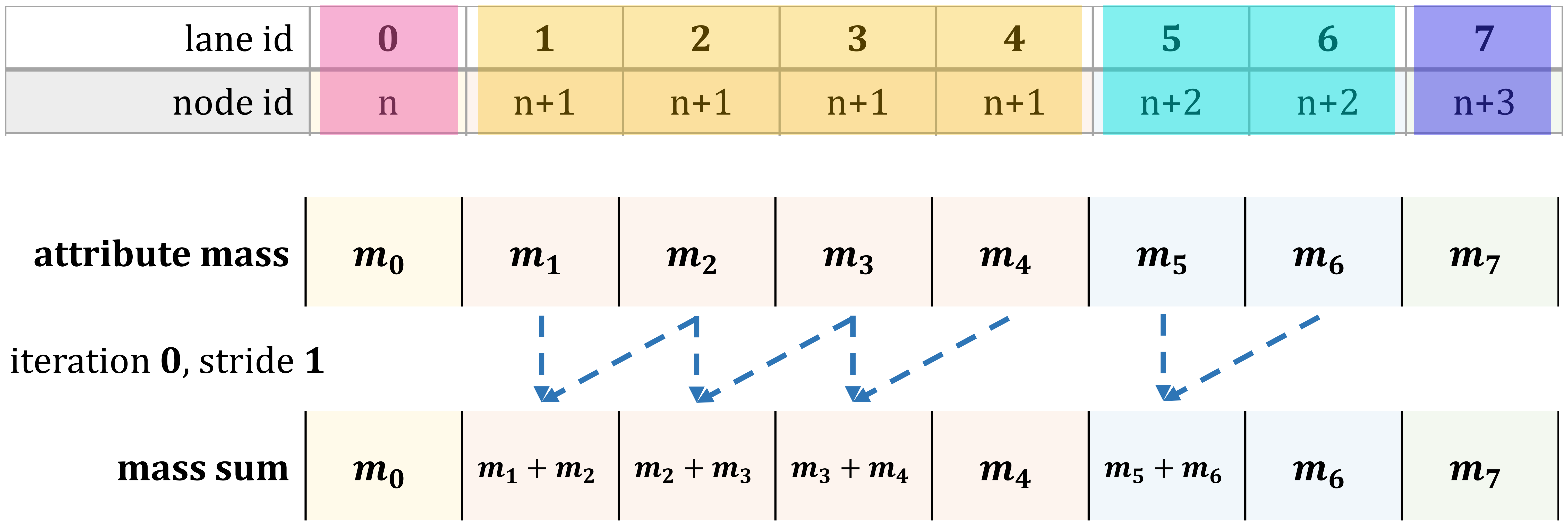


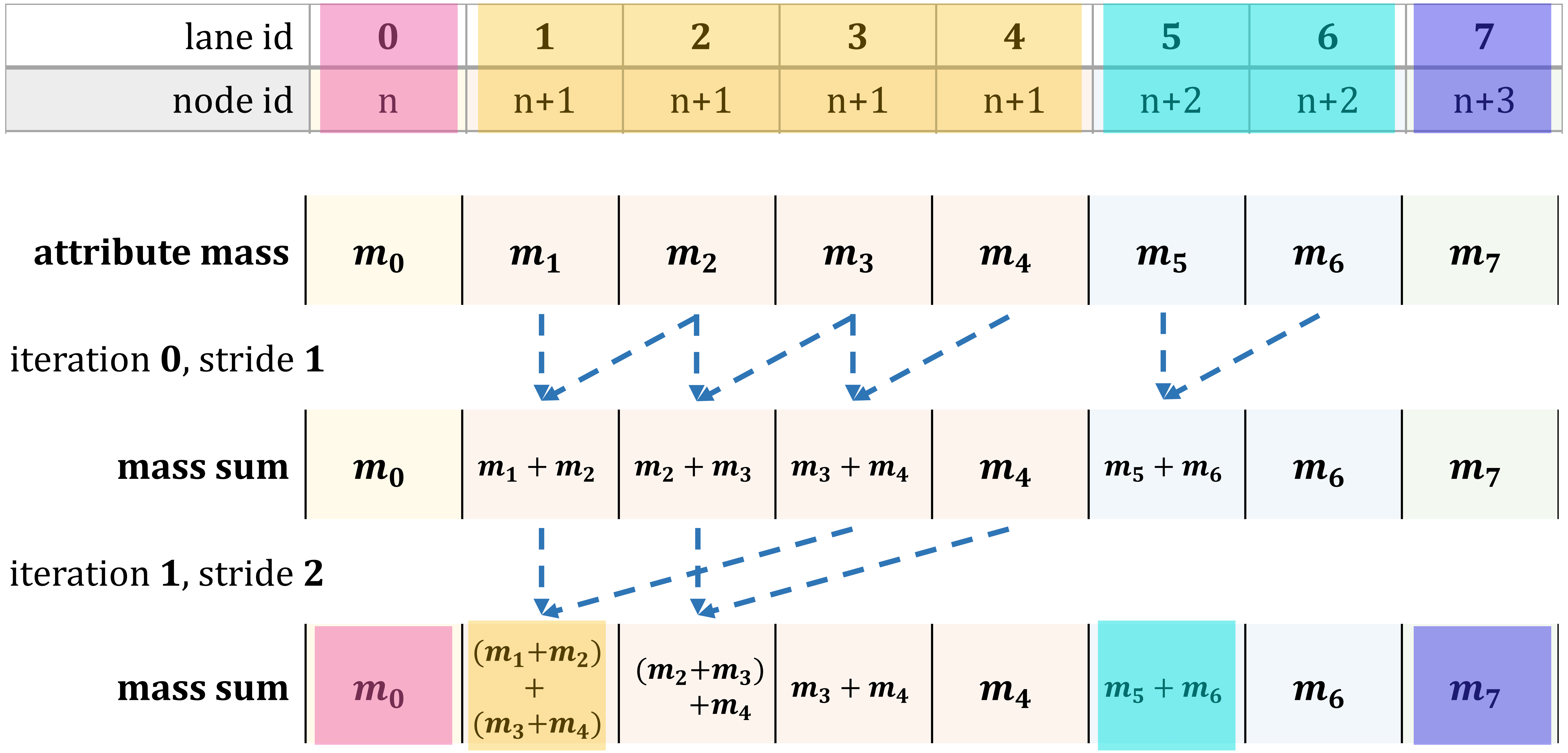
Write hazards

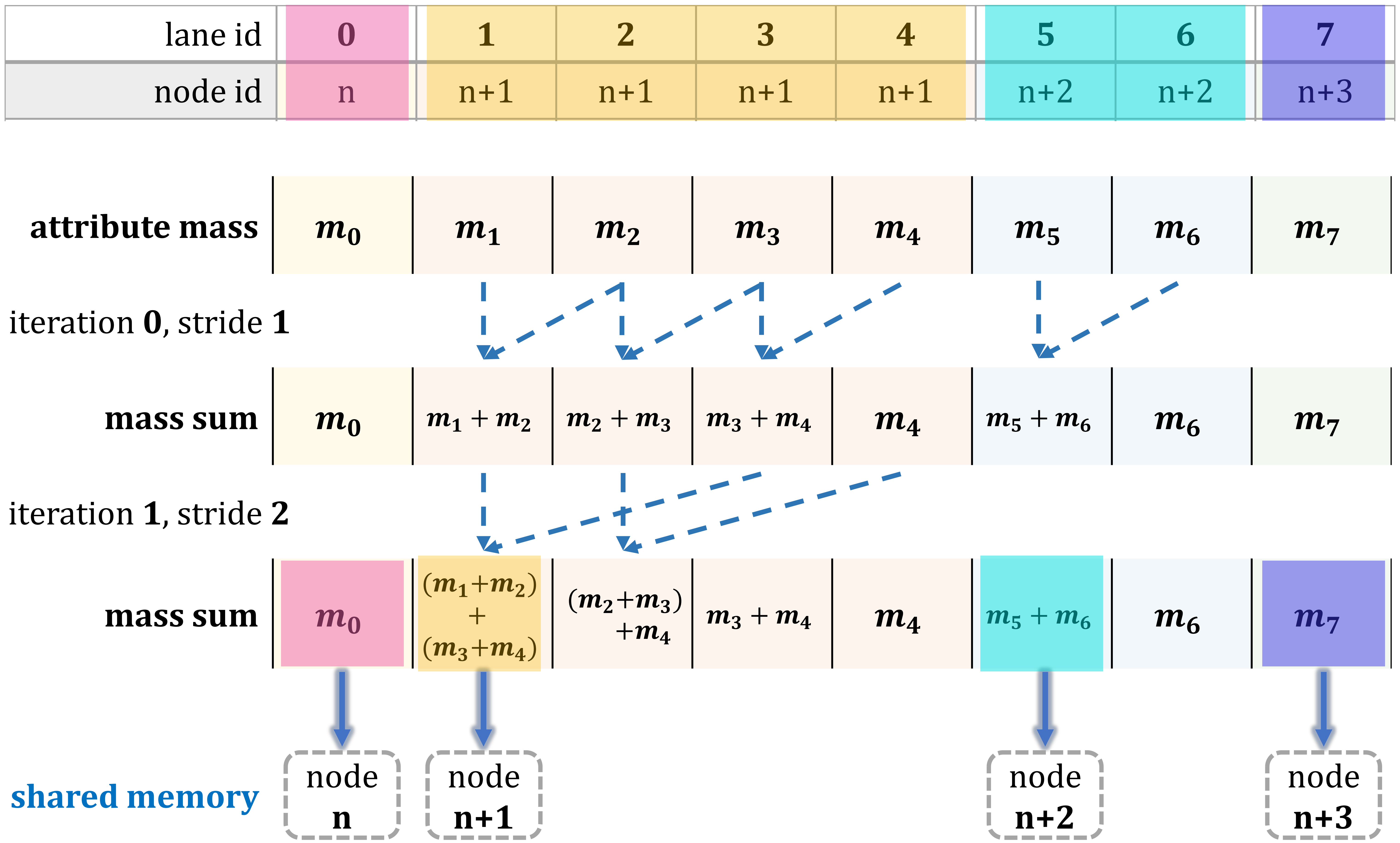


CUDA thread - ●

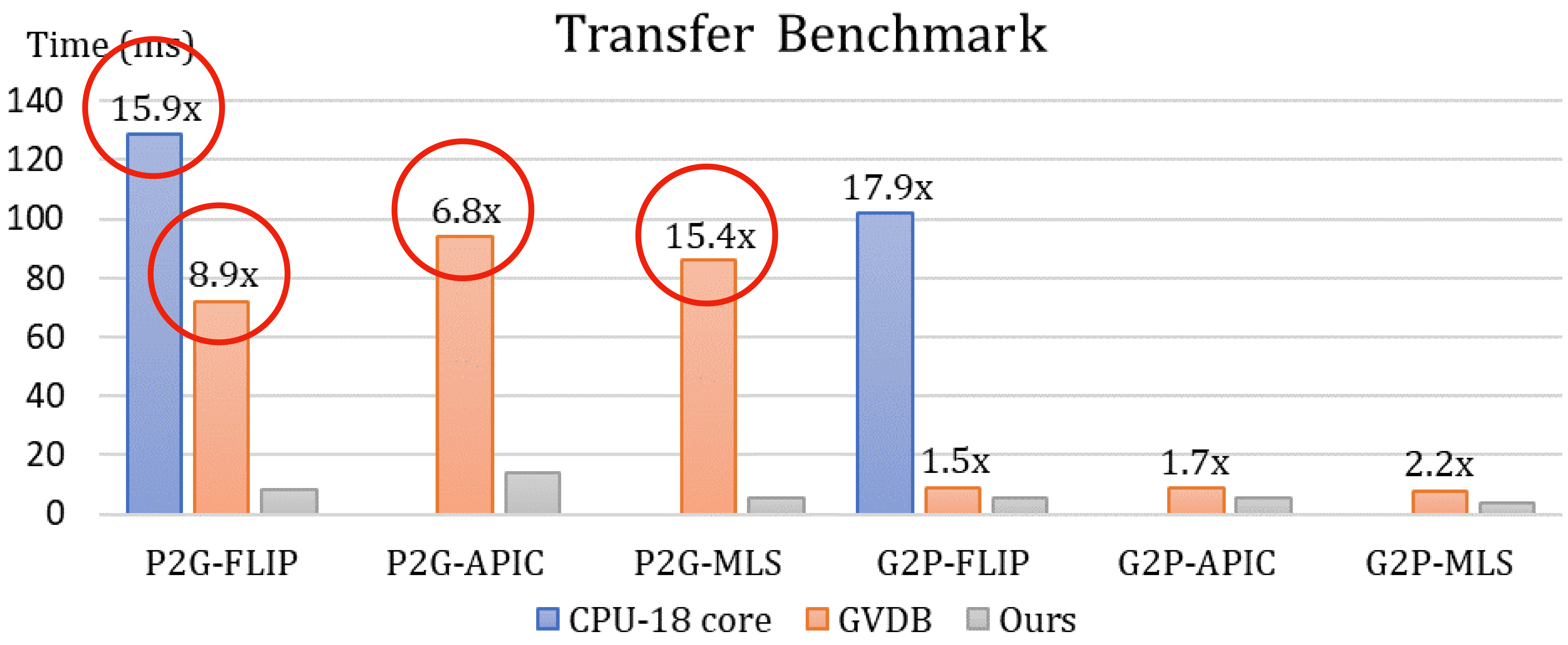
lane id	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
node id	n	n+1	n+1	n+1	n+1	n+2	n+2	n+3


lane id
0
1
2
3
4
5
6
7
node id
n
n+1
n+1
n+1
n+1
n+2
n+2
n+3
attribute mass
m_0
m_1
m_2
m_3
m_4
m_5
m_6
m_7
iteration **0**, stride **1**
mass sum
m_0
$m_1 + m_2$
$m_2 + m_3$
$m_3 + m_4$
m_4
$m_5 + m_6$
m_6
m_7


lane id
0
1
2
3
4
5
6
7
node id
n
n+1
n+1
n+1
n+1
n+2
n+2
n+3
attribute mass
m_0 m_1 m_2 m_3 m_4 m_5 m_6 m_7
iteration 0, stride 1
mass sum
m_0 m_1+m_2 m_2+m_3 m_3+m_4 m_4 m_5+m_6 m_6 m_7
iteration 1, stride 2
mass sum
m_0 $(m_1+m_2)+(m_3+m_4)$ $(m_2+m_3)+m_4$ m_3+m_4 m_4 m_5+m_6 m_6 m_7


lane id
0
1
2
3
4
5
6
7
node id
n
n+1
n+1
n+1
n+1
n+2
n+2
n+3
attribute mass
m_0 m_1 m_2 m_3 m_4 m_5 m_6 m_7
iteration 0, stride 1
mass sum
m_0 $m_1 + m_2$ $m_2 + m_3$ $m_3 + m_4$ m_4 $m_5 + m_6$ m_6 m_7
iteration 1, stride 2
mass sum
m_0 $(m_1+m_2) + (m_3+m_4)$ $(m_2+m_3) + m_4$ $m_3 + m_4$ m_4 $m_5 + m_6$ m_6 m_7
shared memory
node n
node n+1
node n+2
node n+3

Benchmark


Transfer Benchmark
Time (ms)
140
120
100
80
60
40
20
0
15.9x
8.9x
6.8x
15.4x
17.9x
1.5x
1.7x
2.2x
P2G-FLIP
P2G-APIC
P2G-MLS
G2P-FLIP
G2P-APIC
G2P-MLS
CPU-18 core
GVDB
Ours

Additional contributions

- Accelerated particle sorting
- Avoiding explicit particle reordering
- A new sand model for semi-implicit integration
- A MPM-based heat solver

Conclusion

SPGrid
GSPGrid

Fluid

Goo

Jello

Sand

Mixture

Conclusion

Breadth of simulation ⟷ Parallel efficiency

Price paid: Platform-specific solutions

Optimizations far from automatic

Focus on **visual** appeal

Future work

SPGrid
GSPGrid

Fluid
Goo
Modeling
Jello
Sand
Rendering
Mixture
3D printing

Acknowledgement

- My advisor: Prof. Eftychios Sifakis
- Committee:
  - Prof. Michael Gleicher, Dan Negrut, Jean-Luc Thiffeault and Chenfanfu Jiang
- UW Visual Computing Lab
- External collaborators
- National Science Foundation for funding