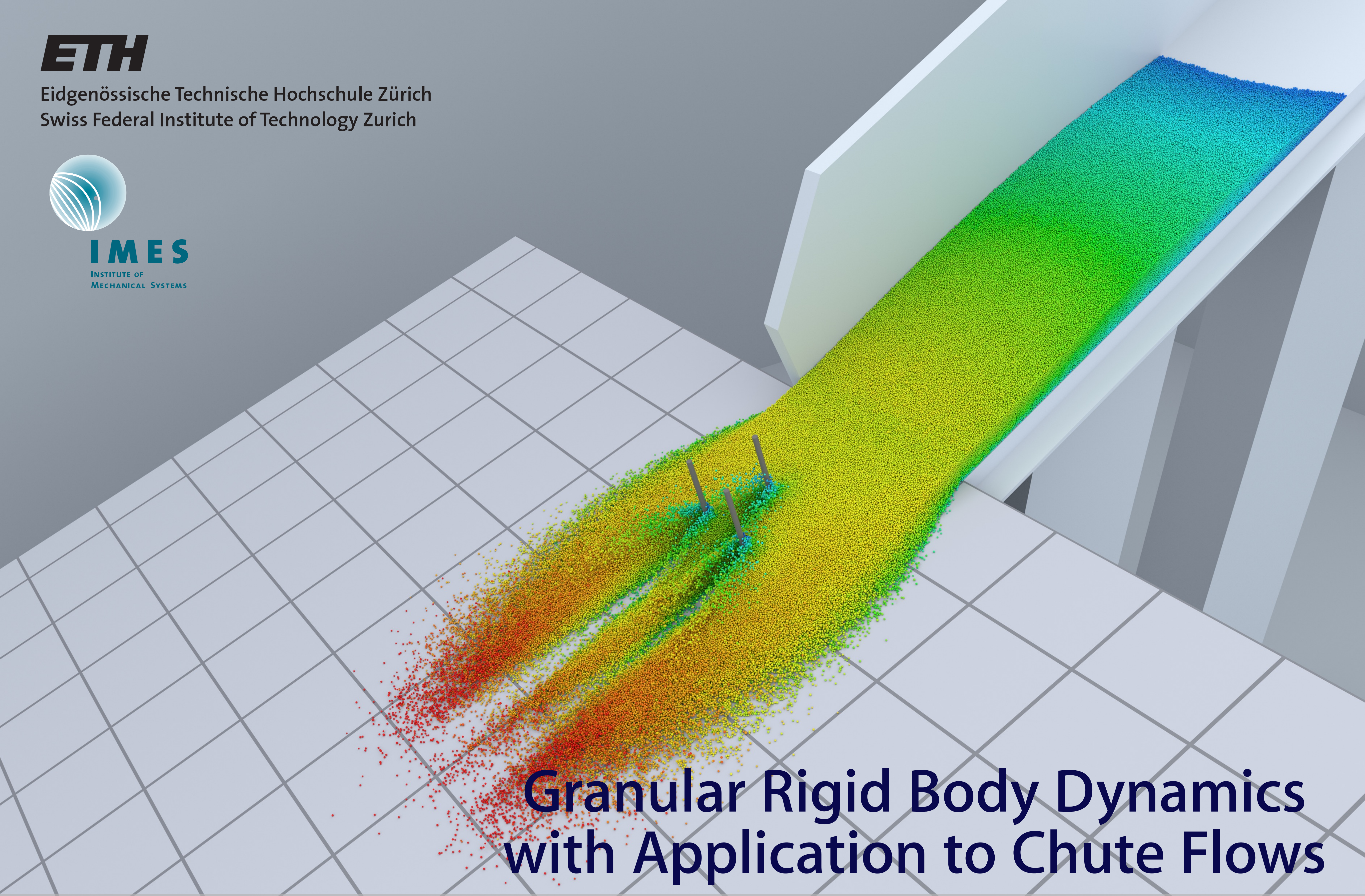
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

IMES
Institute of
Mechanical Systems



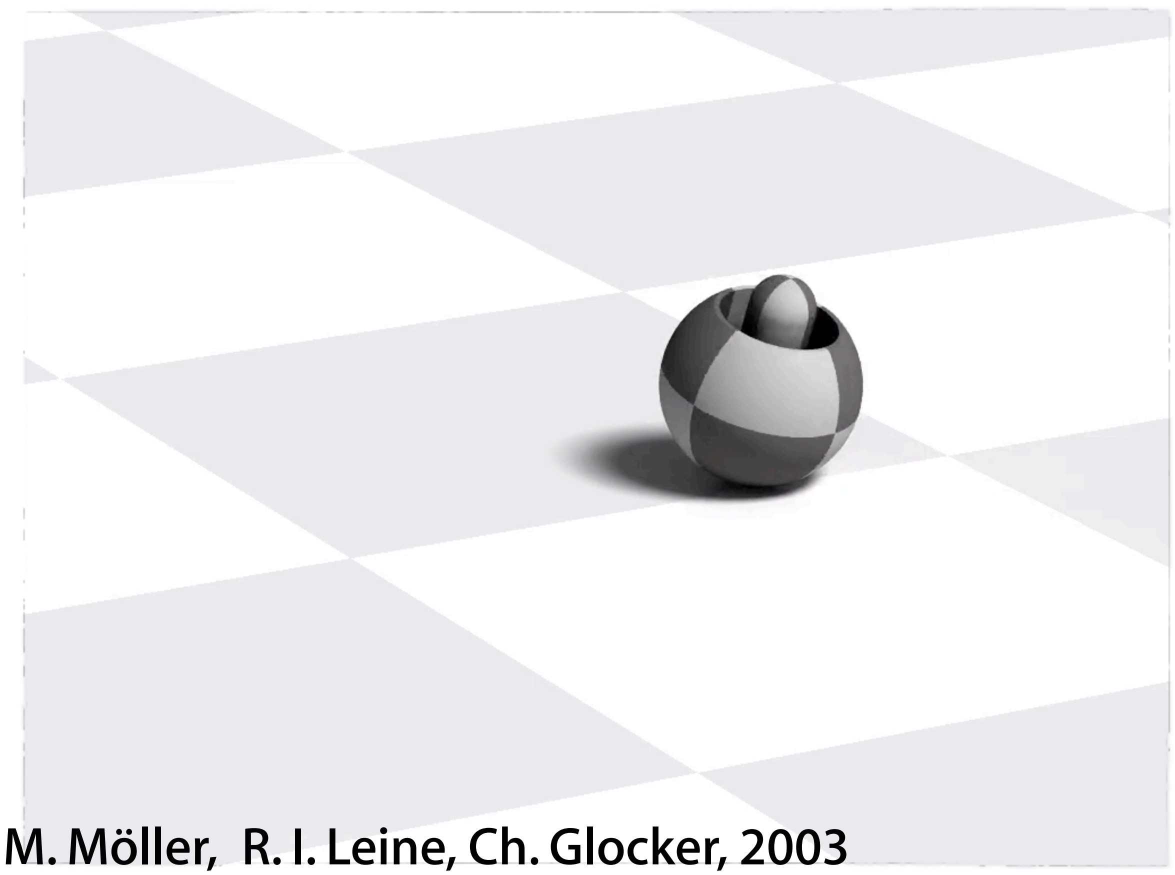
Granular Rigid Body Dynamics with Application to Chute Flows

MSc ETH ME Gabriel Nützi

Examiner: Prof. Dr. Ch. Glocker
Co-Examiner: Prof. Dr. J. Dual, Prof. Dr. R. Leine

Mechanics

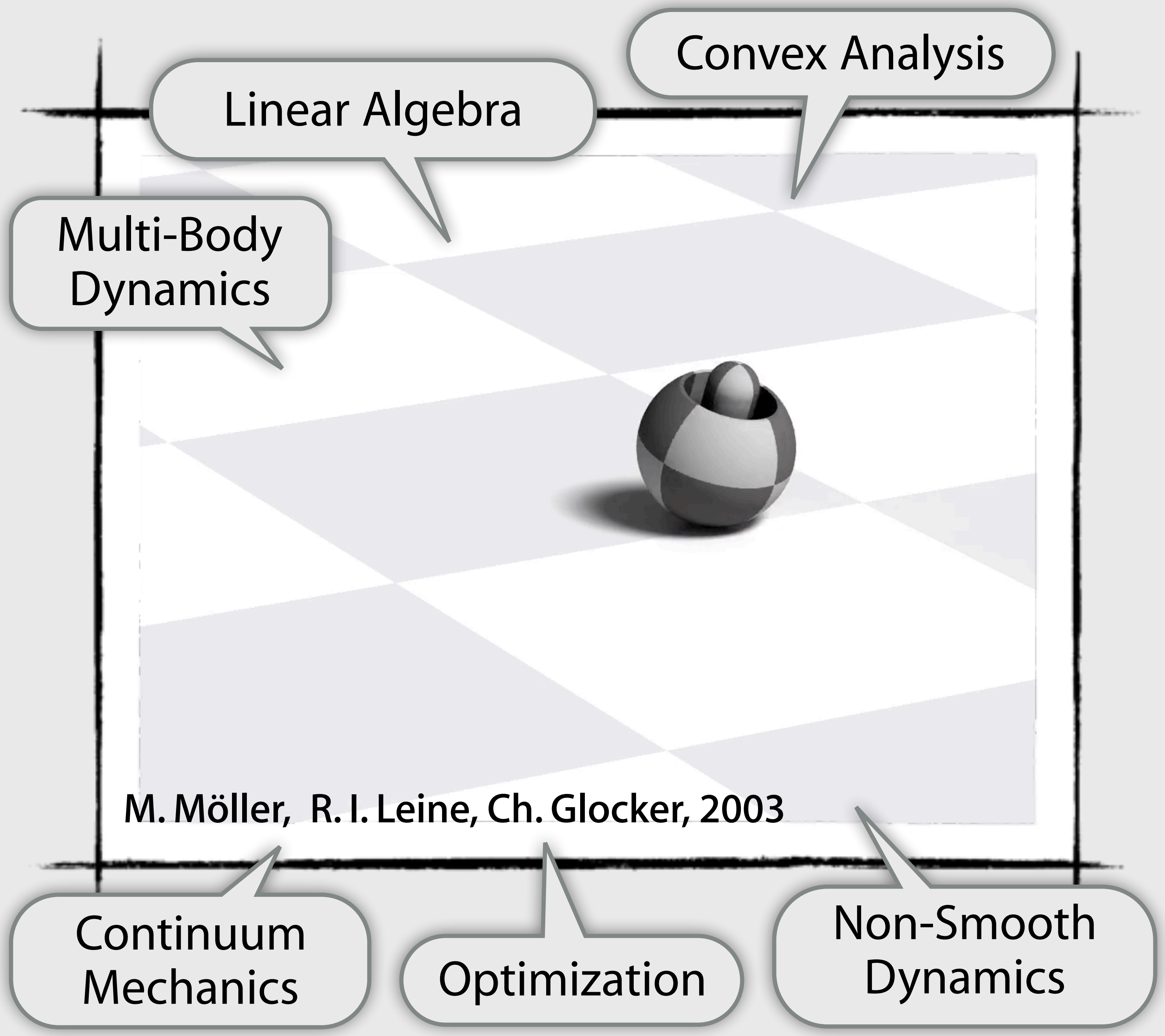
Computer Science



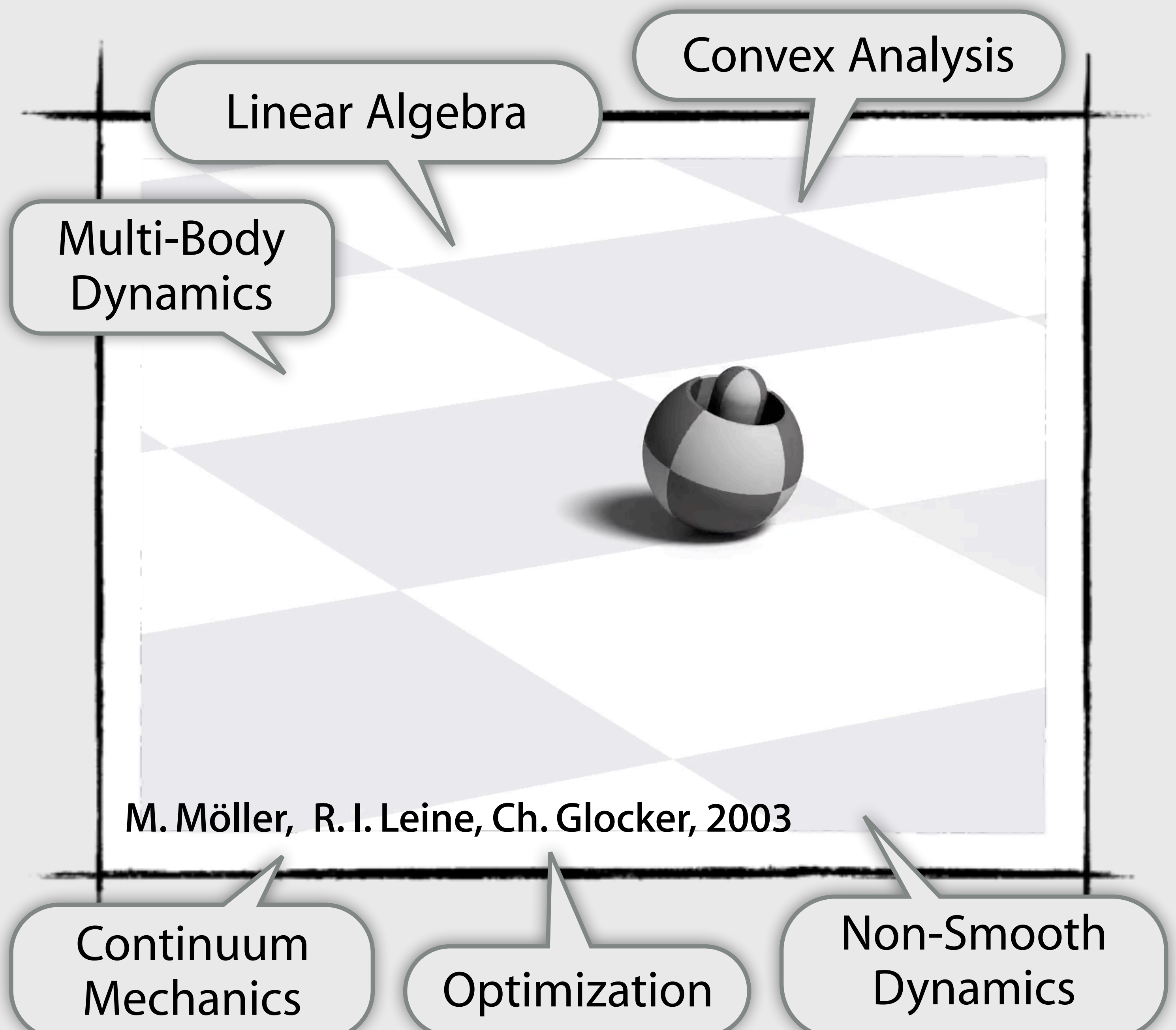
M. Möller, R. I. Leine, Ch. Glocker, 2003

Mechanics

Computer Science



Mechanics



Computer Science

```
#include <unistd.h>
float o=0.075,h=1.5,T,r,O,l,I;int
_,L=80,s=3200;main(){for(;s%L||
(h-=o,T= -2),s;4 -(r=O*O)<(l=I*I)|
++ _==L&&write(1,(--s%L?_<L?--_
%6:6:7)+"World! \n",
1)&&(O=I=l=_=r=0,T+=o /2))O=I*2*O
+h,I=l+T-r;}
```

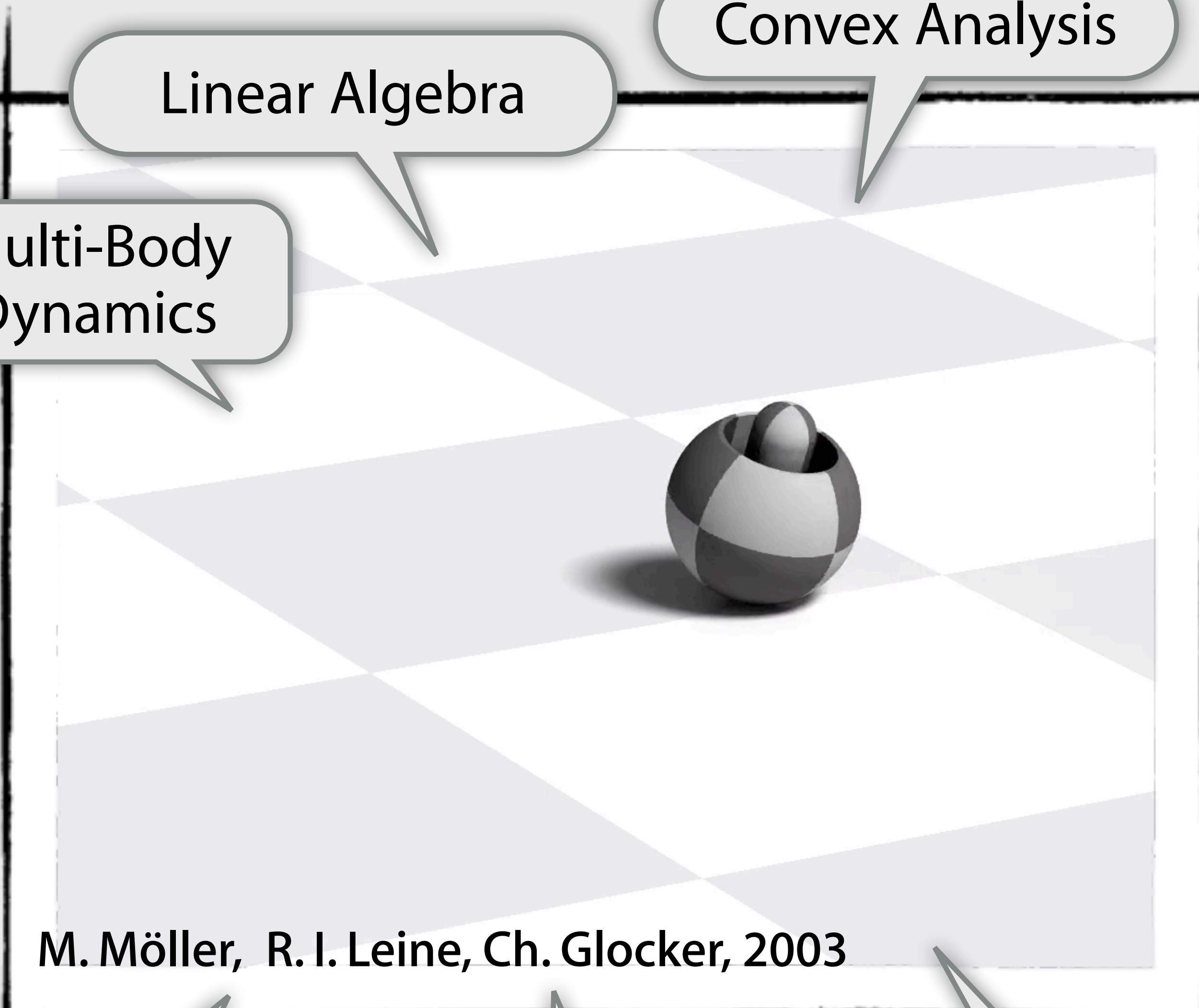
Algorithm (noun.)
Word used by programmers when they don't want to explain what they did.

Mechanics

Linear Algebra

Convex Analysis

Multi-Body Dynamics



M. Möller, R. I. Leine, Ch. Glocker, 2003

Continuum Mechanics

Optimization

Non-Smooth Dynamics

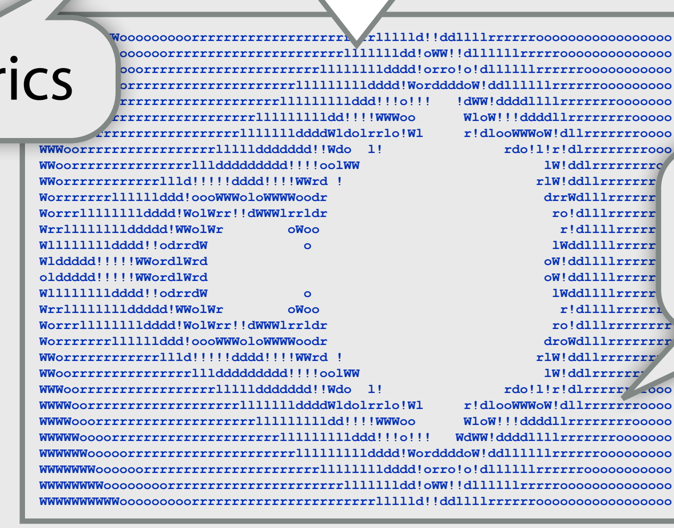
Computer Science

Algorithms

Logic & Data

```
#include <unistd.h>
float o=0.075,h=1.5,T,r,O,l,I;int
_,L=80,s=3200;main(){for(;s%L||
(h-=o,T= -2),s;4 -(r=O*O)<(l=I*I)|
++ _==L&&write(1,(--s%L?_<L?--_
%6:6:7)+"World! \n",
1)&&(O=I=l=_=r=0,T+=o /2))O=I*2*O
+h,I=l+T-r;}
```

Numerics



Computer Graphics

Algorithm (noun.)
Word used by programmers when they don't want to explain what they did.

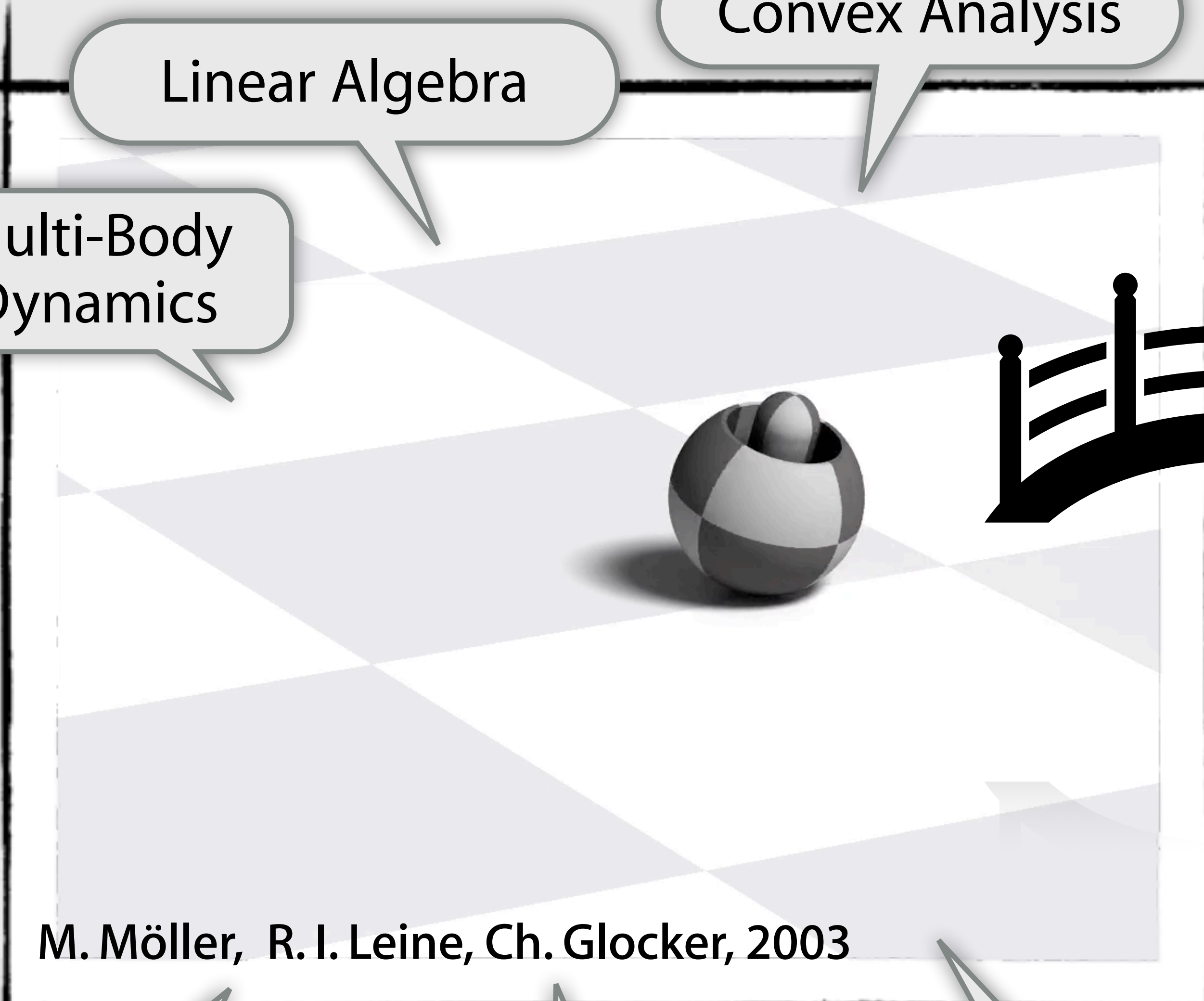
Mechanics

Computer Science

Linear Algebra

Convex Analysis

Multi-Body Dynamics

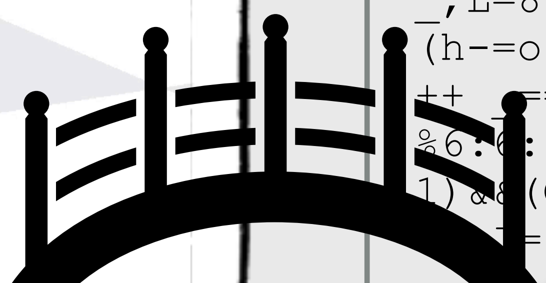


M. Möller, R. I. Leine, Ch. Glocker, 2003

Continuum Mechanics

Optimization

Non-Smooth Dynamics



Algorithms

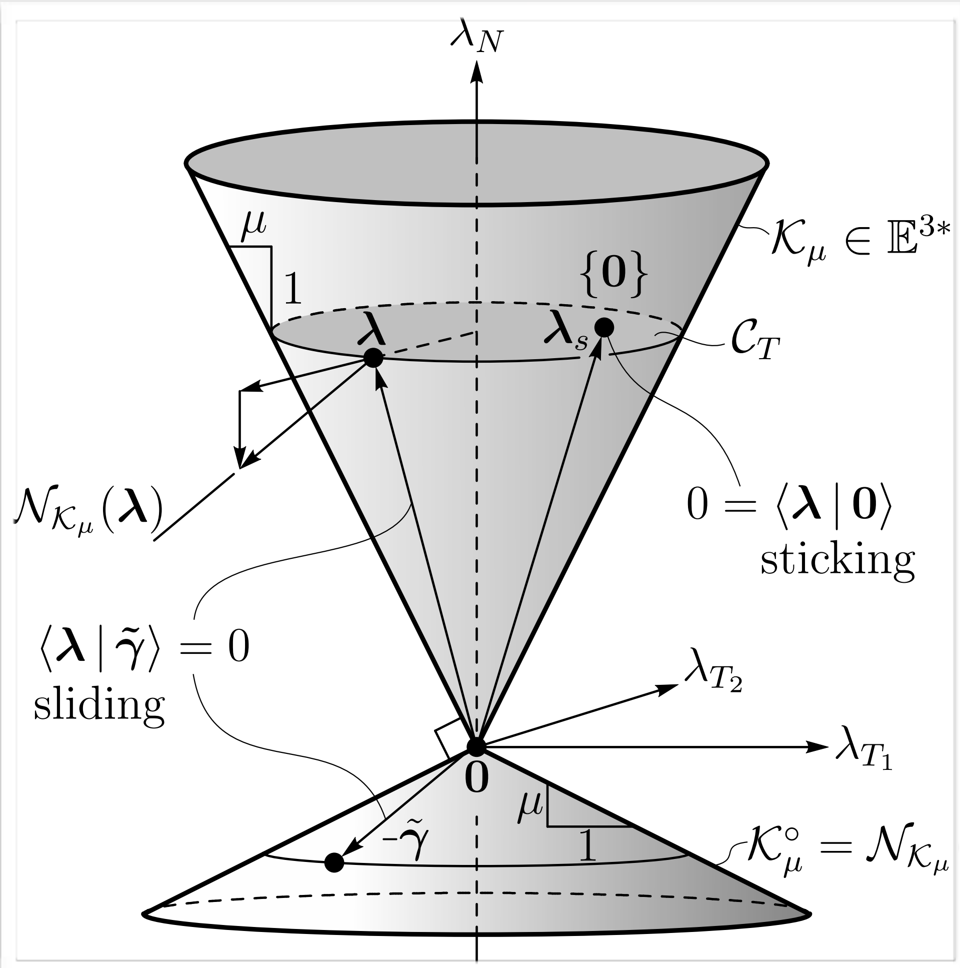
Logic & Data

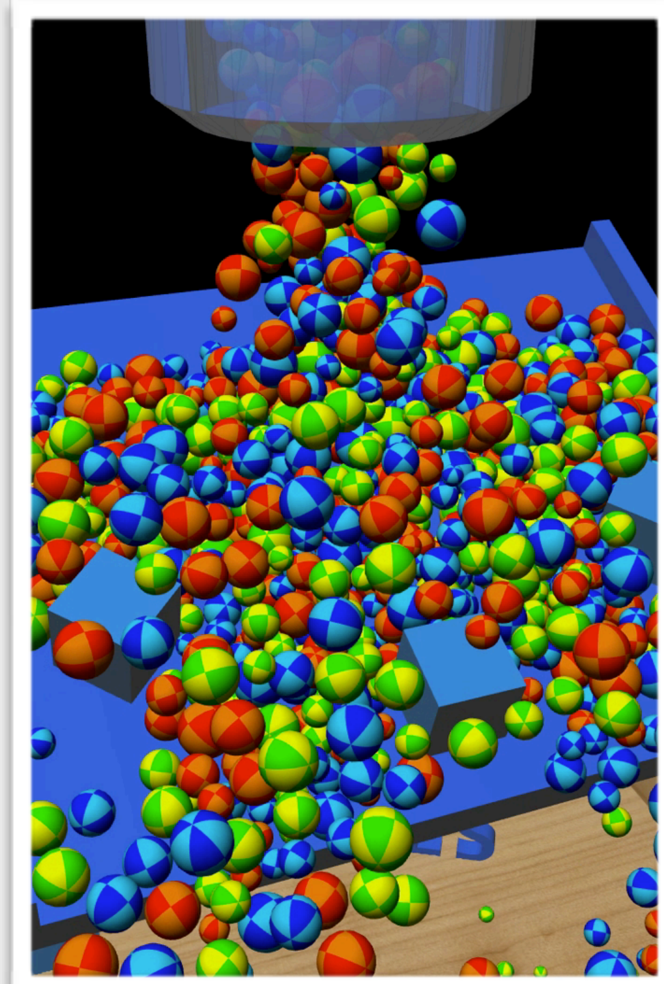
```
#include <unistd.h>
float o=0.075,h=1.5,T,r,O,l,I;int
_,L=80,s=3200;main(){for(;s%L||
(h-=o,T= -2),s;4 -(r=O*O)<(l=I*I)|
++ _==L&&write(1,(--s%L?_<L?--_
%6:6:7)+"World! \n",
1)&&(O=I=l=_=r=0,T+=o /2))O=I*2*O
+h,I=l+T-r;}
```

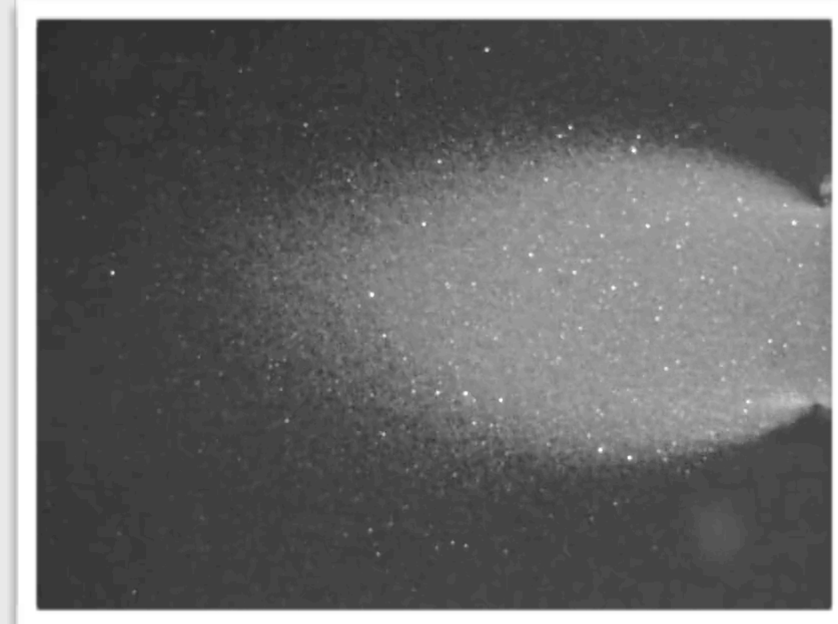
Numerics

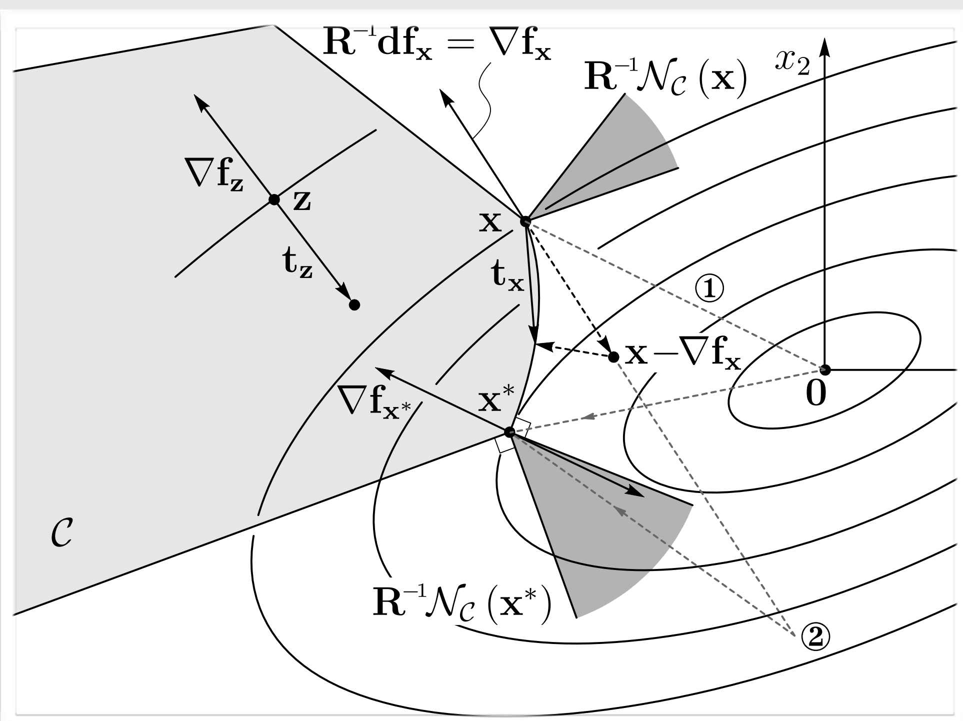
Computer Graphics

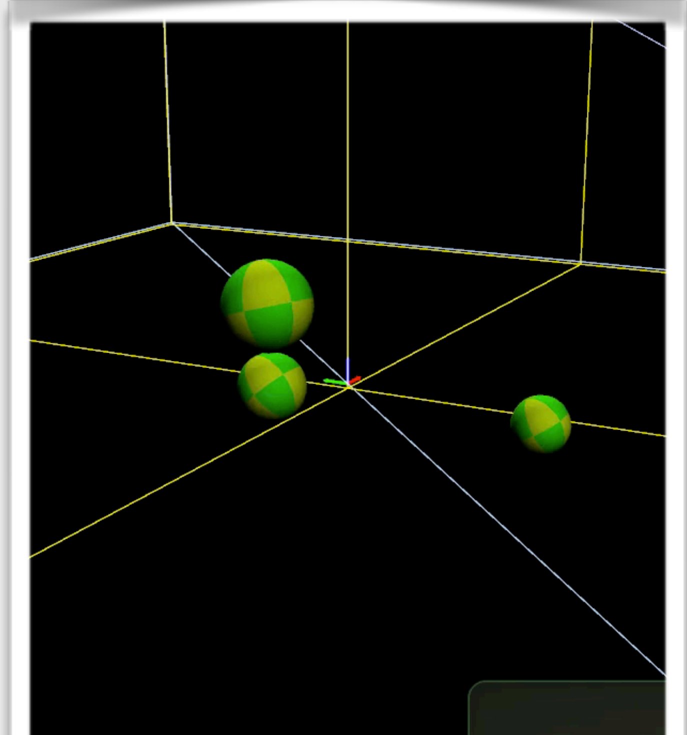
Algorithm (noun.)
Word used by programmers when they don't want to explain what they did.

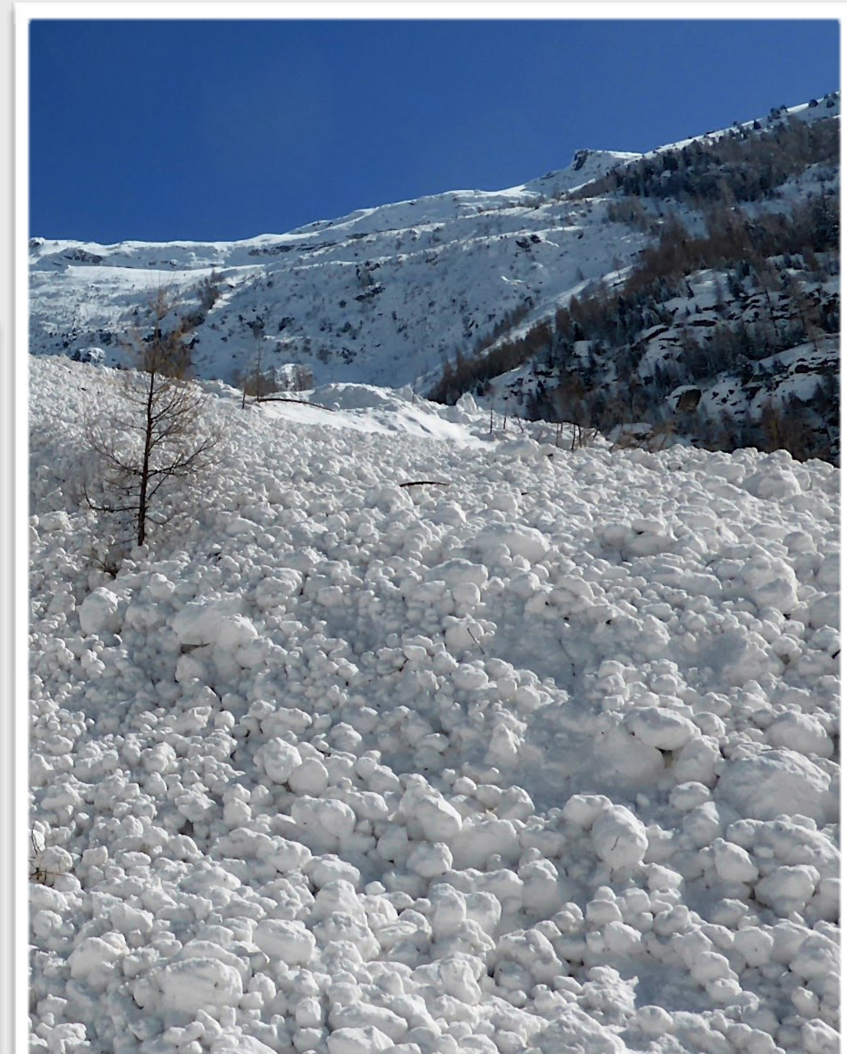

λ_N
μ
1
$\{0\}$
$\mathcal{K}_\mu \in \mathbb{E}^{3*}$
$\boldsymbol{\lambda}$
$\boldsymbol{\lambda}_s$
$\mathcal{C}_T$
$\mathcal{N}_{\mathcal{K}_\mu}(\boldsymbol{\lambda})$
$0 = \langle \boldsymbol{\lambda} \,|\, \mathbf{0} \rangle$
sticking
$\langle \boldsymbol{\lambda} \,|\, \tilde{\boldsymbol{\gamma}} \rangle = 0$
sliding
λ_{T_2}
λ_{T_1}
$\mathbf{0}$
$-\tilde{\boldsymbol{\gamma}}$
$\mathcal{K}^\circ_\mu = \mathcal{N}_{\mathcal{K}_\mu}$

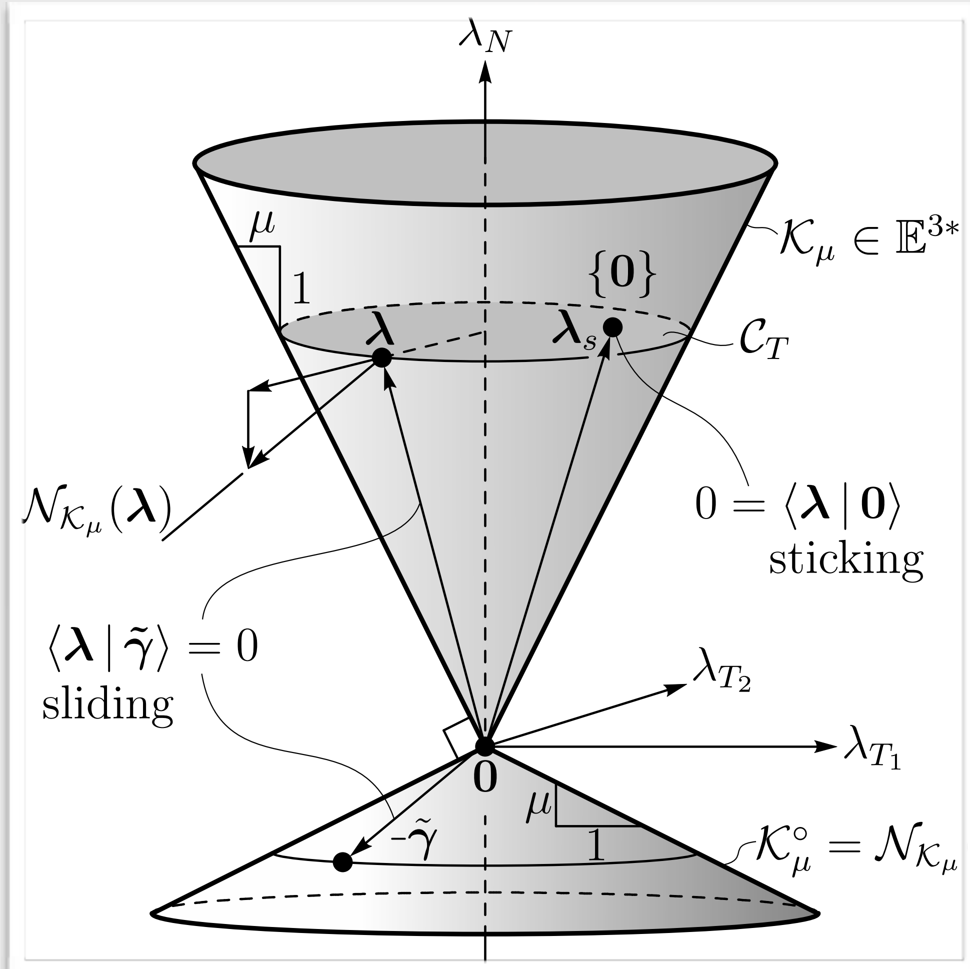


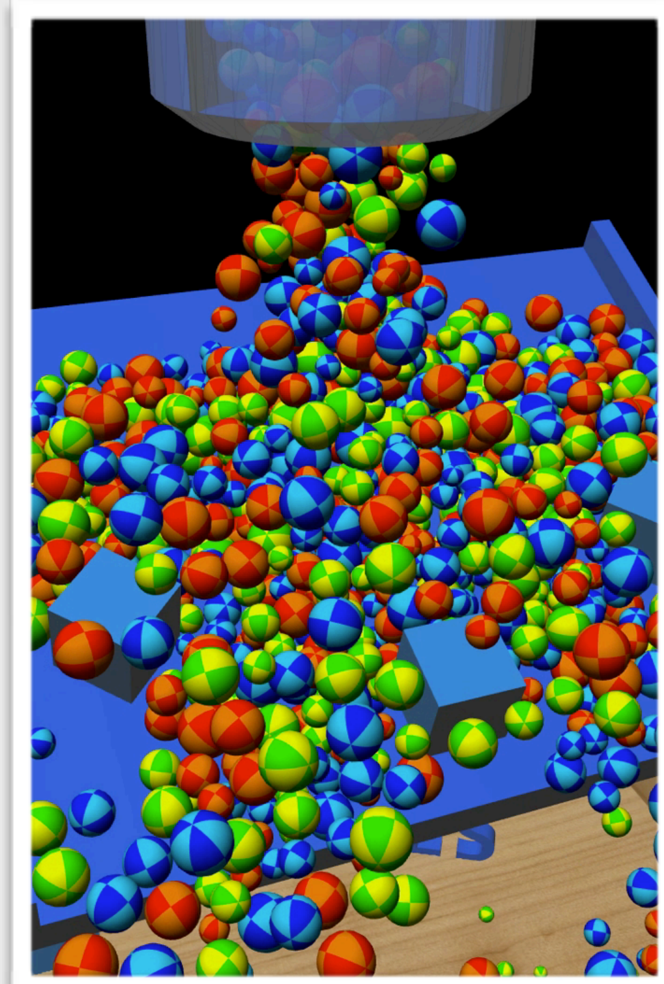


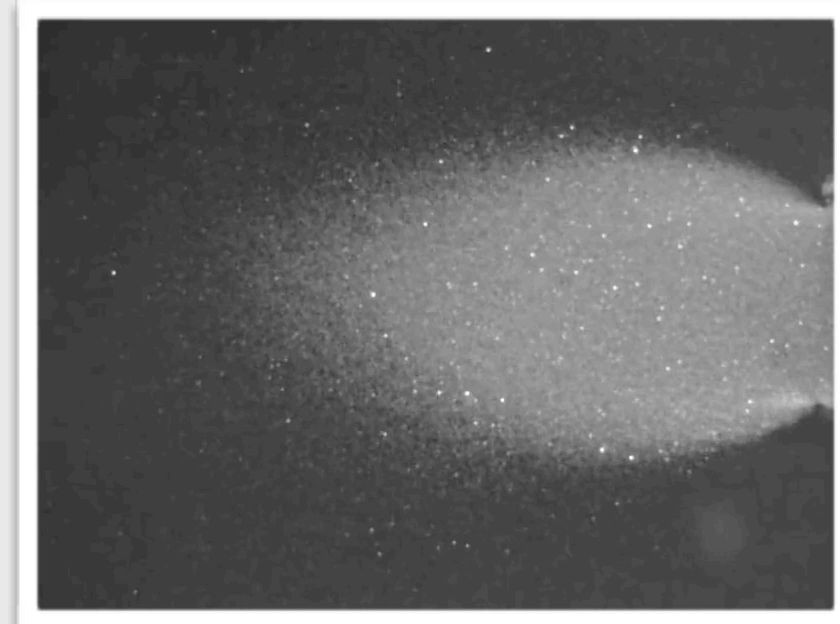

$\mathbf{R}^{-1}\mathbf{df}_\mathbf{x} = \nabla \mathbf{f}_\mathbf{x}$
$\mathbf{R}^{-1}\mathcal{N}_\mathcal{C}(\mathbf{x})$
x_2
$\nabla \mathbf{f}_\mathbf{z}$
$\mathbf{z}$
$\mathbf{t}_\mathbf{z}$
$\mathbf{x}$
$\mathbf{t}_\mathbf{x}$
①
$\mathbf{x} - \nabla \mathbf{f}_\mathbf{x}$
$\nabla \mathbf{f}_{\mathbf{x}^*}$
$\mathbf{x}^*$
$\mathbf{0}$
$\mathcal{C}$
$\mathbf{R}^{-1}\mathcal{N}_\mathcal{C}(\mathbf{x}^*)$
②

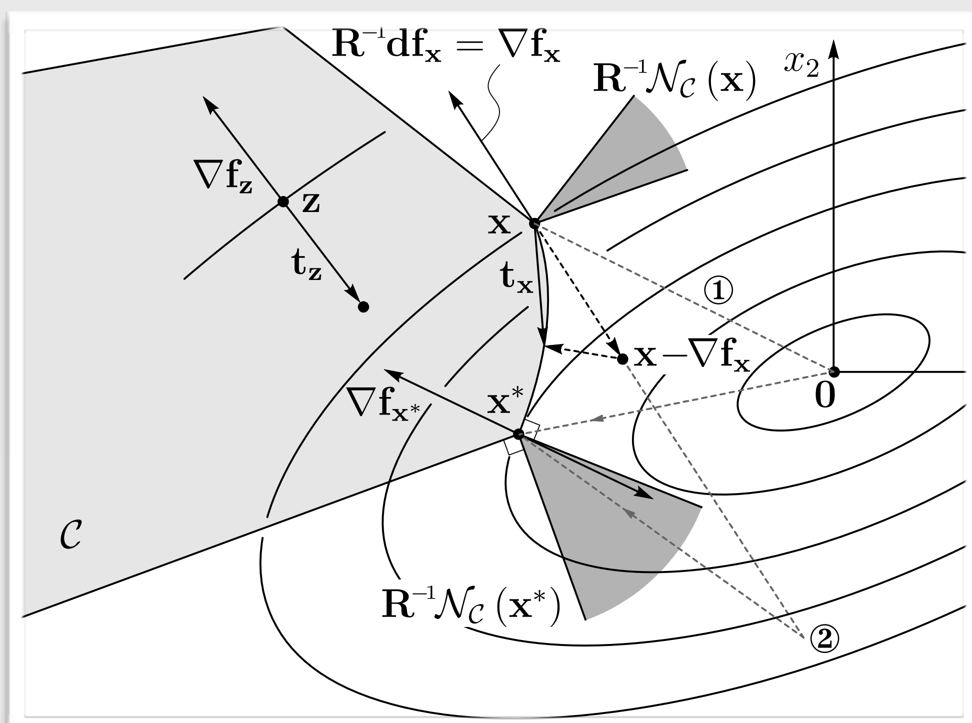


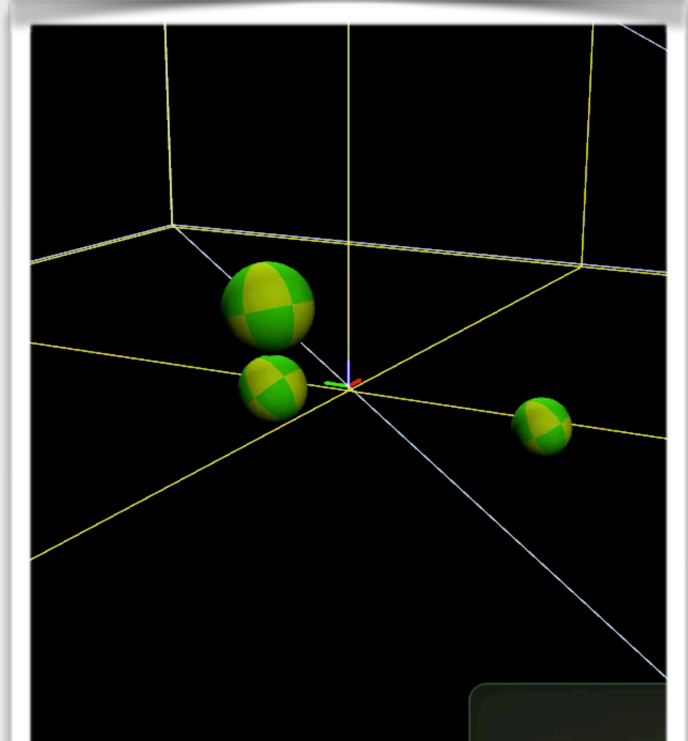


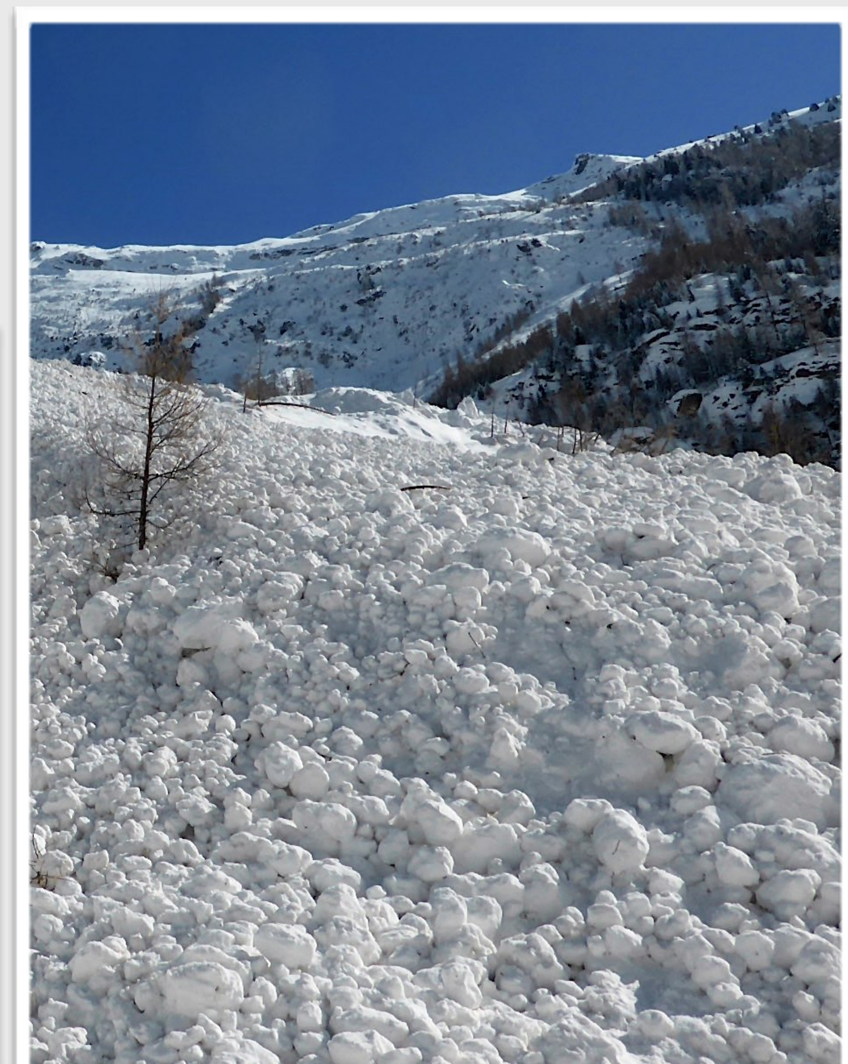

λ_N
μ
1
$\{0\}$
$\mathcal{K}_\mu \in \mathbb{E}^{3*}$
$\boldsymbol{\lambda}$
$\boldsymbol{\lambda}_s$
$\mathcal{C}_T$
$\mathcal{N}_{\mathcal{K}_\mu}(\boldsymbol{\lambda})$
$0 = \langle \boldsymbol{\lambda} \,|\, \mathbf{0} \rangle$
sticking
$\langle \boldsymbol{\lambda} \,|\, \tilde{\gamma} \rangle = 0$
sliding
λ_{T_2}
λ_{T_1}
$\mathbf{0}$
$-\tilde{\gamma}$
μ
1
$\mathcal{K}_\mu^\circ = \mathcal{N}_{\mathcal{K}_\mu}$

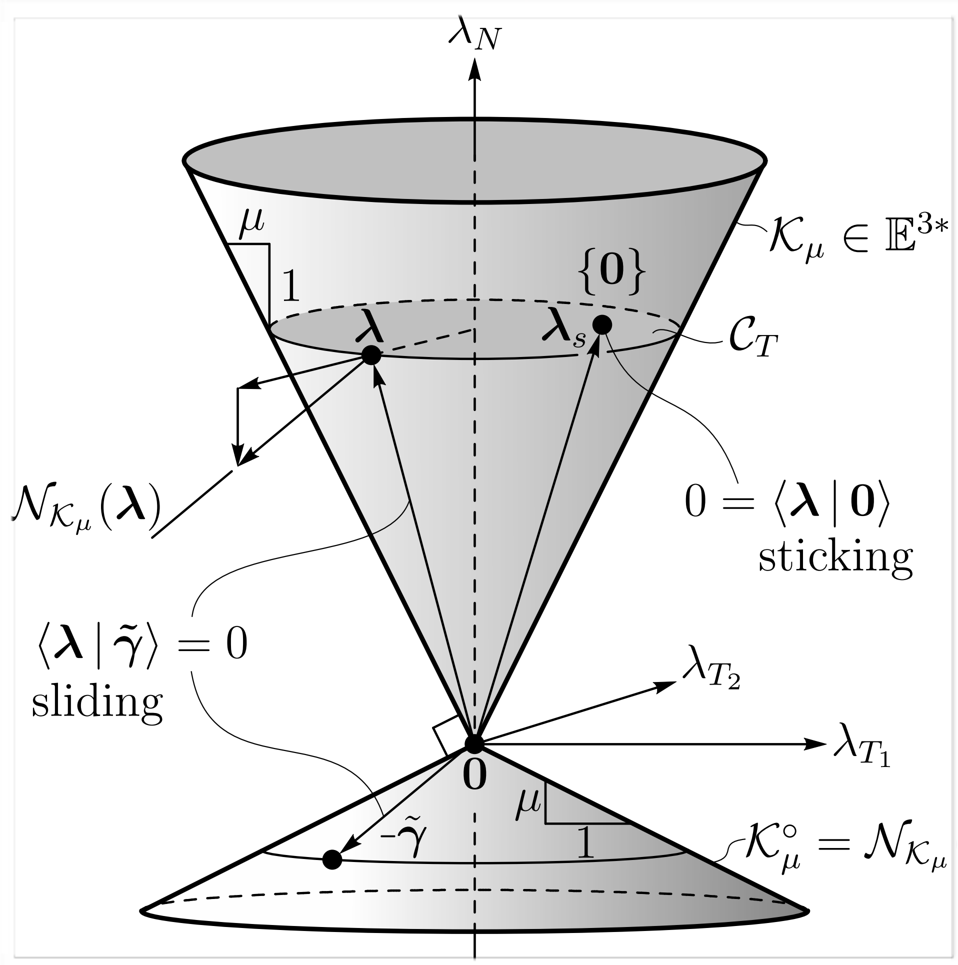


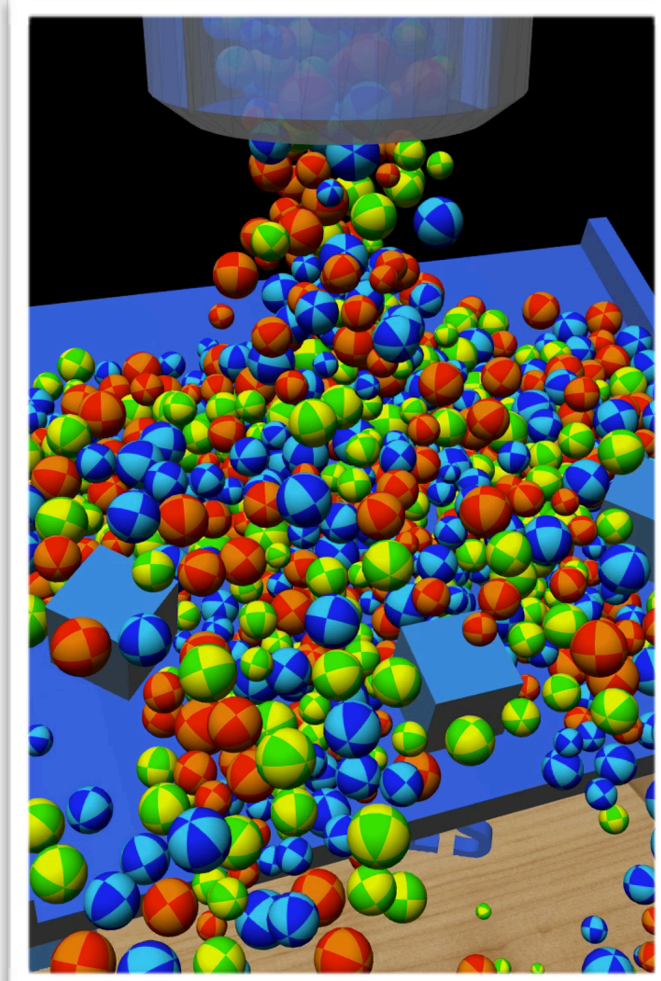


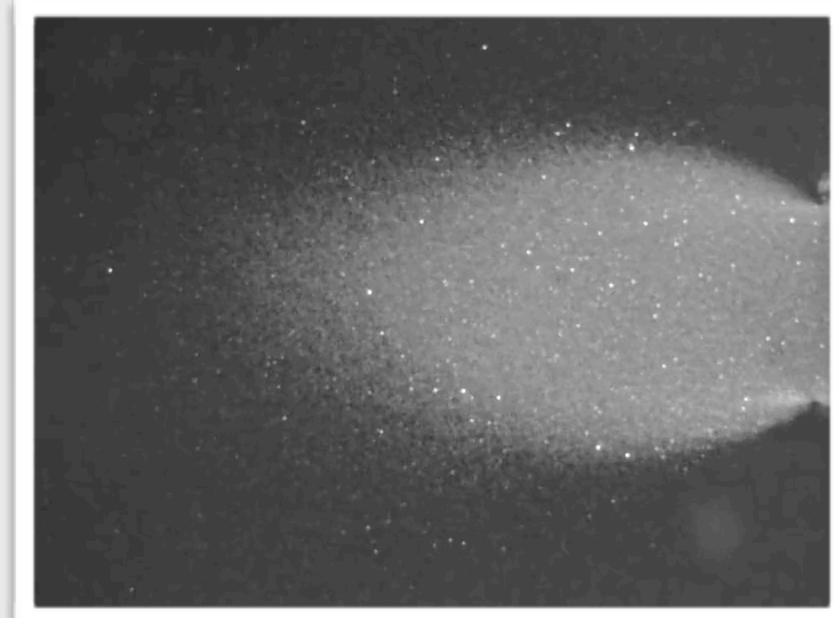

$\mathbf{R}^{-1}\mathbf{df_x} = \nabla \mathbf{f_x}$
$\mathbf{R}^{-1}\mathcal{N}_\mathcal{C}(\mathbf{x})$
x_2
$\nabla \mathbf{f_z}$
$\mathbf{z}$
$\mathbf{t_z}$
$\mathbf{x}$
$\mathbf{t_x}$
①
$\mathbf{x} - \nabla \mathbf{f_x}$
$\nabla \mathbf{f_{x^*}}$
$\mathbf{x}^*$
$\mathbf{0}$
$\mathcal{C}$
$\mathbf{R}^{-1}\mathcal{N}_\mathcal{C}(\mathbf{x}^*)$
②

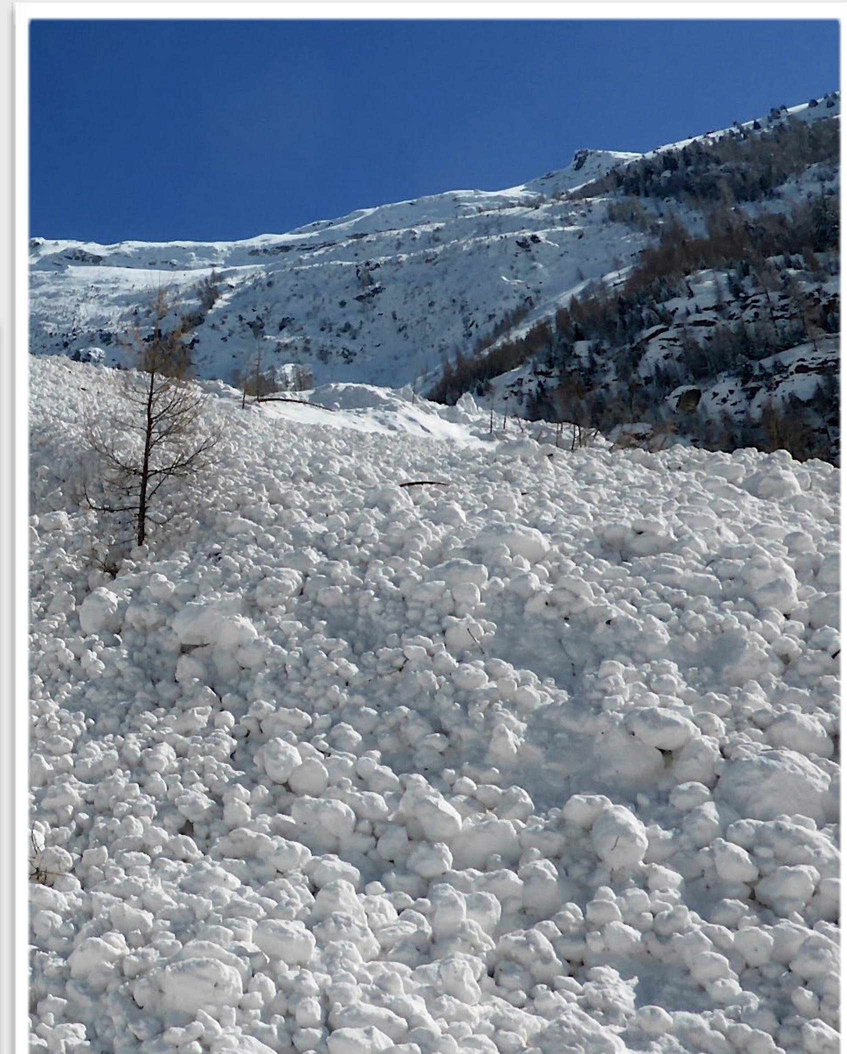


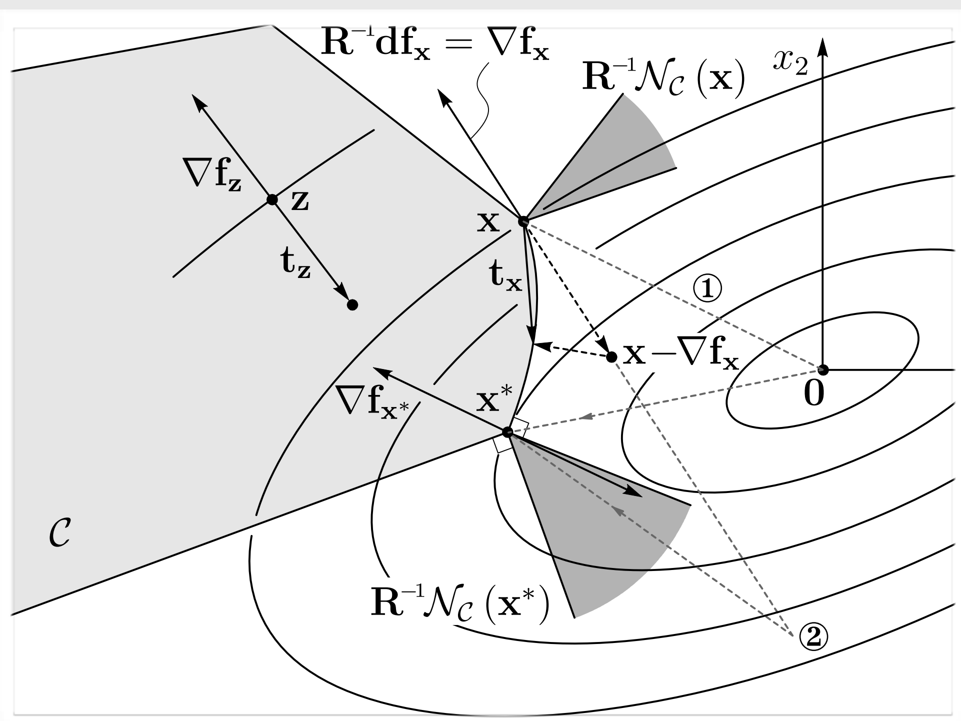


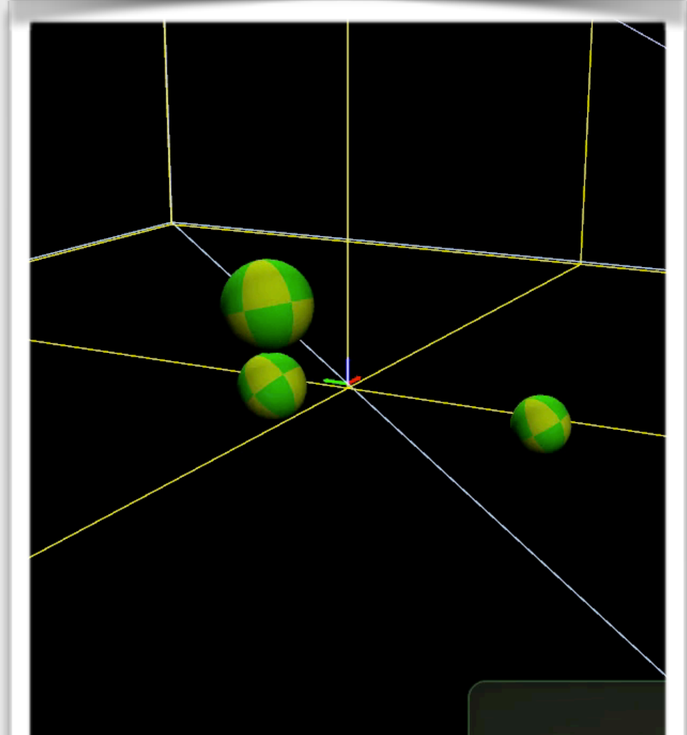

λ_N
μ
1
$\{\mathbf{0}\}$
$\mathcal{K}_\mu \in \mathbb{E}^{3*}$
$\boldsymbol{\lambda}$
$\boldsymbol{\lambda}_s$
$\mathcal{C}_T$
$\mathcal{N}_{\mathcal{K}_\mu}(\boldsymbol{\lambda})$
$0 = \langle \boldsymbol{\lambda} \,|\, \mathbf{0} \rangle$
sticking
$\langle \boldsymbol{\lambda} \,|\, \tilde{\gamma} \rangle = 0$
sliding
λ_{T_2}
λ_{T_1}
$\mathbf{0}$
$-\tilde{\gamma}$
$\mathcal{K}^\circ_\mu = \mathcal{N}_{\mathcal{K}_\mu}$



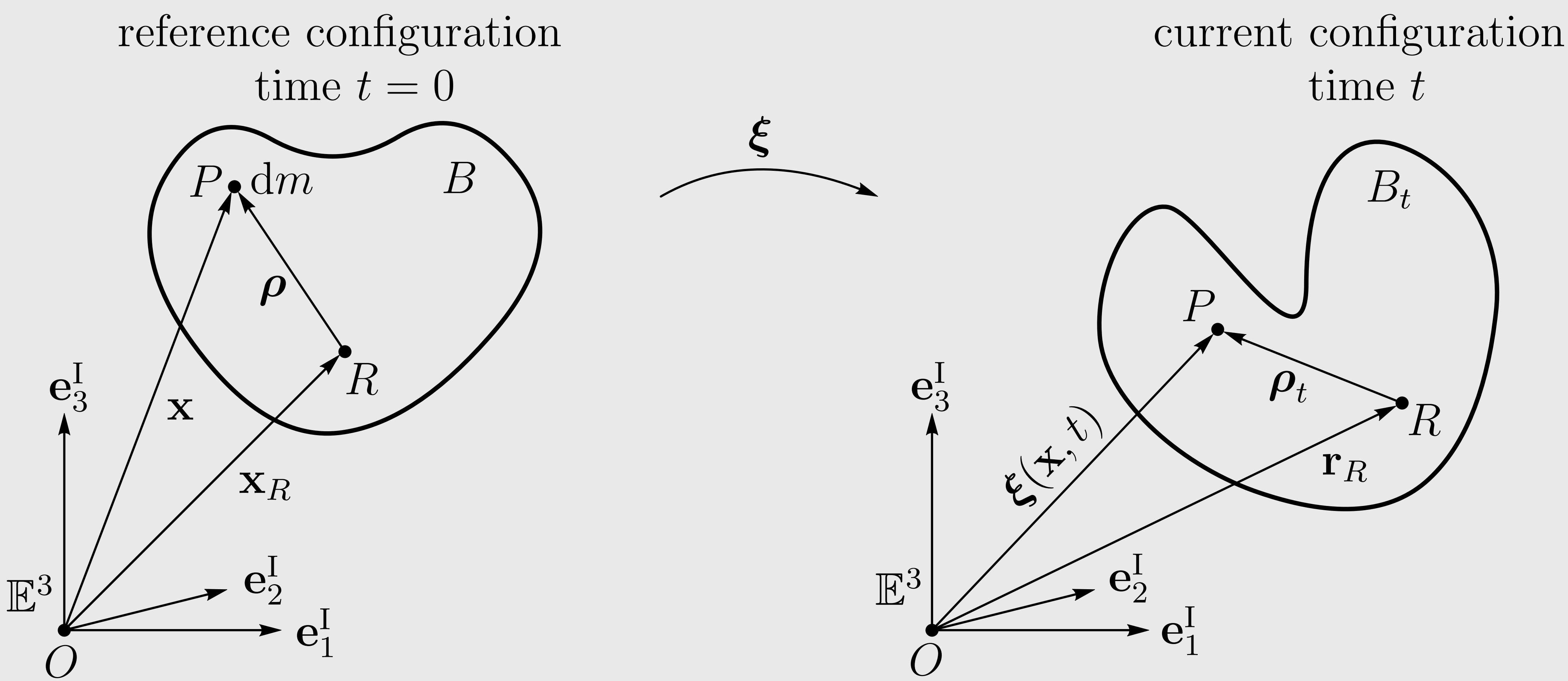




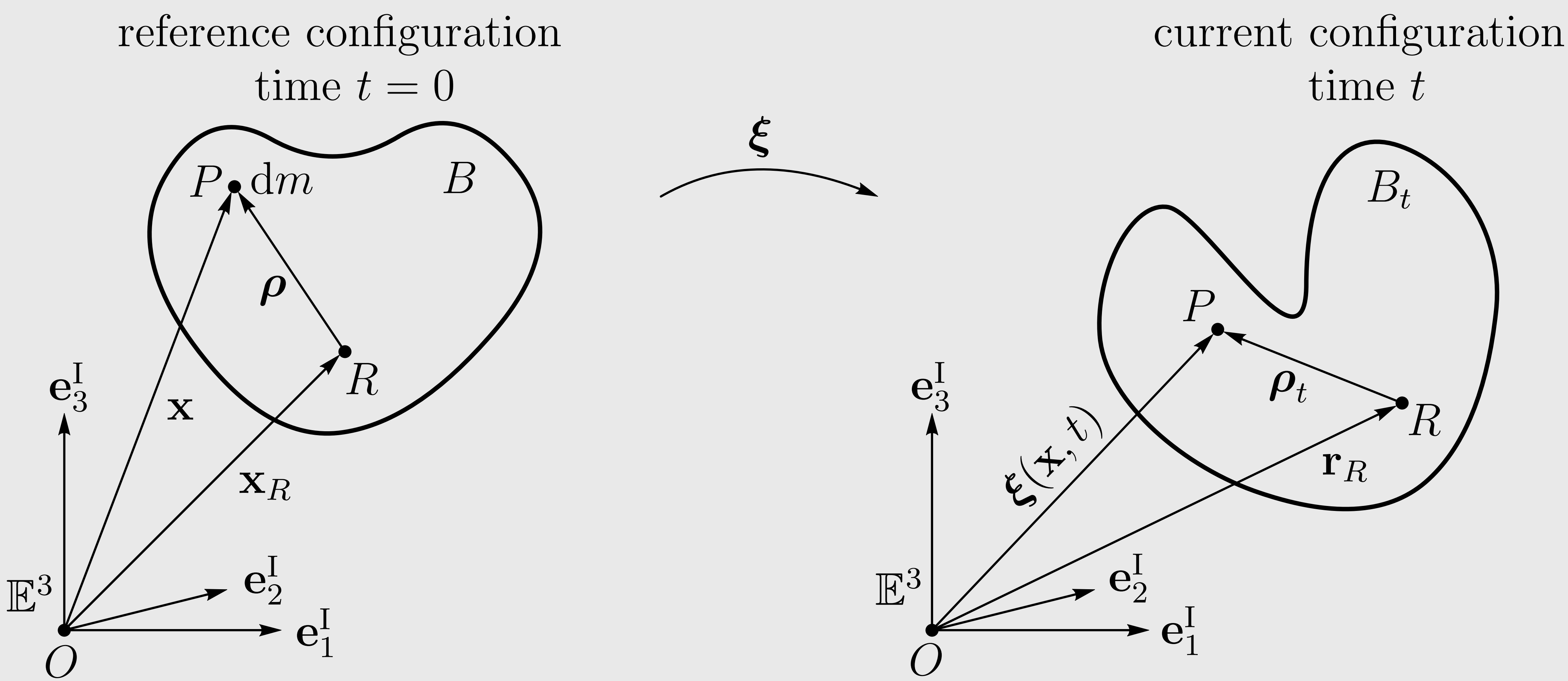

$\mathbf{R}^{-1}\mathbf{df}_\mathbf{x} = \nabla\mathbf{f}_\mathbf{x}$
$\mathbf{R}^{-1}\mathcal{N}_\mathcal{C}(\mathbf{x})$
x_2
$\nabla\mathbf{f}_\mathbf{z}$
$\mathbf{z}$
$\mathbf{t}_\mathbf{z}$
$\mathbf{x}$
$\mathbf{t}_\mathbf{x}$
①
$\mathbf{x} - \nabla\mathbf{f}_\mathbf{x}$
$\nabla\mathbf{f}_{\mathbf{x}^*}$
$\mathbf{x}^*$
$\mathbf{0}$
$\mathcal{C}$
$\mathbf{R}^{-1}\mathcal{N}_\mathcal{C}(\mathbf{x}^*)$
②



Body Motion


reference configuration
time $t = 0$
current configuration
time t
$\boldsymbol{\xi}$
P
$\mathrm{d}m$
B
$\boldsymbol{\rho}$
R
$\mathbf{x}$
$\mathbf{x}_R$
$\mathbf{e}_3^I$
$\mathbf{e}_2^I$
$\mathbf{e}_1^I$
$\mathbb{E}^3$
O
B_t
$\boldsymbol{\rho}_t$
$\mathbf{r}_R$
$\boldsymbol{\xi}(\mathbf{x}, t)$

Body Motion

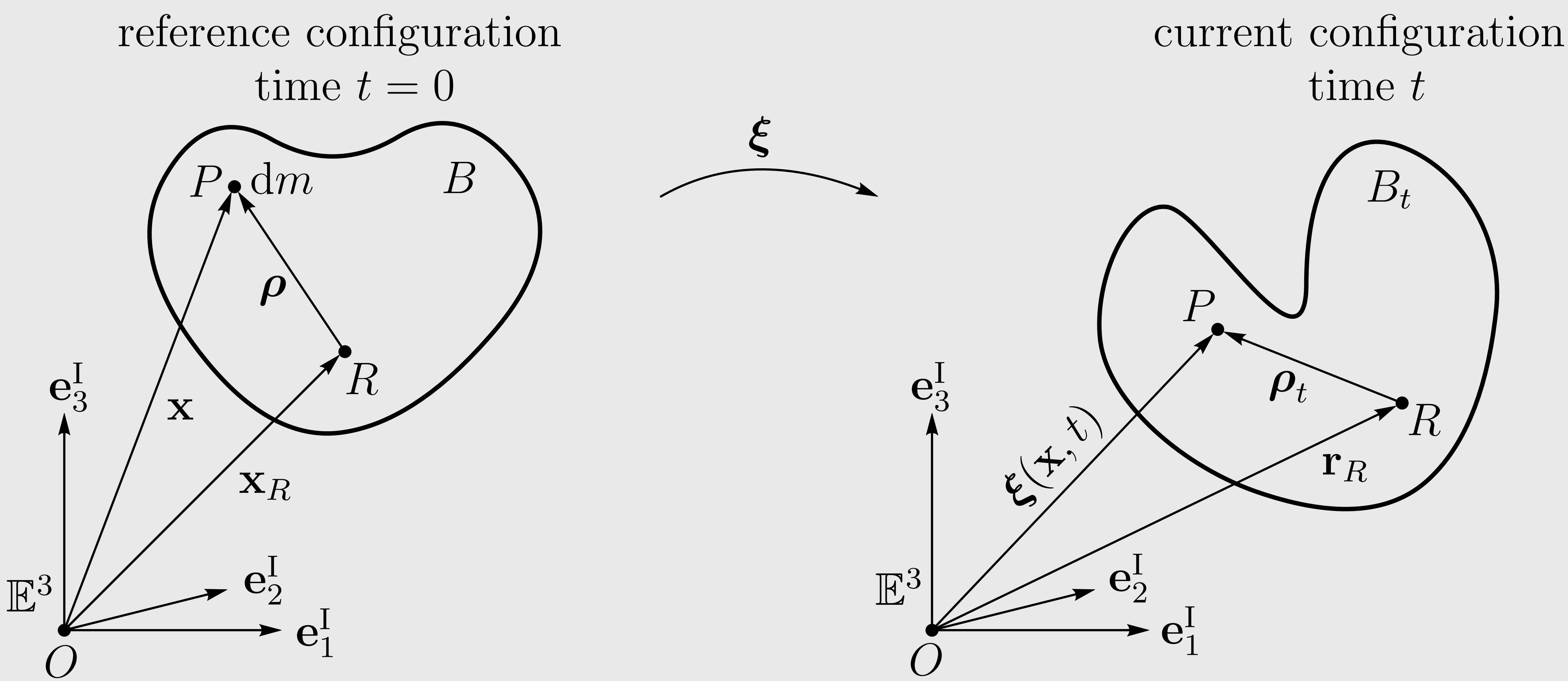


Motion: $\boldsymbol{\xi}(\cdot,t) : B \rightarrow B_t \subset \mathbb{E}^3$

$$\boldsymbol{\rho} \mapsto \boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{\rho},t)$$

Inner Product: $(\mathbf{x}\,|\,\mathbf{y}) \Rightarrow \|\mathbf{x}\|_2 := \sqrt{(\mathbf{x}\,|\,\mathbf{x})}$

Body Motion



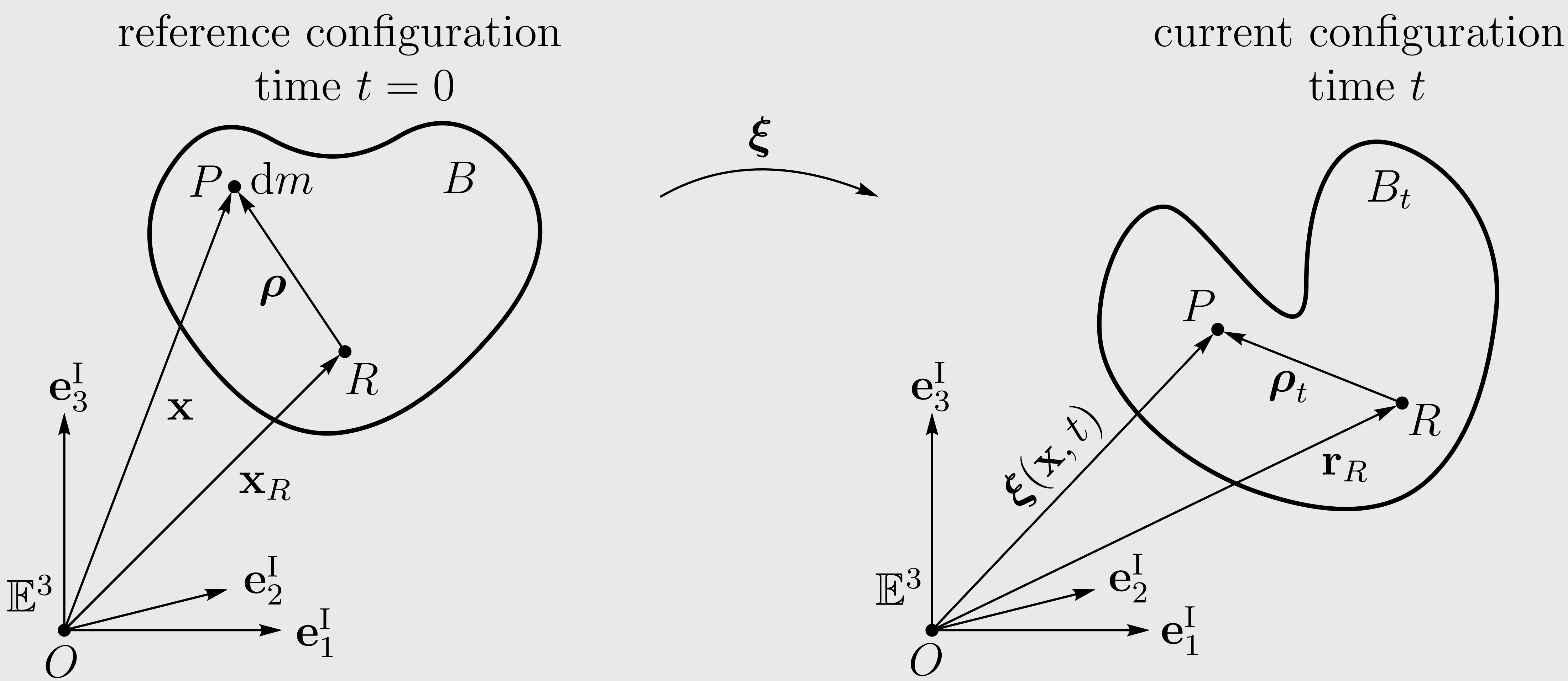
Motion: $\boldsymbol{\xi}(\cdot, t) : B \rightarrow B_t \subset \mathbb{E}^3$

$$\boldsymbol{\rho} \mapsto \boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{\rho}, t)$$

Inner Product: $(\mathbf{x} \,|\, \mathbf{y}) \Rightarrow \|\mathbf{x}\|_2 := \sqrt{(\mathbf{x} \,|\, \mathbf{x})}$

Isometry: $\|\boldsymbol{\xi}(\boldsymbol{\rho}_1, t) - \boldsymbol{\xi}(\boldsymbol{\rho}_2, t)\|_2 = \|\boldsymbol{\rho}_1 - \boldsymbol{\rho}_2\|_2 \quad \forall \boldsymbol{\rho}_1, \boldsymbol{\rho}_2 \in B \subset \mathbb{E}^3$

Body Motion



Motion: $\boldsymbol{\xi}(\cdot, t) : B \to B_t \subset \mathbb{E}^3$

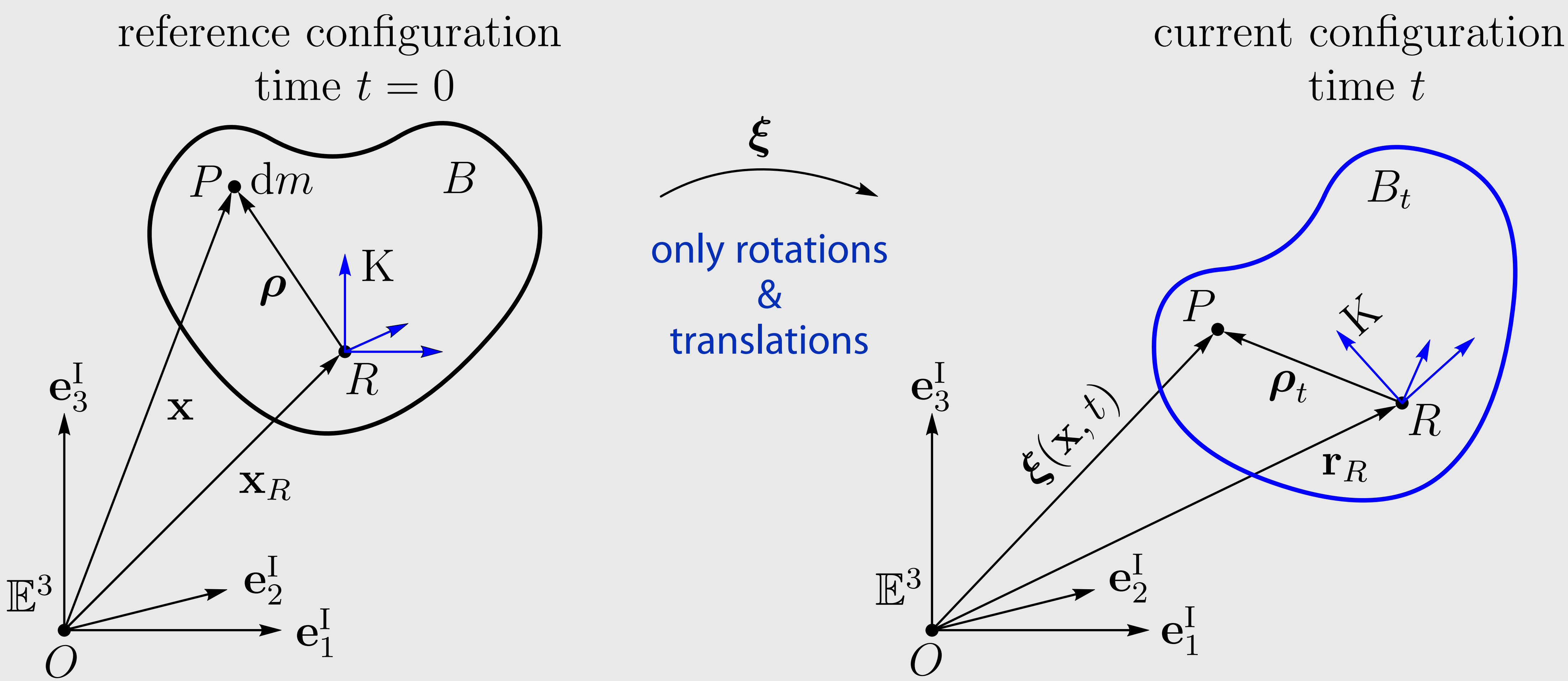
$$\boldsymbol{\rho} \mapsto \boldsymbol{\xi}_{\text{aff}}(\boldsymbol{\rho}, t) := \mathcal{T}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

Mazur-Ulam

Inner Product: $(\mathbf{x} \,|\, \mathbf{y}) \Rightarrow \|\mathbf{x}\|_2 := \sqrt{(\mathbf{x} \,|\, \mathbf{x})}$

Isometry: $\|\boldsymbol{\xi}(\boldsymbol{\rho}_1, t) - \boldsymbol{\xi}(\boldsymbol{\rho}_2, t)\|_2 = \|\boldsymbol{\rho}_1 - \boldsymbol{\rho}_2\|_2 \quad \forall \boldsymbol{\rho}_1, \boldsymbol{\rho}_2 \in B \subset \mathbb{E}^3$

Body Motion ⇒ Rigid Body Motion



Motion: $\boldsymbol{\xi}(\cdot, t) : B \;\to\; B_t \subset \mathbb{E}^3$

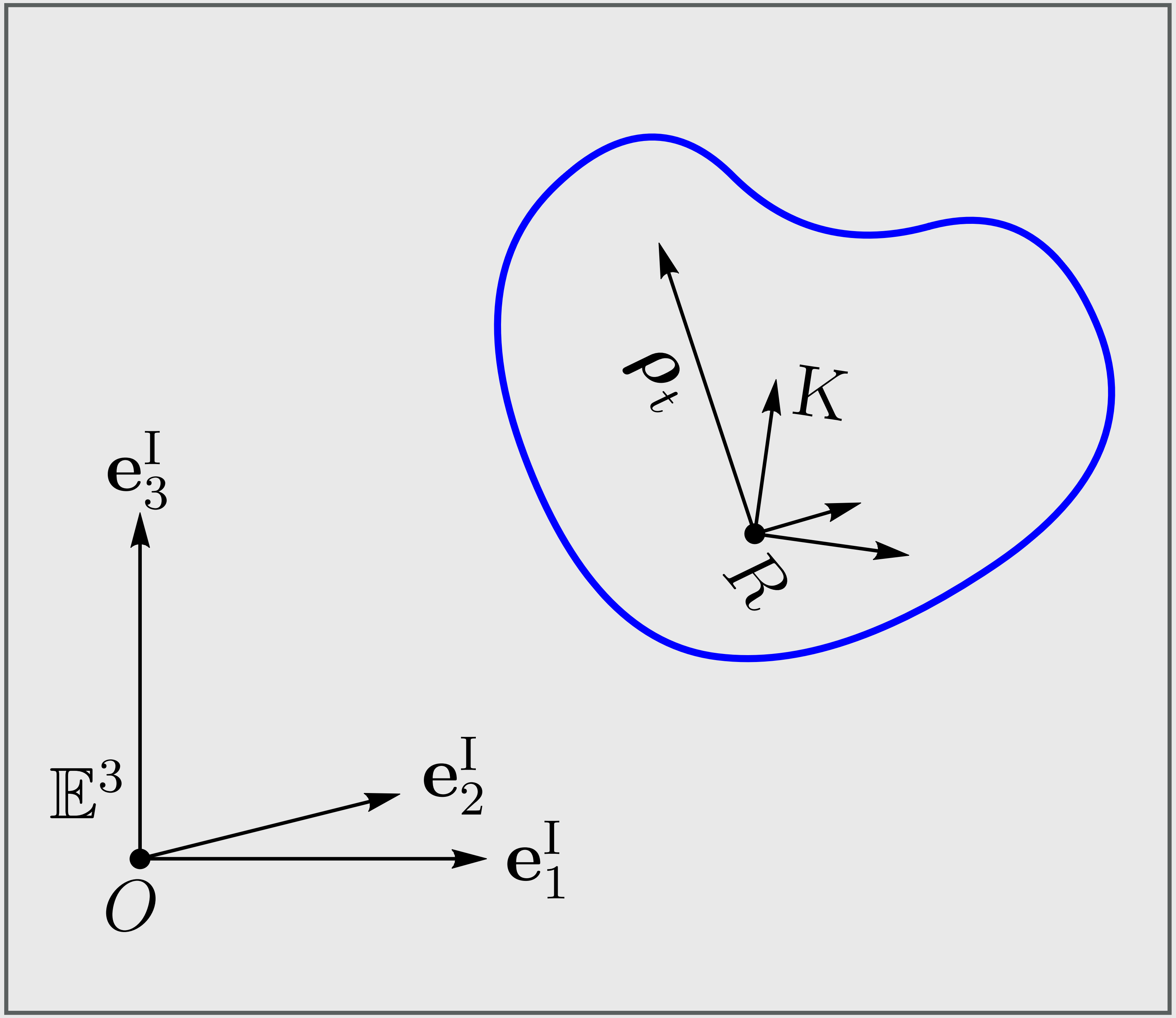
$$\boldsymbol{\rho} \;\mapsto\; \boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

Inner Product: $(\mathbf{x}\,|\,\mathbf{y}) \Rightarrow \|\mathbf{x}\|_2 := \sqrt{(\mathbf{x}\,|\,\mathbf{x})}$

Mazur-Ulam

Isometry: $\|\boldsymbol{\xi}(\boldsymbol{\rho}_1, t) - \boldsymbol{\xi}(\boldsymbol{\rho}_2, t)\|_2 = \|\boldsymbol{\rho}_1 - \boldsymbol{\rho}_2\|_2 \quad \forall \boldsymbol{\rho}_1, \boldsymbol{\rho}_2 \in B \subset \mathbb{E}^3$

What is a Rotation?



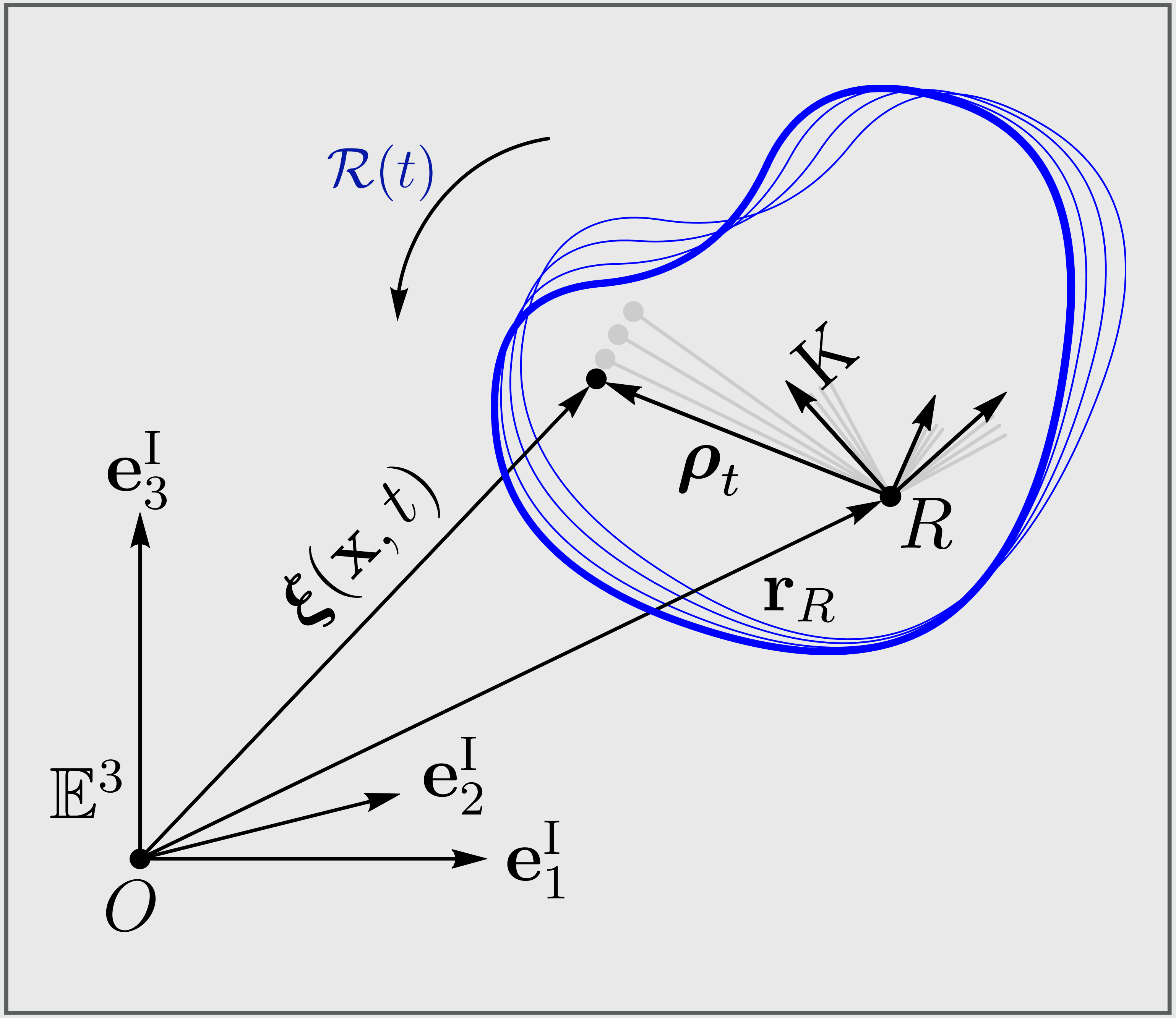
$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

A rotation is a linear map which:

- preserves the inner product (preserving angles)
- preserves the induced norm (preserves length)

If we preserve orientation (determinant = 1) too, we look at proper rotations.

What is a Rotation?



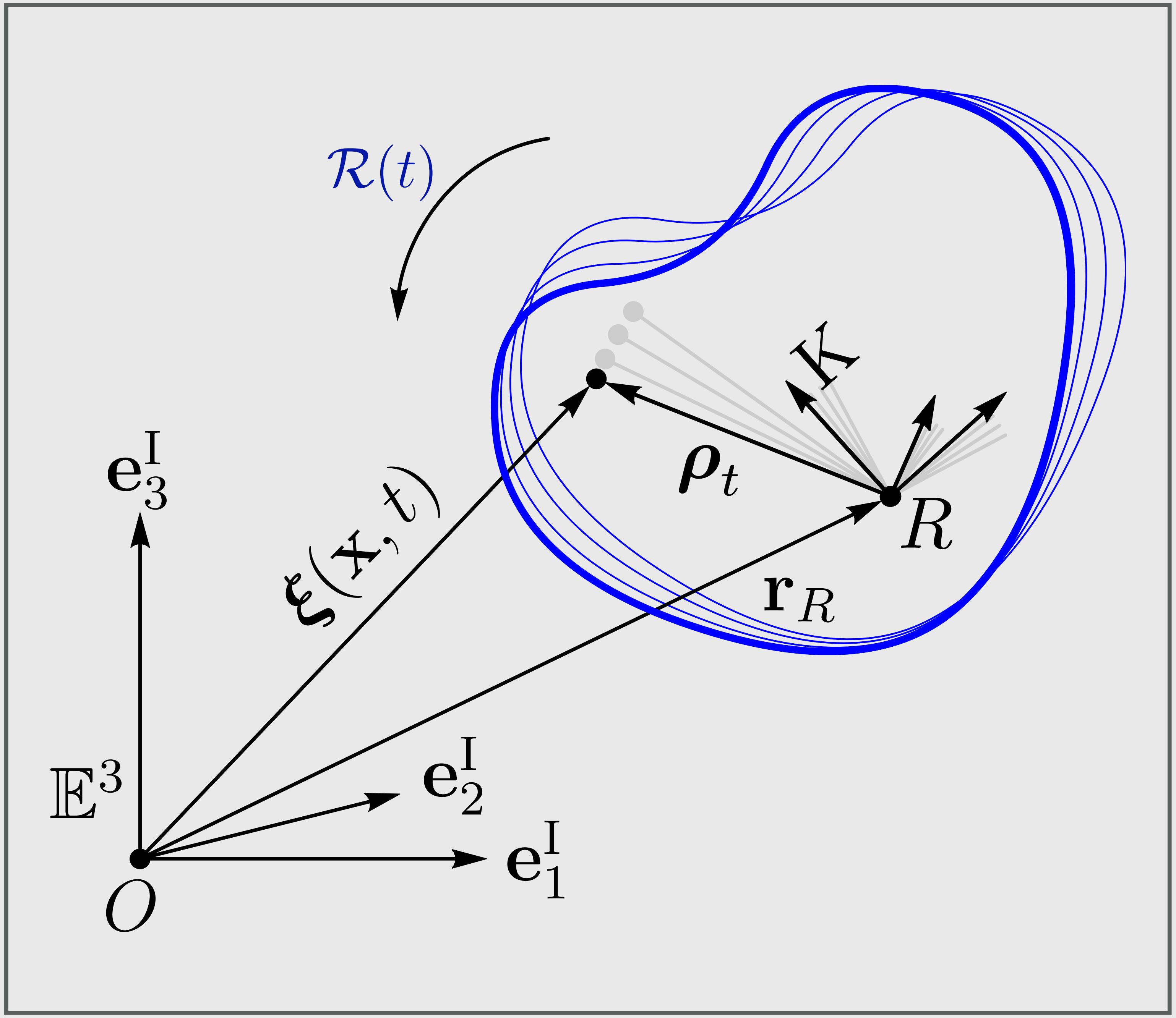
$$\xi_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

A rotation is a linear map which:

- preserves the inner product (preserving angles)
- preserves the induced norm (preserves length)

If we preserve orientation (determinant = 1) too, we look at proper rotations.

What is a Rotation?



$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho},t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

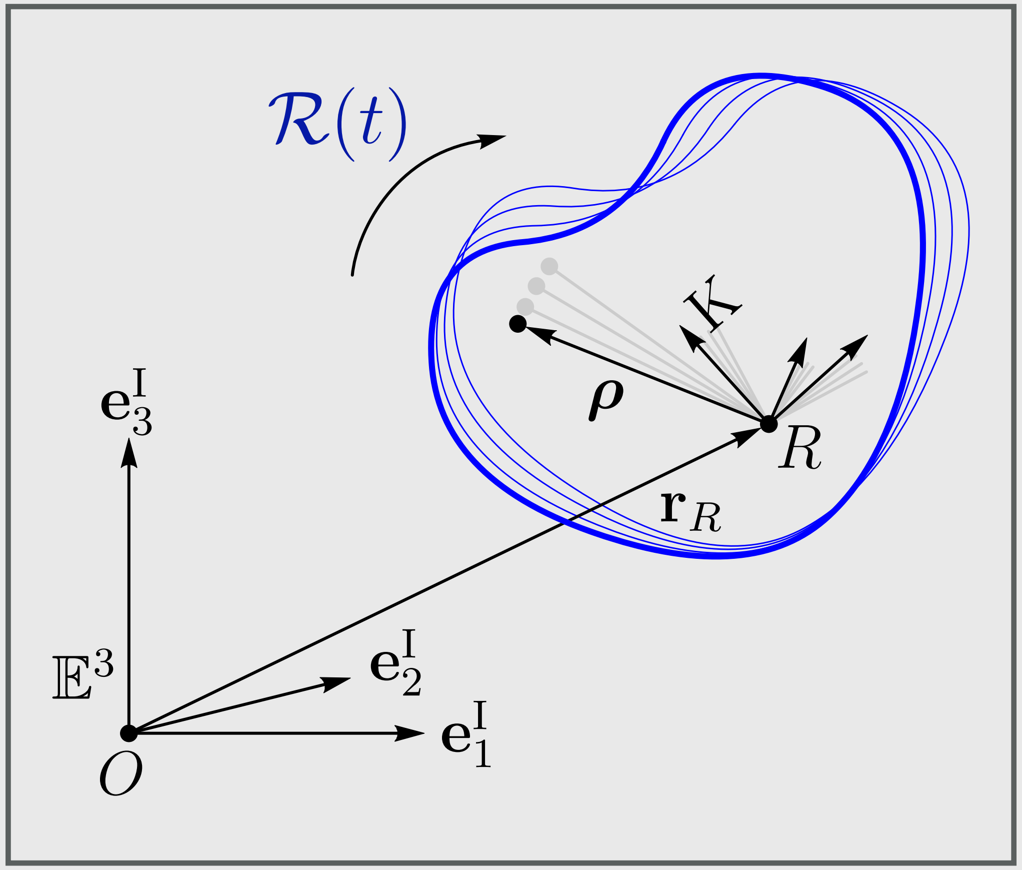
A rotation is a linear map which:

- preserves the inner product (preserving angles)
- preserves the induced norm (preserves length)

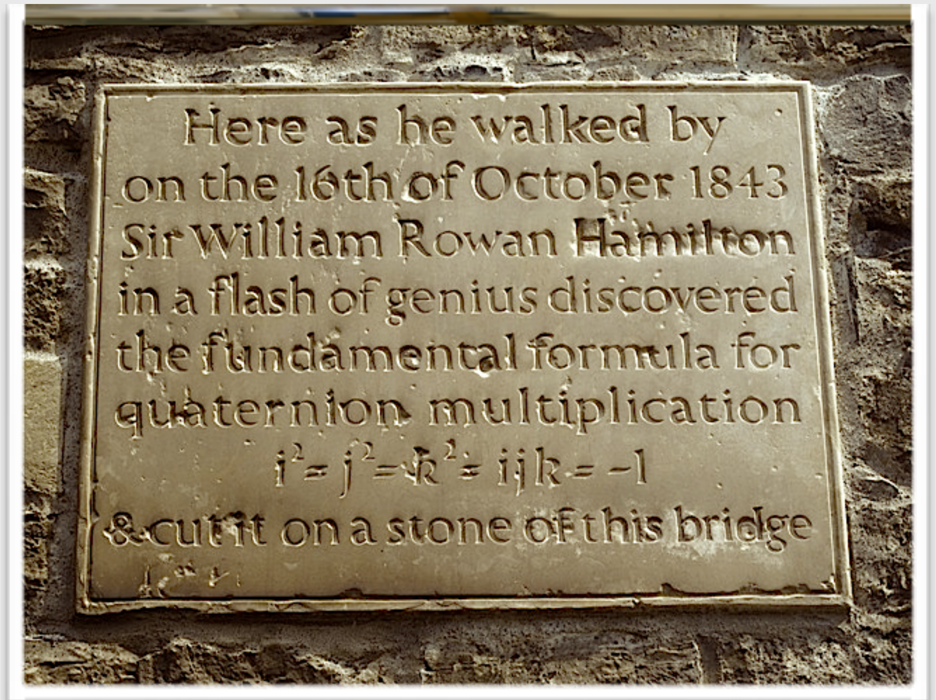
If we preserve orientation (determinant = 1) too, we look at proper rotations.

special orthogonal group: $SO(3) := \{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^{\top}\mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = 1\}$

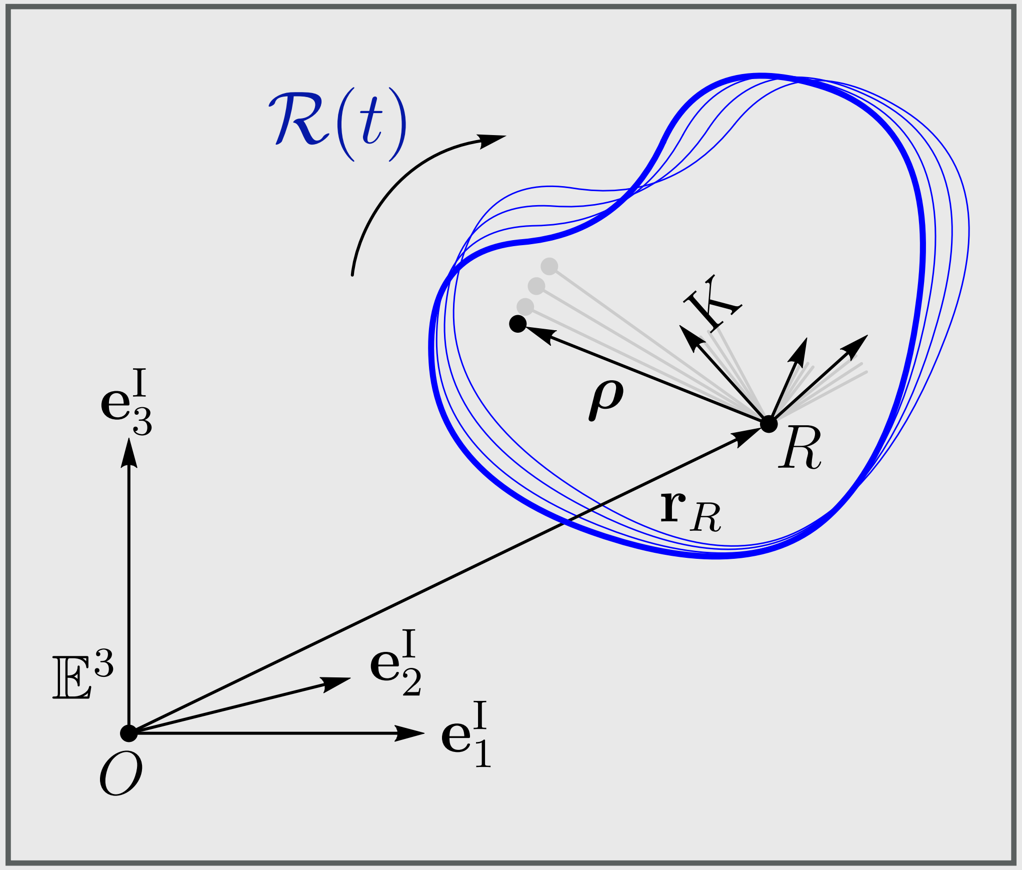
Quaternions for Rotations (Rodrigues 1840, Hamilton 1843)



$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$



Quaternions for Rotations (Rodrigues 1840, Hamilton 1843)



$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

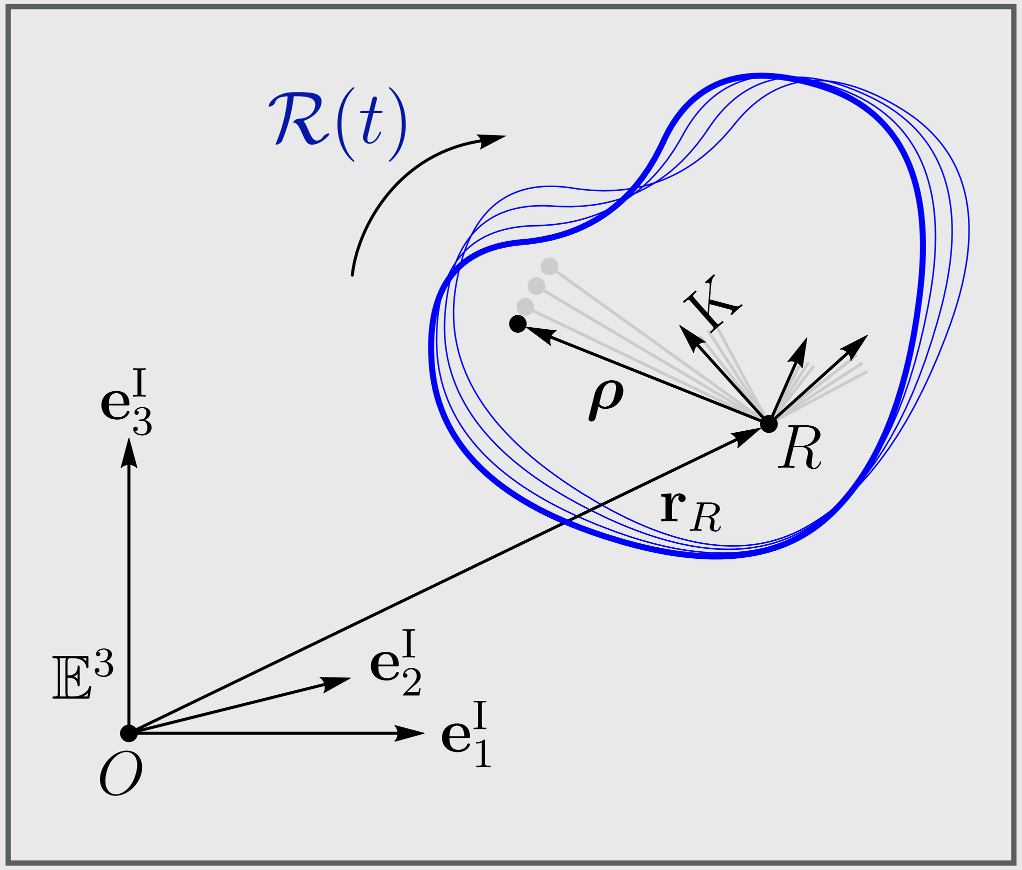
Theory of Couplets (Complex Numbers):

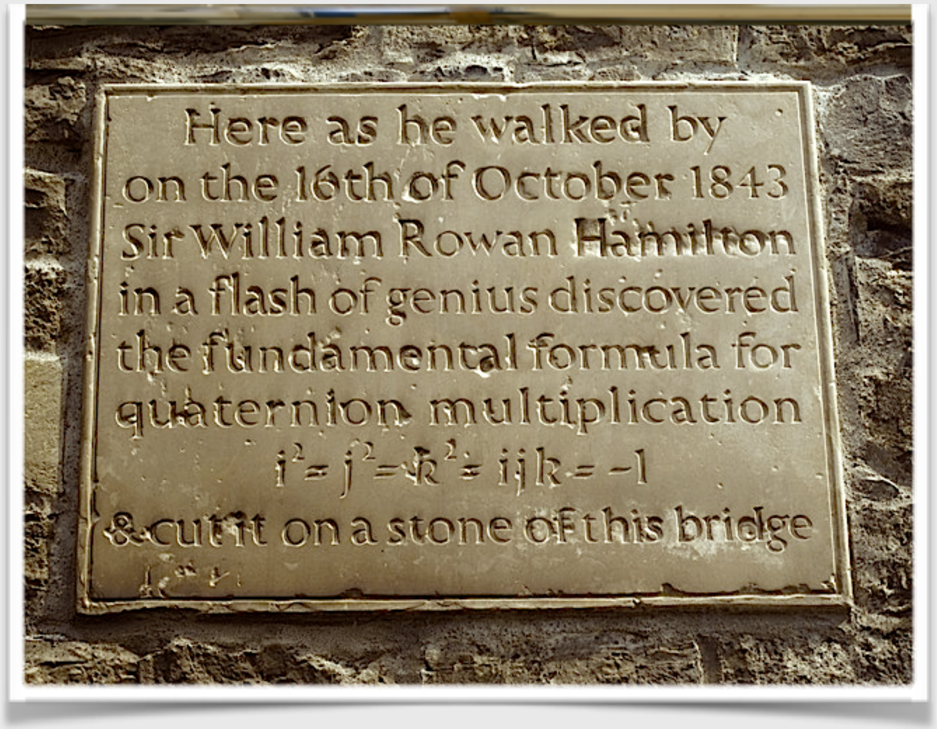
$a + bi \in \mathbb{C} \quad \Leftrightarrow \quad re^{i\varphi} \in \mathbb{C}$ with $i^2 = -1$

Add: $(a, b) + (c, d) = (a + c, b + d)$

Mult: $(a, b) \cdot (c, d) = (ac - bd, ad - bc)$

Quaternions for Rotations (Rodrigues 1840, Hamilton 1843)





$$\boldsymbol{\xi}_{\text{rig}}(\boldsymbol{\rho}, t) := \boxed{\mathcal{R}(t)}(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

Theory of Couplets (Complex Numbers):

$a + bi \in \mathbb{C} \quad \Leftrightarrow \quad re^{i\varphi} \in \mathbb{C}$ with $i^2 = -1$

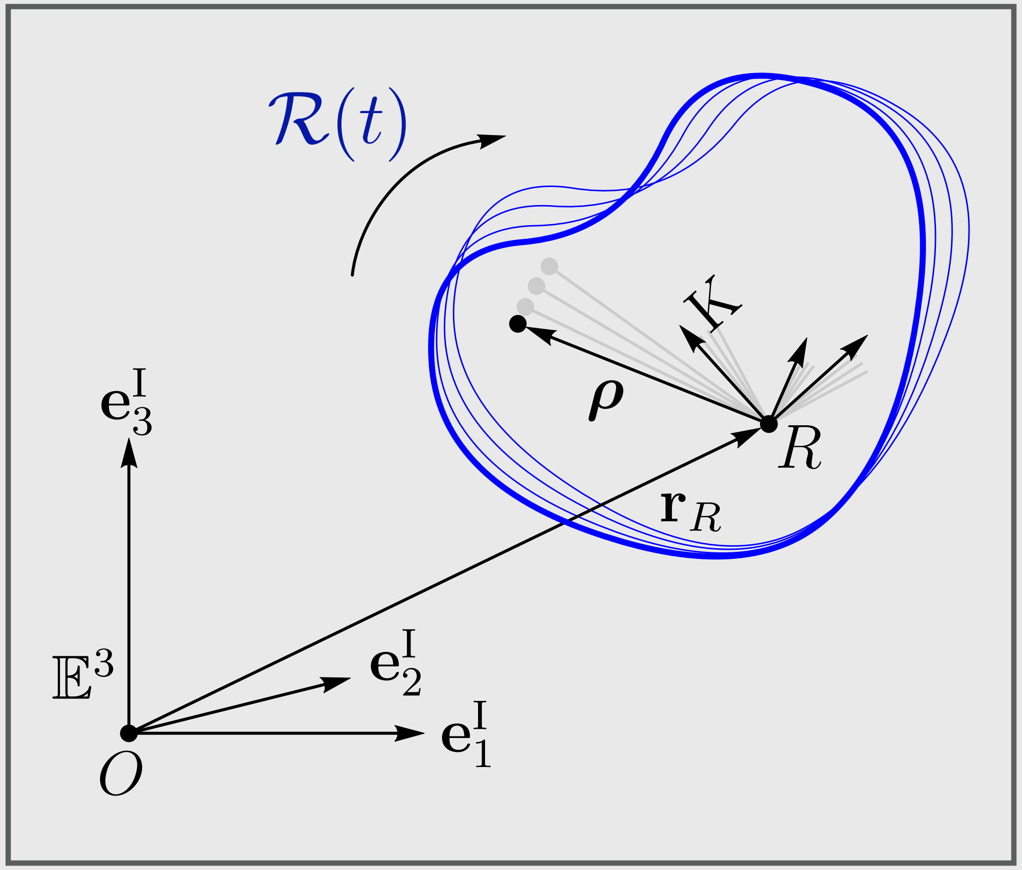
Add: $(a, b) + (c, d) = (a + c, b + d)$

Mult: $(a, b) \cdot (c, d) = (ac - bd, ad - bc)$

Addition and multiplication leads to rotations in 2D. **How to make it work for triples?**

$a + bi + cj$ with $i^2 = j^2 = -1$ doesn't work!

Quaternions for Rotations (Rodrigues 1840, Hamilton 1843)



$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

Theory of Couplets (Complex Numbers):

$a + bi \in \mathbb{C} \quad \Leftrightarrow \quad re^{i\varphi} \in \mathbb{C}$ with $i^2 = -1$

Add: $(a, b) + (c, d) = (a + c, b + d)$

Mult: $(a, b) \cdot (c, d) = (ac - bd, ad - bc)$

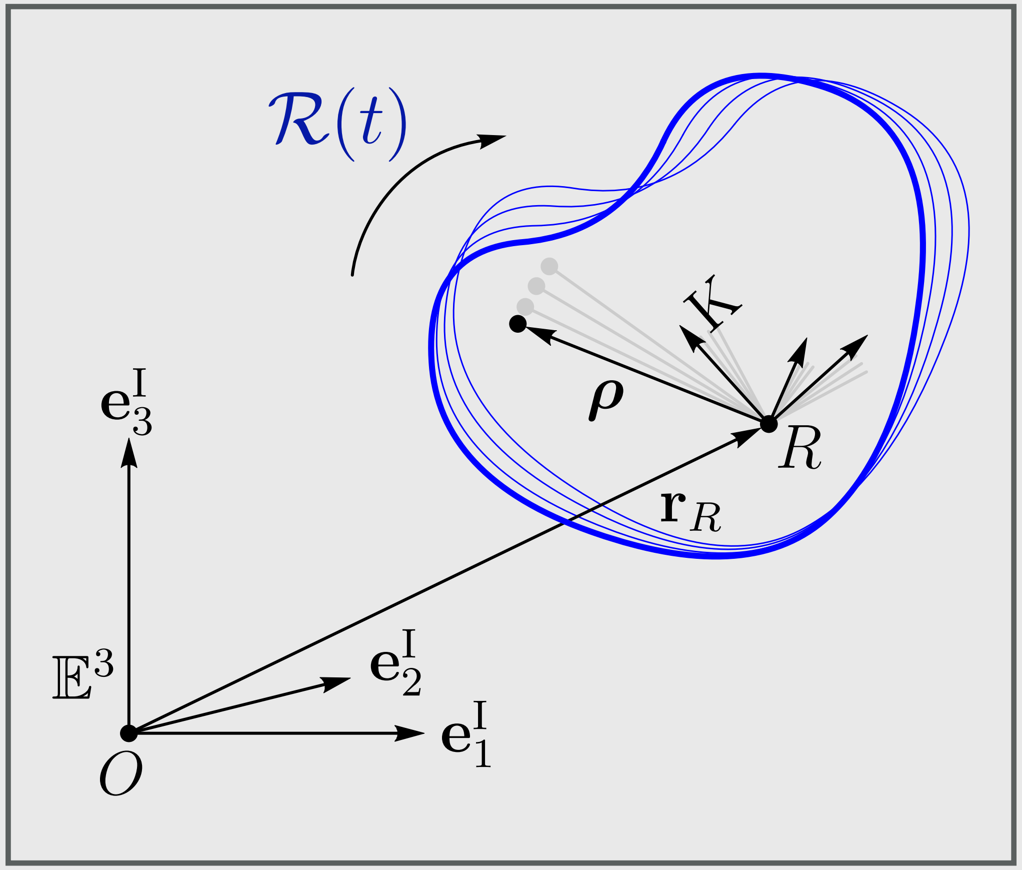
Addition and multiplication leads to rotations in 2D. **How to make it work for triples?**

$a + bi + cj$ with $i^2 = j^2 = -1$ doesn't work!

$$\mathbb{H} := \{p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k} \ , p_i \in \mathbb{R}\}$$

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ikj} = -1$$

Quaternions for Rotations (Rodrigues 1840, Hamilton 1843)



$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

Theory of Couplets (Complex Numbers):

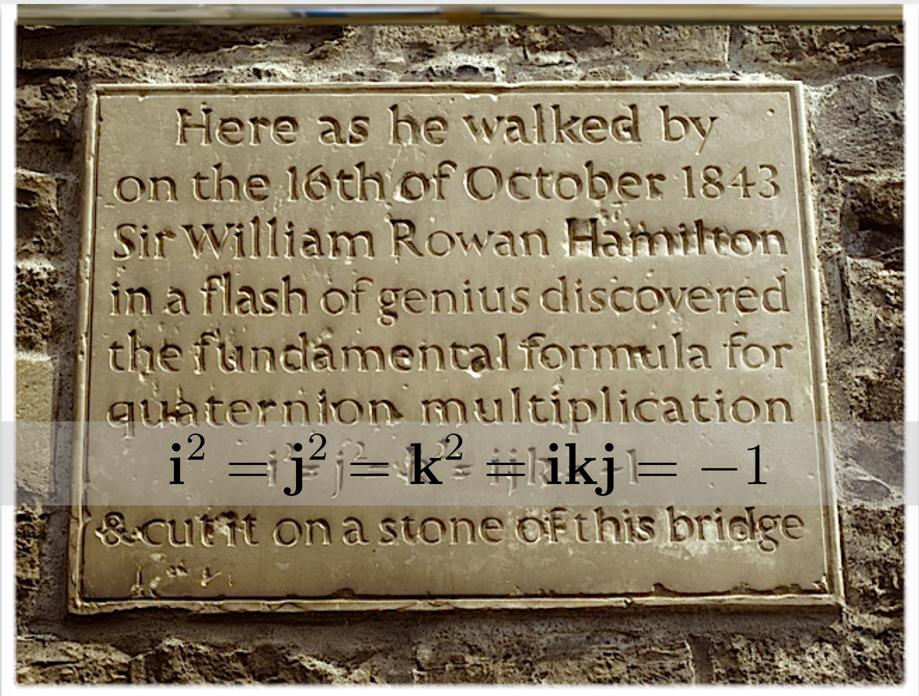
$a + bi \in \mathbb{C} \quad \Leftrightarrow \quad re^{i\varphi} \in \mathbb{C}$ with $i^2 = -1$

Add: $(a, b) + (c, d) = (a + c, b + d)$

Mult: $(a, b) \cdot (c, d) = (ac - bd, ad - bc)$

Addition and multiplication leads to rotations in 2D. **How to make it work for triples?**

$a + bi + cj$ with $i^2 = j^2 = -1$ doesn't work!



$$\mathbb{H} := \{p_0\mathbf{I} + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}\ , p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

with:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{i} = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}, \quad \mathbf{j} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$$

Quaternions for Rotations

$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

$$\mathbf{P} \in \mathbb{H} := \{\underbrace{p_0\mathbf{I}}_{\text{scalar part}} + \underbrace{p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}}_{\text{pure part}}\ , p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

Quaternions for Rotations

$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

$$\mathbf{P} \in \mathbb{H} := \{\underbrace{p_0\mathbf{I}}_{\text{scalar part}} + \underbrace{p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}}_{\text{pure part}}, p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

Rotations in $\mathbb{E}^4$ where P, Q are unit quaternions (= 6 min. parameters):

$$\begin{array}{ccc} \mathbb{H} & \to & \mathbb{H} \\ \mathbf{X} & \mapsto & \mathbf{PXQ} \end{array} \quad \xrightarrow{\mathcal{K}} \quad \begin{array}{ccc} \mathbb{R}^4 & \to & \mathbb{R}^4 \\ \mathbf{x} & \mapsto & \underbrace{\varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})}_{\in\, \mathbb{R}^{4\times 4}}\underbrace{\mathbf{x}}_{\in\, \mathbb{R}^4} \end{array}$$

Quaternions for Rotations

$$\boldsymbol{\xi}_{\text{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

$$\mathbf{P} \in \mathbb{H} := \{\underbrace{p_0\mathbf{I}}_{\text{scalar part}} + \underbrace{p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}}_{\text{pure part}}, p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

Rotations in $\mathbb{E}^4$ where P, Q are unit quaternions (= 6 min. parameters):

$$\begin{array}{ccc} \mathbb{H} & \to & \mathbb{H} \\ \mathbf{X} & \mapsto & \mathbf{PXQ} \end{array} \xrightarrow{\mathcal{K}} \begin{array}{ccc} \mathbb{R}^4 & \to & \mathbb{R}^4 \\ \mathbf{x} & \mapsto & \underbrace{\boldsymbol{\varphi}_L(\mathbf{P})\boldsymbol{\varphi}_R(\mathbf{Q})}_{\in \mathbb{R}^{4\times 4}}\underbrace{\mathbf{x}}_{\in \mathbb{R}^4} \end{array}$$

make $\mathbf{x}$ pure and $\mathbf{Q} := \mathbf{P}^{-1}$

rotations in $\mathbb{E}^3$

Quaternions for Rotations

$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

$$\mathbf{P} \in \mathbb{H} := \{\underbrace{p_0\mathbf{I}}_{\text{scalar part}} + \underbrace{p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}}_{\text{pure part}}, p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

Rotations in $\mathbb{E}^4$ where $\mathbf{P}, \mathbf{Q}$ are unit quaternions (= 6 min. parameters):

$$\begin{array}{ccc} \mathbb{H} & \to & \mathbb{H} \\ \mathbf{X} & \mapsto & \mathbf{PXQ} \end{array} \xrightarrow{\quad\mathcal{K}\quad} \begin{array}{ccc} \mathbb{R}^4 & \to & \mathbb{R}^4 \\ \mathbf{x} & \mapsto & \underset{\mathbb{R}^{4\times 4}}{\underset{\cup}{\boldsymbol{\varphi}_L(\mathbf{P})\boldsymbol{\varphi}_R(\mathbf{Q})}}\underset{\mathbb{R}^4}{\underset{\cup}{\mathbf{x}}} \end{array}$$

make $\mathbf{x}$ pure and $\mathbf{Q} := \mathbf{P}^{-1}$

rotations in $\mathbb{E}^3$

scalable body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{scal}} = s\ {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

scaling factor: $s := \|\mathbf{p}\|$

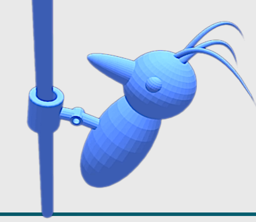
rigid body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

constraint: $\|\mathbf{p}\| = 1 \quad \forall t$



Quaternions for Rotations

$$\boldsymbol{\xi}_{\text{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

$$\mathbf{P} \in \mathbb{H} := \{\underbrace{p_0\mathbf{I}}_{\text{scalar part}} + \underbrace{p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}}_{\text{pure part}}, p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

Rotations in $\mathbb{E}^4$ where $\mathbf{P}, \mathbf{Q}$ are unit quaternions (= 6 min. parameters):

$$\begin{array}{ccc} \mathbb{H} & \to & \mathbb{H} \\ \mathbf{X} & \mapsto & \mathbf{PXQ} \end{array} \xrightarrow{\quad\mathcal{K}\quad} \begin{array}{ccc} \mathbb{R}^4 & \to & \mathbb{R}^4 \\ \mathbf{x} & \mapsto & \underset{\mathbb{R}^{4\times 4}}{\underset{\cup}{\boldsymbol{\varphi}_L(\mathbf{P})}}\boldsymbol{\varphi}_R(\mathbf{Q})\underset{\mathbb{R}^4}{\underset{\cup}{\mathbf{x}}} \end{array}$$

make $\mathbf{x}$ pure and $\mathbf{Q} := \mathbf{P}^{-1}$

rotations in $\mathbb{E}^3$

scalable body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\text{scal}} = s\ {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

scaling factor: $s := \|\mathbf{p}\|$

rigid body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\text{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

constraint: $\|\mathbf{p}\| = 1 \quad \forall t$

equation of motions by principles

The Principle of Virtual Work

What the hack is a force?

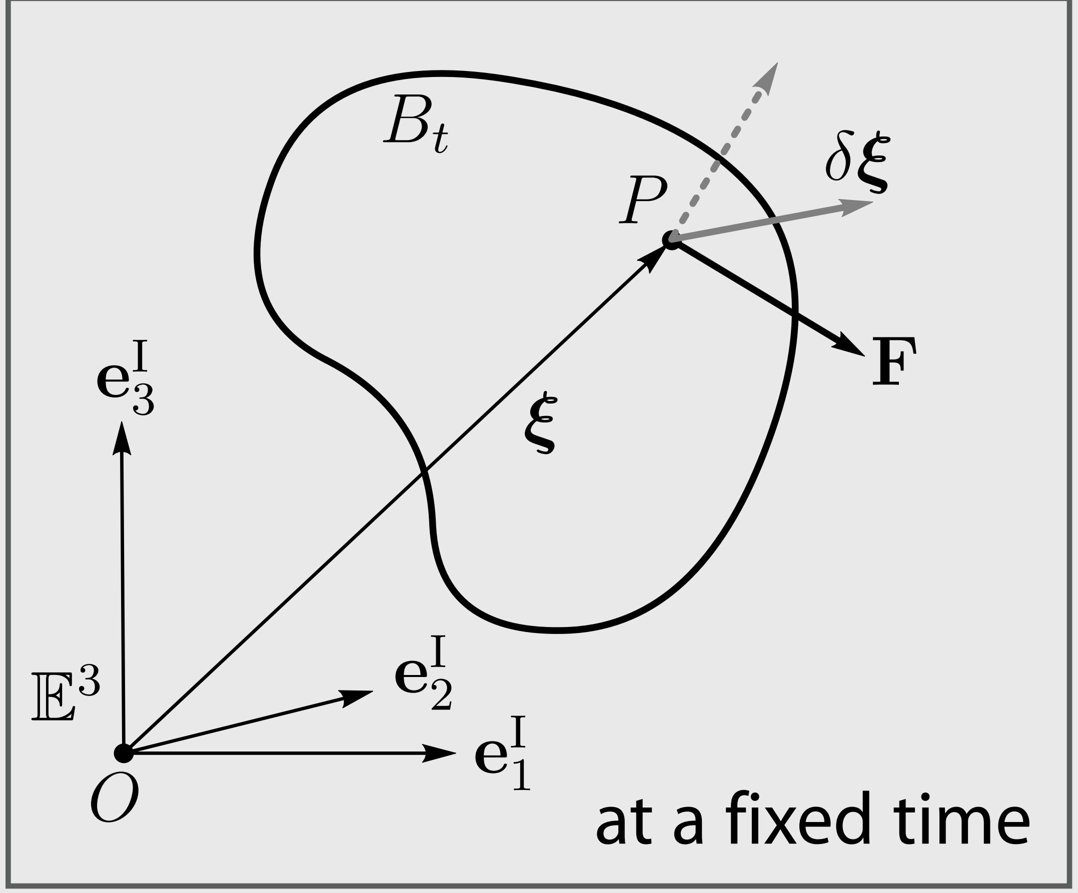
linear functional ⇒ *producing virtual work*

when given a virtual displacement.

$\langle \mathbf{F} \,|\, \delta\boldsymbol{\xi} \rangle =$ $\mathbf{F}(\delta\boldsymbol{\xi}) = \delta W$

duality pairing

force virtual displacement



Principle of Virtual Work:

At any instant of time t, the virtual work δW of a body $B \in \mathbb{E}^3$ vanishes for all virtual displacements $\delta\boldsymbol{\xi}$, that is,

$$\delta W(\delta\boldsymbol{\xi}) = \int_B \langle \mathrm{d}\mathbf{F} \,|\, \delta\boldsymbol{\xi}(\mathbf{x}) \rangle = \delta W^{\mathrm{dyn}}(\delta\boldsymbol{\xi}) + \delta W^{\mathrm{int}}(\delta\boldsymbol{\xi}) + \delta W^{\mathrm{ext}}(\delta\boldsymbol{\xi}) = 0 \quad \forall \delta\boldsymbol{\xi} \;.$$

The Principle of Virtual Work

What the hack is a force?

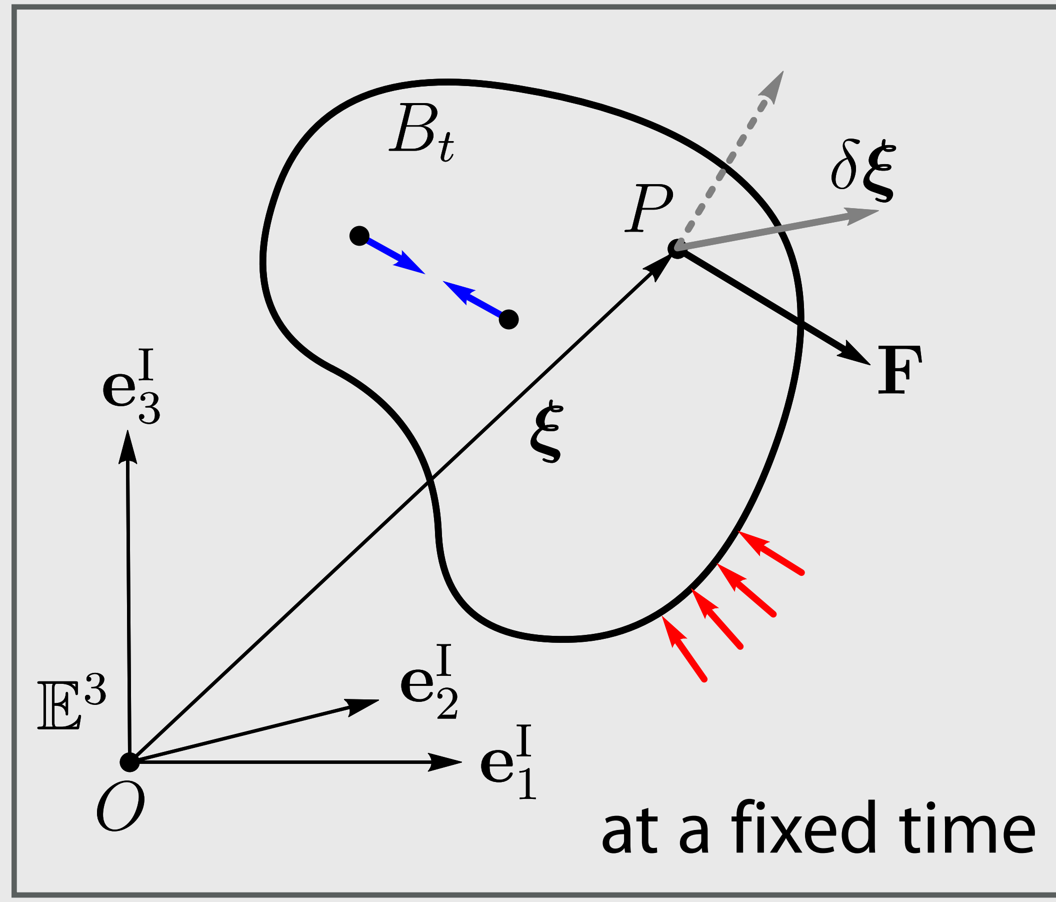
linear functional ⇒ *producing virtual work*

when given a virtual displacement.

$\langle \mathbf{F} \,|\, \delta\boldsymbol{\xi} \rangle =$ $\mathbf{F}(\delta\boldsymbol{\xi}) = \delta W$

duality pairing

force virtual displacement



Principle of Virtual Work:

At any instant of time t, the virtual work δW of a body $B \in \mathbb{E}^3$ vanishes for all virtual displacements $\delta\boldsymbol{\xi}$, that is,

$$\delta W(\delta\boldsymbol{\xi}) = \int_B \langle \mathrm{d}\mathbf{F} \,|\, \delta\boldsymbol{\xi}(\mathbf{x}) \rangle = \delta W^{\mathrm{dyn}}(\delta\boldsymbol{\xi}) + \delta W^{\mathrm{int}}(\delta\boldsymbol{\xi}) + \delta W^{\mathrm{ext}}(\delta\boldsymbol{\xi}) = 0 \quad \forall \delta\boldsymbol{\xi} \ .$$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\boxed{{}_\mathrm{I}\boldsymbol{\xi}_\mathrm{rig} = {}_\mathrm{I}\mathbf{R}(\mathbf{p})\ {}_\mathrm{I}\boldsymbol{\rho} + {}_\mathrm{I}\mathbf{r}_R}$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_\mathrm{I}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$\boxed{0 = \delta W({}_\mathrm{I}\delta\boldsymbol{\xi}_\mathrm{rig}) = \delta W^\mathrm{dyn} + \delta W^\mathrm{ext} + \delta W^\mathrm{int} \qquad \forall_\mathrm{I}\delta\boldsymbol{\xi}_\mathrm{rig},\ t}$$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W^{\mathrm{ext}} + \delta W^{\mathrm{int}} \qquad \forall_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t$$

$$0 = \boxed{-\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}^{\top}\ \boxed{{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\mathrm{d}m}}$$

constitutive force law:
Newton's second law

Applying the Fundamental Axioms

Equations of motion for the rigid body:

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

rigid body motion: $${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W^{\mathrm{ext}} + \delta W^{\mathrm{int}} \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t$$

$$0 = -\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\, {}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\,\mathrm{d}m + \int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\, {}_{\mathrm{I}}\mathrm{d}\mathbf{F}^{\mathrm{e}}$$

constitutive force law:
Newton's second law

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion:

$$ {}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R $$

$$ \|\mathbf{p}\| = 1 \quad \forall t $$

$$ \mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7 $$

$$ 0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W^{\mathrm{ext}} + \delta W^{\mathrm{int}} \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t $$

law of interaction &
perfect constraint
forces ($\infty \rightarrow 6$)

$$ 0 = -\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\ \boxed{{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\,\mathrm{d}m} + \int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\ {}_{\mathrm{I}}\mathrm{d}\mathbf{F}^{\mathrm{e}} + 0 \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t $$

constitutive force law:
Newton's second law

Applying the Fundamental Axioms

Equations of motion for the rigid body:

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

rigid body motion: $\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W^{\mathrm{ext}} + \delta W^{\mathrm{int}} \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t$$

law of interaction & **perfect** constraint forces ($\infty \rightarrow 6$)

$$0 = -\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\ \boxed{{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\mathrm{d}m} + \int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\ {}_{\mathrm{I}}\mathrm{d}\mathbf{F}^{\mathrm{e}} + 0 \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t$$

constitutive force law:
Newton's second law

velocities: $\quad {}_{\mathrm{I}}\dot{\boldsymbol{\xi}}_{\mathrm{rig}} = {}_{\mathrm{K}}\mathbf{R}\ {}_{\mathrm{K}}\tilde{\boldsymbol{\Omega}}\ {}_{\mathrm{K}}\boldsymbol{\rho}_t + {}_{\mathrm{I}}\dot{\mathbf{r}}_R$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_{R}$$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_{R} \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W^{\mathrm{ext}} + \delta W^{\mathrm{int}} \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t$$

law of interaction & perfect constraint forces ($\infty \to 6$)

$$0 = -\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}^{\top}\ \boxed{{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\mathrm{d}m} + \int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}^{\top}\ {}_{\mathrm{I}}\mathrm{d}\mathbf{F}^{\mathrm{e}} + 0 \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t$$

constitutive force law:
Newton's second law

velocities: $${}_{\mathrm{I}}\dot{\boldsymbol{\xi}}_{\mathrm{rig}} = {}_{\mathrm{K}}\mathbf{R}\ {}_{\mathrm{K}}\tilde{\boldsymbol{\Omega}}\ {}_{\mathrm{K}}\boldsymbol{\rho}_t + {}_{\mathrm{I}}\dot{\mathbf{r}}_{R}$$

generalized velocities: $$\mathbf{u} := \begin{bmatrix} {}_{\mathrm{I}}\dot{\mathbf{r}}_{R} \\ {}_{\mathrm{K}}\boldsymbol{\Omega} \end{bmatrix} \in \mathbb{R}^6$$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_I\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

rigid body motion: $\boxed{{}_I\boldsymbol{\xi}_{\text{rig}} = {}_I\mathbf{R}(\mathbf{p})\ {}_I\boldsymbol{\rho} + {}_I\mathbf{r}_R}$

$$0 = \delta W({}_I\delta\boldsymbol{\xi}_{\text{rig}}) = \delta W^{\text{dyn}} + \delta W^{\text{ext}} + \delta W^{\text{int}} \qquad \forall {}_I\delta\boldsymbol{\xi}_{\text{rig}},\ t$$

law of interaction & **perfect** constraint forces ($\infty \rightarrow 6$)

$$0 = -\int_B {}_I\delta\boldsymbol{\xi}_{\text{rig}}^{\top}\ \boxed{{}_I\ddot{\boldsymbol{\xi}}_{\text{rig}}\mathrm{d}m} + \int_B {}_I\delta\boldsymbol{\xi}_{\text{rig}}^{\top}\ {}_I\mathrm{d}\mathbf{F}^{\text{e}} + 0 \qquad \forall {}_I\delta\boldsymbol{\xi}_{\text{rig}},\ t$$

constitutive force law:
Newton's second law

$\dfrac{\mathrm{d}}{\mathrm{d}t}$

velocities: $\quad {}_I\dot{\boldsymbol{\xi}}_{\text{rig}} = {}_K\mathbf{R}\ {}_K\tilde{\boldsymbol{\Omega}}\ {}_K\boldsymbol{\rho}_t + {}_I\dot{\mathbf{r}}_R$

generalized velocities: $\quad \mathbf{u} := \begin{bmatrix} {}_I\dot{\mathbf{r}}_R \\ {}_K\boldsymbol{\Omega} \end{bmatrix} \in \mathbb{R}^6$

Non-Smooth Dynamics in a Nutshell

Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

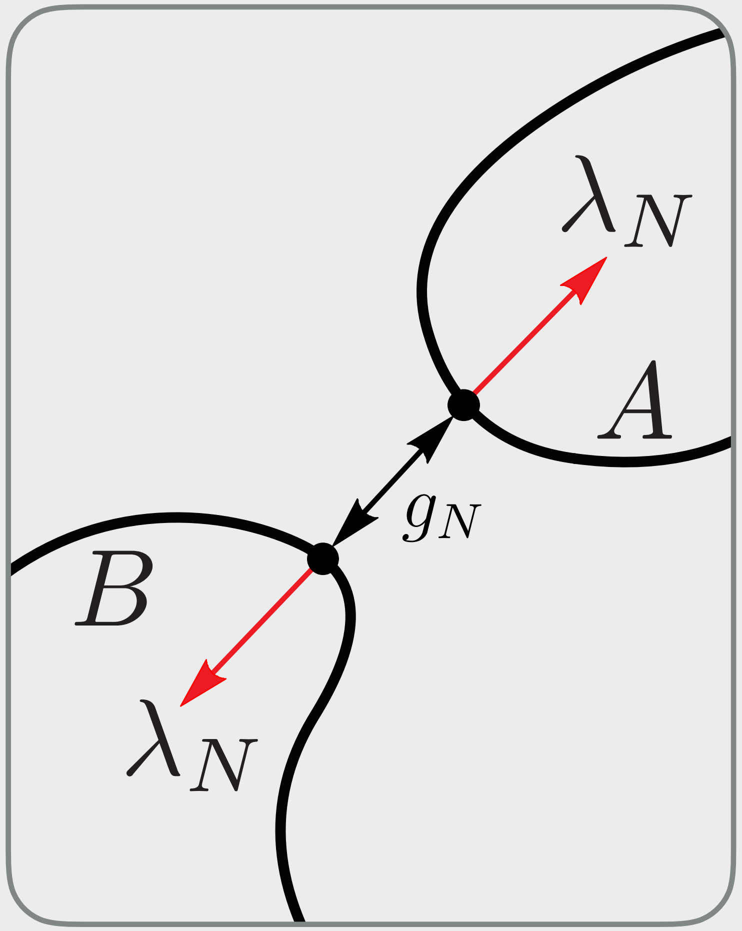
$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$

Non-Smooth Dynamics in a Nutshell

Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

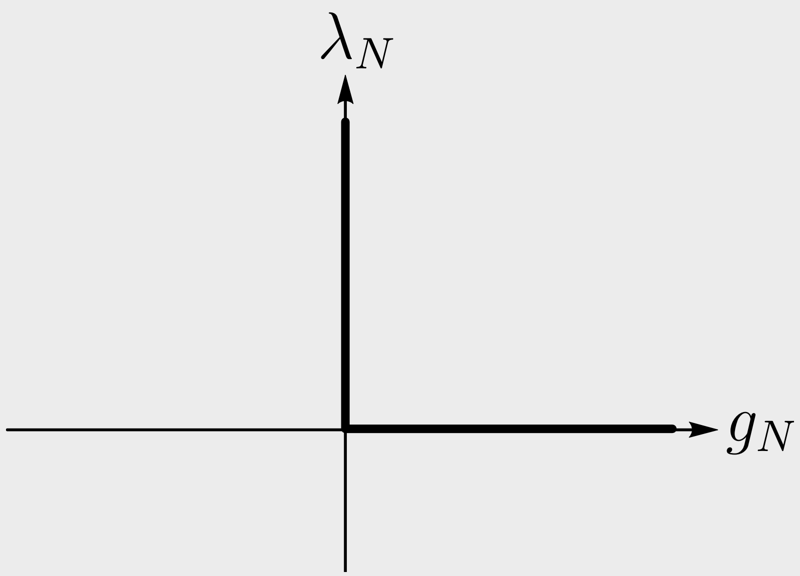
$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$



contact configuration

Unilateral Contact:

$$-g_N \in \mathcal{N}_{\mathbb{R}_0^+}(\lambda_N)$$



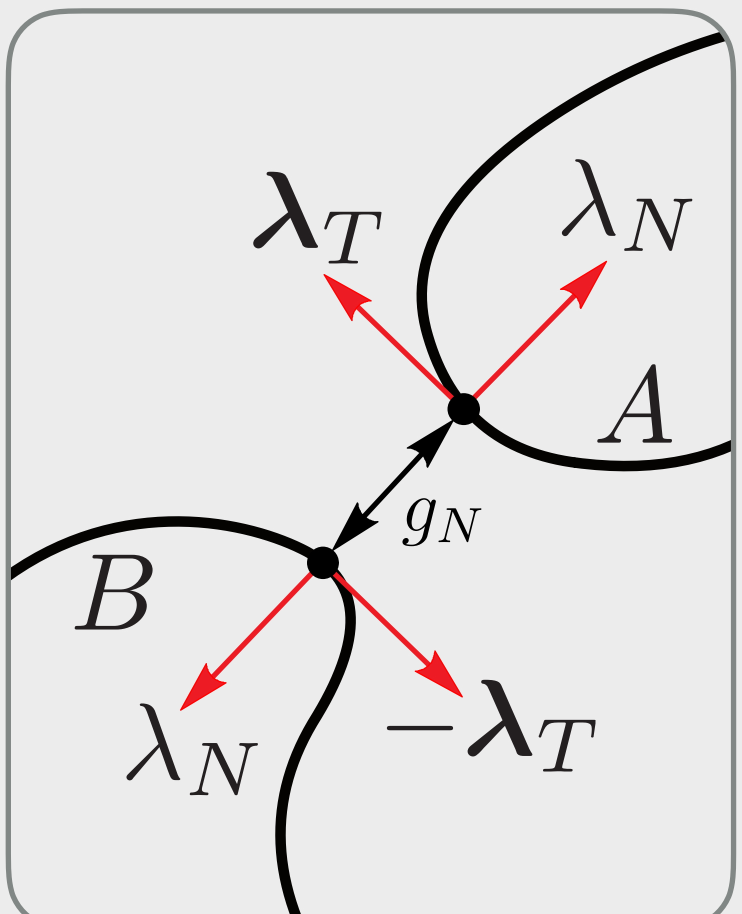
force laws as normal cone inclusions

Non-Smooth Dynamics in a Nutshell

Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

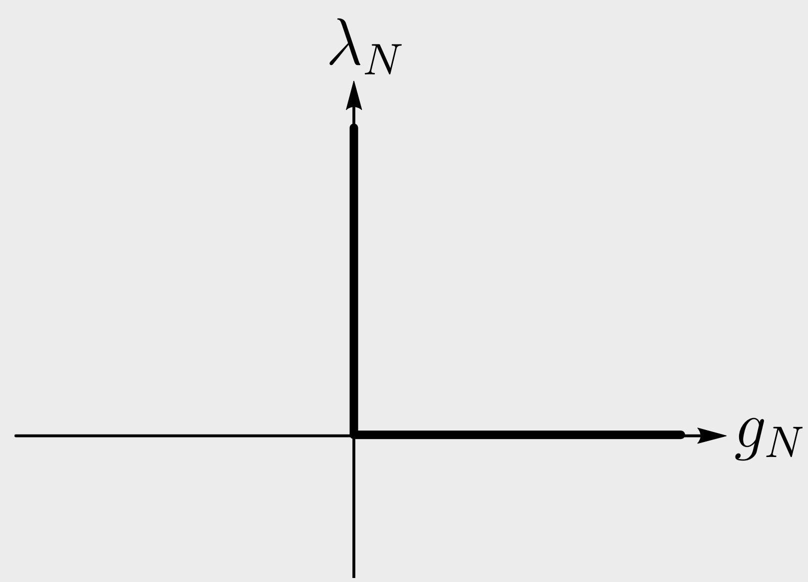
$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$



contact configuration

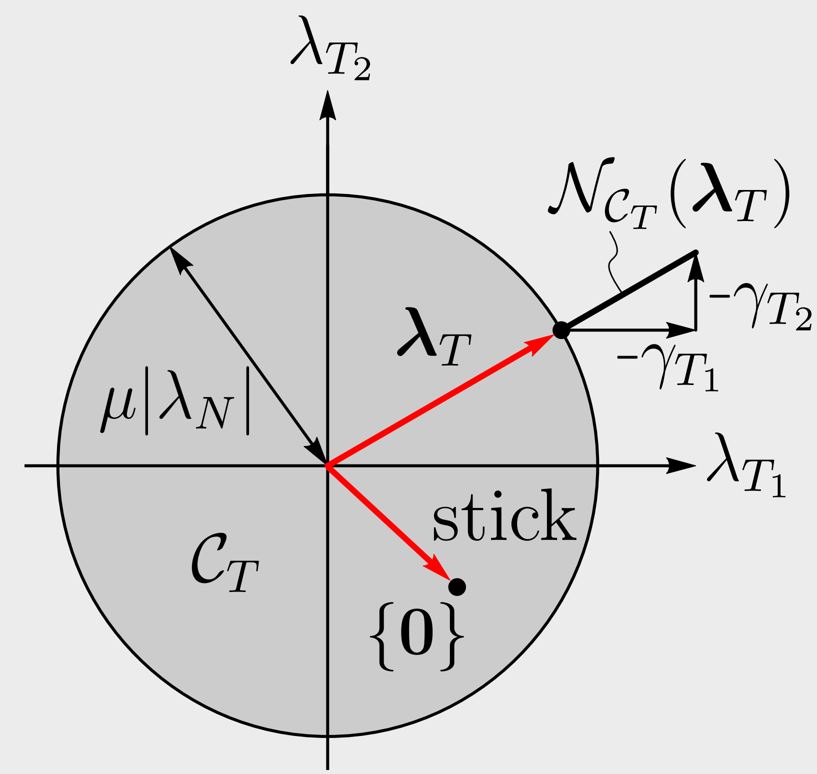
Unilateral Contact:

$$-g_N \in \mathcal{N}_{\mathbb{R}_0^+}(\lambda_N)$$



Coulomb Friction:

$$-\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(\boldsymbol{\lambda}_T)$$



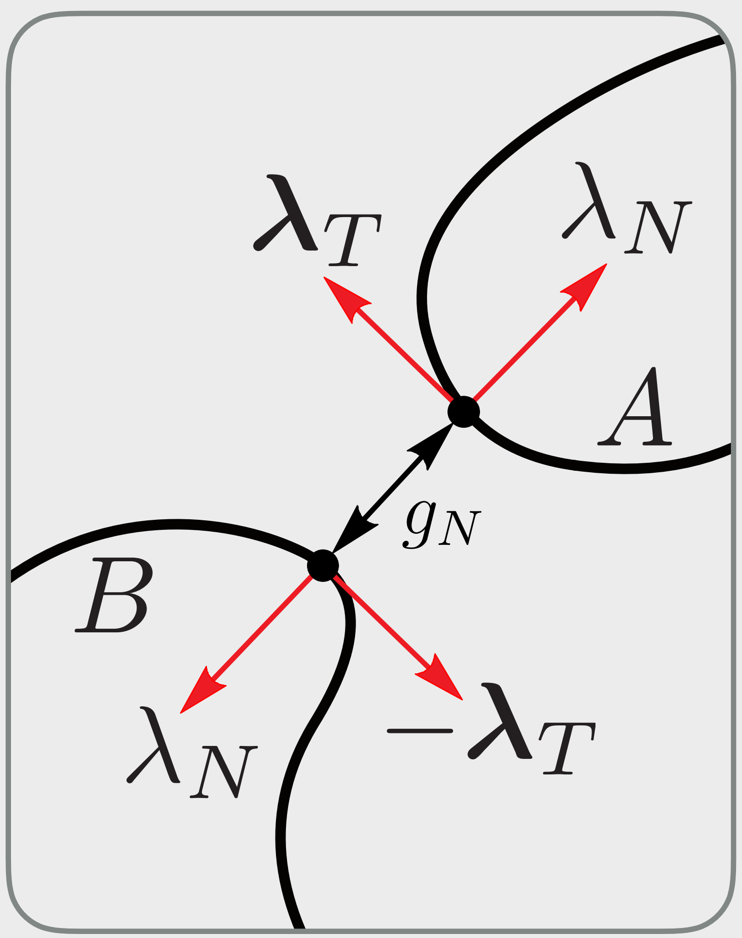
force laws as normal cone inclusions

Non-Smooth Dynamics in a Nutshell

Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\;\dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\;\mathbf{u}$$



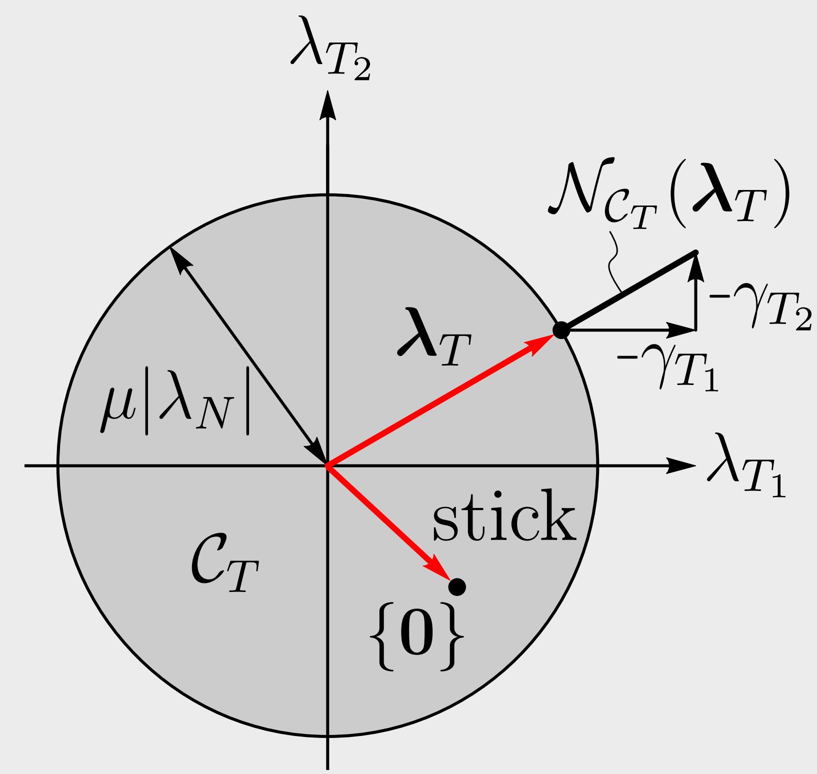
contact configuration

Coulomb Friction:

$$-\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(\boldsymbol{\lambda}_T)$$

Proximal Point Equation:

$$\boldsymbol{\lambda}_T = \mathrm{prox}_{\mathcal{C}_T}^{r_T}(\boldsymbol{\lambda}_T - r_T\boldsymbol{\gamma}_T)$$



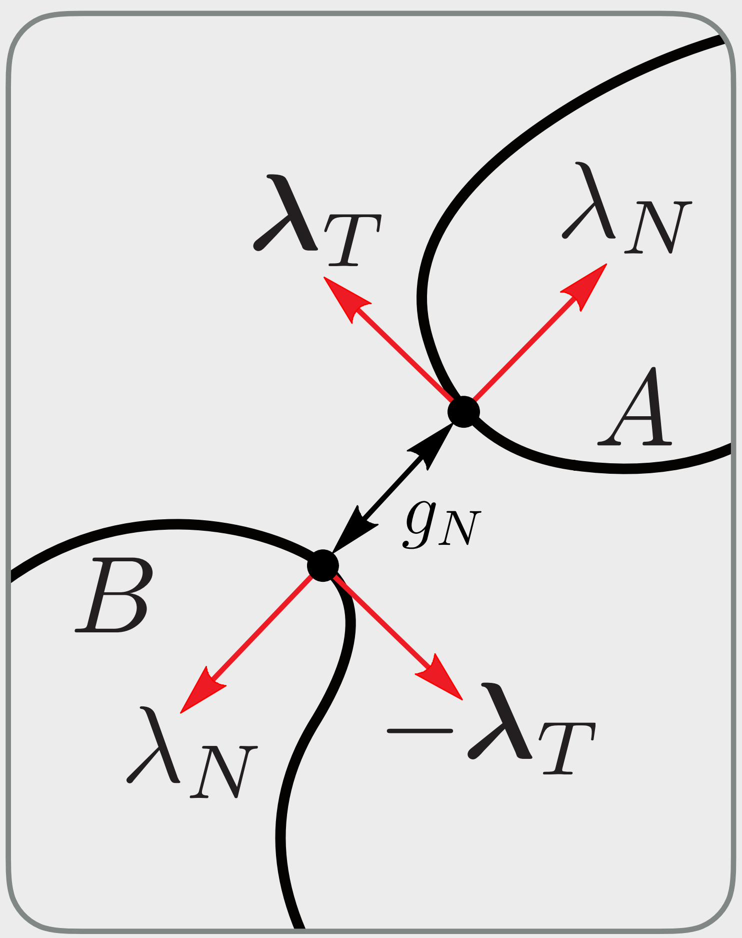
force laws as normal cone inclusions

Non-Smooth Dynamics in a Nutshell

Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$



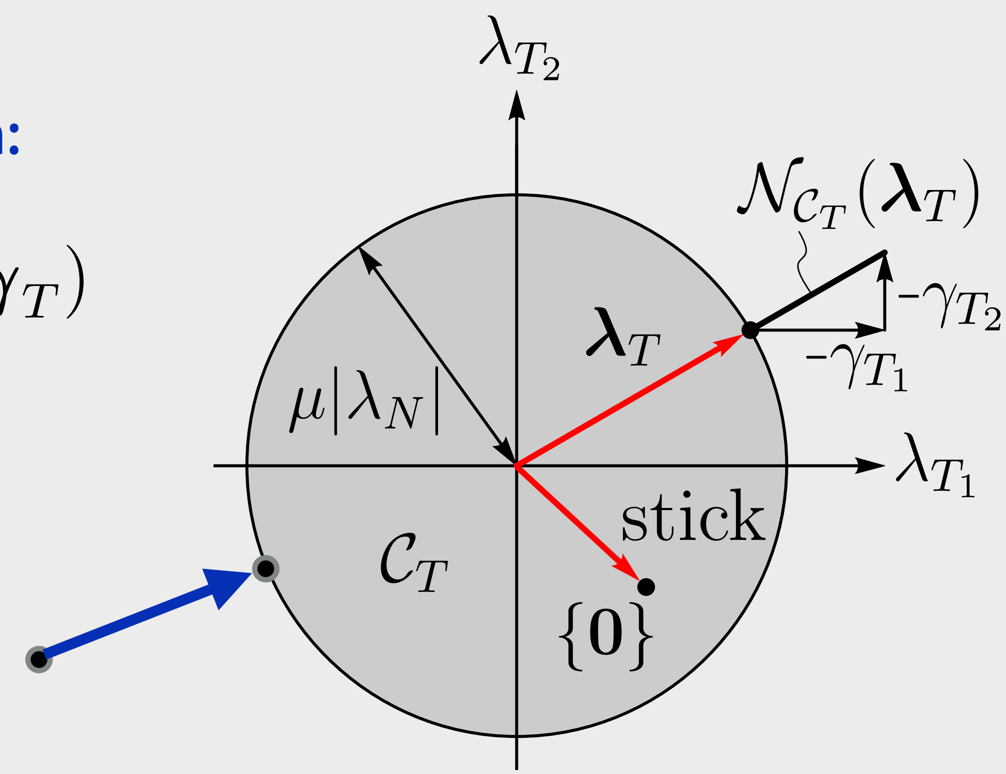
contact configuration

Coulomb Friction:

$$-\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(\boldsymbol{\lambda}_T)$$

Proximal Point Equation:

$$\boldsymbol{\lambda}_T = \mathrm{prox}_{\mathcal{C}_T}^{r_T}(\boldsymbol{\lambda}_T - r_T\boldsymbol{\gamma}_T)$$



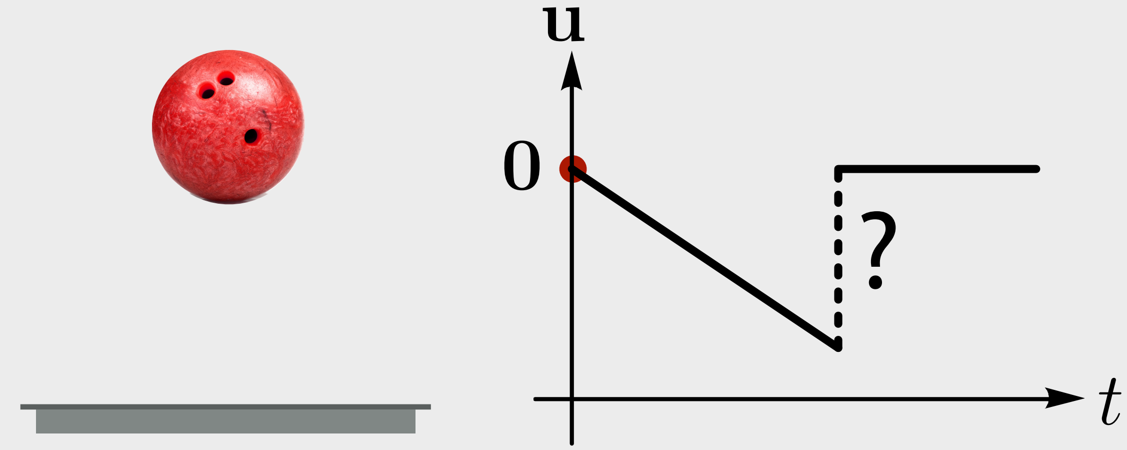
force laws as normal cone inclusions

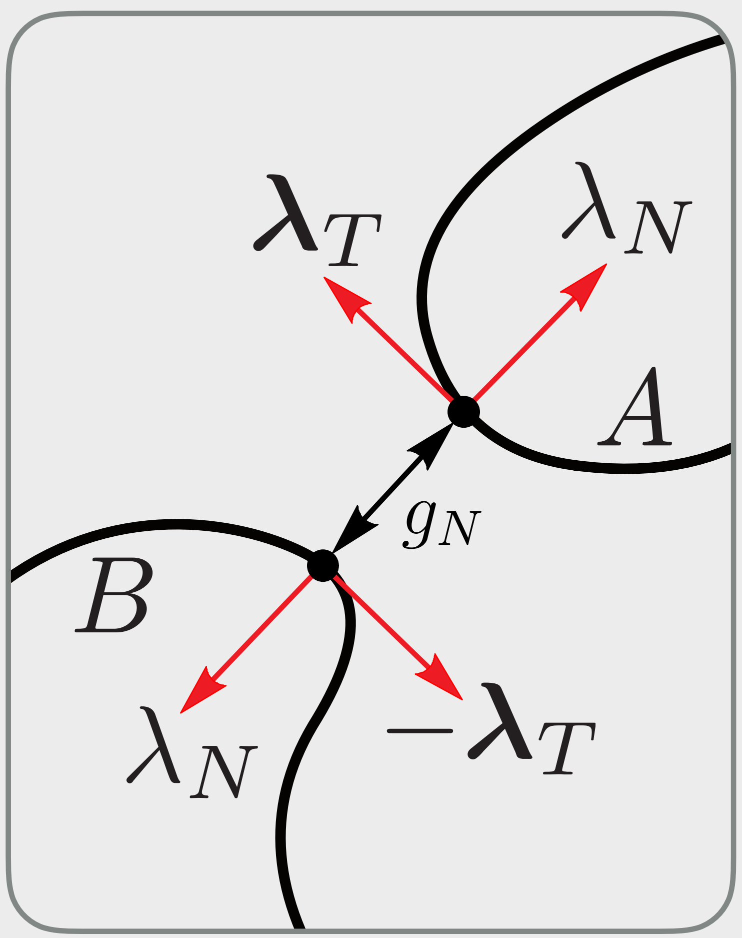
Non-Smooth Dynamics in a Nutshell

Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$





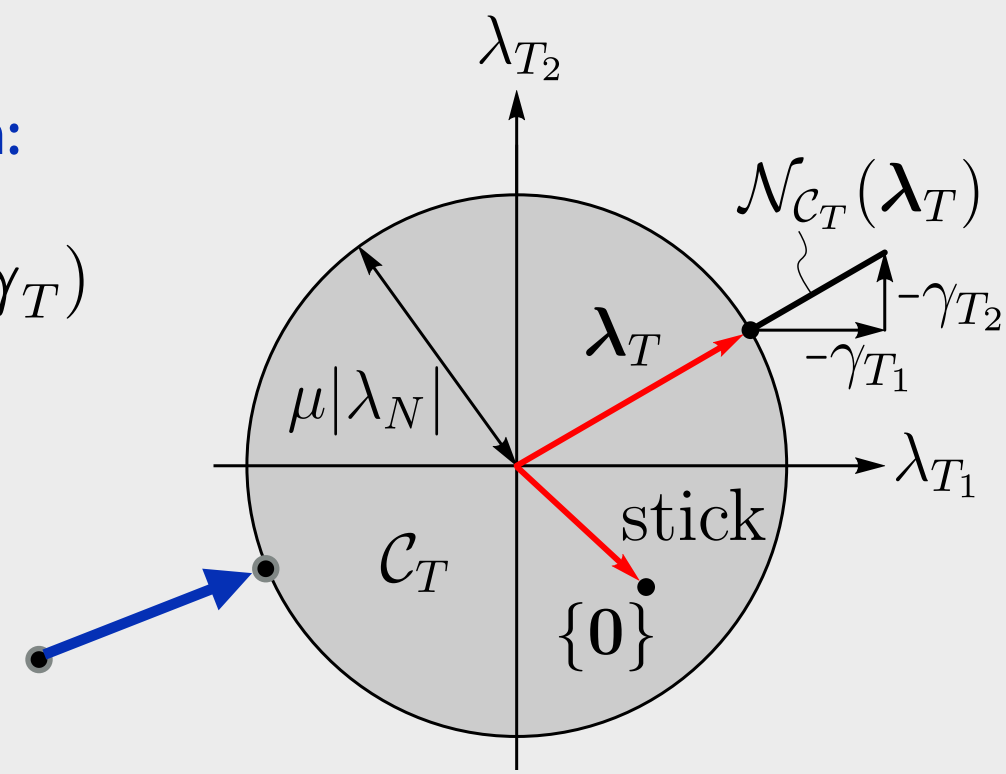
contact configuration

Coulomb Friction:

$$-\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(\boldsymbol{\lambda}_T)$$

Proximal Point Equation:

$$\boldsymbol{\lambda}_T = \mathrm{prox}_{\mathcal{C}_T}^{r_T}(\boldsymbol{\lambda}_T - r_T\boldsymbol{\gamma}_T)$$



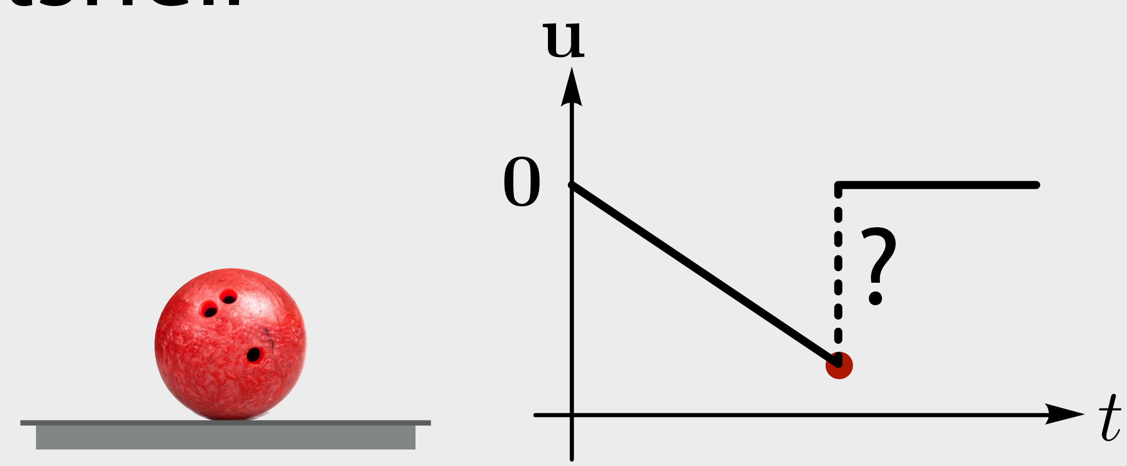
force laws as normal cone inclusions

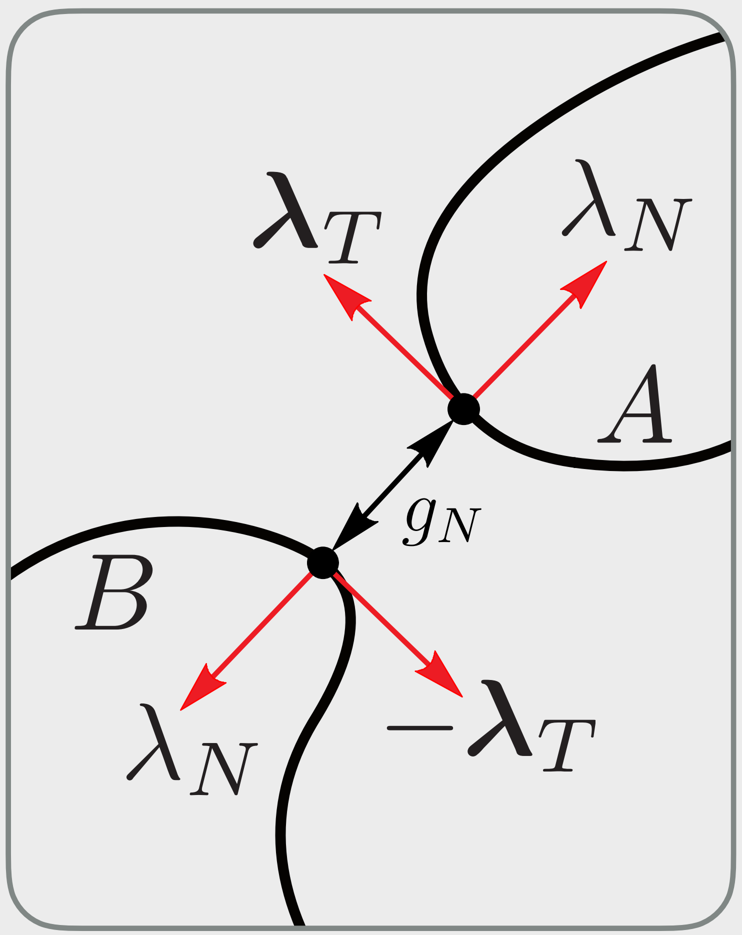
Non-Smooth Dynamics in a Nutshell

Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$





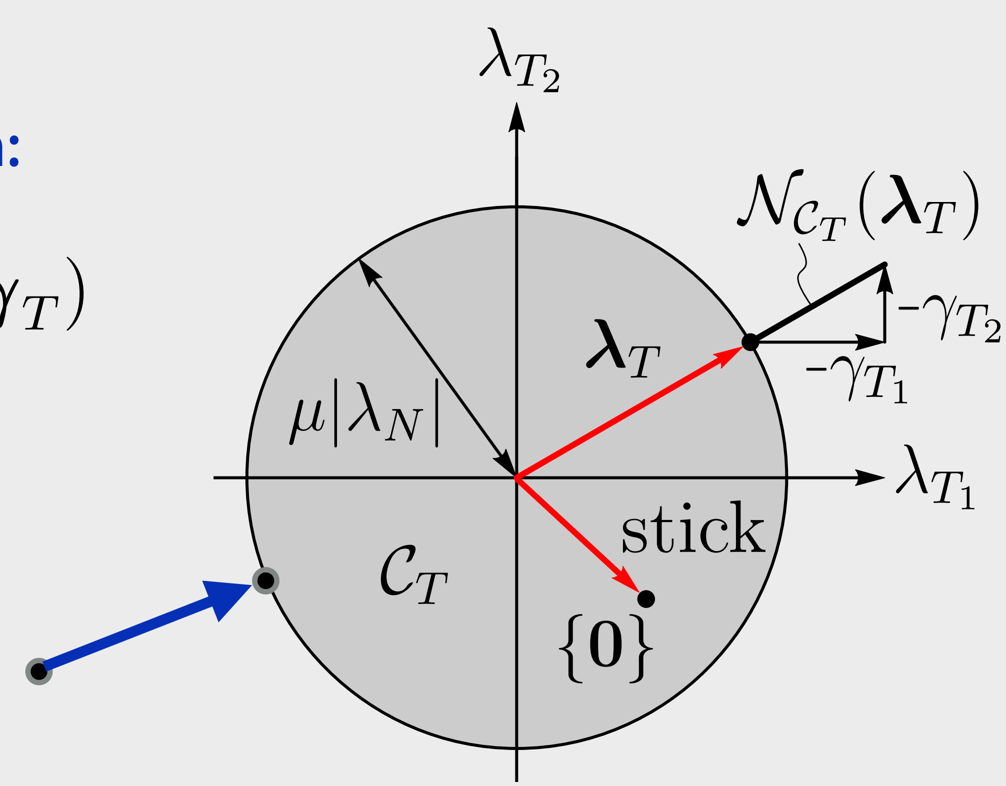
contact configuration

Coulomb Friction:

$$-\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(\boldsymbol{\lambda}_T)$$

Proximal Point Equation:

$$\boldsymbol{\lambda}_T = \mathrm{prox}_{\mathcal{C}_T}^{r_T}(\boldsymbol{\lambda}_T - r_T\boldsymbol{\gamma}_T)$$



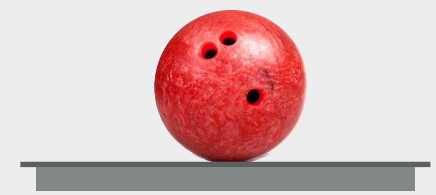
force laws as normal cone inclusions

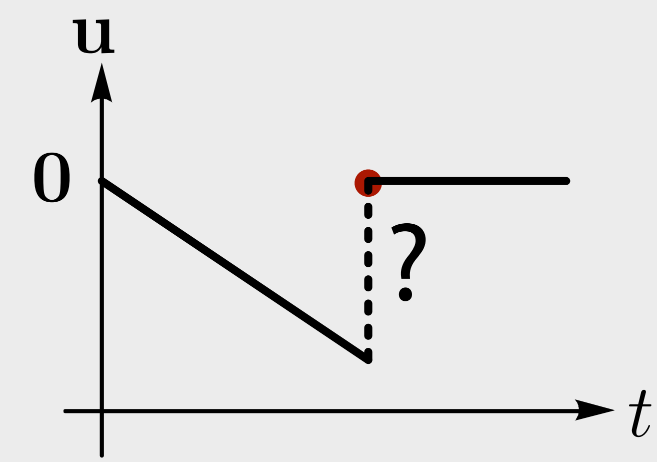
Non-Smooth Dynamics in a Nutshell

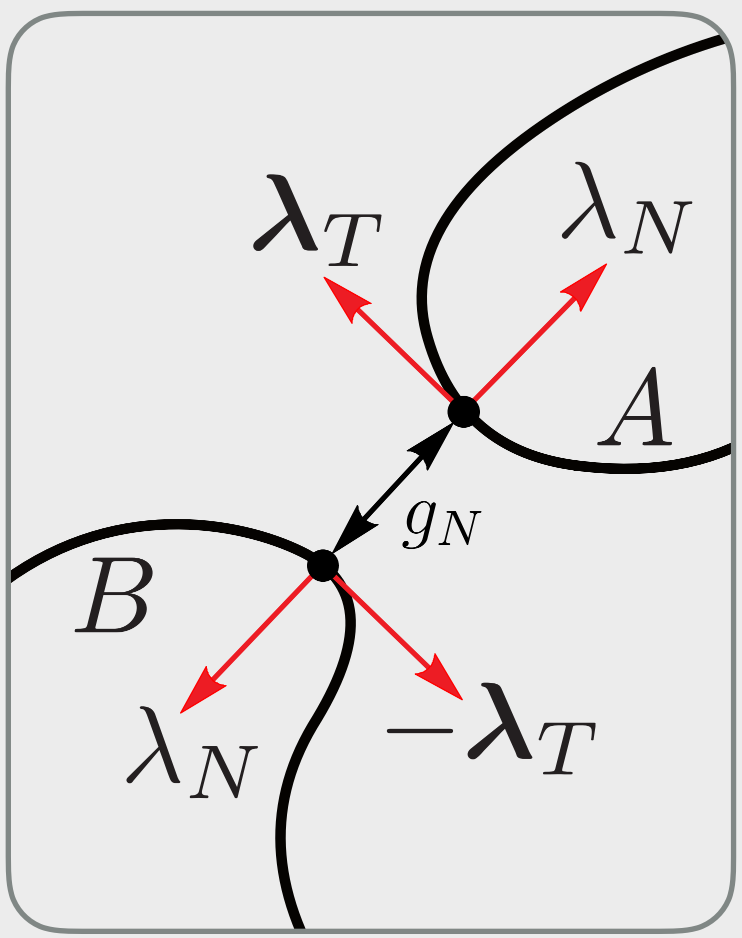
Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$







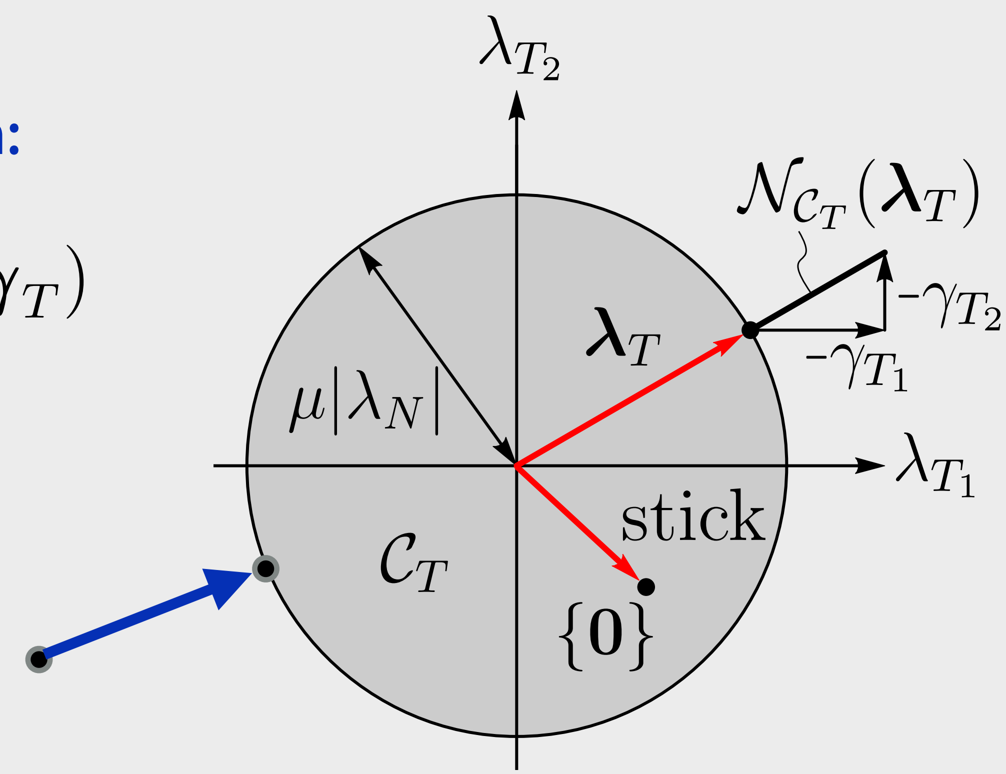
contact configuration

Coulomb Friction:

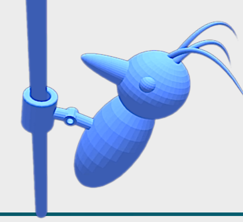
$$-\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(\boldsymbol{\lambda}_T)$$

Proximal Point Equation:

$$\boldsymbol{\lambda}_T = \mathrm{prox}_{\mathcal{C}_T}^{r_T}(\boldsymbol{\lambda}_T - r_T\boldsymbol{\gamma}_T)$$



force laws as normal cone inclusions



Non-Smooth Dynamics in a Nutshell

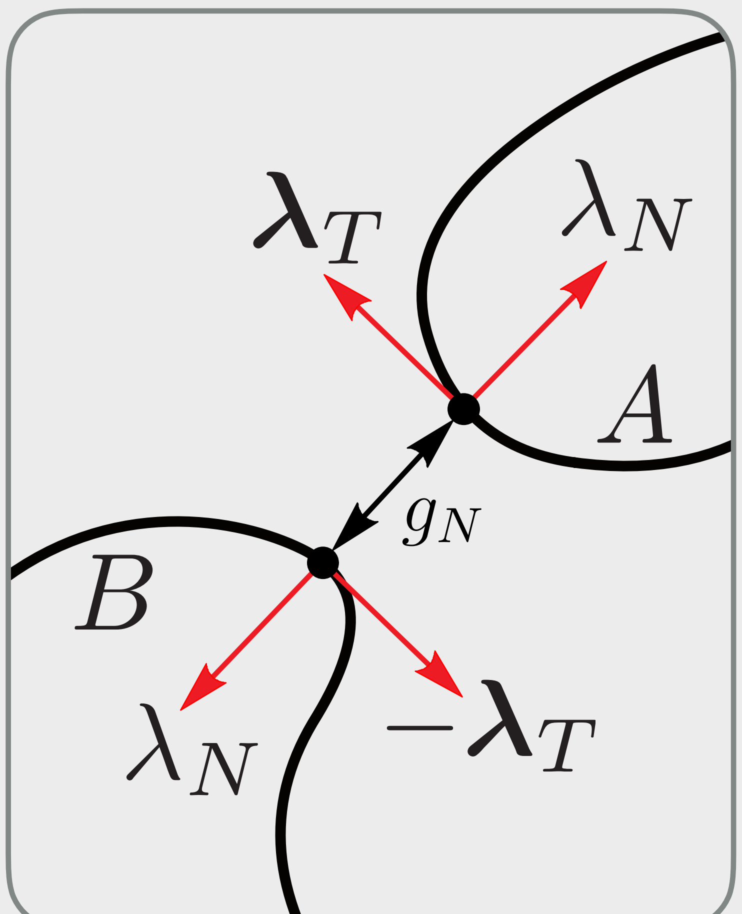
Continuous Equations of Motion:

$$\mathbf{M}(\mathbf{q},t)\ \dot{\mathbf{u}} - \mathbf{h}(\mathbf{q},\mathbf{u},t) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\lambda} = \mathbf{0}$$
$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q})\ \mathbf{u}$$

Impact Equation:

$$\mathbf{M}(\mathbf{q},t)\left(\mathbf{u}^{+} - \mathbf{u}^{-}\right) - \mathbf{W}(\mathbf{q},t)\boldsymbol{\Lambda} = \mathbf{0}$$

Newton-type impact law



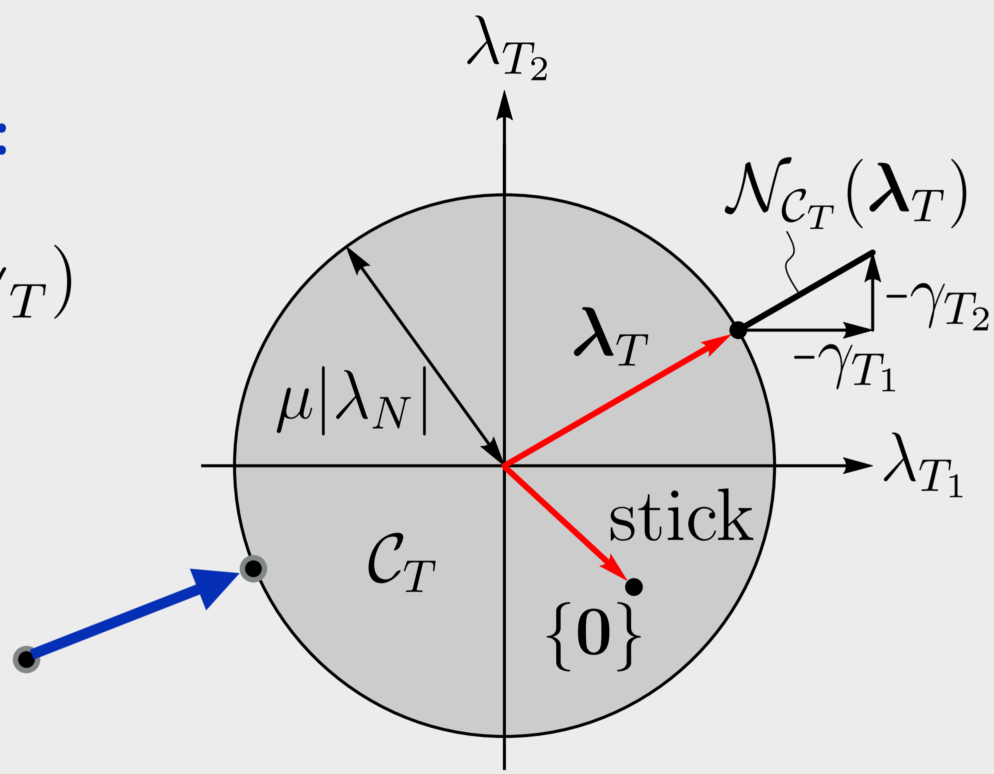
contact configuration

Coulomb Friction:

$$-\boldsymbol{\gamma}_T \in \mathcal{N}_{\mathcal{C}_T}(\boldsymbol{\lambda}_T)$$

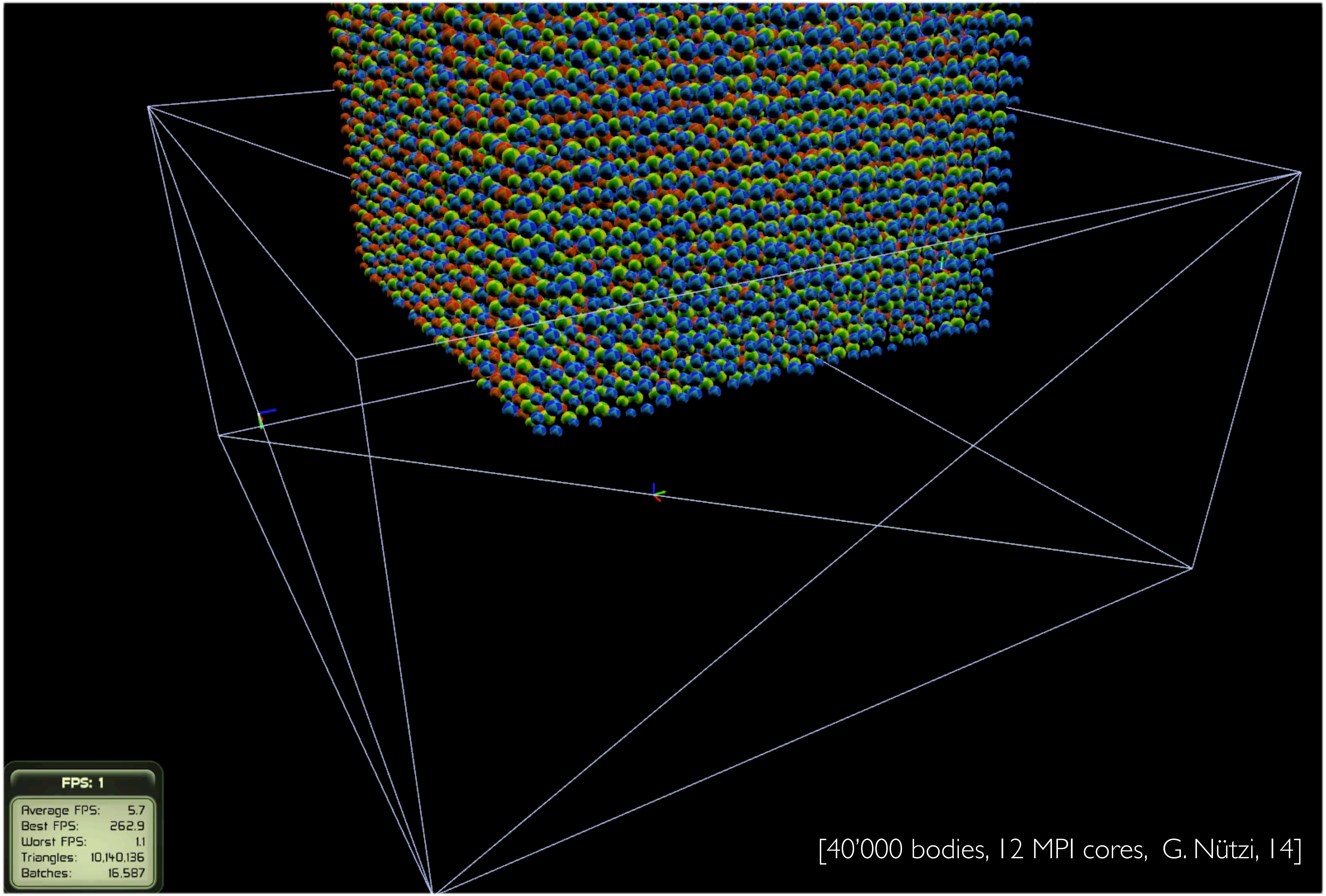
Proximal Point Equation:

$$\boldsymbol{\lambda}_T = \mathrm{prox}_{\mathcal{C}_T}^{r_T}(\boldsymbol{\lambda}_T - r_T\boldsymbol{\gamma}_T)$$

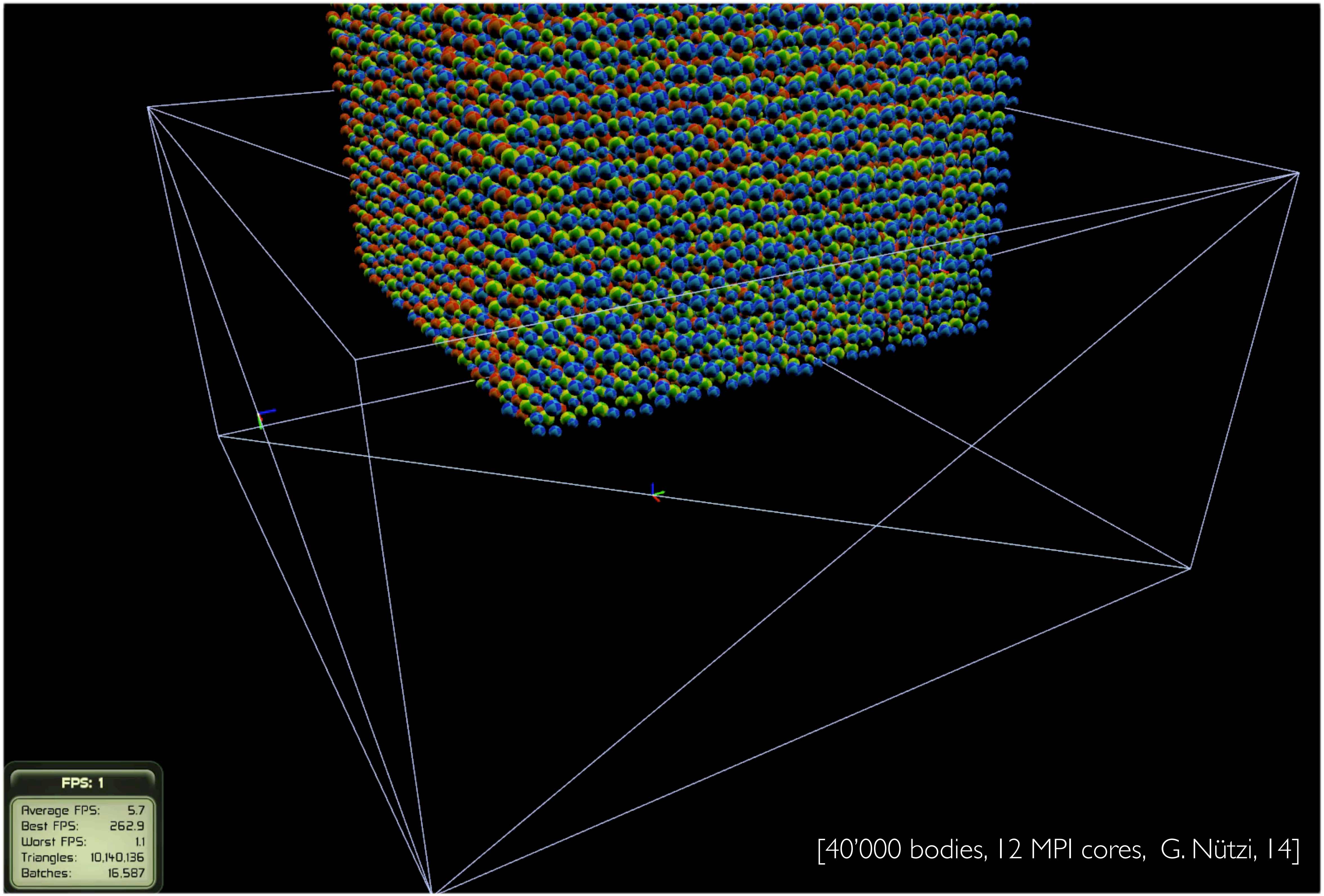


force laws as normal cone inclusions

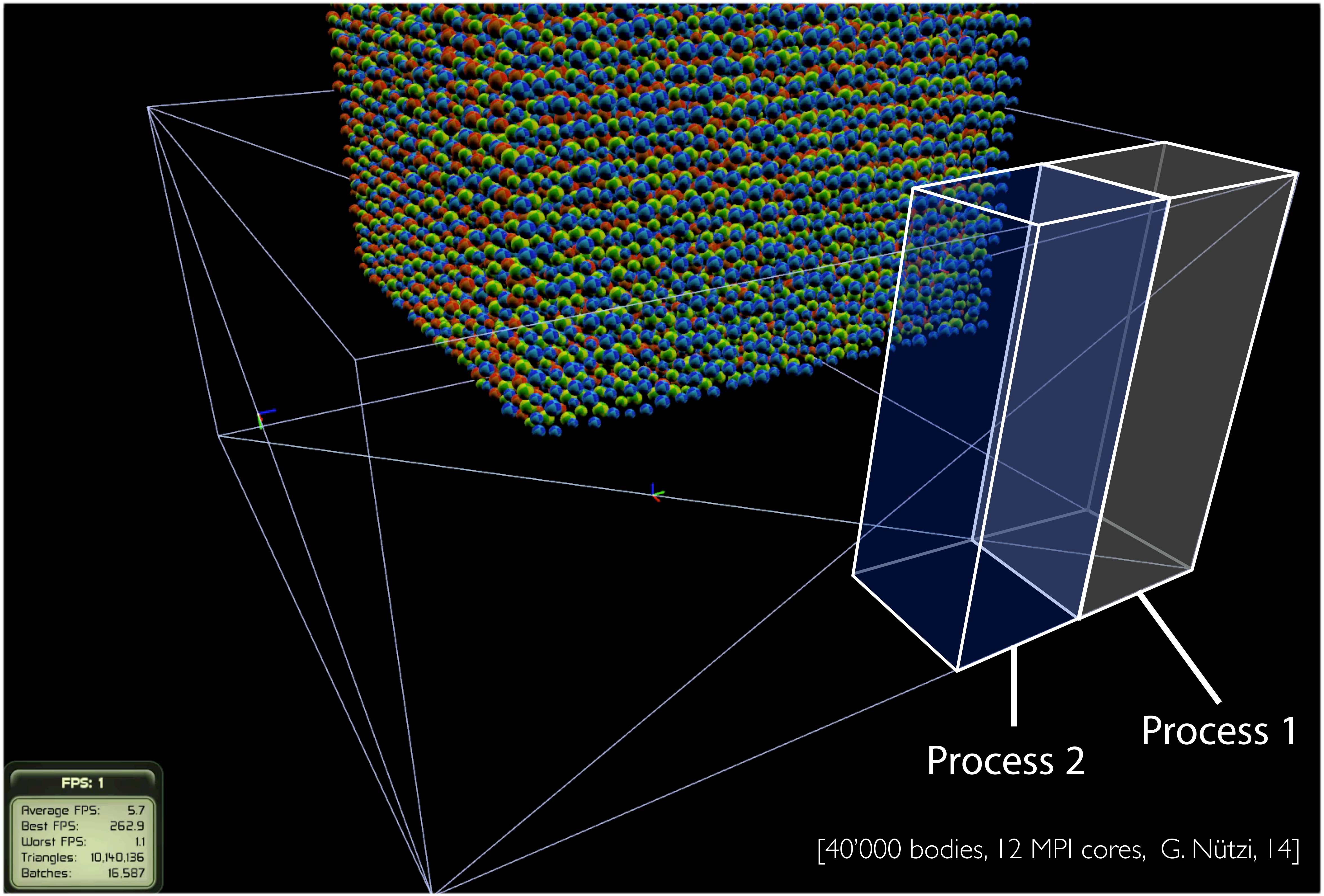
Non-Smooth Dynamics with Friction and Impacts



Non-Smooth Dynamics with Friction and Impacts



Non-Smooth Dynamics with Friction and Impacts



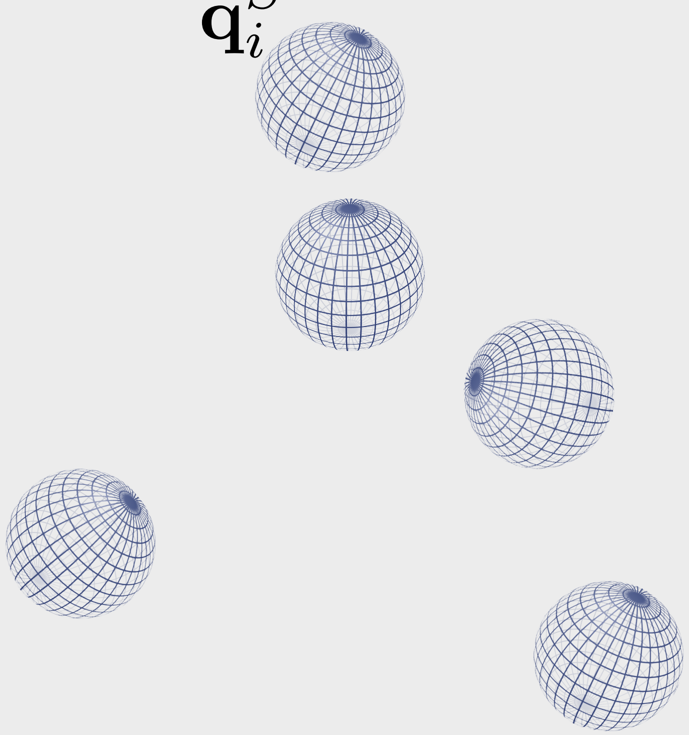
Moreau's Time-Stepping (Jean & Moreau 1986)

$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\quad\quad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

$\mathbf{q}_i^S$



Moreau's Time-Stepping (Jean & Moreau 1986)

$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

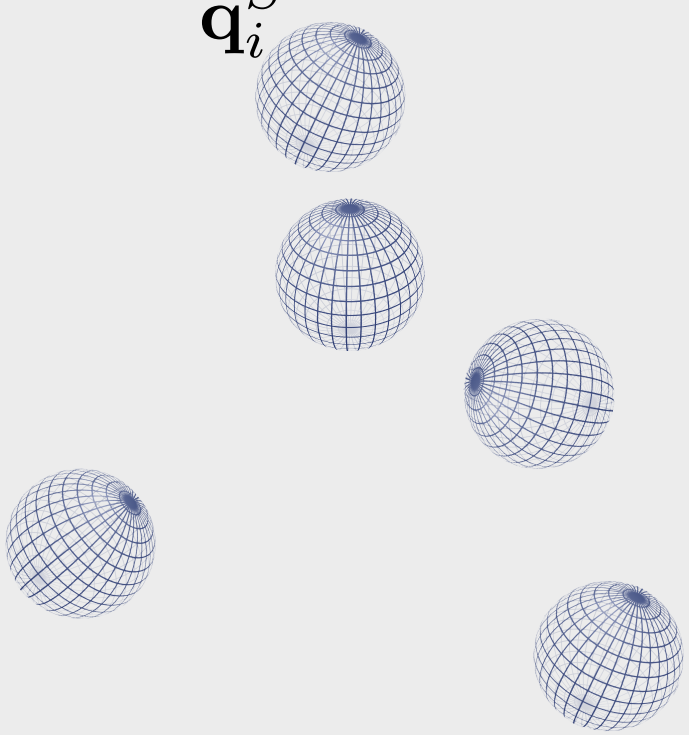
$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\qquad\qquad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

update to midpoint

$$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$$

$\mathbf{q}_i^S$



Moreau's Time-Stepping (Jean & Moreau 1986)

$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

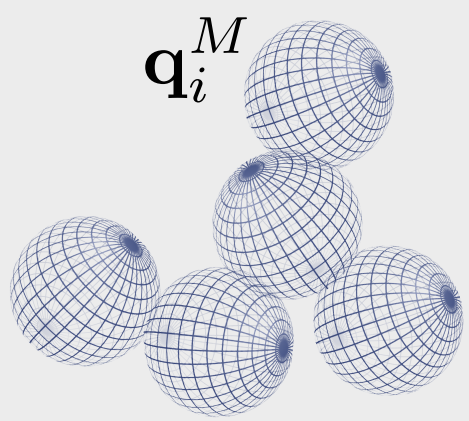
$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

update to midpoint

$$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$$

$\mathbf{q}_i^M$



Moreau's Time-Stepping (Jean & Moreau 1986)

$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\qquad\qquad\qquad\qquad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → **Collision Detection** → Solve Contact Problem → Timestep $\Delta t/2$

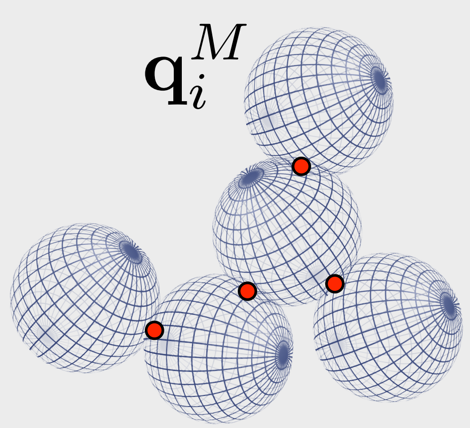
update to midpoint

$$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$$

find all contacts

collision routines:

- Spheres-Sphere
- Mesh-Sphere
- etc...



Moreau's Time-Stepping (Jean & Moreau 1986)

$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\qquad\qquad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

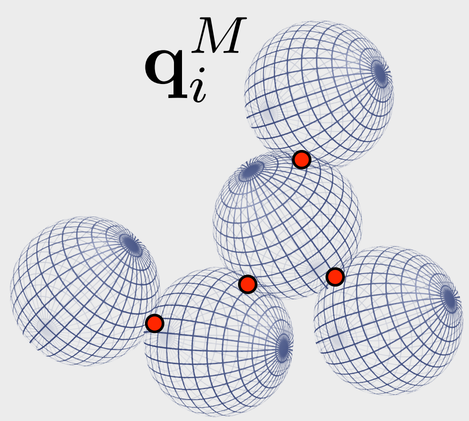
Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

Timestep $\Delta t/2$

update to midpoint

$$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$$

$\mathbf{q}_i^M$



Collision Detection

find all contacts

collision routines:

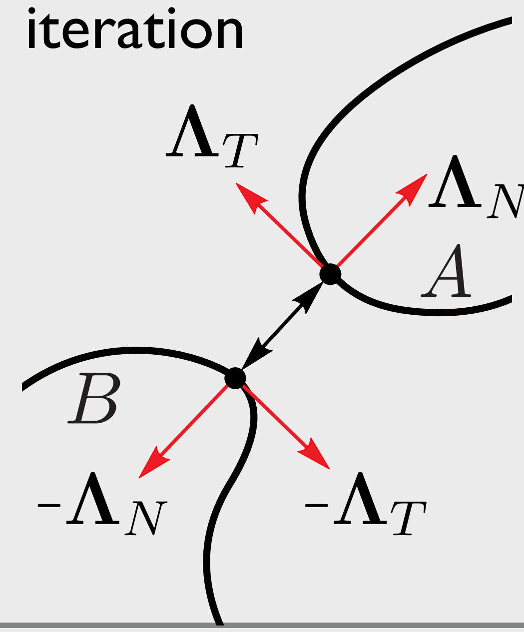
- Spheres-Sphere
- Mesh-Sphere
- etc...

Solve Contact Problem

find percussions and velocities: $\mathbf{u}_i^E, \mathbf{\Lambda}$

contact graph iteration:

- proj. Jacobi iteration
- proj. Gauss-Seidel iteration



Moreau's Time-Stepping (Jean & Moreau 1986)

$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\qquad\qquad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → **Collision Detection** → **Solve Contact Problem** → **Timestep $\Delta t/2$**

update to midpoint

$$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$$

find all contacts

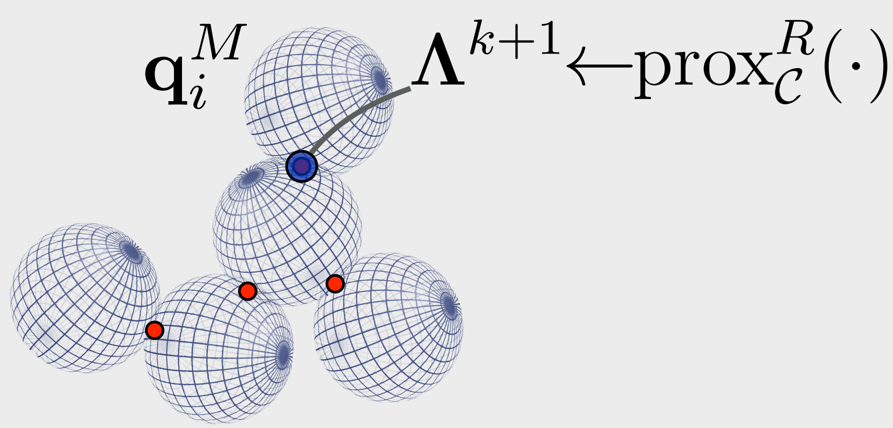
collision routines:

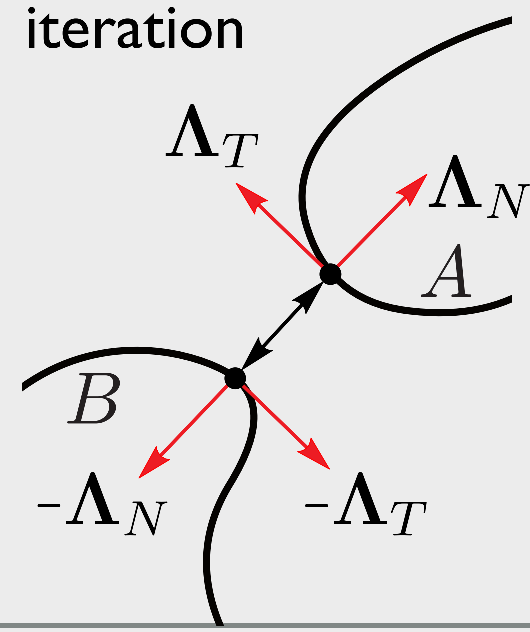
- Spheres-Sphere
- Mesh-Sphere
- etc...

find percussions and velocities: $\mathbf{u}_i^E, \boldsymbol{\Lambda}$

contact graph iteration:

- proj. Jacobi iteration
- proj. Gauss-Seidel iteration





Moreau's Time-Stepping (Jean & Moreau 1986)

$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ → $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

update to midpoint

$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$

find all contacts

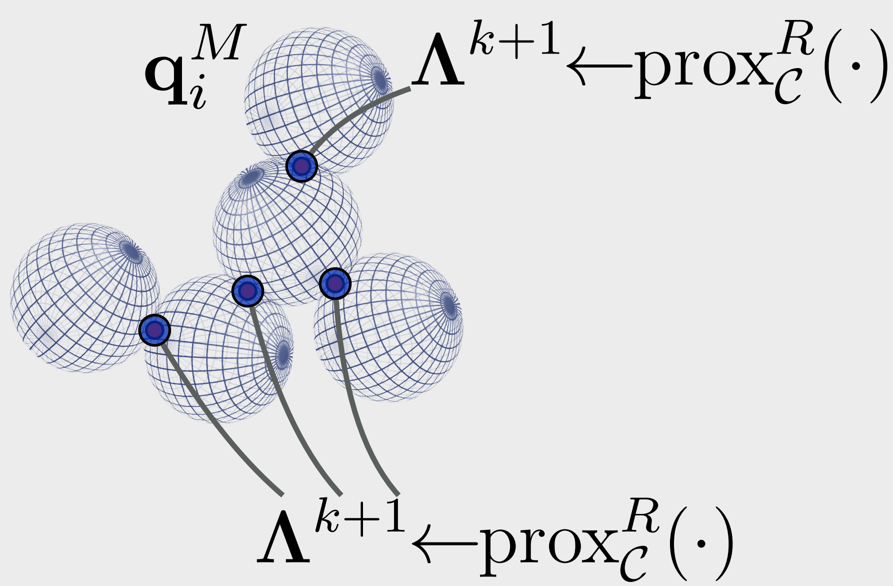
collision routines:

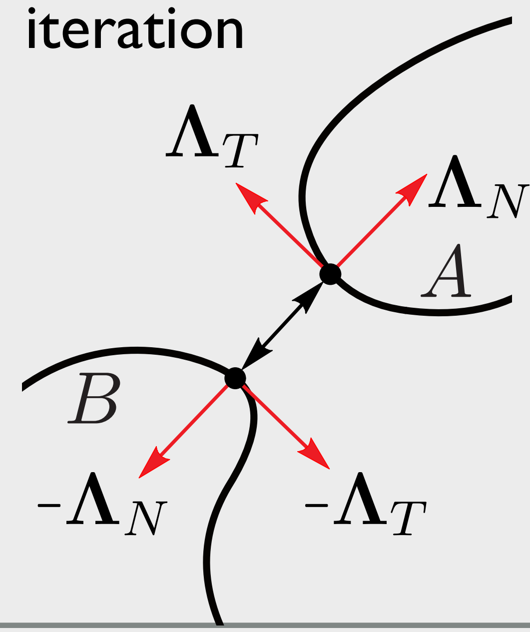
- Spheres-Sphere
- Mesh-Sphere
- etc...

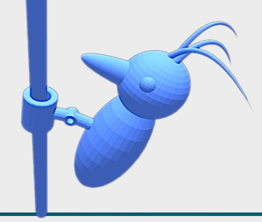
find percussions and velocities: $\mathbf{u}_i^E, \mathbf{\Lambda}$

contact graph iteration:

- proj. Jacobi iteration
- proj. Gauss-Seidel iteration







Moreau's Time-Stepping (Jean & Moreau 1986)

$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\qquad\qquad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

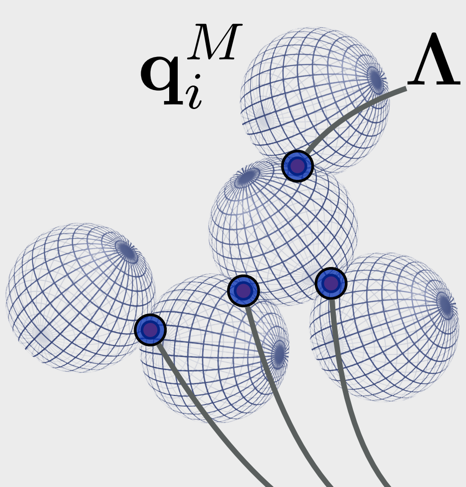
update to midpoint
$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$

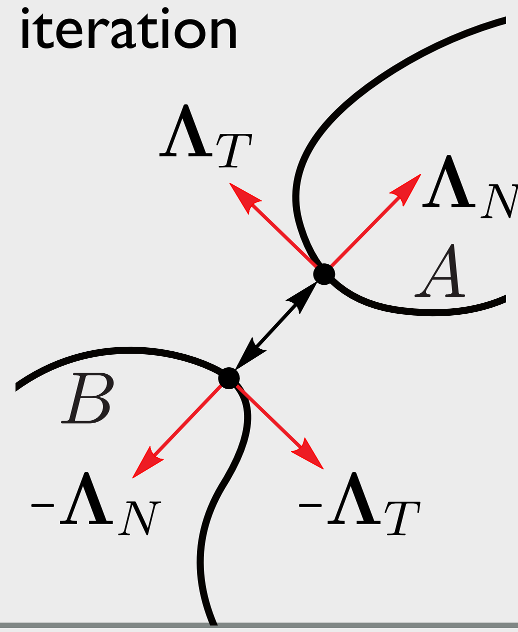
find all contacts

collision routines:
- Spheres-Sphere
- Mesh-Sphere
- etc...

find percussions and velocities: $\mathbf{u}_i^E, \boldsymbol{\Lambda}$

contact graph iteration:
- proj. Jacobi iteration
- proj. Gauss-Seidel iteration





Moreau's Time-Stepping (Jean & Moreau 1986)

$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ → $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → **Collision Detection** → **Solve Contact Problem** → **Timestep $\Delta t/2$**

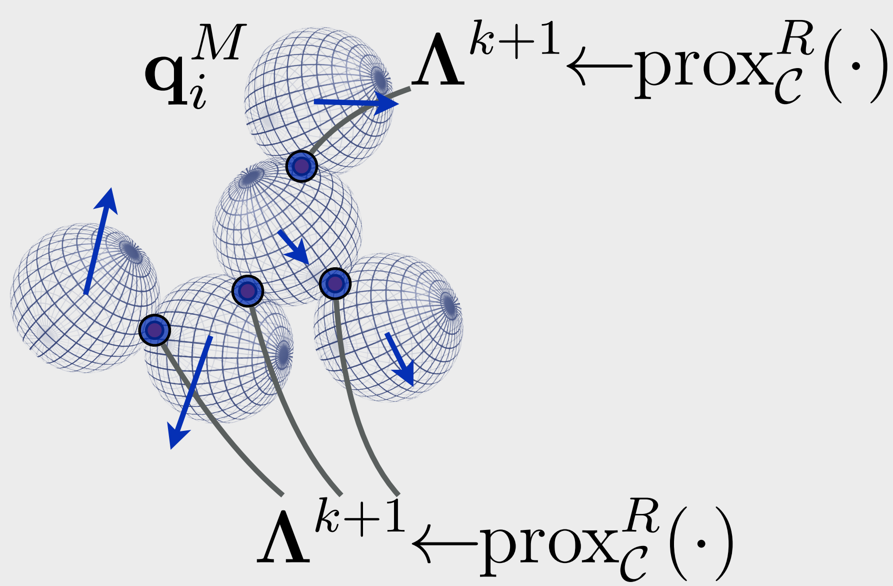
update to midpoint
$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$

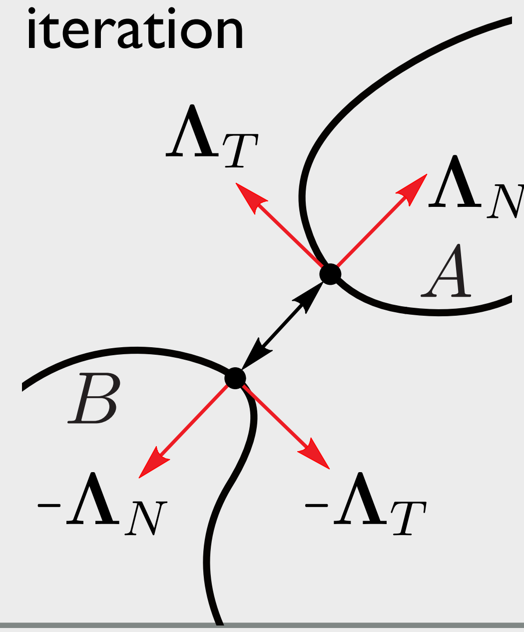
find all contacts

collision routines:
- Spheres-Sphere
- Mesh-Sphere
- etc...

find percussions and velocities: $\mathbf{u}_i^E, \boldsymbol{\Lambda}$

contact graph iteration:
- proj. Jacobi iteration
- proj. Gauss-Seidel iteration





Moreau's Time-Stepping (Jean & Moreau 1986)

$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ → $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → **Collision Detection** → **Solve Contact Problem** → **Timestep $\Delta t/2$**

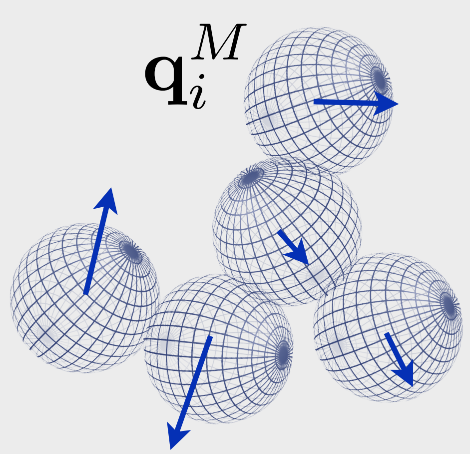
update to midpoint
$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$

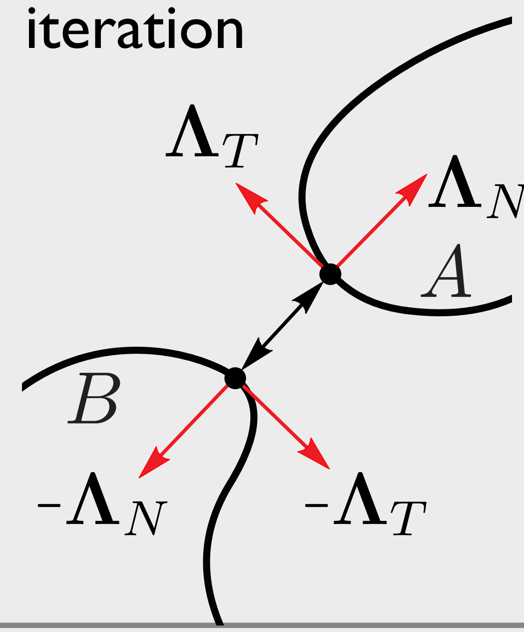
find all contacts

collision routines:
- Spheres-Sphere
- Mesh-Sphere
- etc...

find percussions and velocities: $\mathbf{u}_i^E, \boldsymbol{\Lambda}$

contact graph iteration:
- proj. Jacobi iteration
- proj. Gauss-Seidel iteration





Moreau's Time-Stepping (Jean & Moreau 1986)

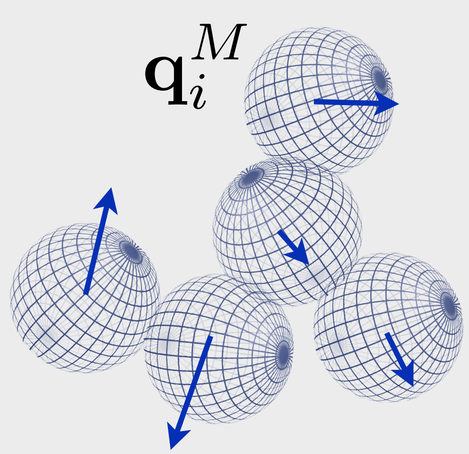
$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\qquad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

Timestep $\Delta t/2$

update to midpoint
$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$



Collision Detection

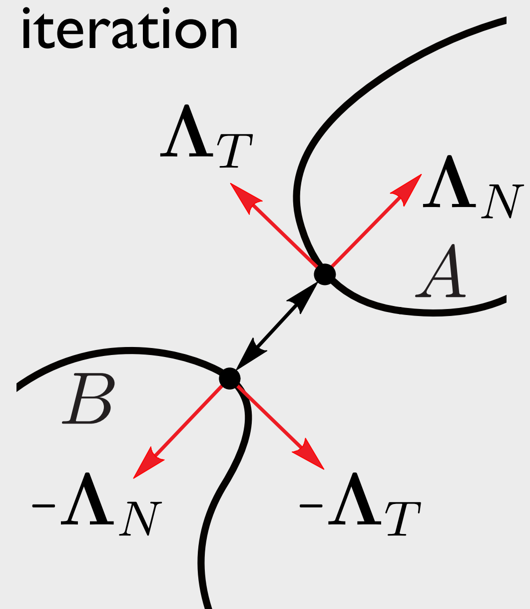
find all contacts

collision routines:
- Spheres-Sphere
- Mesh-Sphere
- etc...

Solve Contact Problem

find percussions and velocities: $\mathbf{u}_i^E, \mathbf{\Lambda}$

contact graph iteration:
- proj. Jacobi iteration
- proj. Gauss-Seidel iteration



Timestep $\Delta t/2$

update to end point
$(\mathbf{\Lambda}, \mathbf{u}_i^E, \mathbf{q}_i^M) \rightarrow \mathbf{q}_i^E$

Moreau's Time-Stepping (Jean & Moreau 1986)

$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ $\qquad$ $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → Collision Detection → Solve Contact Problem → Timestep $\Delta t/2$

Timestep $\Delta t/2$

update to midpoint
$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$

Collision Detection

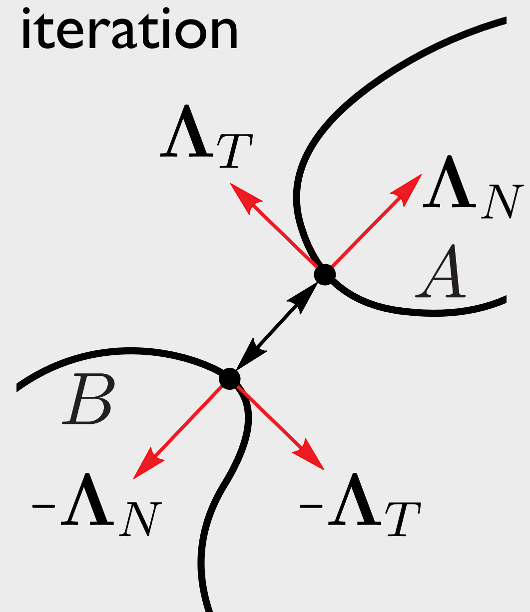
find all contacts

collision routines:
- Spheres-Sphere
- Mesh-Sphere
- etc...

Solve Contact Problem

find percussions and velocities: $\mathbf{u}_i^E, \mathbf{\Lambda}$

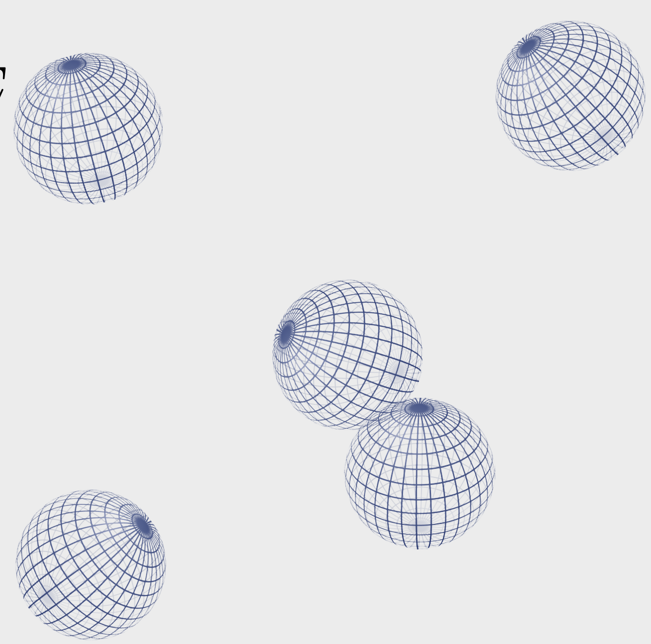
contact graph iteration:
- proj. Jacobi iteration
- proj. Gauss-Seidel iteration



Timestep $\Delta t/2$

update to end point
$(\mathbf{\Lambda}, \mathbf{u}_i^E, \mathbf{q}_i^M) \rightarrow \mathbf{q}_i^E$

$\mathbf{q}_i^E$



Moreau's Time-Stepping (Jean & Moreau 1986)

$$t_S \xrightarrow{\Delta t/2} t_M \xrightarrow{\Delta t/2} t_E$$

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$ — body communication — $(\mathbf{q}_i^E, \mathbf{u}_i^E)$

Timestep $\Delta t/2$ → **Collision Detection** → **Solve Contact Problem** → **Timestep $\Delta t/2$**

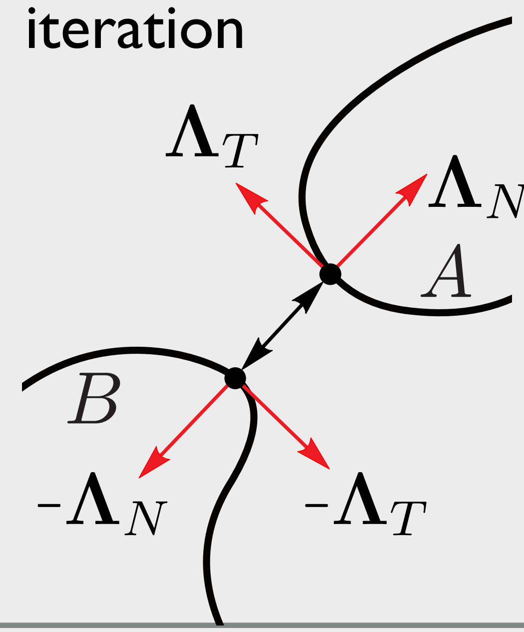
update to midpoint
$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$

find all contacts

collision routines:
- Spheres-Sphere
- Mesh-Sphere
- etc...

find percussions and velocities: $\mathbf{u}_i^E, \mathbf{\Lambda}$

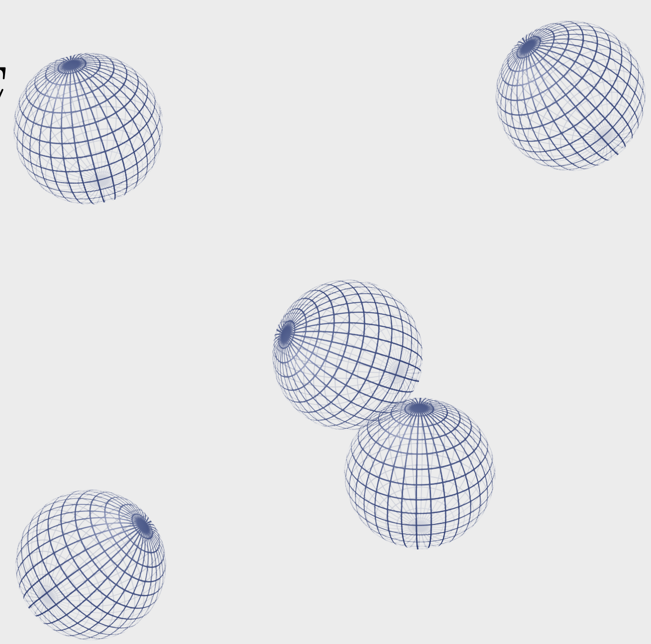
contact graph iteration:
- proj. Jacobi iteration
- proj. Gauss-Seidel iteration



communication

update to end point
$(\mathbf{\Lambda}, \mathbf{u}_i^E, \mathbf{q}_i^M) \rightarrow \mathbf{q}_i^E$

$\mathbf{q}_i^E$



Moreau's Time-Stepping (Jean & Moreau 1986)

t_S —— $\Delta t/2$ ——→ t_M —— $\Delta t/2$ ——→ t_E

$(\mathbf{q}_i^S, \mathbf{u}_i^S)$

body communication

Timestep $\Delta t/2$ → **Collision Detection** → **Solve Contact Problem** → **Timestep $\Delta t/2$**

$(\mathbf{q}_i^E, \mathbf{u}_i^E)$

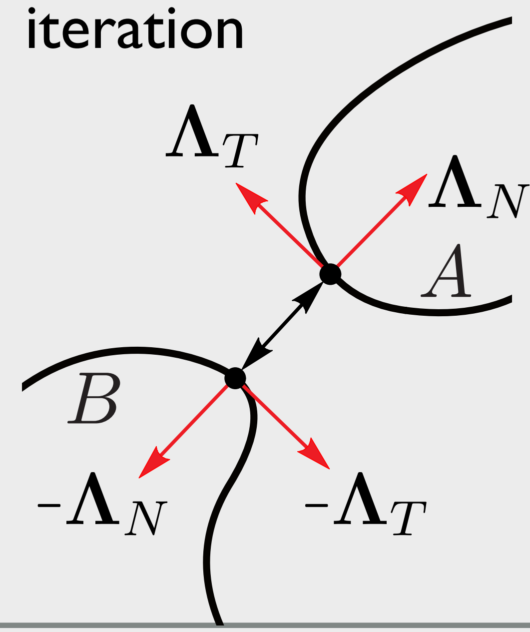
update to midpoint
$\mathbf{q}_i^S \rightarrow \mathbf{q}_i^M$

find all contacts

collision routines:
- Spheres-Sphere
- Mesh-Sphere
- etc...

find percussions and velocities: $\mathbf{u}_i^E, \mathbf{\Lambda}$

contact graph iteration:
- proj. Jacobi iteration
- proj. Gauss-Seidel iteration

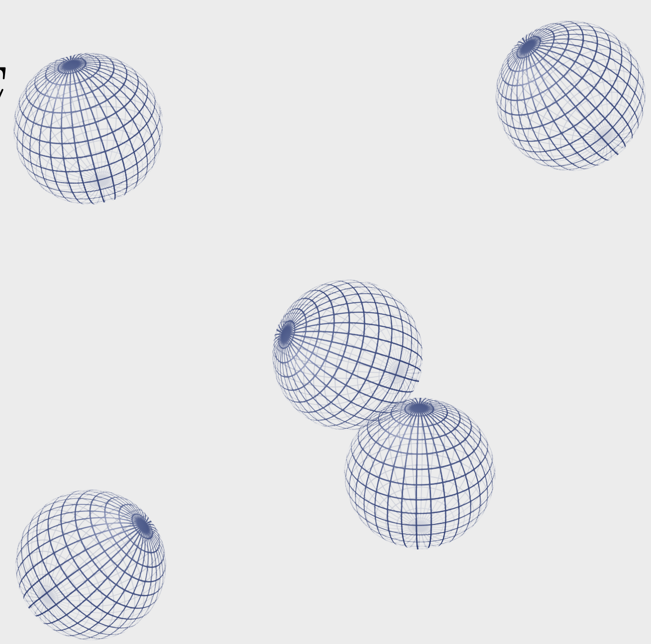


communication

update to end point
$(\mathbf{\Lambda}, \mathbf{u}_i^E, \mathbf{q}_i^M) \rightarrow \mathbf{q}_i^E$

rebuild domain decomposition

$\mathbf{q}_i^E$

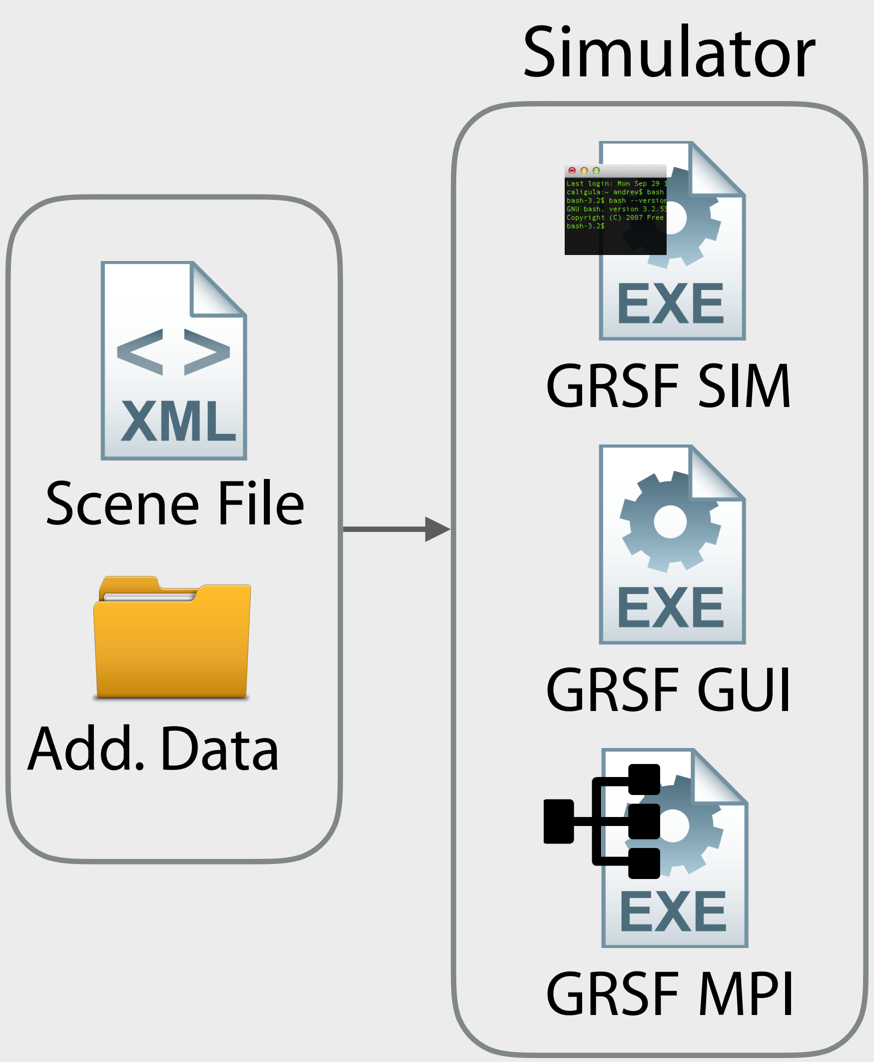


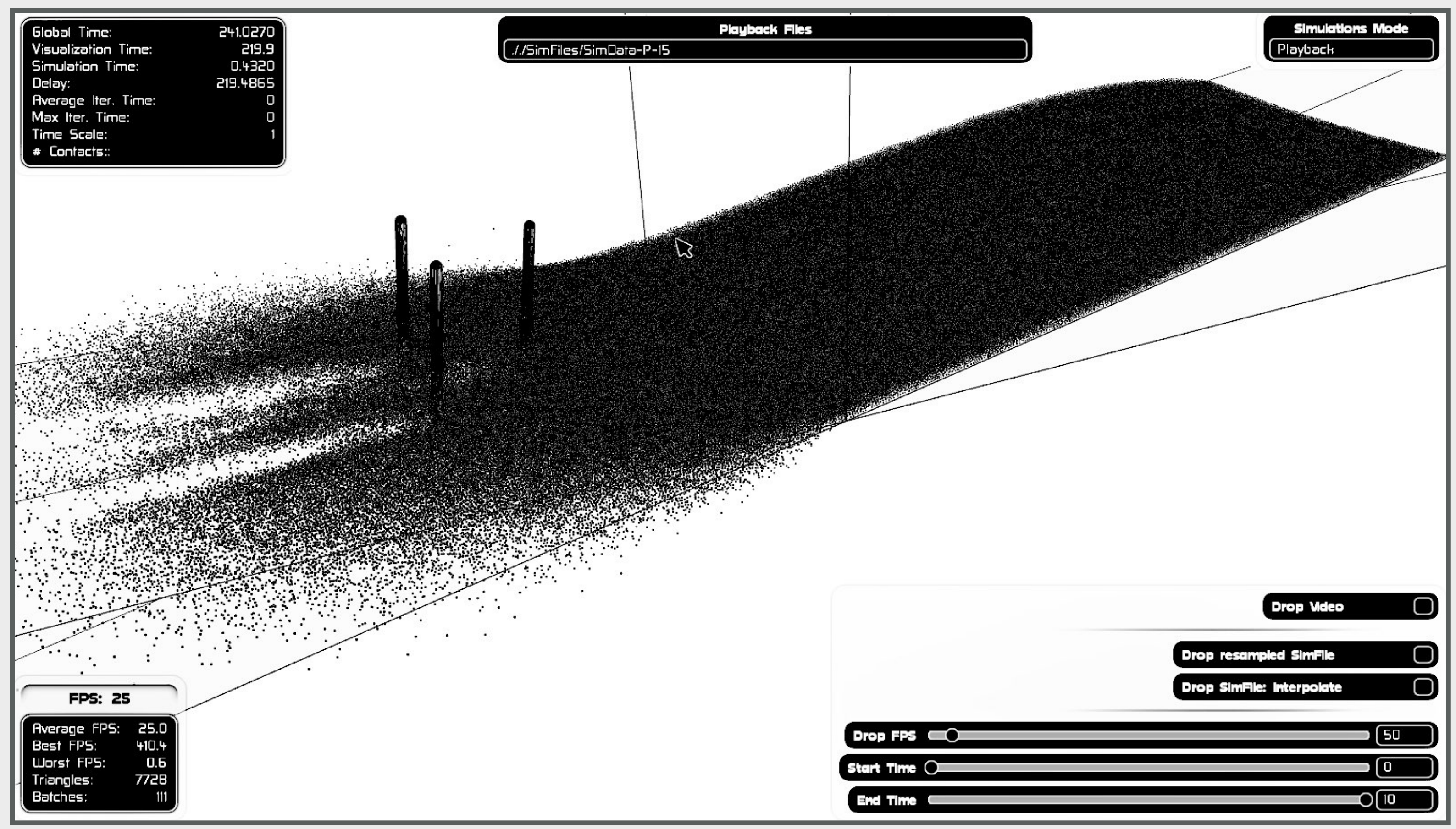
The Granular Rigid Body Simulation Framework

Language: C++11/14 will appear on **github.com**

Features: Large-Scale Non-Smooth Rigid Body Simulations

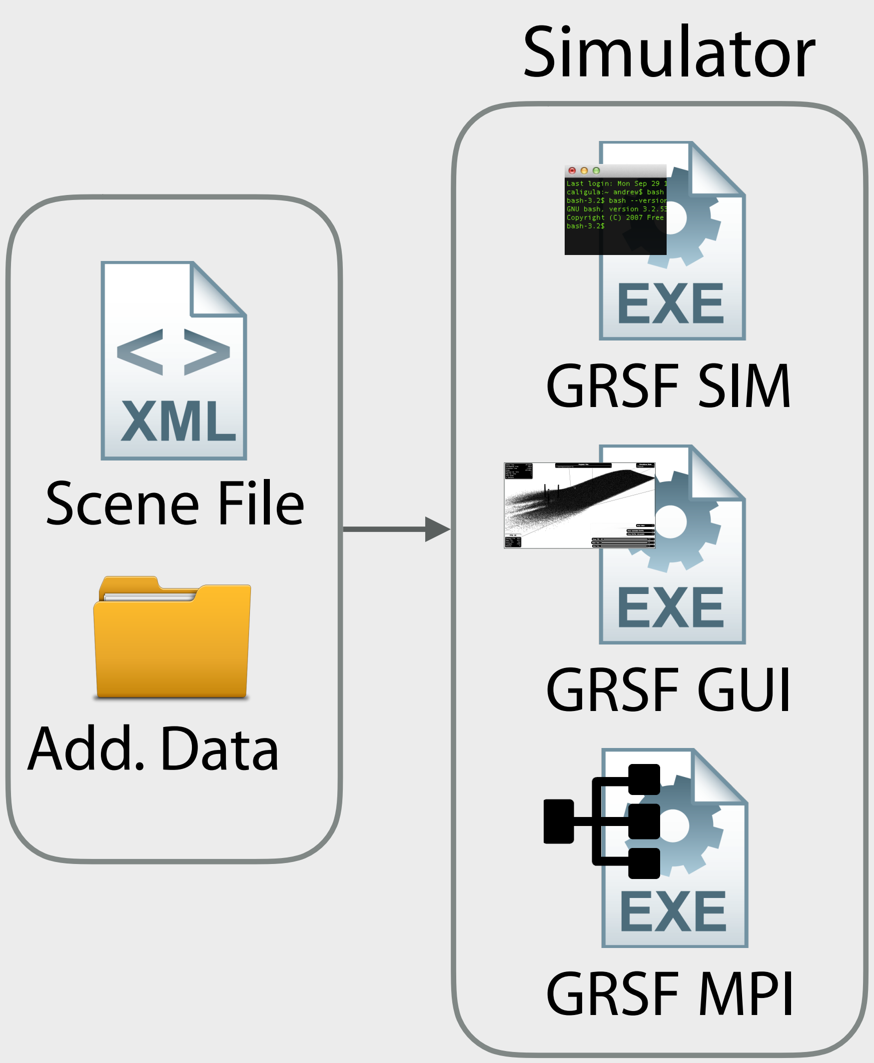
Dependencies: Eigen, boost, OpenMPI, HDF5, pugixml, Ogre, ApproxMVBB





The Granular Rigid Body Simulation Framework

Language: C++11/14 will appear on **github.com**
Features: Large-Scale Non-Smooth Rigid Body Simulations
Dependencies: Eigen, boost, OpenMPI, HDF5, pugixml, Ogre, ApproxMVBB

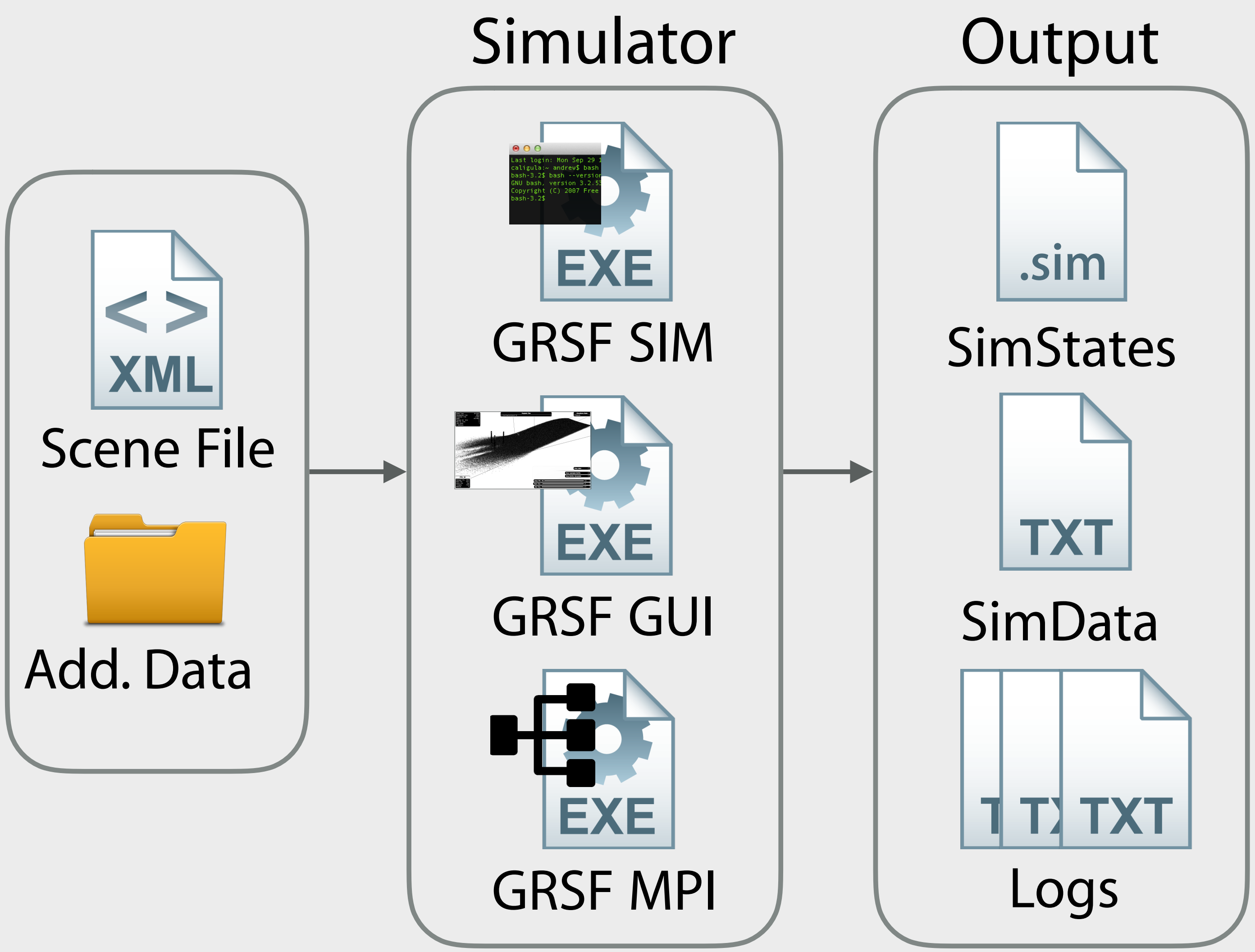


The Granular Rigid Body Simulation Framework

Language: C++11/14 will appear on **github.com**

Features: Large-Scale Non-Smooth Rigid Body Simulations

Dependencies: Eigen, boost, OpenMPI, HDF5, pugixml, Ogre, ApproxMVBB

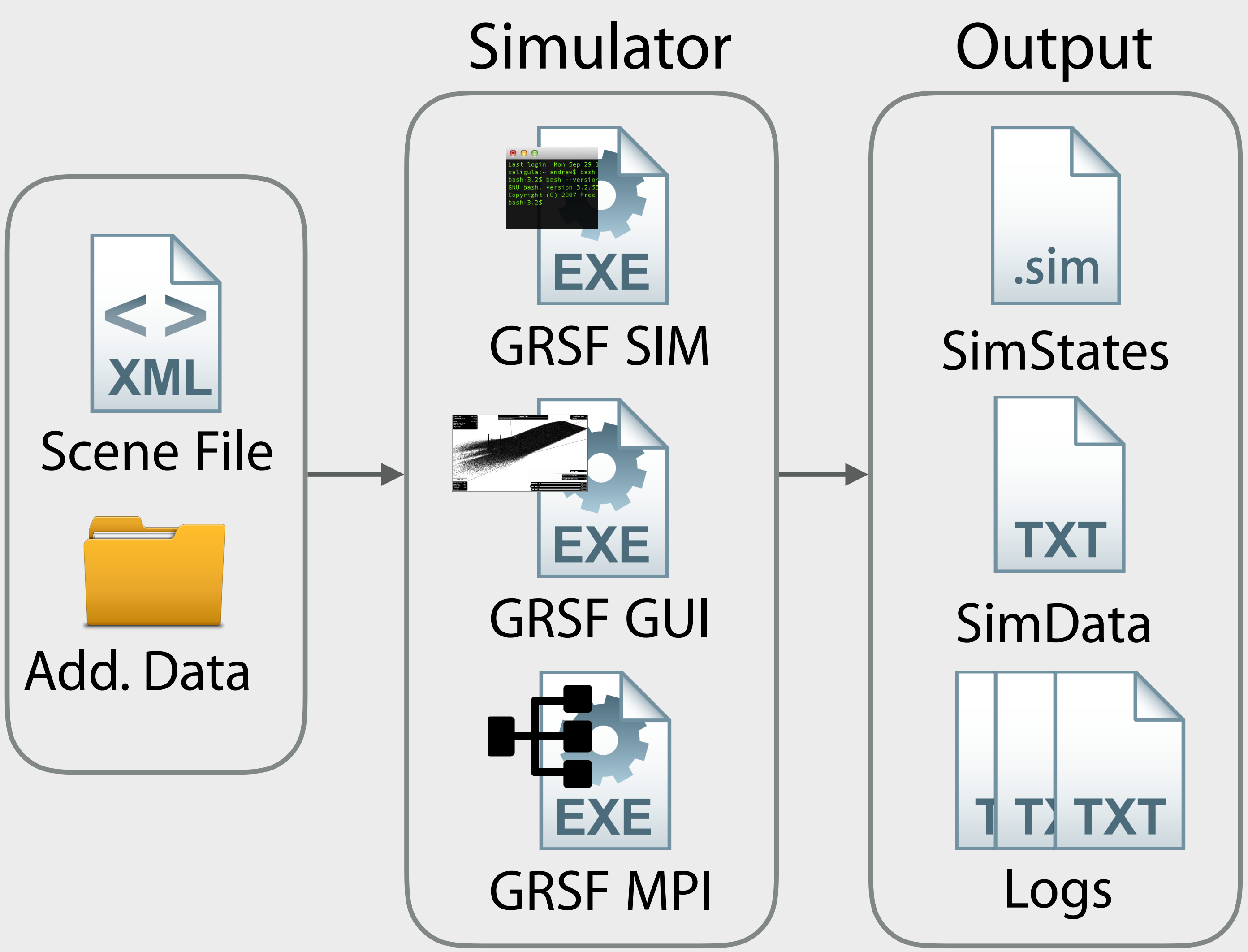


The Granular Rigid Body Simulation Framework

Language: C++11/14 will appear on **github.com**

Features: Large-Scale Non-Smooth Rigid Body Simulations

Dependencies: Eigen, boost, OpenMPI, HDF5, pugixml, Ogre, ApproxMVBB



- position/velocity for each body
- add. data per body

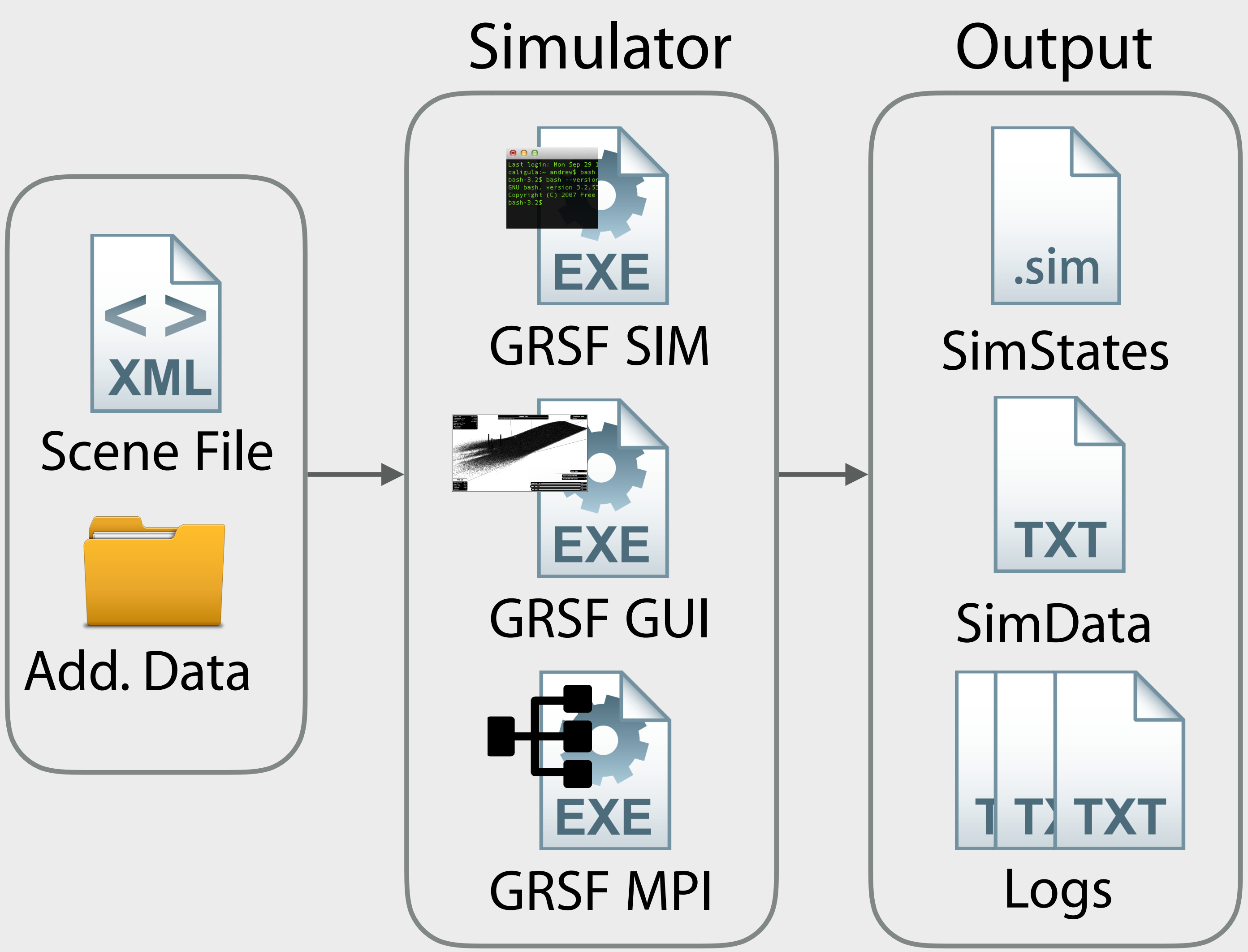
$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \quad \mathbf{u}(t) := \begin{bmatrix} {}_{\mathrm{I}}\dot{\mathbf{r}}_R \\ {}_{\mathrm{K}}\boldsymbol{\Omega} \end{bmatrix}$$

The Granular Rigid Body Simulation Framework

Language: C++11/14 will appear on **github.com**

Features: Large-Scale Non-Smooth Rigid Body Simulations

Dependencies: Eigen, boost, OpenMPI, HDF5, pugixml, Ogre, ApproxMVBB



- position/velocity for each body
- add. data per body

$$\mathbf{q}(t) := \begin{bmatrix} {}_{I}\mathbf{r}_{R} \\ \mathbf{p} \end{bmatrix} \quad \mathbf{u}(t) := \begin{bmatrix} {}_{I}\dot{\mathbf{r}}_{R} \\ {}_{K}\boldsymbol{\Omega} \end{bmatrix}$$

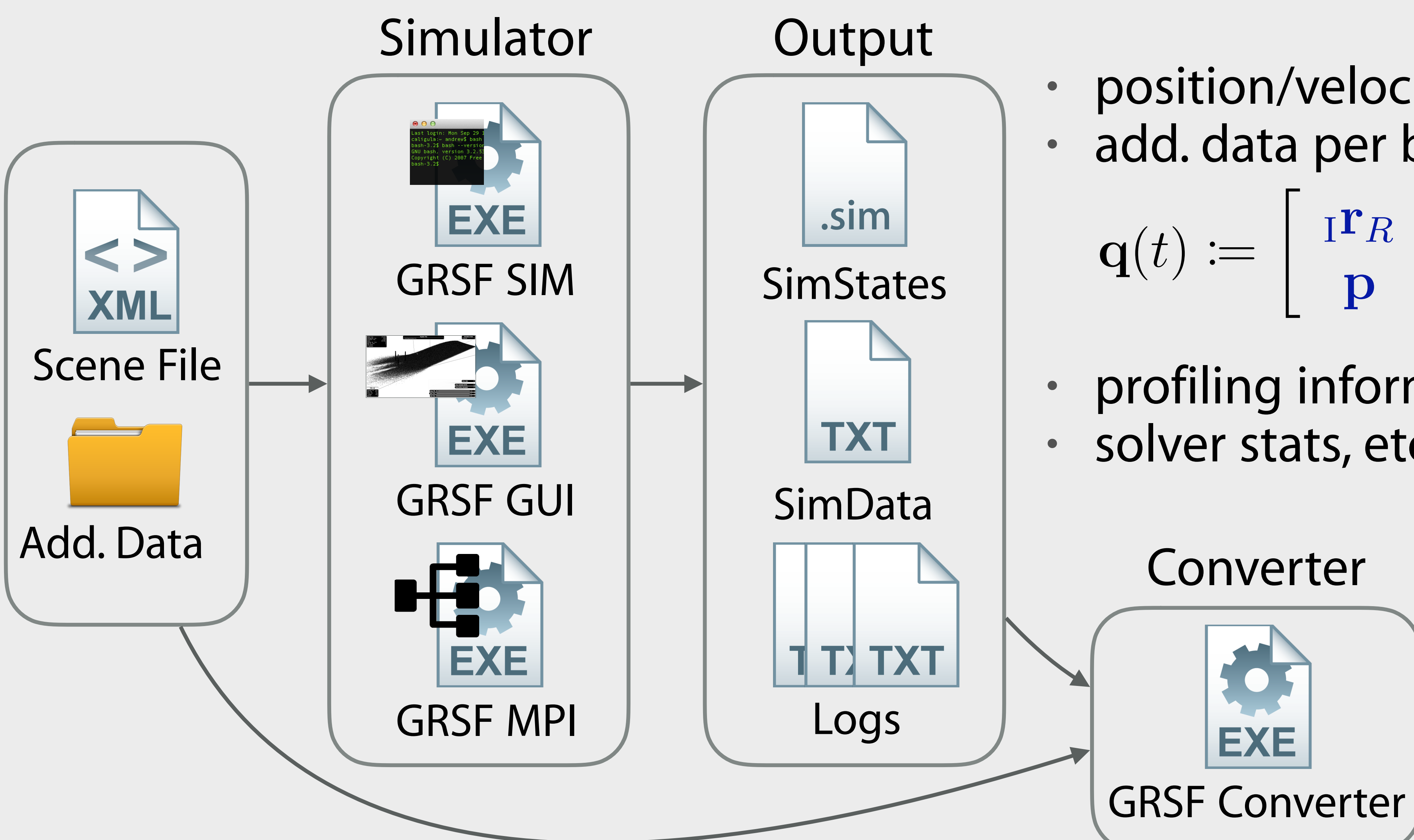
- profiling information
- solver stats, etc.

The Granular Rigid Body Simulation Framework

Language: C++11/14 will appear on **github.com**

Features: Large-Scale Non-Smooth Rigid Body Simulations

Dependencies: Eigen, boost, OpenMPI, HDF5, pugixml, Ogre, ApproxMVBB



- position/velocity for each body
- add. data per body

$$\mathbf{q}(t) := \begin{bmatrix} {}_I\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \quad \mathbf{u}(t) := \begin{bmatrix} {}_I\dot{\mathbf{r}}_R \\ {}_K\boldsymbol{\Omega} \end{bmatrix}$$

- profiling information
- solver stats, etc.

- sim file utilities
- exec. graph based analysis
- gridder

Body Communication among Processes

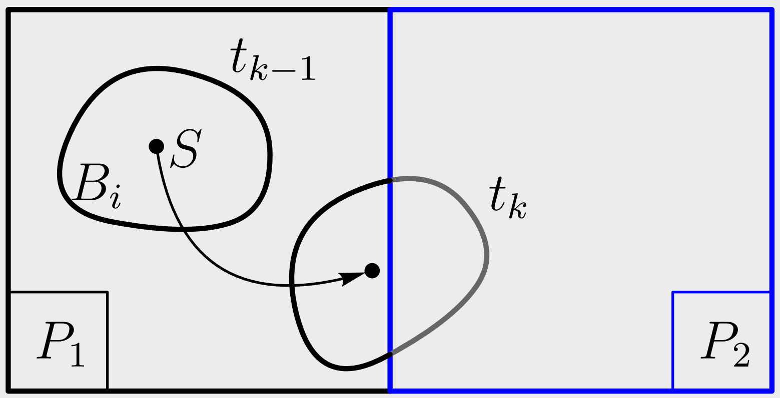
Notification

Update

Removal

Body Communication among Processes

Notification

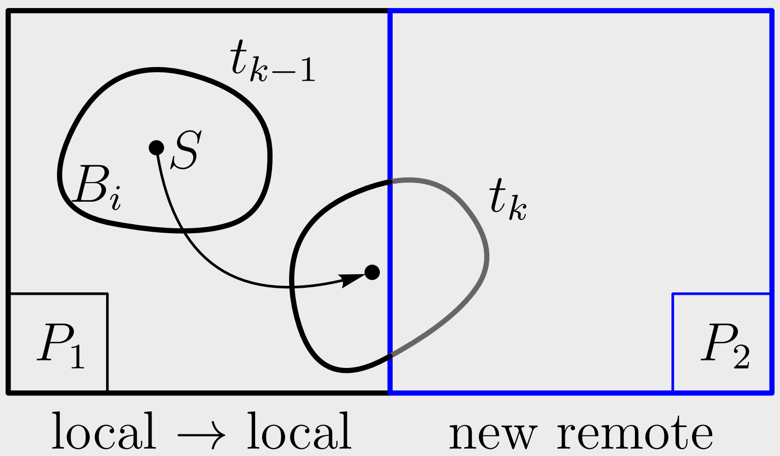


Update

Removal

Body Communication among Processes

Notification



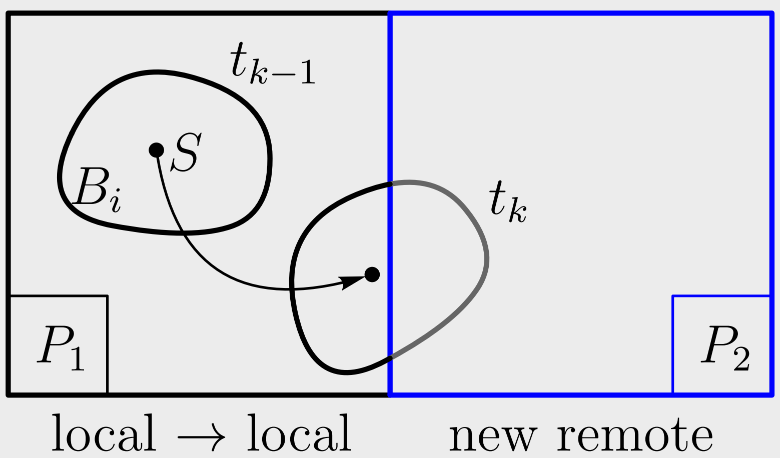
+ 2 more case

Update

Removal

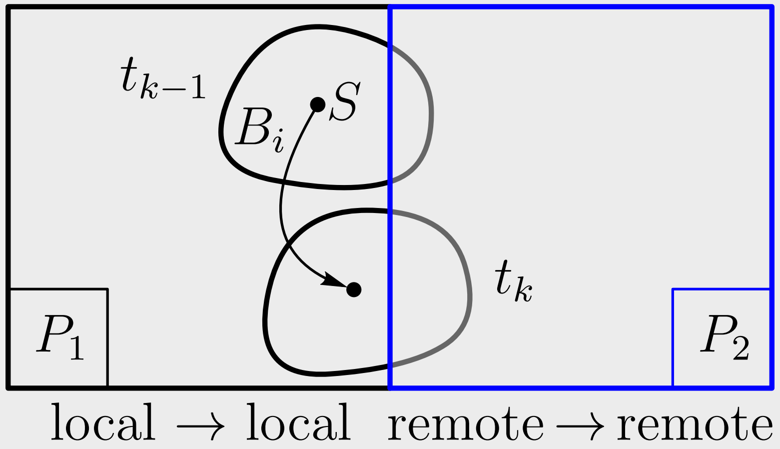
Body Communication among Processes

Notification



+ 2 more case

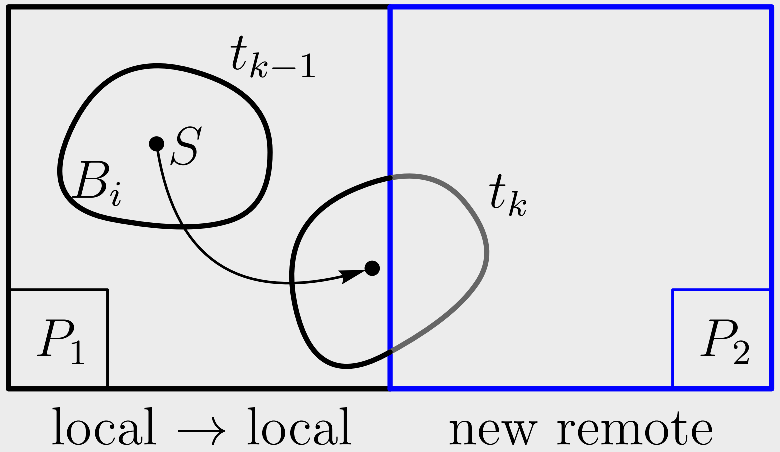
Update



Removal

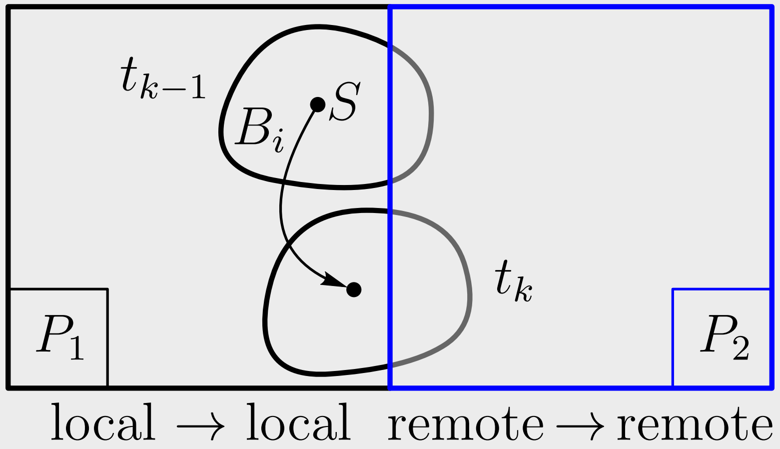
Body Communication among Processes

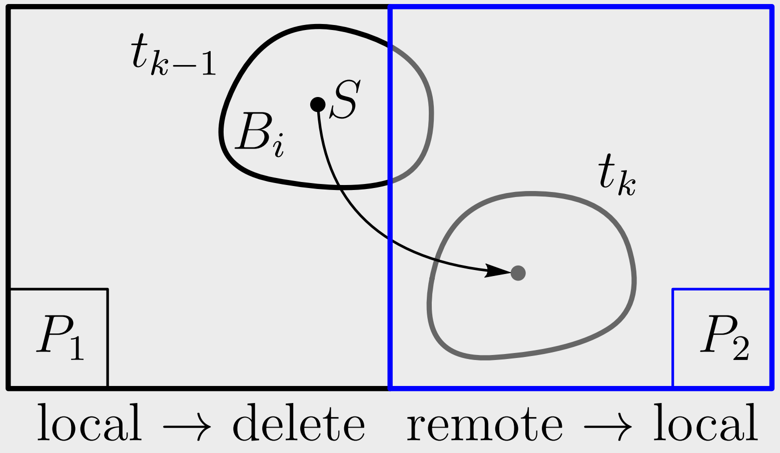
Notification



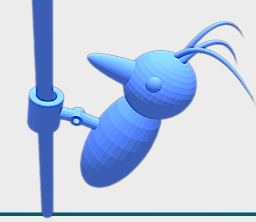
+ 2 more case

Update



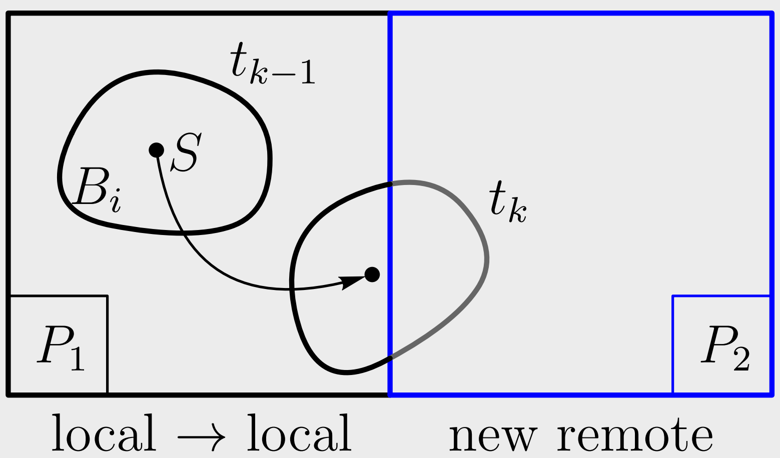


Removal



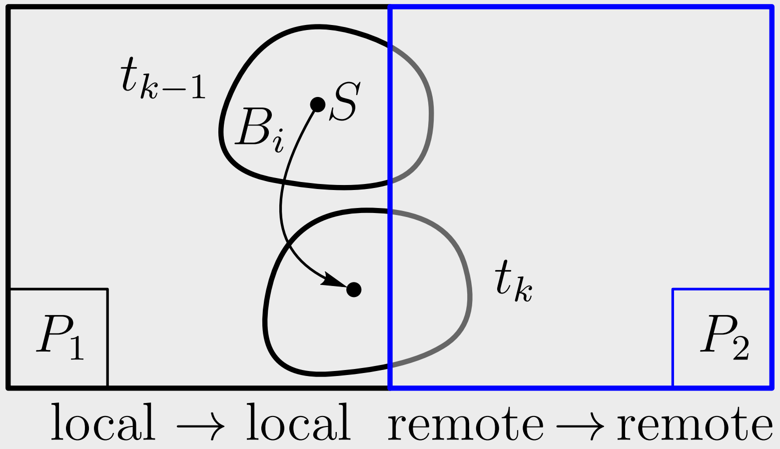
Body Communication among Processes

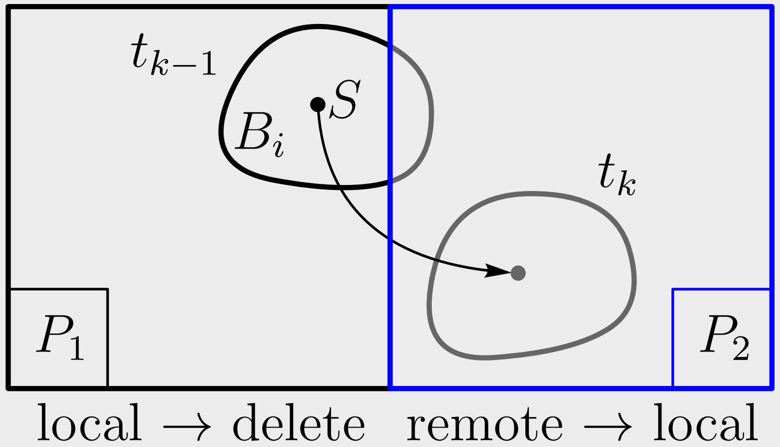
Notification



+ 2 more case

Update



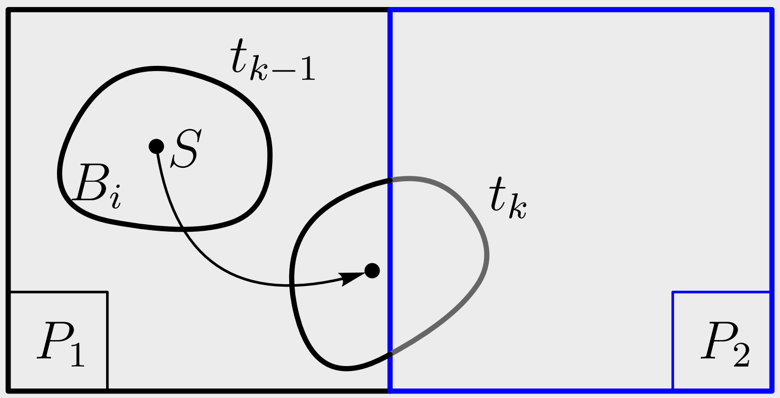


+ 1 more case

Removal

Body Communication among Processes

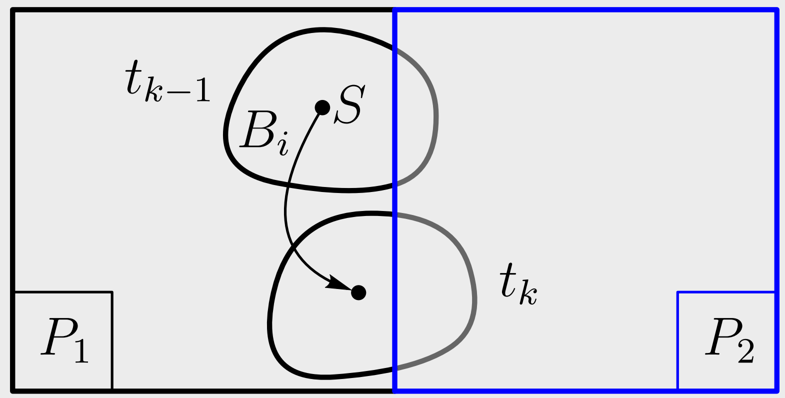
Notification



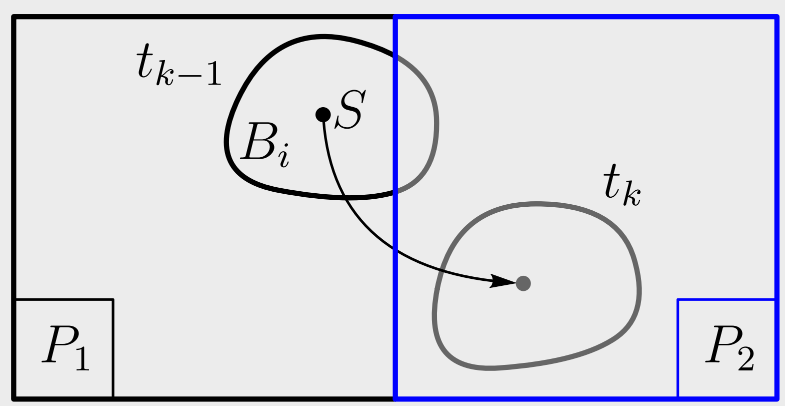
local $\rightarrow$ local new remote

+ 2 more case

Update



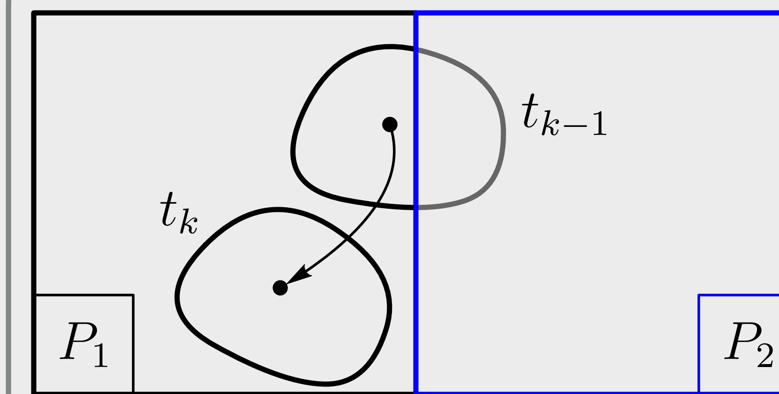
local $\rightarrow$ local remote $\rightarrow$ remote



local $\rightarrow$ delete remote $\rightarrow$ local

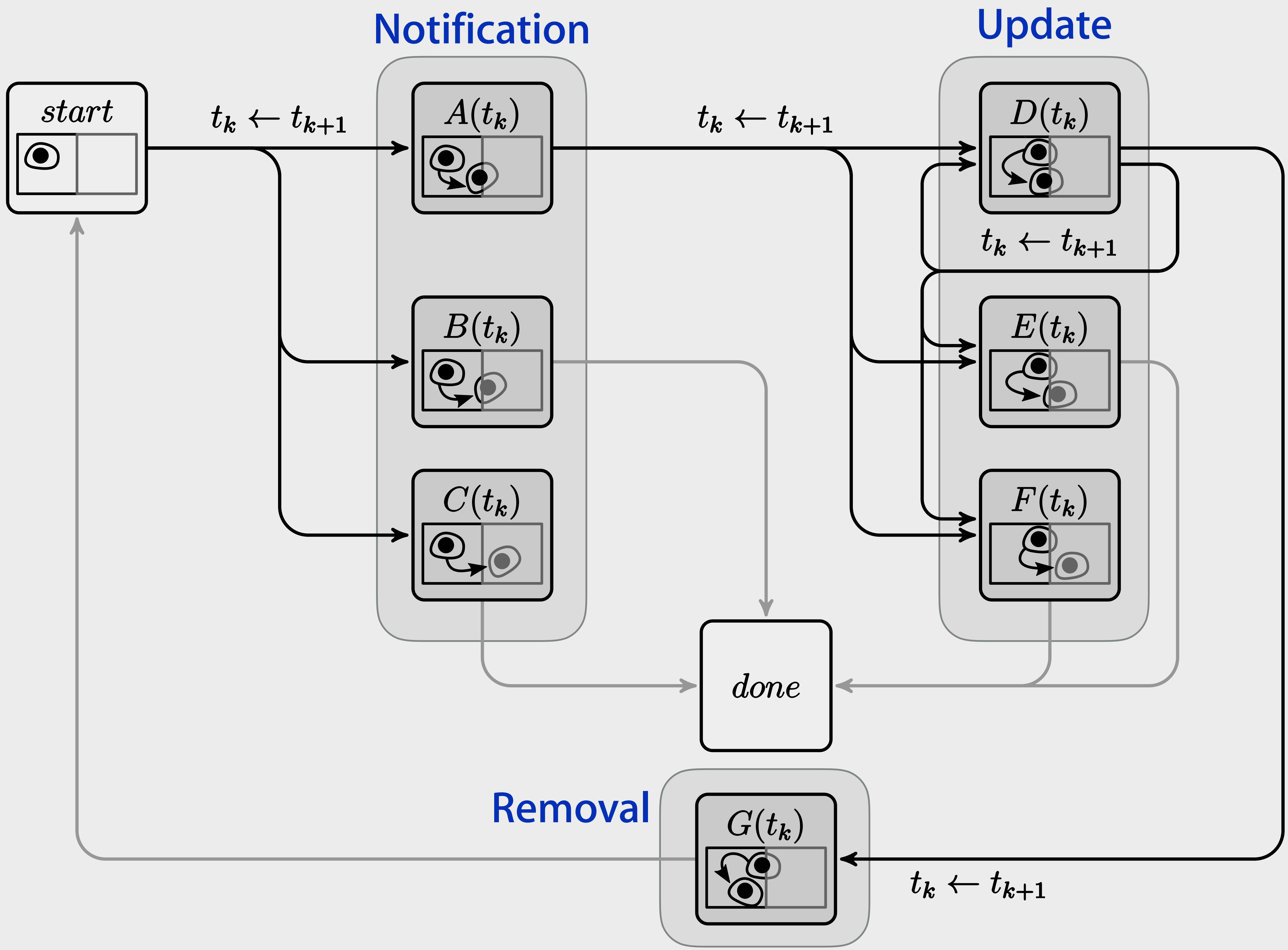
+ 1 more case

Removal



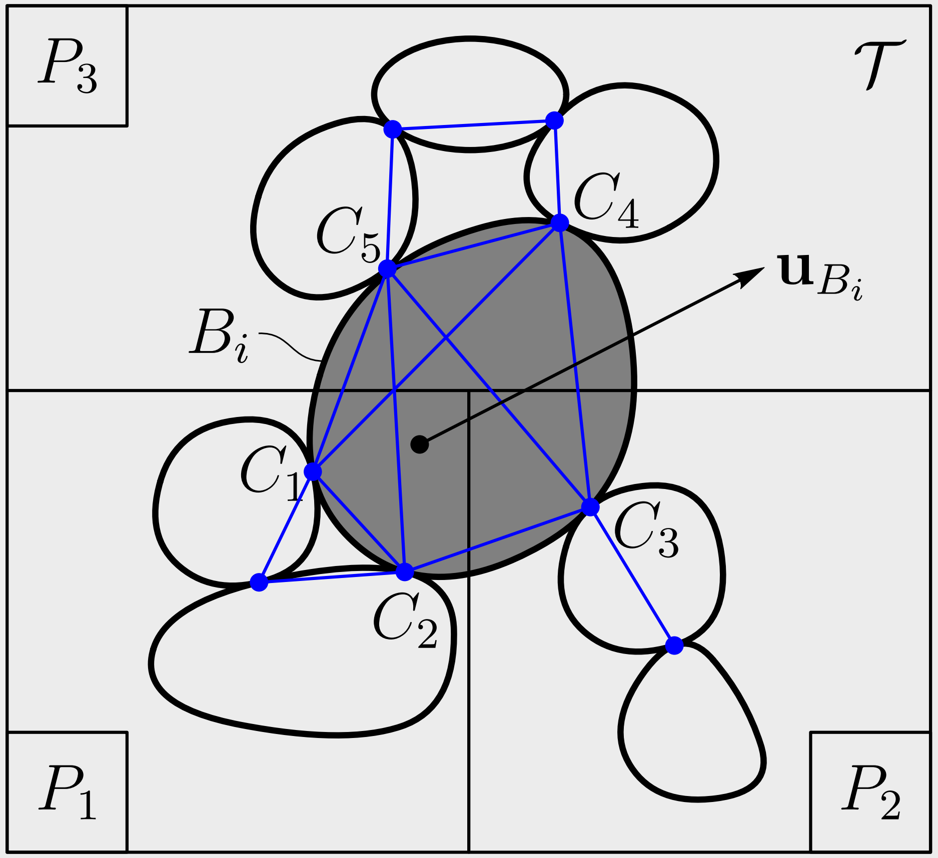
local $\rightarrow$ local remote $\rightarrow$ delete

Body Communication: State Machine



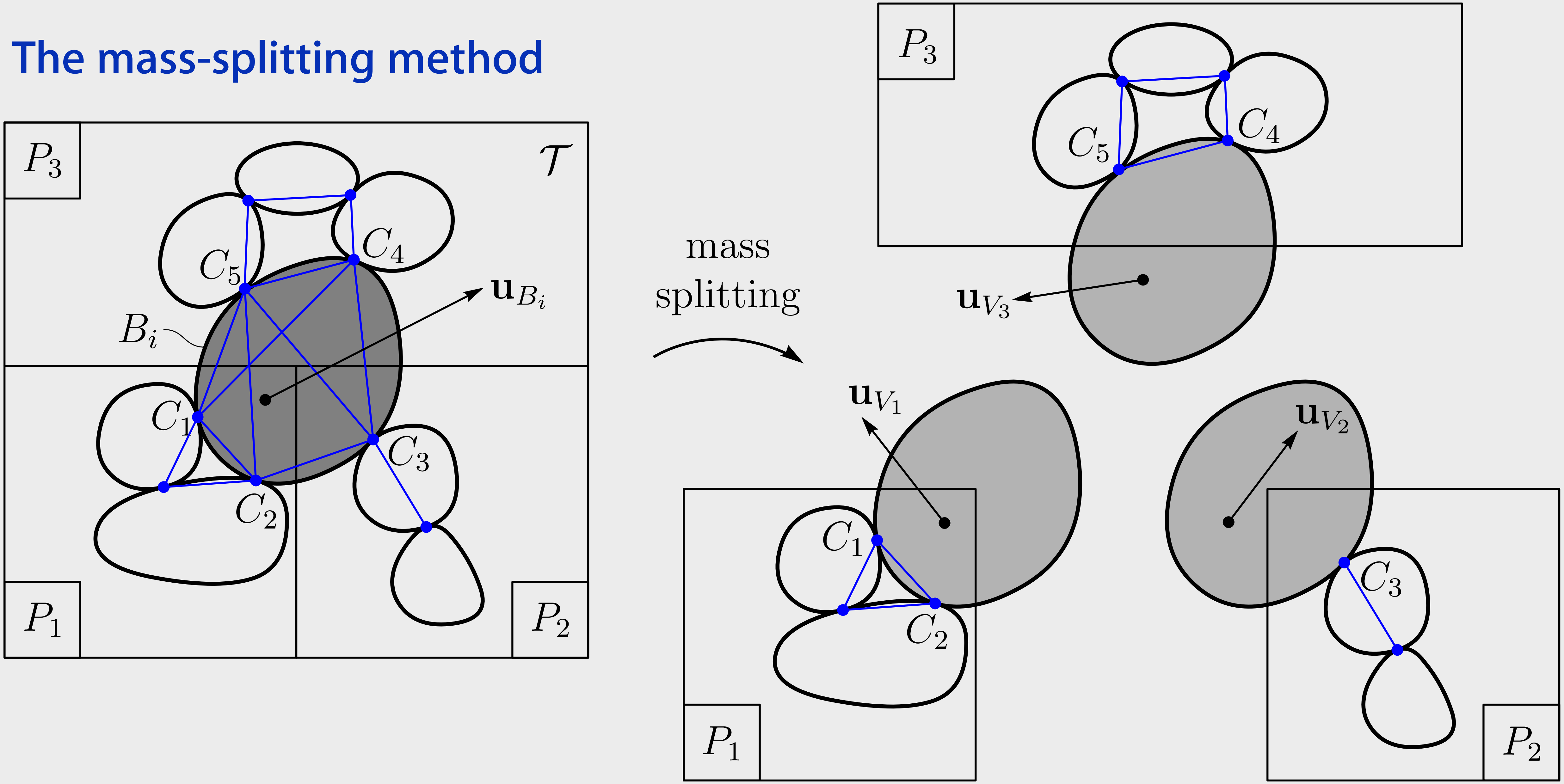
Solving the Discretized Contact Problem

The mass-splitting method



Solving the Discretized Contact Problem

The mass-splitting method

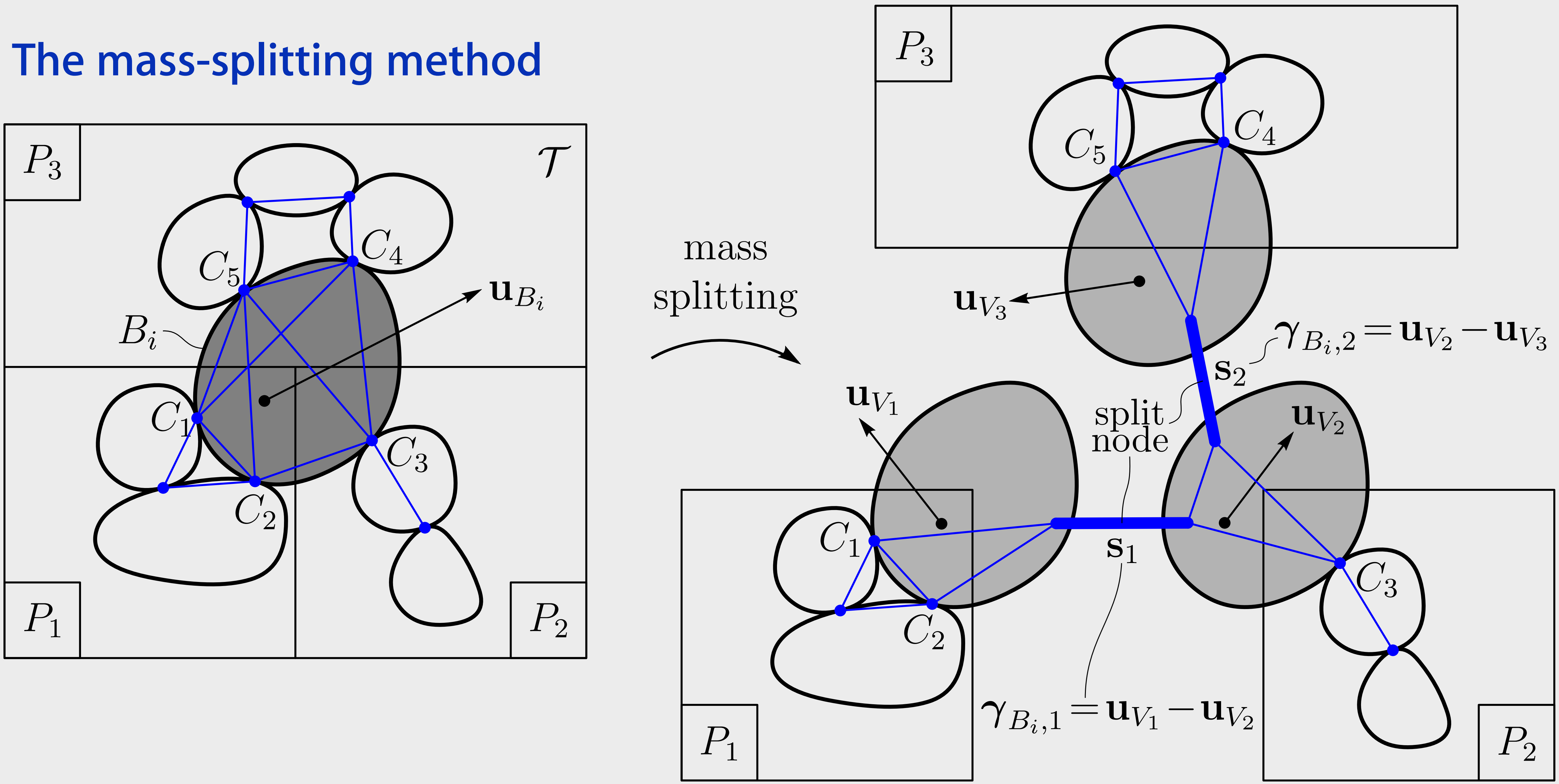


$$\mathbf{M}_{B_i} := \sum_l \alpha_l \mathbf{M}_{V_l}, \quad \mathbf{h}_{B_i} := \sum_l \alpha_l \mathbf{h}_{V_l}$$

with: $\alpha_1 + \cdots + \alpha_p = 1, \quad \alpha_i \geqslant 0$

Solving the Discretized Contact Problem

The mass-splitting method



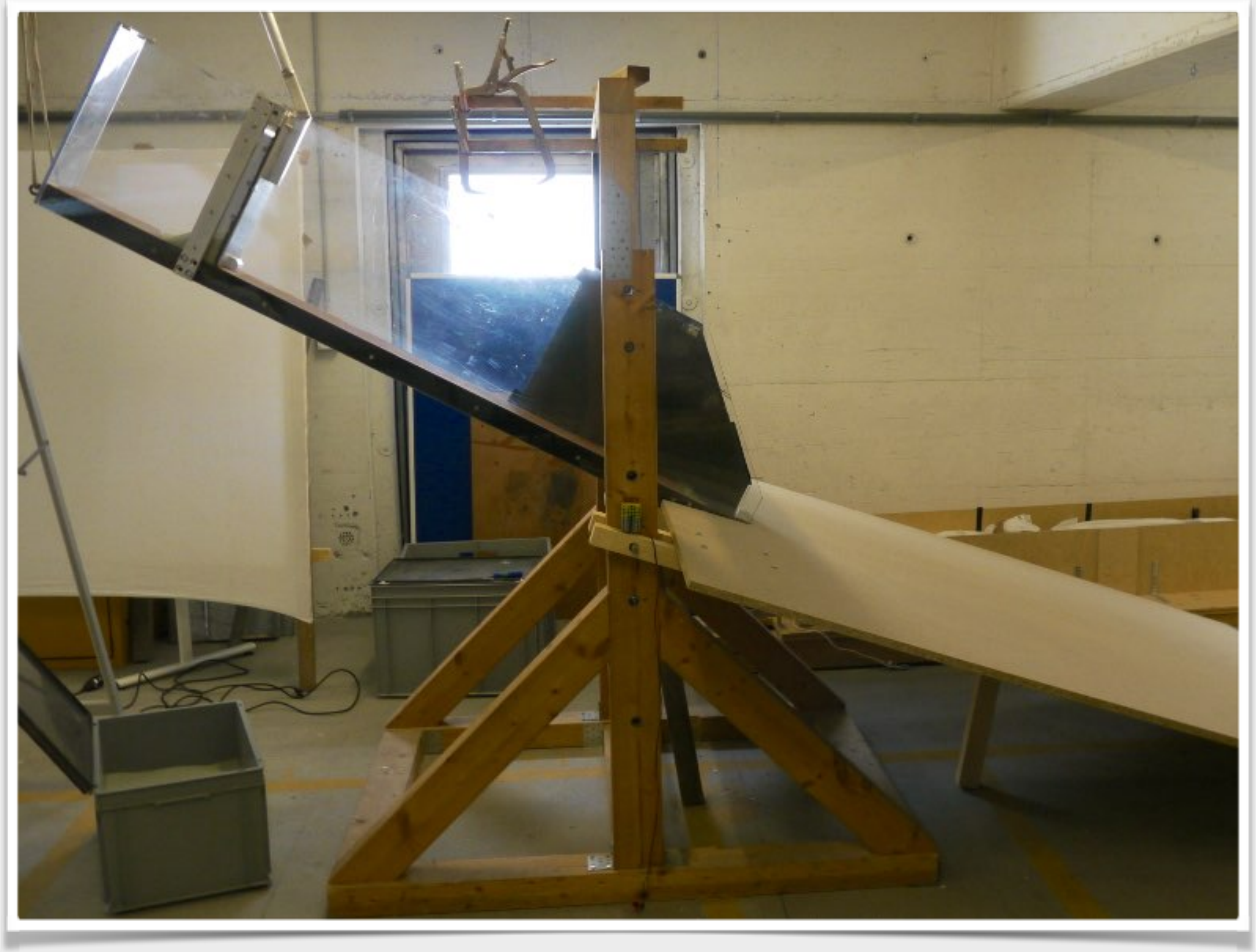
analytical solution of a split-node:
affine combination of velocities

$$\mathbf{M}_{B_i} := \sum_l \alpha_l \mathbf{M}_{V_l}, \quad \mathbf{h}_{B_i} := \sum_l \alpha_l \mathbf{h}_{V_l}$$

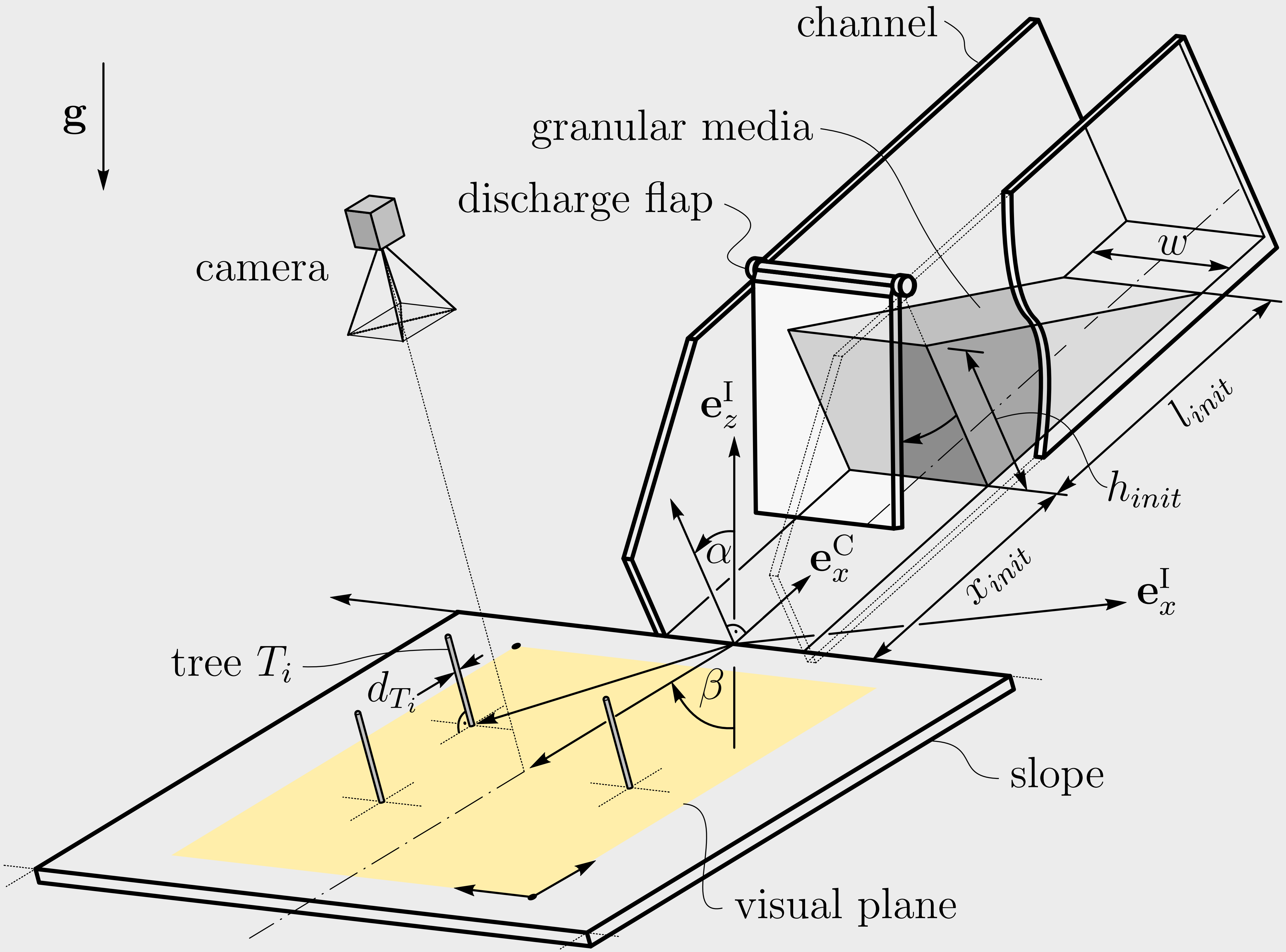
$$\text{with: } \alpha_1 + \cdots + \alpha_p = 1, \quad \alpha_i \geqslant 0$$

Chute Flow Experiments

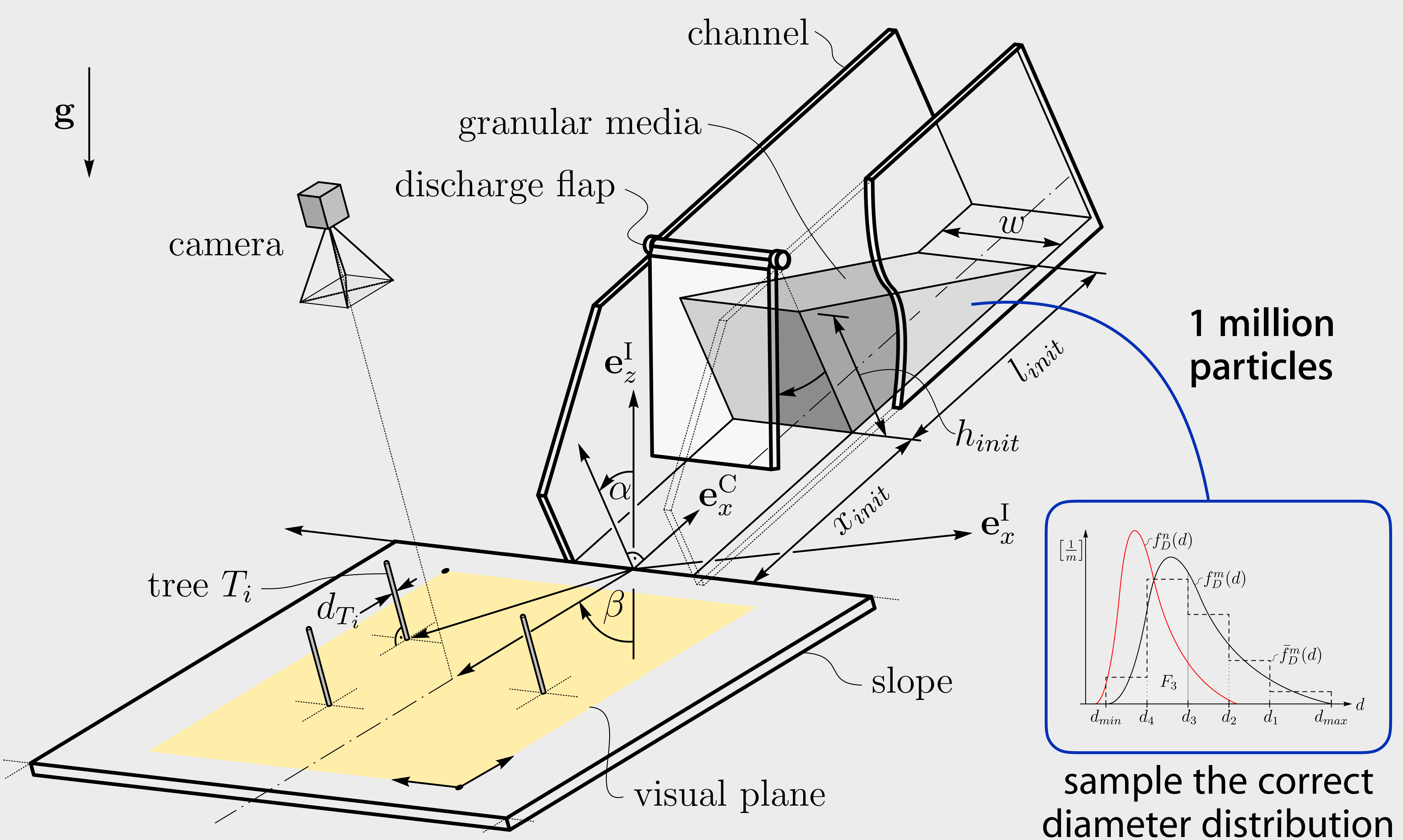
Institute for Snow and Avalanche Research in Davos



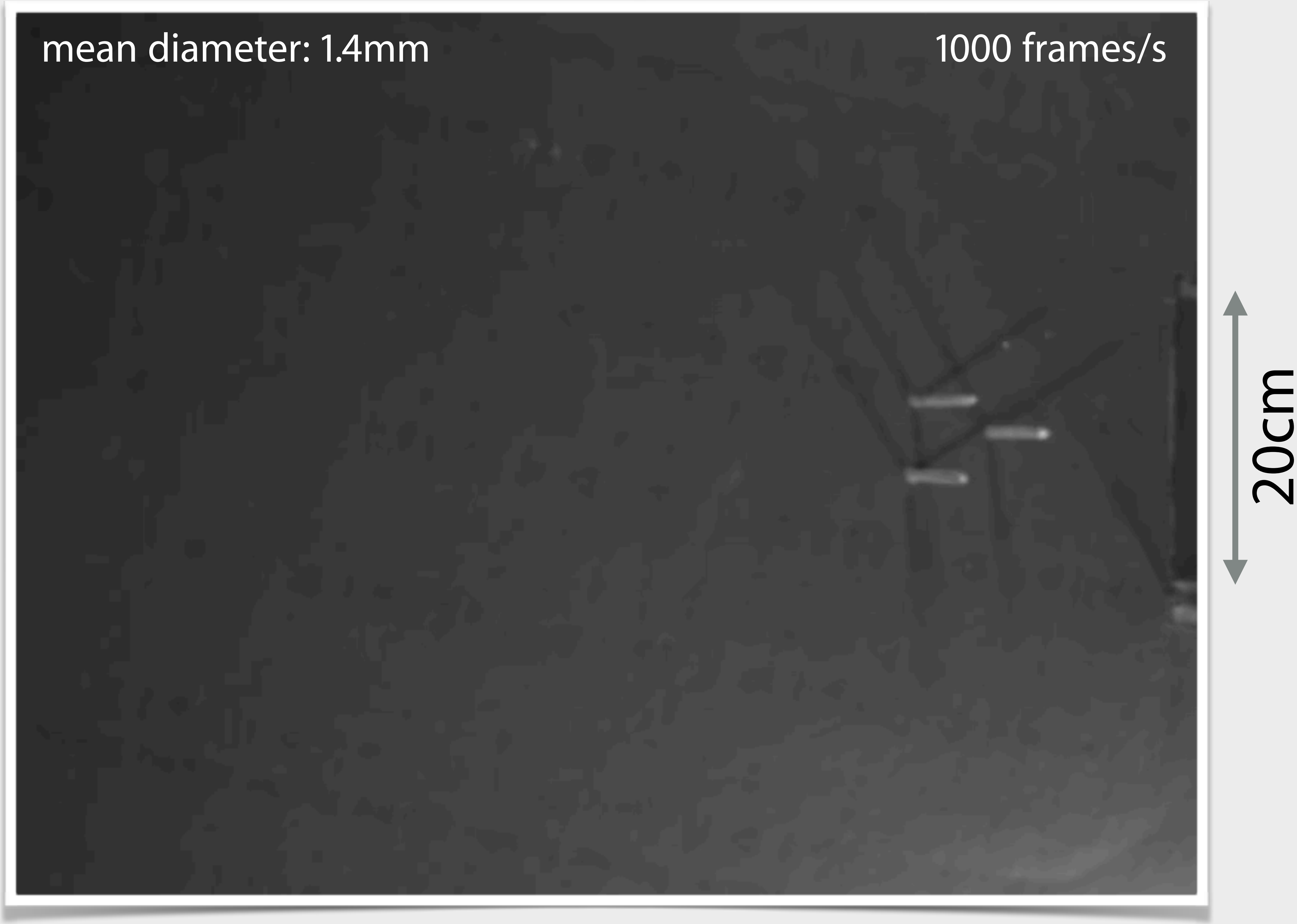
Chute Flow Experiments



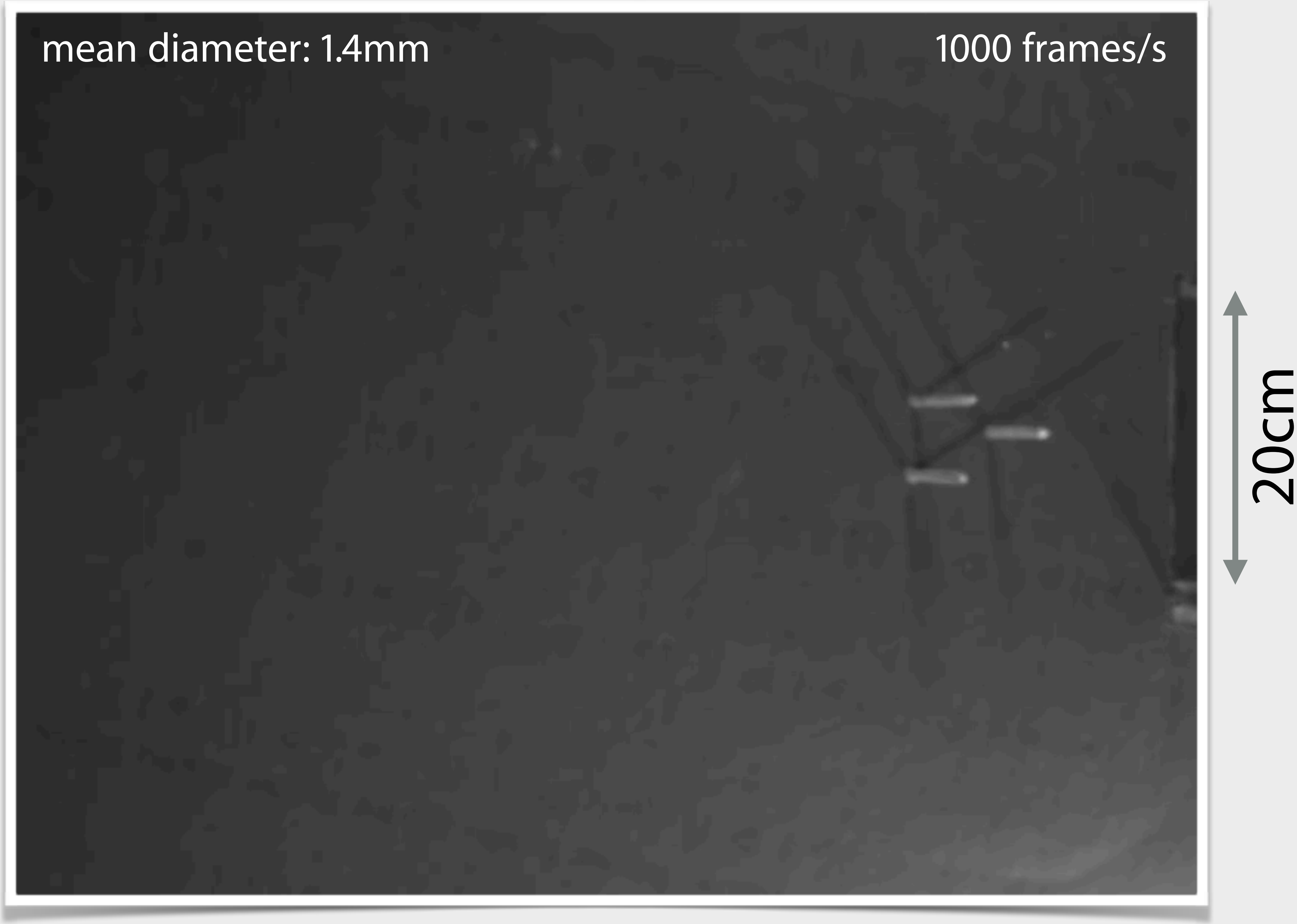
Chute Flow Experiments



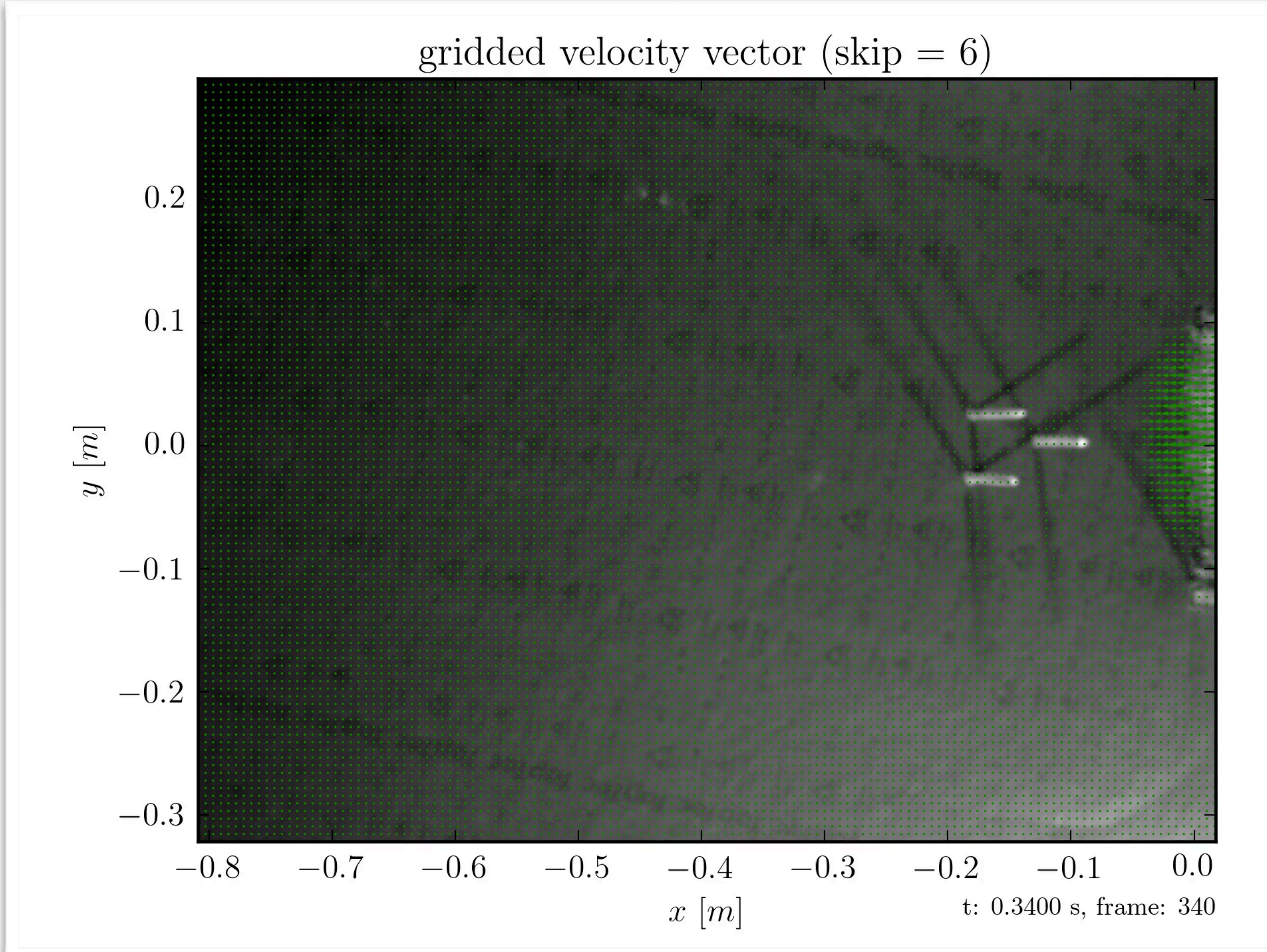
Chute Flow Experiments



Chute Flow Experiments

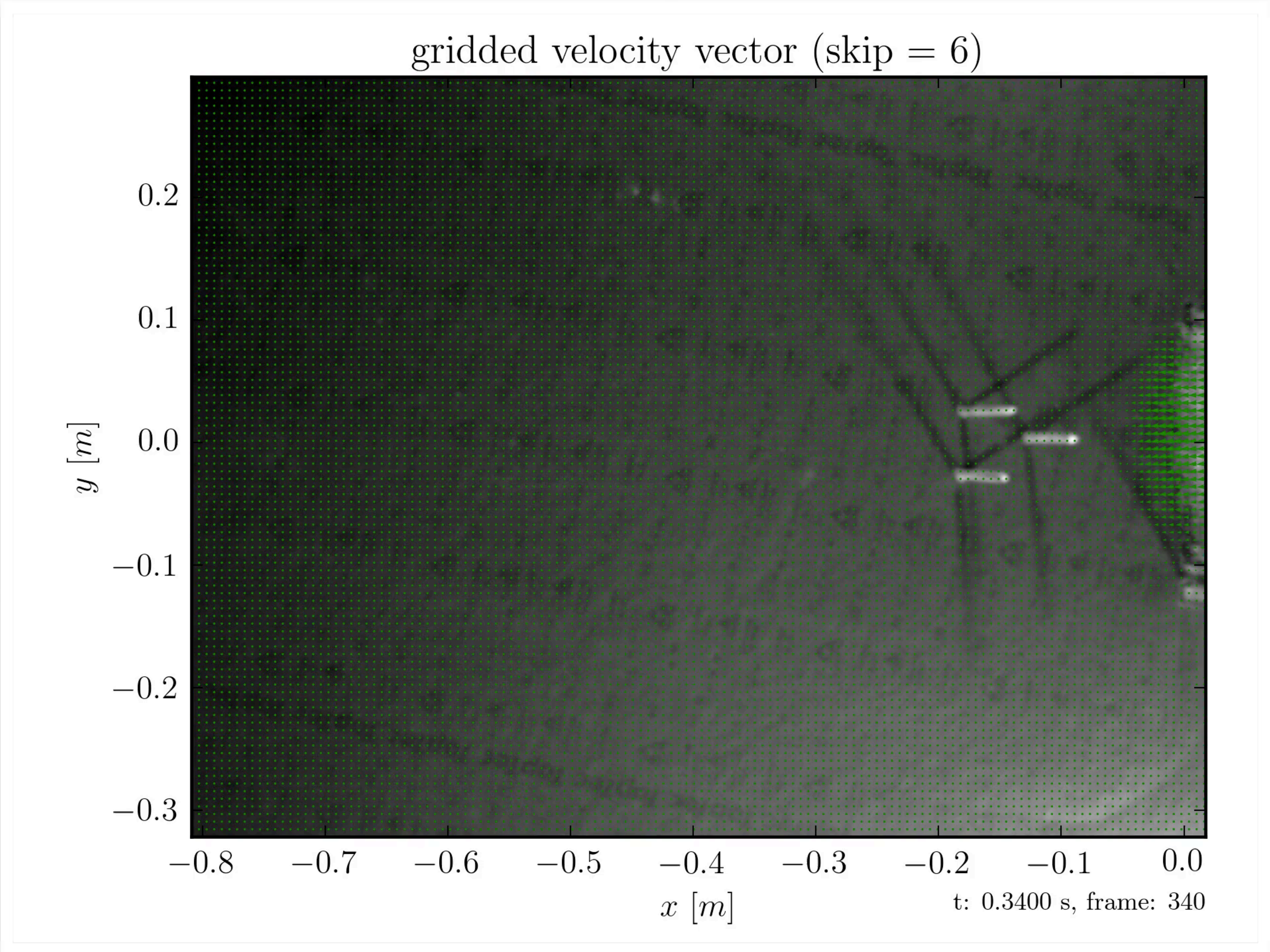


Advection Corrected Velociometry (X. Assay-Davis, 2009)



2000 frames, 4 images/frame, 2 passes = 4h = 75 Gb

Advection Corrected Velociometry (X. Assay-Davis, 2009)

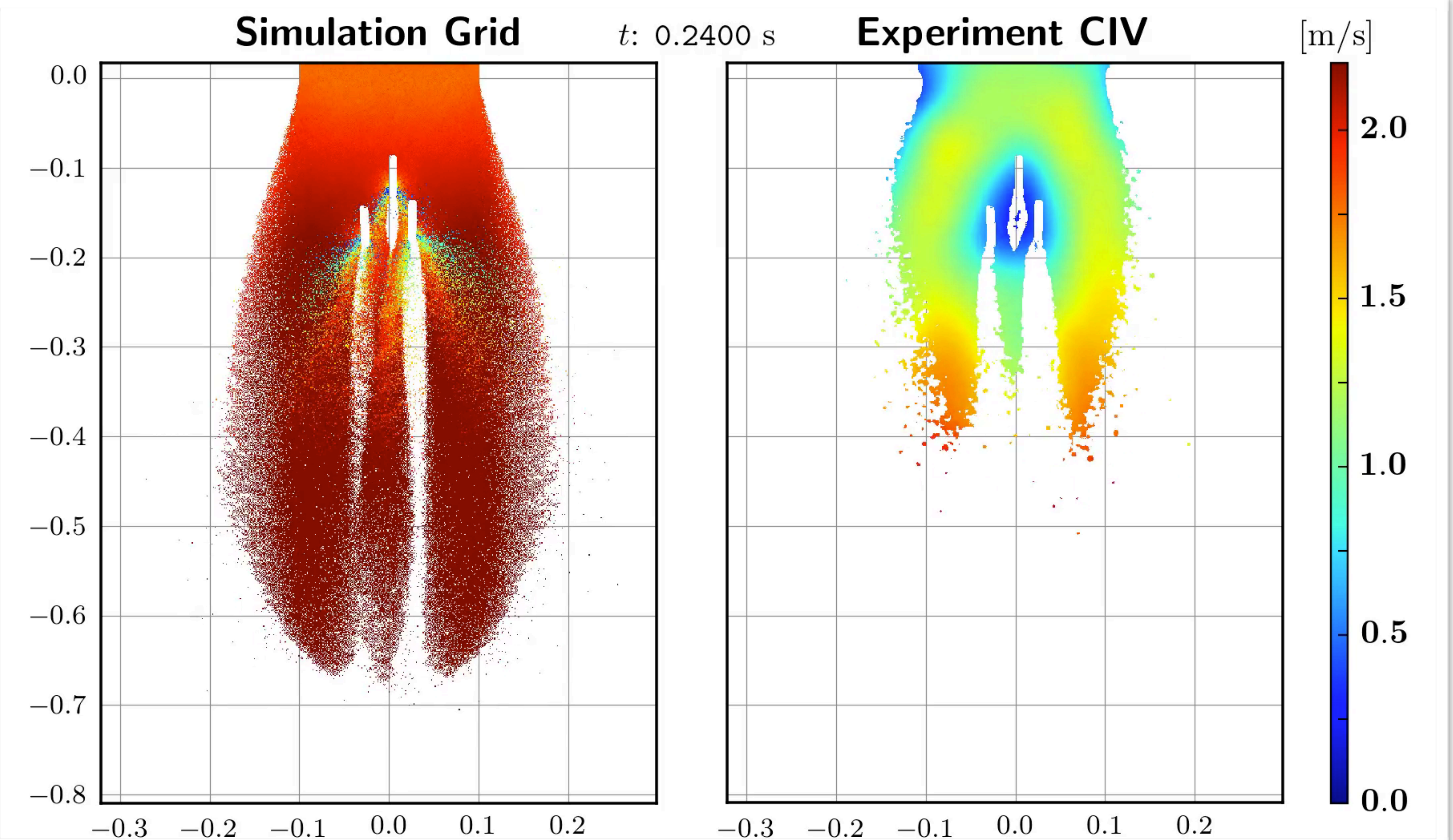


2000 frames, 4 images/frame, 2 passes = 4h = 75 Gb

Comparison

$\mu = 0$

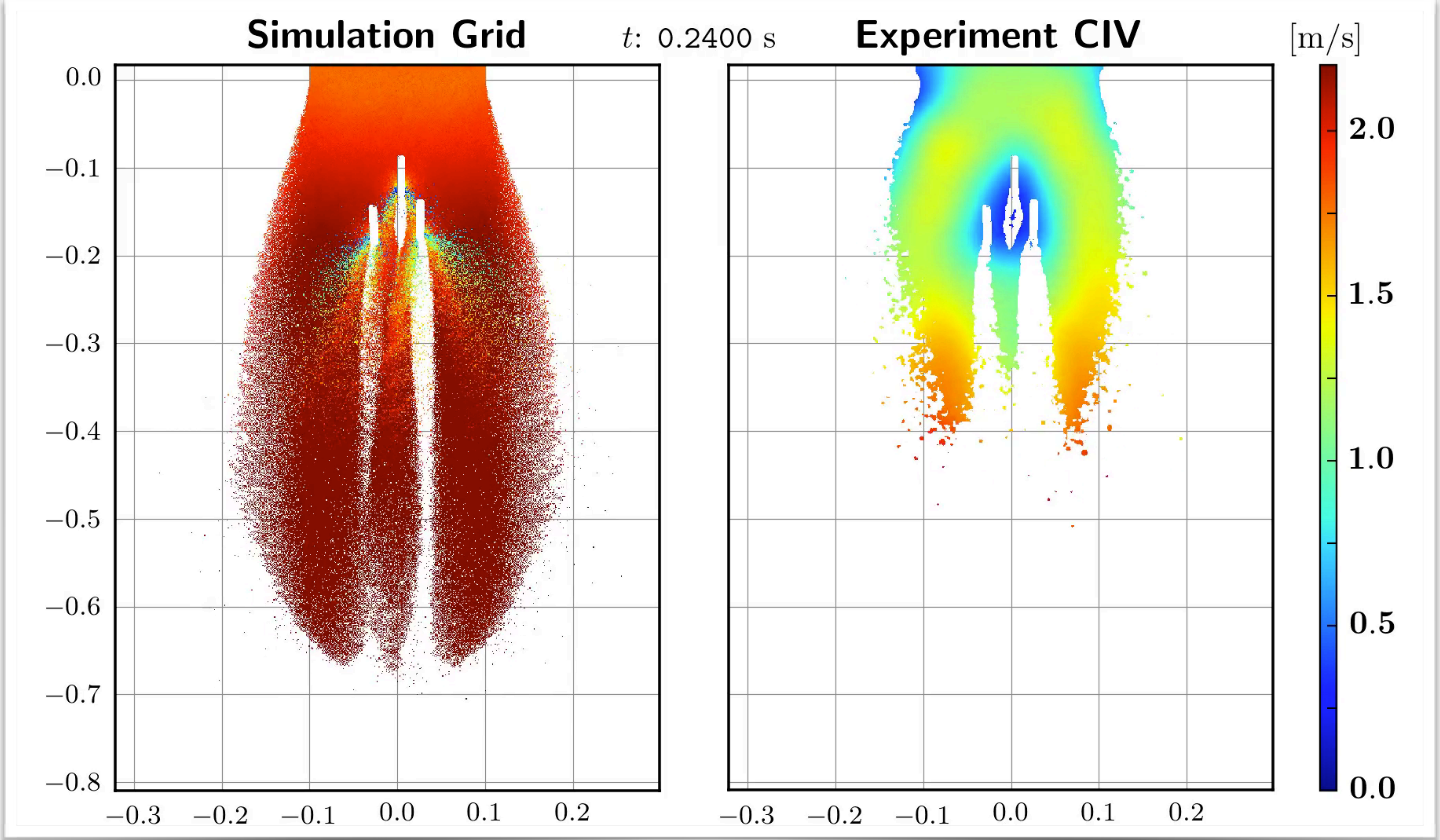
$|\Delta t = 0.0002\,\mathrm{s}\,|\,\mathrm{maxIter}{=}1000\,|\,\varepsilon_N, \varepsilon_T = 0$
$|\,n_T = 10^6$



Comparison

$\mu = 0$

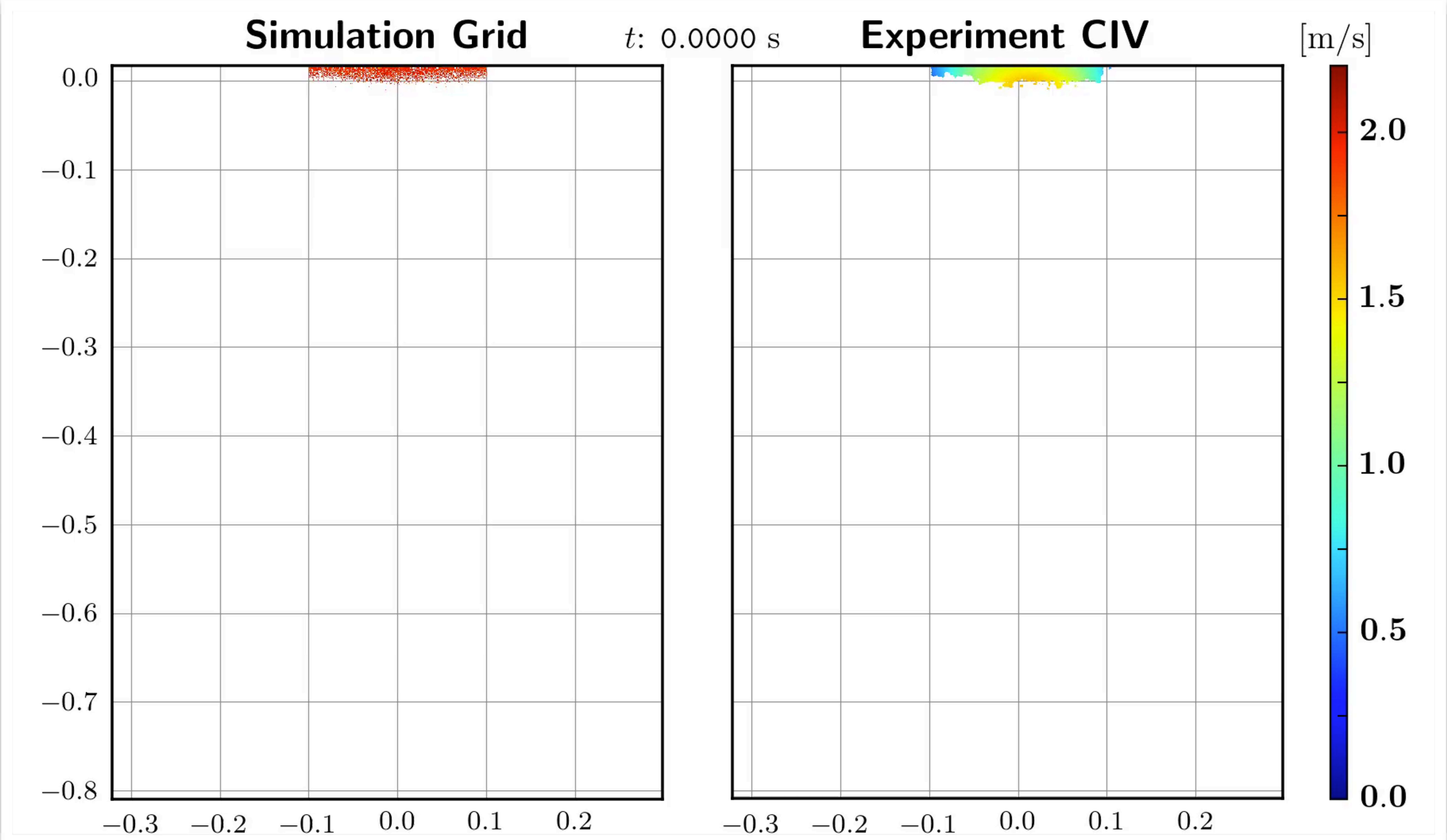
$|\Delta t = 0.0002\,\mathrm{s} \,|\, \mathrm{maxIter}{=}1000 \,|\, \varepsilon_N, \varepsilon_T = 0$
$|\, n_T = 10^6$



Comparison

$\mu = 0.3$

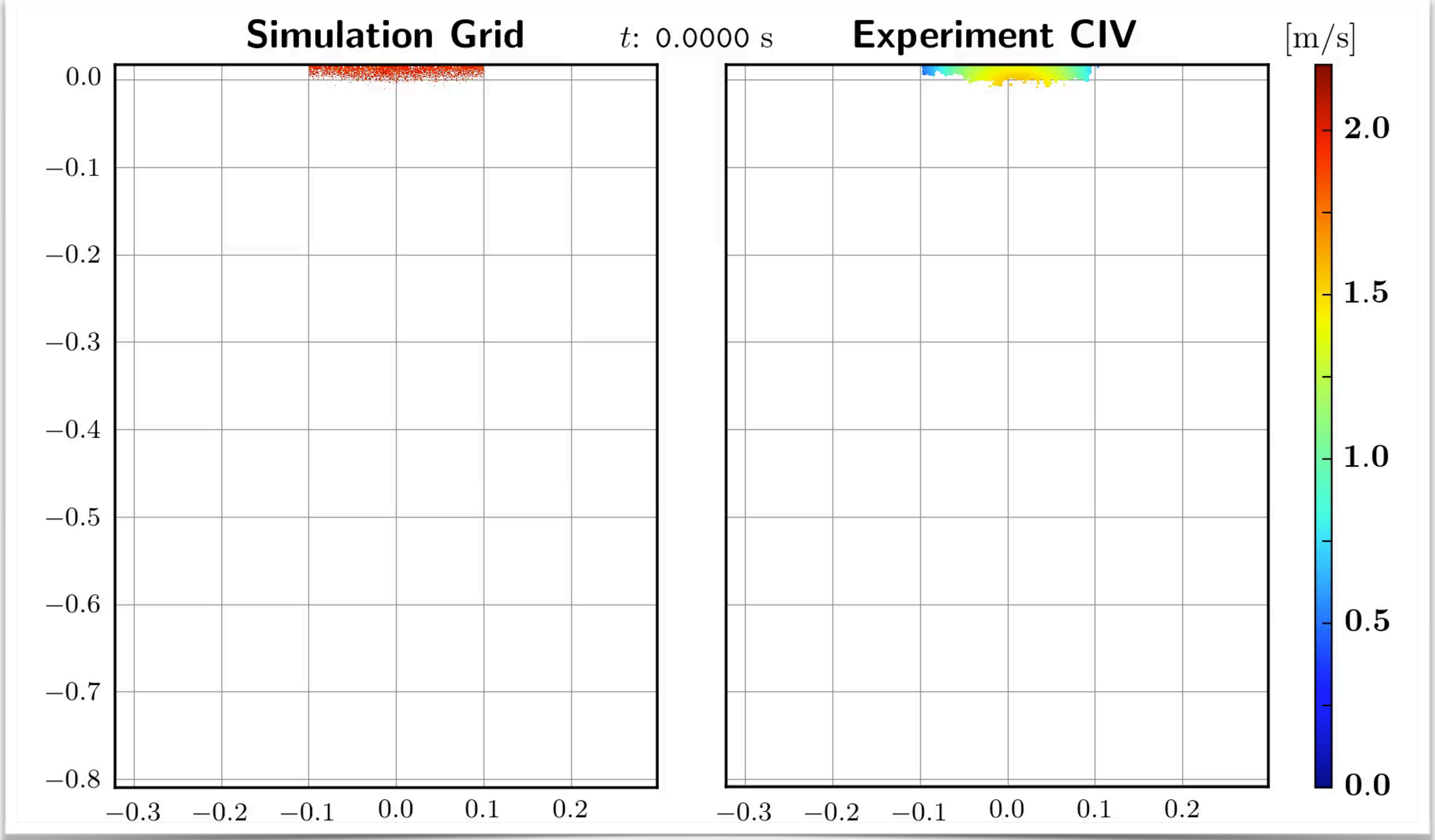
$|\, \Delta t = 0.0002\,\mathrm{s} \,|\, \text{maxIter=1000} \,|\, \varepsilon_N, \varepsilon_T = 0$
$|\, n_T = 10^6$



Comparison

$\mu = 0.3$

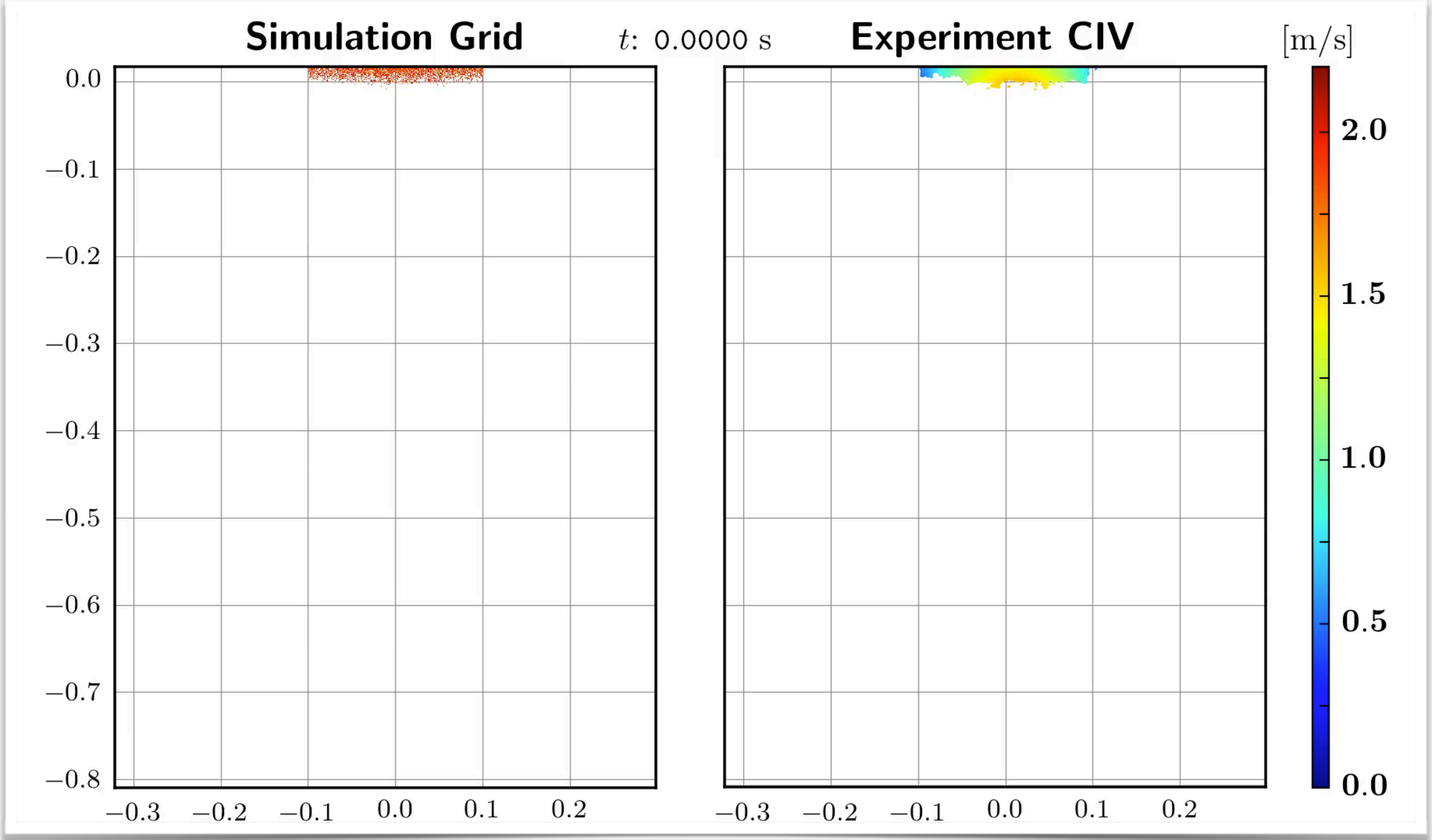
$|\,\Delta t = 0.0002\,\mathrm{s}\,|\,\mathrm{maxIter{=}1000}\,|\,\varepsilon_N, \varepsilon_T = 0$
$|\,n_T = 10^6$



Comparison

$\mu = 0.8$

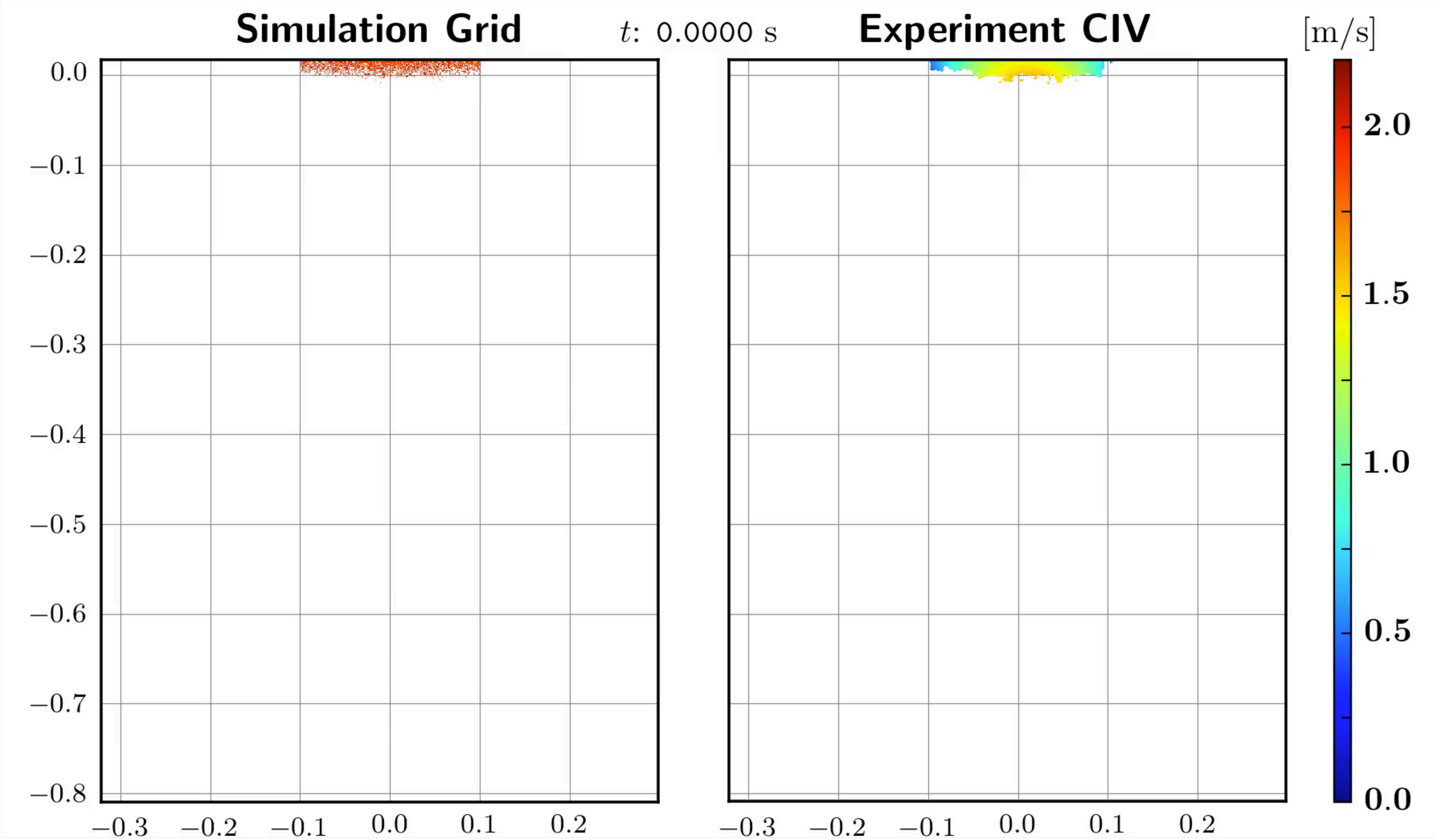
$|\,\Delta t = 0.0002\,\mathrm{s}\,|\,\mathrm{maxIter{=}1000}\,|\,\varepsilon_N, \varepsilon_T = 0$
$|\,n_T = 10^6$ **best match**



Comparison

$\mu = 0.8$

$|\Delta t = 0.0002\,\mathrm{s}\,|\,\text{maxIter}{=}1000\,|\,\varepsilon_N, \varepsilon_T = 0$
$|\, n_T = 10^6$ **best match**

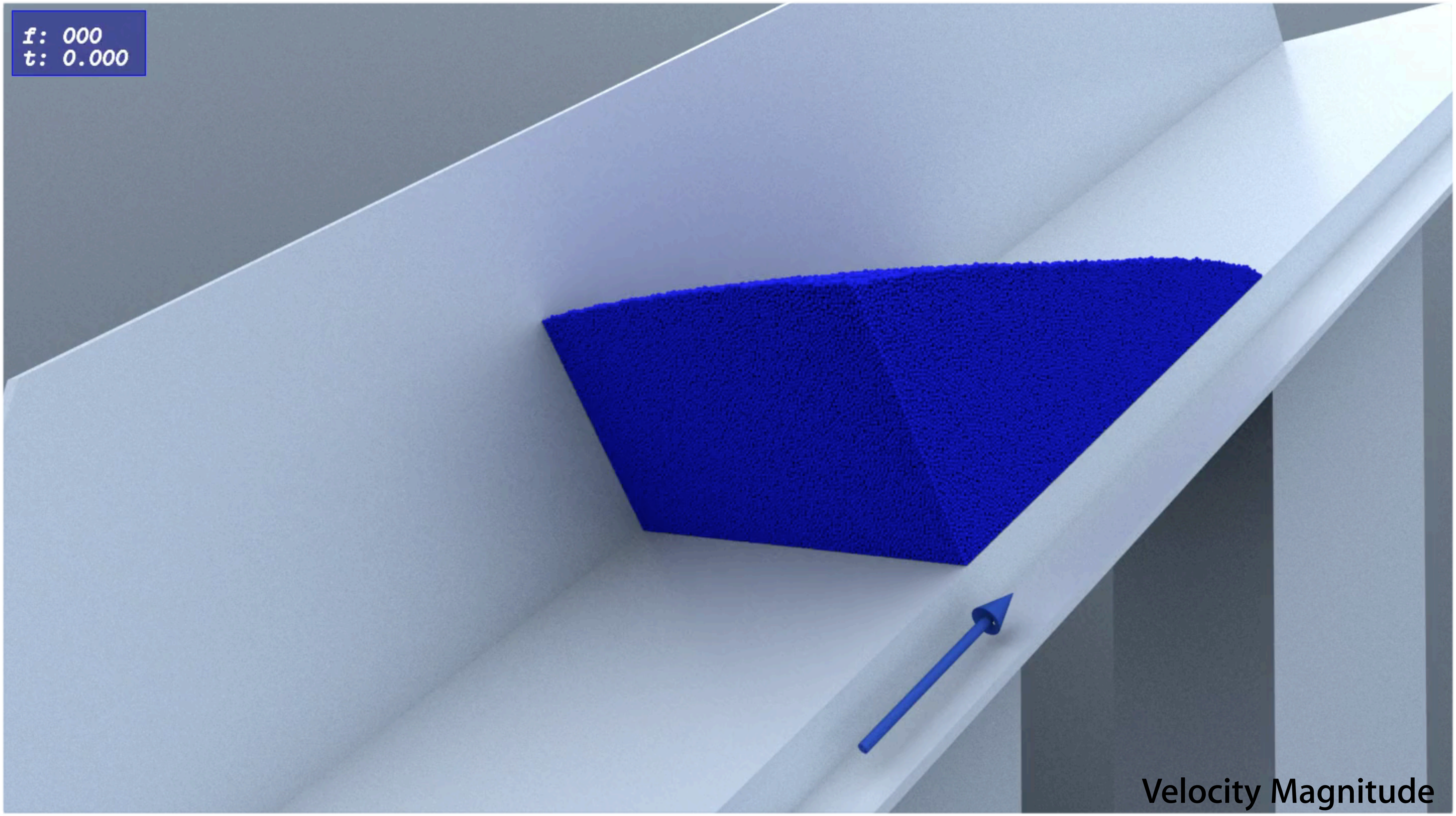


Comparison

ETH Euler: 384 Cores

GRFS MPI: 12h = 120Gb

Converter/Renderman: 780 Frames in 24h



$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$

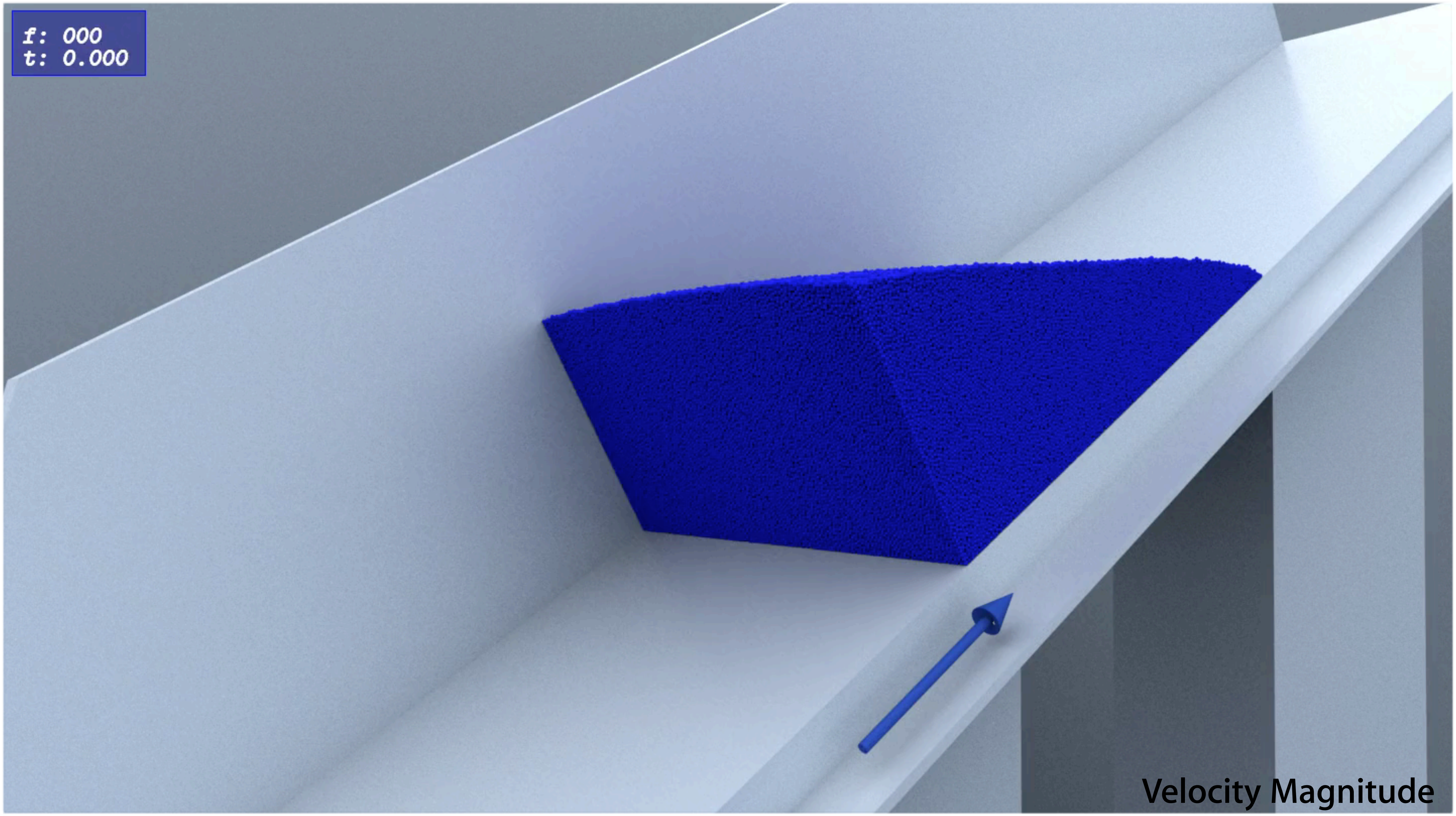
best match

Comparison

ETH Euler: 384 Cores

GRFS MPI: 12h = 120Gb

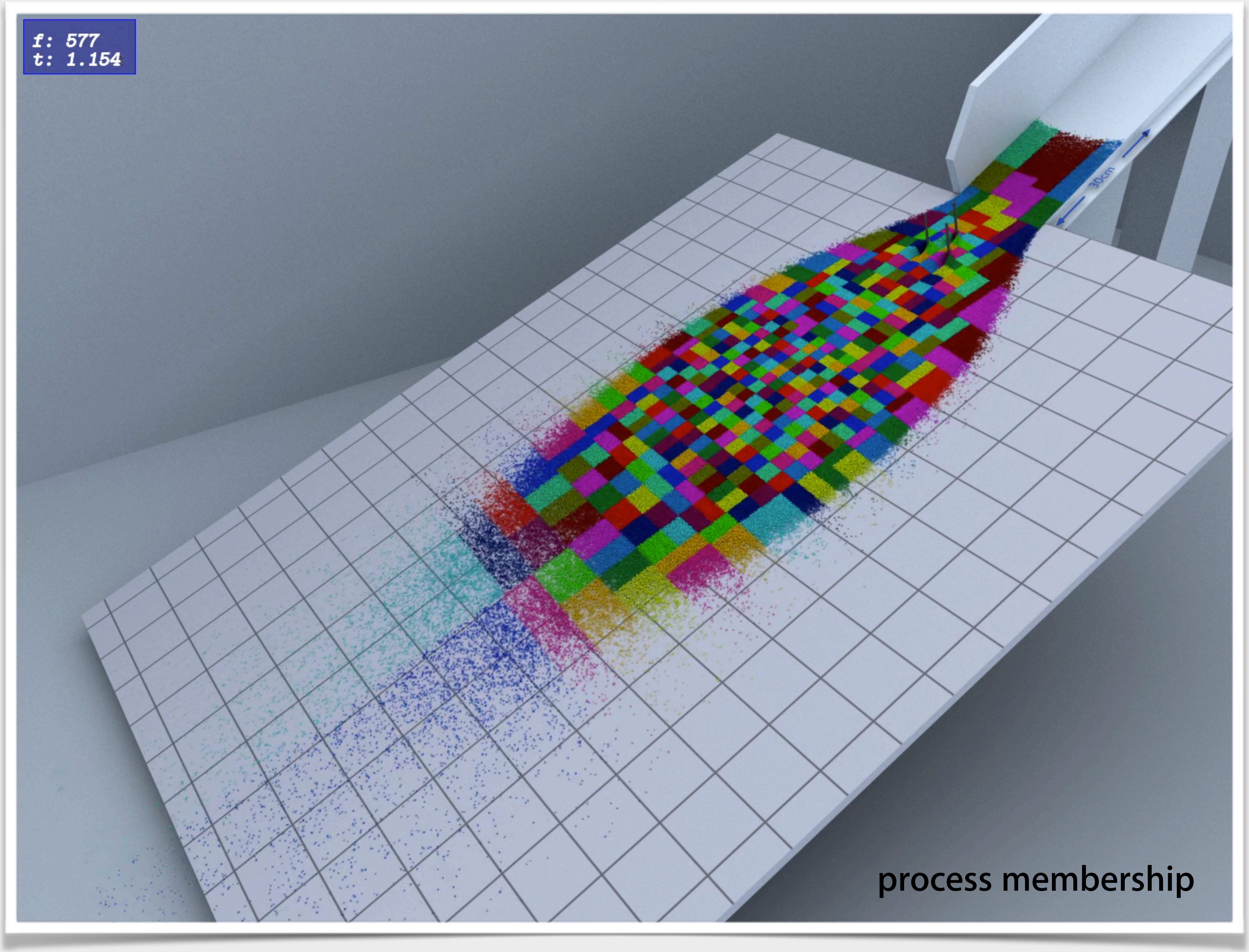
Converter/Renderman: 780 Frames in 24h



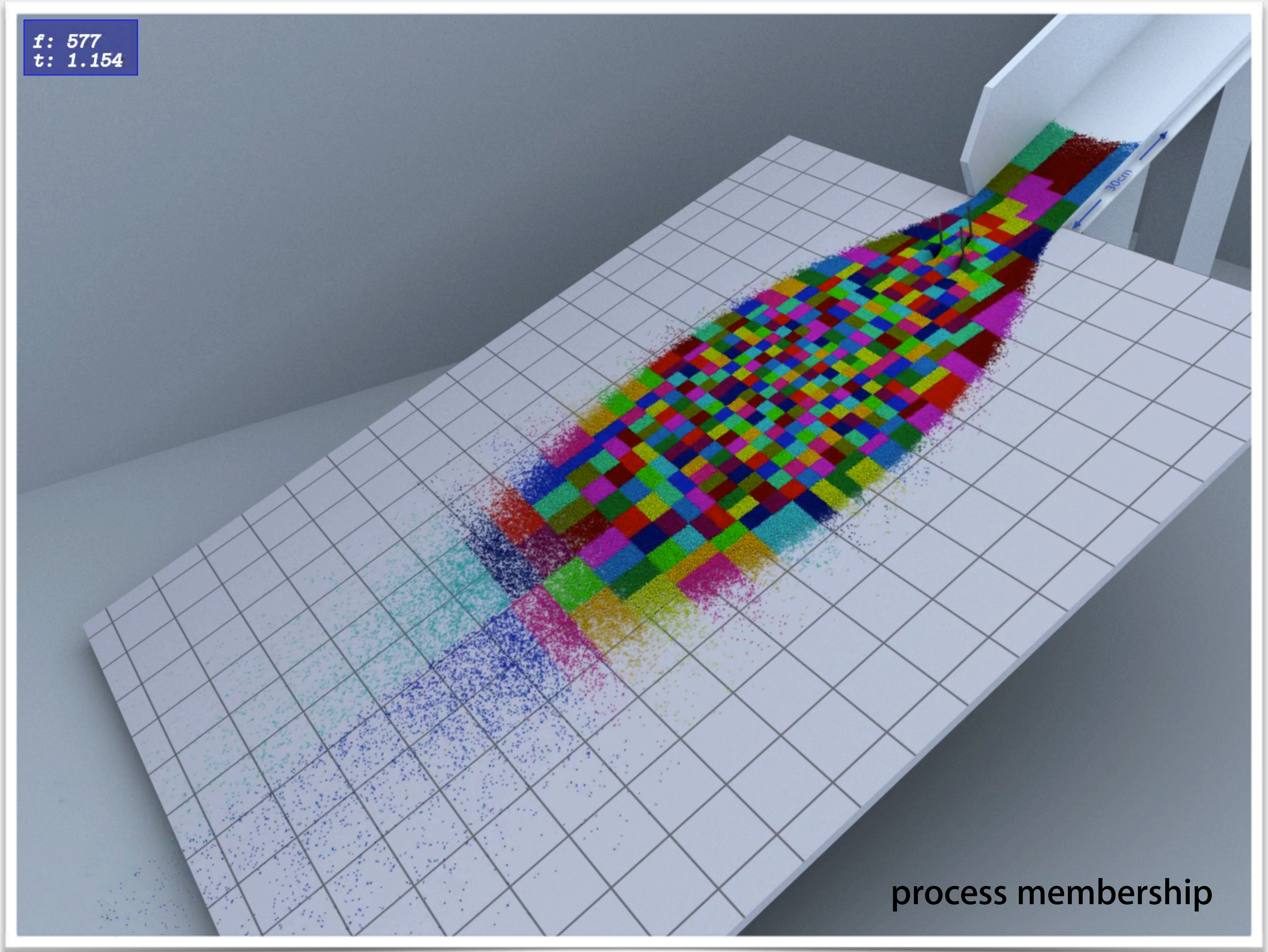
$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$

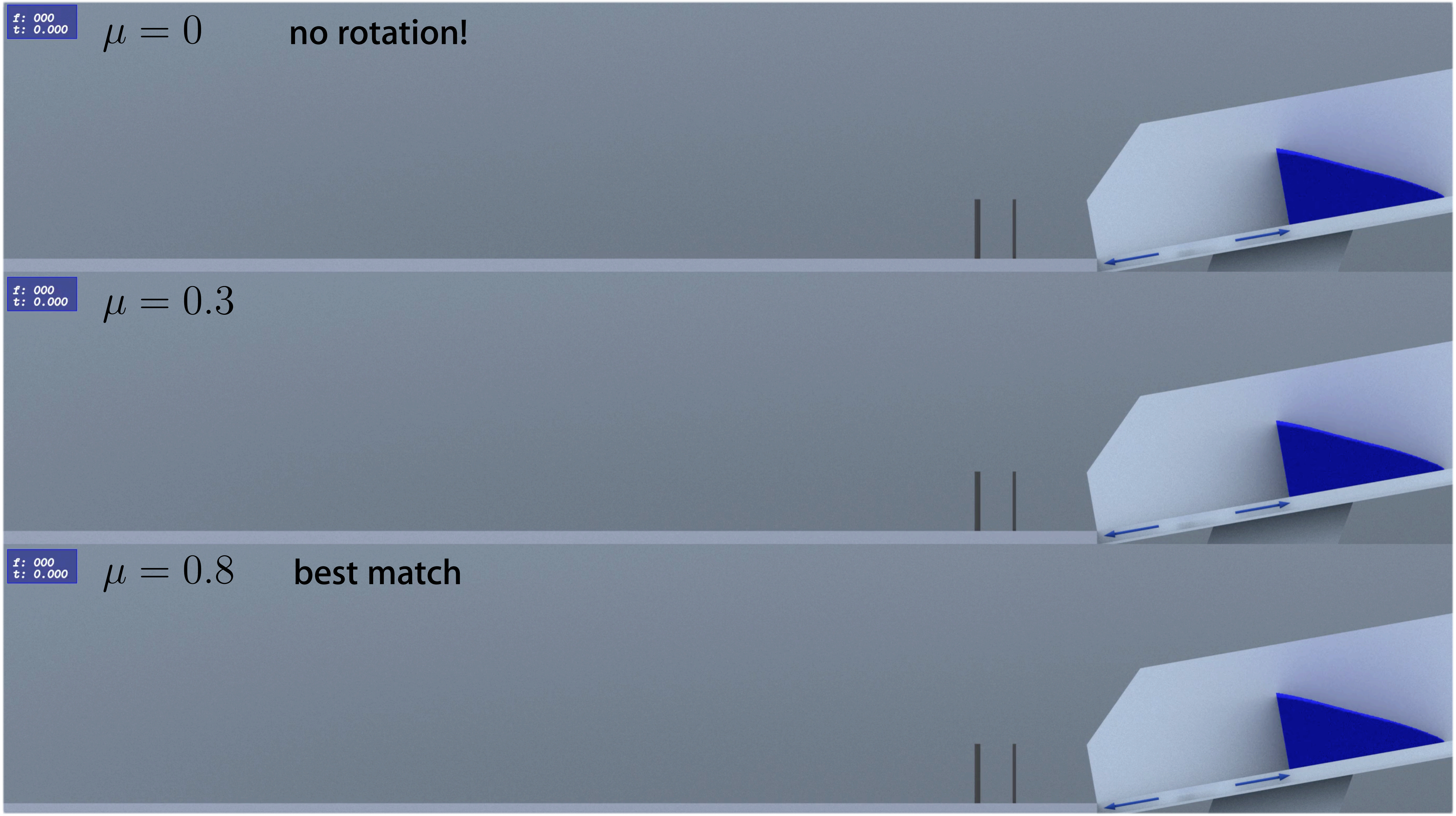
best match

Why Domain Decomposition / Load Balancing?



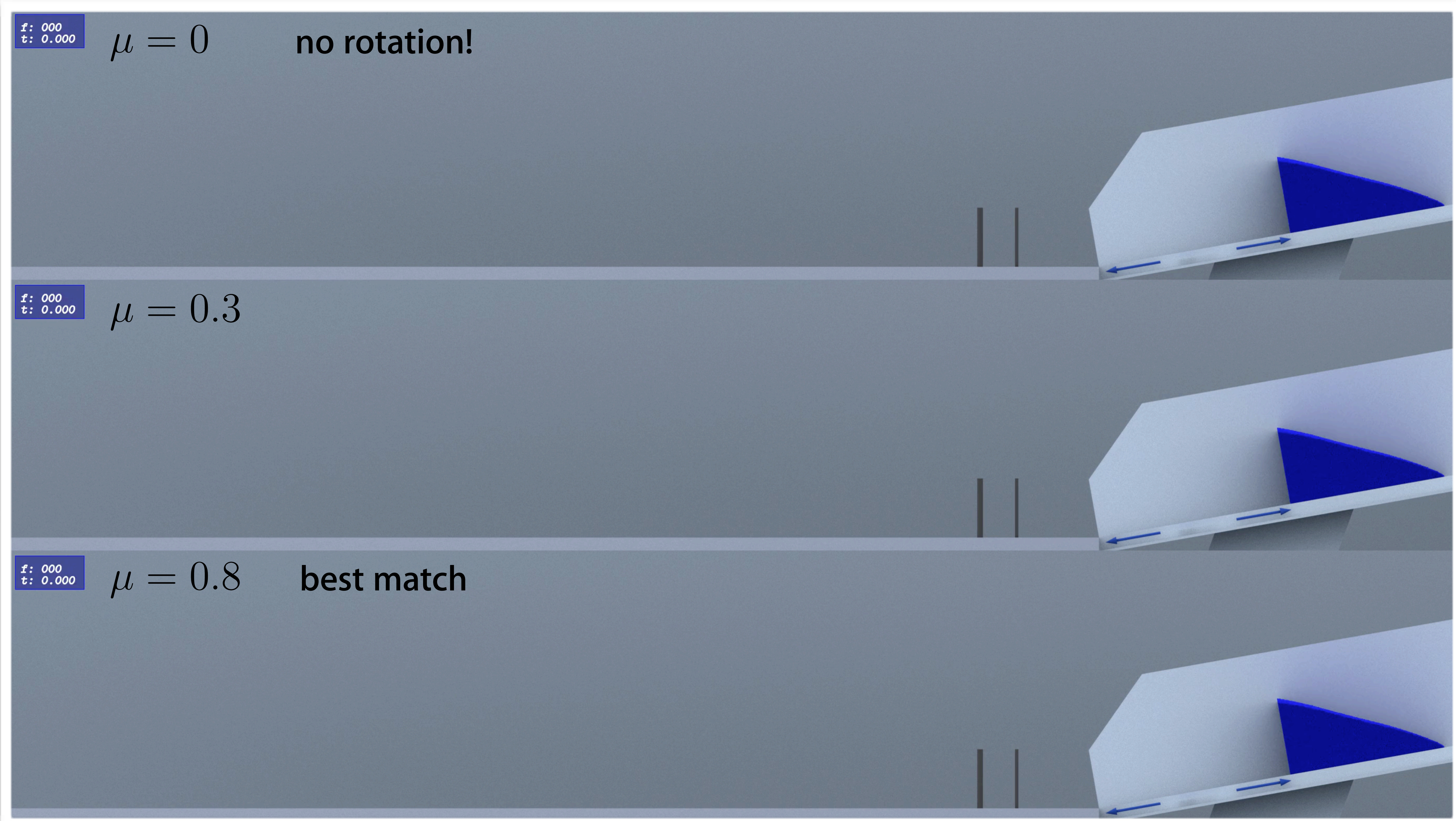
Why Domain Decomposition / Load Balancing?





$\Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter{=}1000} \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$

Velocity Magnitude



$\Delta t = 0.0002\,\mathrm{s}$ | maxIter=1000 | $\varepsilon_N, \varepsilon_T = 0$ | $n_T = 10^6$

Velocity Magnitude

Conclusion

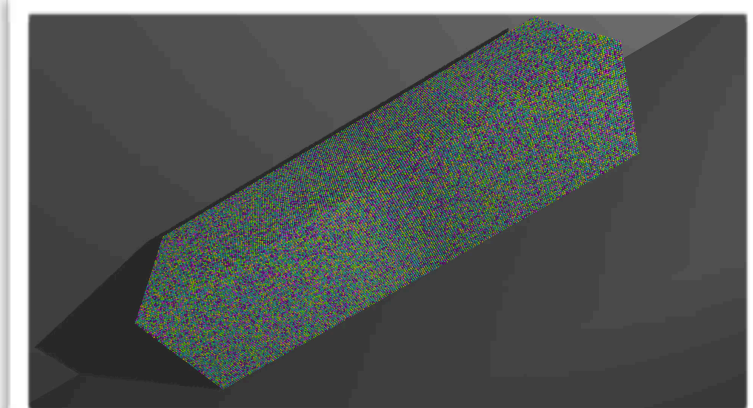
- Modeling granular media within the framework of **non-smooth rigid body dynamics.**
- Quality-conscious **open-source software**:

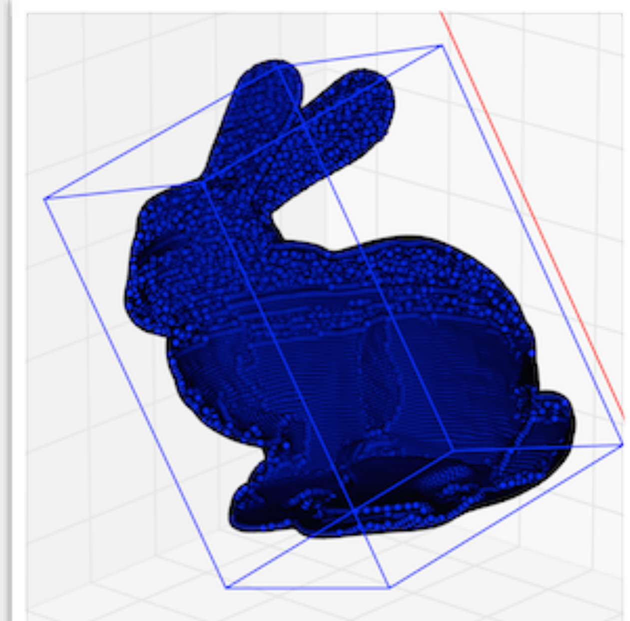
  GRS Framework:
  research tools to simulate granular rigid body dynamics (>10^6 bodies).

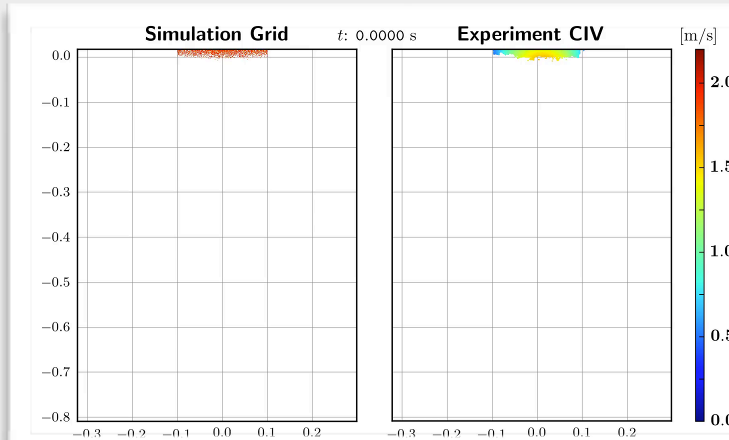
  ApproxMVBB Library:
  approx. minimal volume bounding box, generic kdTree , statistical filtering.

  HPCJobConfigurator:
  job configuration for high-performance computing with **python**

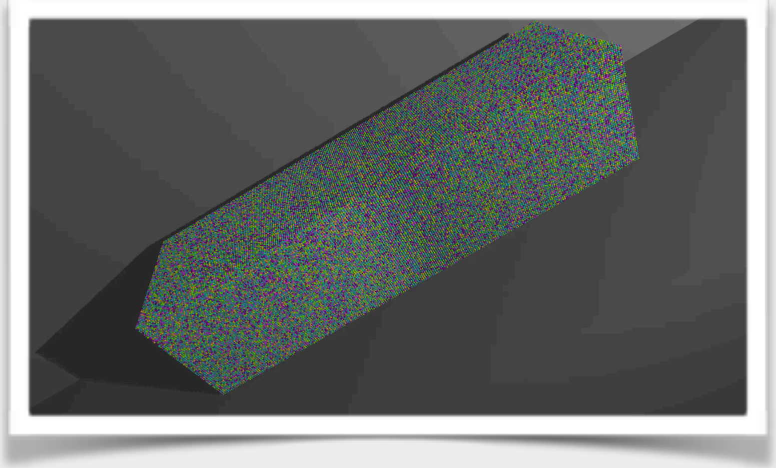
- First step towards visualization and validation of large-scale 3d granular simulations.







Conclusion

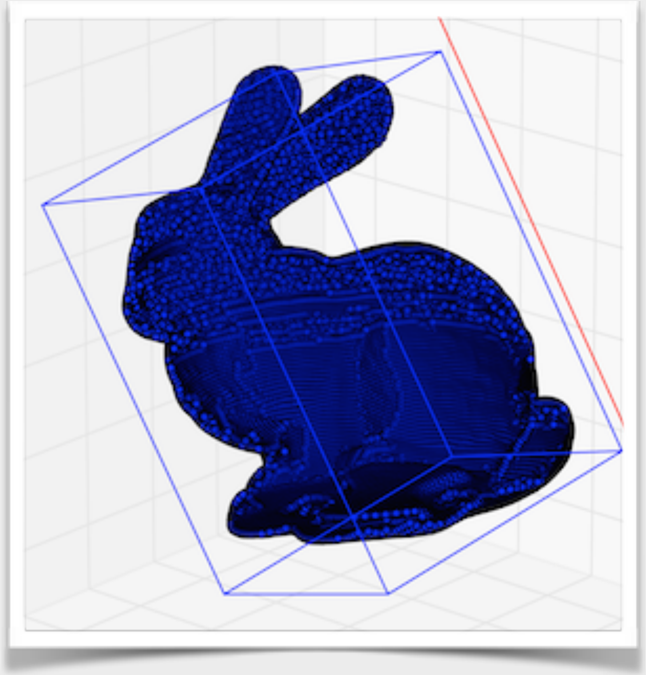


- Modeling granular media within the framework of **non-smooth rigid body dynamics.**
- Quality-conscious **open-source software**:

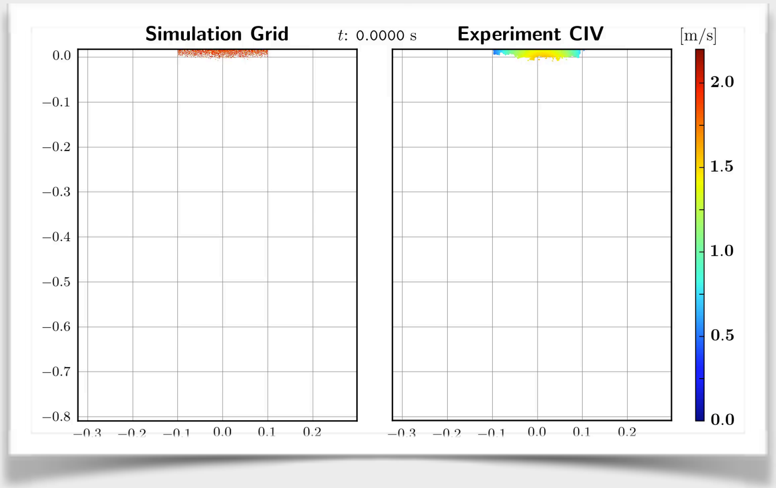
  GRS Framework:
  research tools to simulate granular rigid body dynamics (>10^6 bodies).

  ApproxMVBB Library:
  approx. minimal volume bounding box, generic kdTree , statistical filtering.

  HPCJobConfigurator:
  job configuration for high-performance computing with **python**

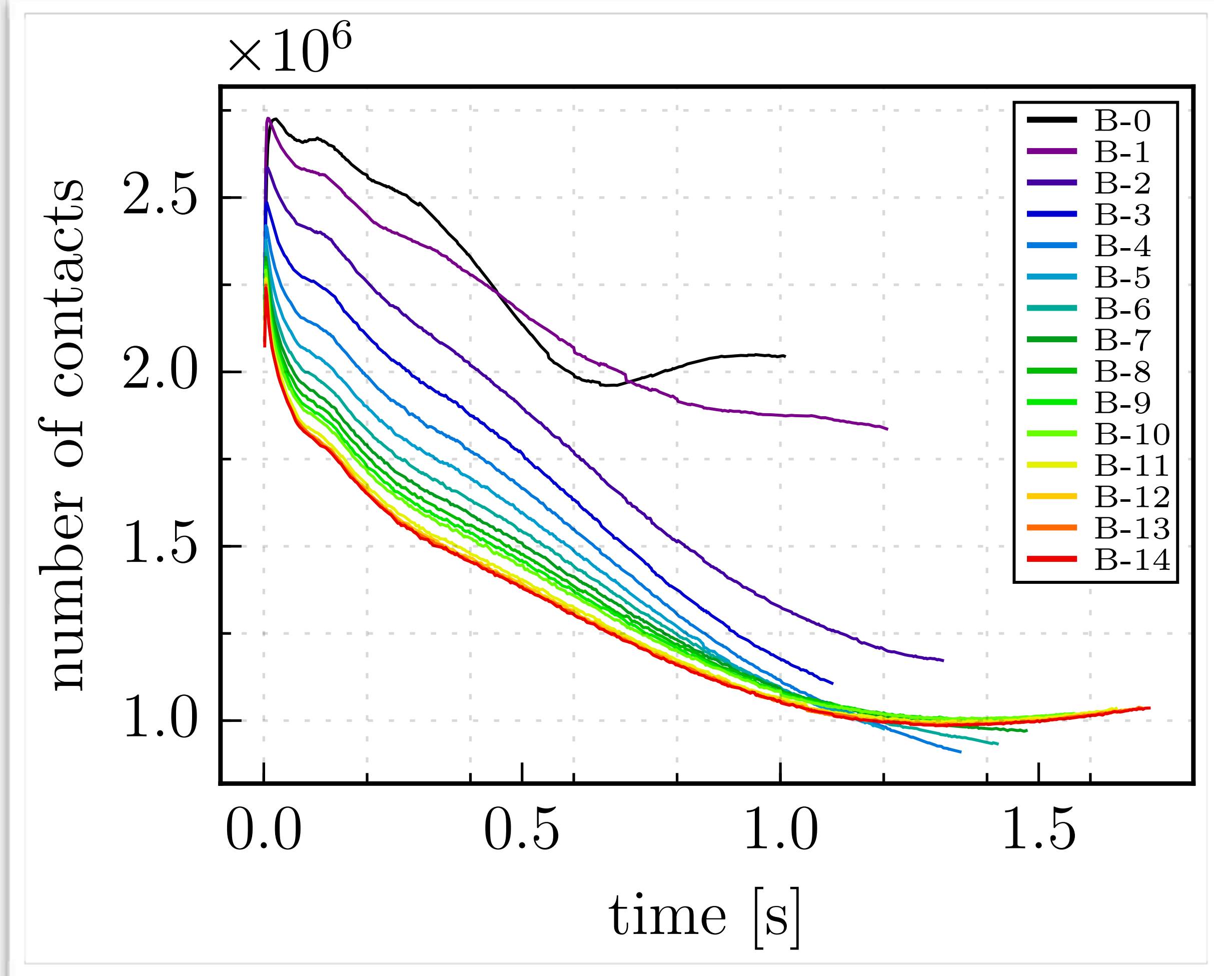


- First step towards visualization and validation of large-scale 3d granular simulations.

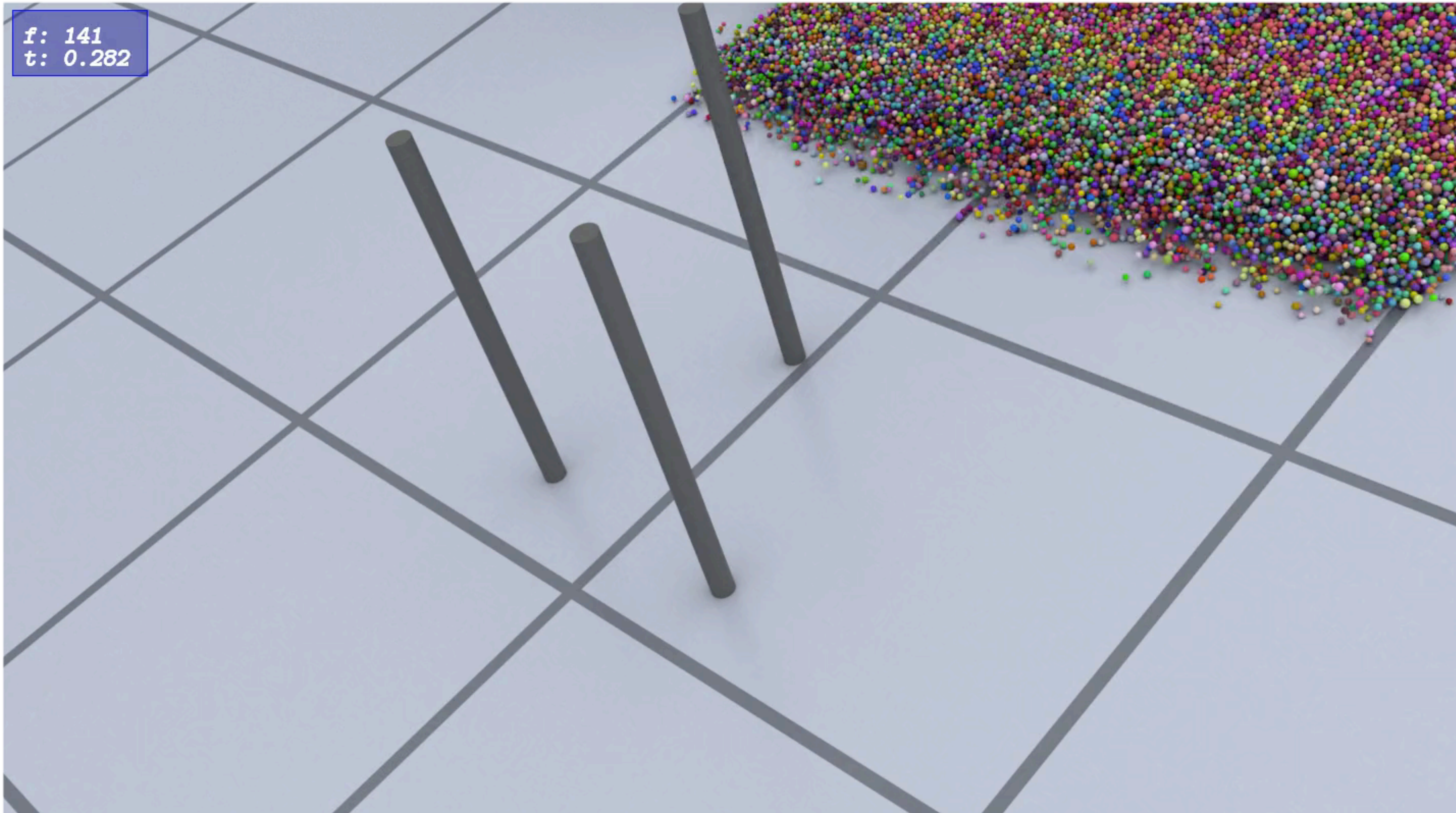


Outlook

- improve **contact problem solvers**:
  - accelerated projected gradient schemes
- more **advanced time-stepping** schemes:
  - variational integration schemes
- improve **experimental evaluations**:
  - velocity fluctuations
  - contact force distribution, stress tensor



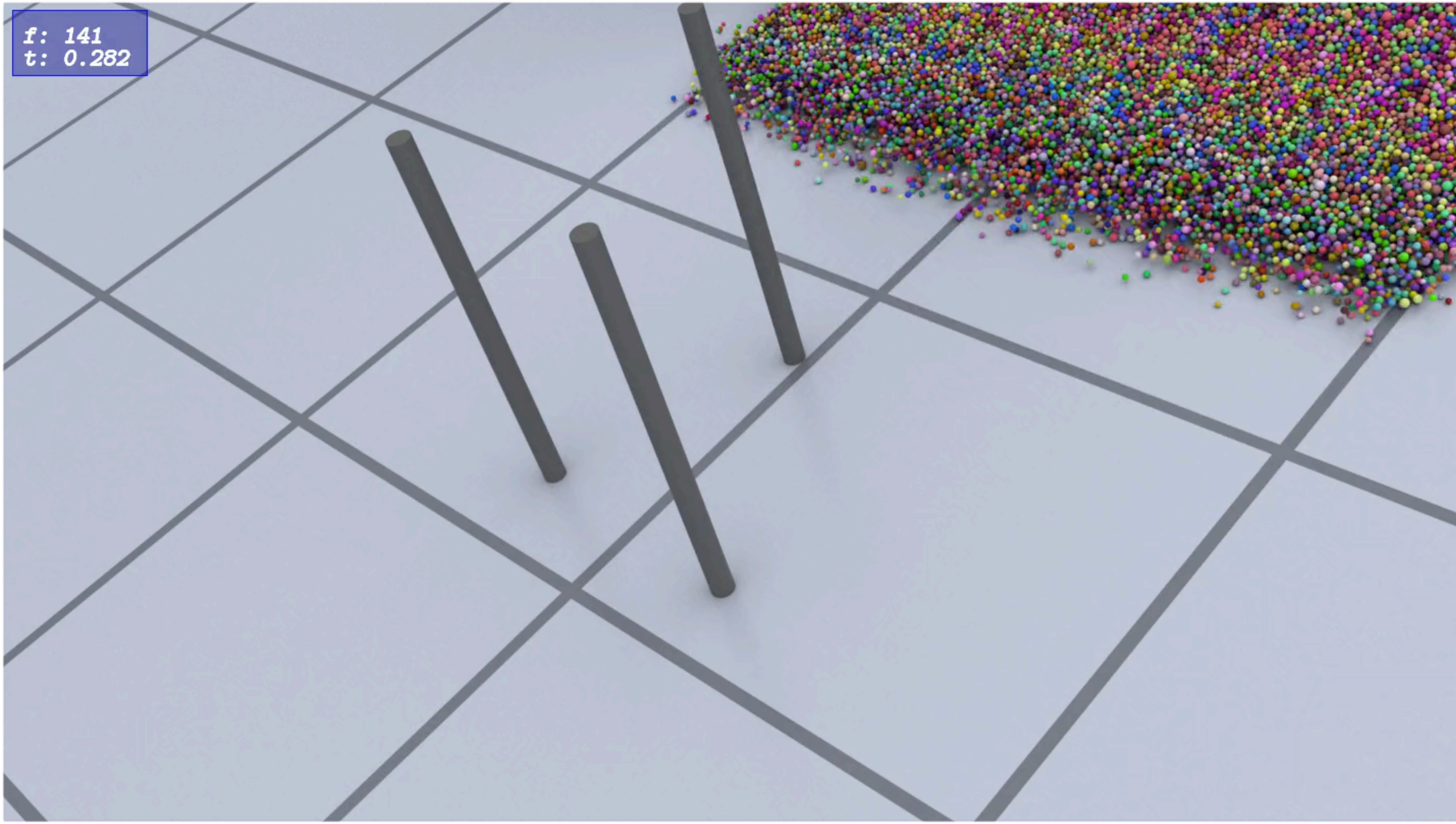
Velocity Magnitude



$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter{=}1000} \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$

best match

Velocity Magnitude

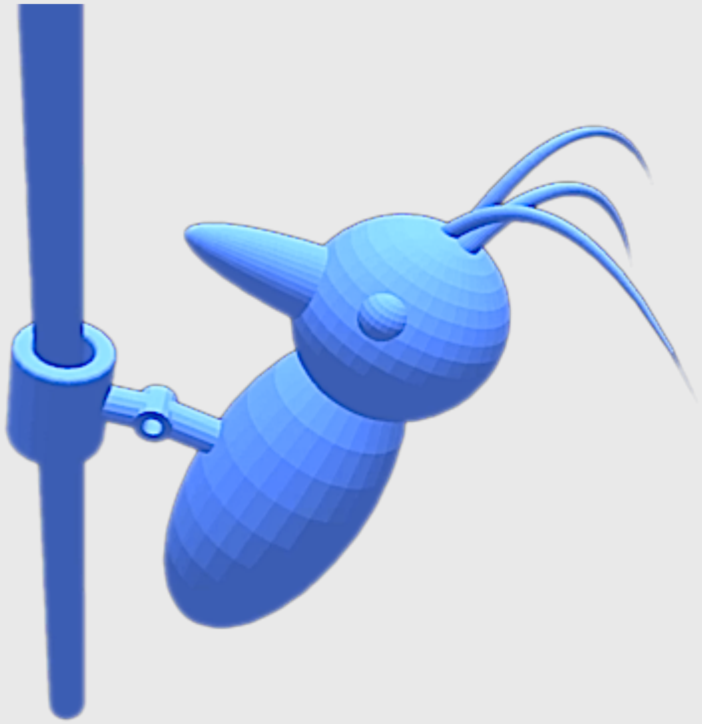


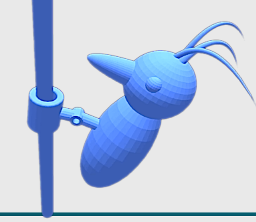
$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$

best match

Thanks for your attention!

Thanks for your attention!





Quaternions for Rotations

$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

The quaternion space is a non-commutative field.

$$\mathbf{P} \in \mathbb{H} := \{p_0\mathbf{I} + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}\ , p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

scalar part

pure part

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

Left multiplication: $\mathbf{PX} \xrightarrow{\mathcal{K}} \varphi_L(\mathbf{P})\mathbf{x}$

Right multiplication: $\mathbf{XP} \xrightarrow[\mathcal{K}]{} \varphi_R(\mathbf{P})\mathbf{x}$

$$\varphi_R(\mathbf{P}) \in \mathbb{R}^{4\times 4}, \quad \mathbf{x} \in \mathbb{R}^4$$

Quaternions for Rotations

$$\boldsymbol{\xi}_{\mathrm{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

The quaternion space is a non-commutative field.

$$\mathbf{P} \in \mathbb{H} := \{\underbrace{p_0 \mathbf{I}}_{\text{scalar part}} + \underbrace{p_1 \mathbf{i} + p_2 \mathbf{j} + p_3 \mathbf{k}}_{\text{pure part}}\ , p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

$$\left.\begin{array}{lll} \text{Left multiplication:} & \mathbf{PX} & \xrightarrow{\mathcal{K}} \varphi_L(\mathbf{P})\mathbf{x} \\ \text{Right multiplication:} & \mathbf{XP} & \xrightarrow[\mathcal{K}]{} \varphi_R(\mathbf{P})\mathbf{x} \end{array}\right\} \text{rotations in } \mathbb{E}^4 \text{ if } \|\mathbf{P}\| = 1$$

$$\varphi_R(\mathbf{P}) \in \mathbb{R}^{4\times 4}, \quad \mathbf{x} \in \mathbb{R}^4$$

Quaternions for Rotations

$$\boldsymbol{\xi}_{\text{rig}}(\boldsymbol{\rho}, t) := \mathcal{R}(t)(\boldsymbol{\rho}) + \mathbf{r}_R(t)$$

The quaternion space is a non-commutative field.

$$\mathbf{P} \in \mathbb{H} := \{\underbrace{p_0 \mathbf{I}}_{\text{scalar part}} + \underbrace{p_1 \mathbf{i} + p_2 \mathbf{j} + p_3 \mathbf{k}}_{\text{pure part}}, p_i \in \mathbb{R}\} \subset \mathbb{C}^{2\times 2}$$

Basis: $(\mathbf{I}, \mathbf{i}, \mathbf{j}, \mathbf{k})$

Coordinates:

$$\mathcal{K}(\mathbf{P}) = \begin{bmatrix} p_0 \\ \mathbf{p}_r \end{bmatrix} \in \mathbb{R}^4$$

$$\left.\begin{array}{lll} \text{Left multiplication:} & \mathbf{PX} \xrightarrow{\mathcal{K}} & \varphi_L(\mathbf{P})\mathbf{x} \\ \text{Right multiplication:} & \mathbf{XP} \xrightarrow[\mathcal{K}]{} & \varphi_R(\mathbf{P})\mathbf{x} \end{array}\right\} \text{rotations in } \mathbb{E}^4 \text{ if } \|\mathbf{P}\| = 1$$

$$\varphi_R(\mathbf{P}) \in \mathbb{R}^{4\times 4}, \quad \mathbf{x} \in \mathbb{R}^4$$

Rotations in $\mathbb{E}^4$ where $\mathbf{P}, \mathbf{Q}$ are unit quaternions (8 parameters + 2 constraints):

$$\begin{array}{rcl} \mathbb{H} & \to & \mathbb{H} \\ \mathbf{X} & \mapsto & \mathbf{PXQ} \end{array} \xrightarrow{\quad\mathcal{K}\quad} \begin{array}{rcl} \mathbb{R}^4 & \to & \mathbb{R}^4 \\ \mathbf{x} & \mapsto & \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x} \end{array}$$

Quaternions for Rotations

The awesome quaternion sandwich: $\mathbf{PXQ} \longleftrightarrow \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x}$

Quaternions for Rotations

The awesome quaternion sandwich: $\mathbf{PXQ} \longleftrightarrow \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x}$

Rotations in $\mathbb{E}^3$ where P is some quaternion and $\mathrm{Q} = \mathrm{P}^{-1}$:

Quaternions for Rotations

The awesome quaternion sandwich: $\mathbf{PXQ} \longleftrightarrow \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x}$

Rotations in $\mathbb{E}^3$ where P is some quaternion and $\mathrm{Q} = \mathrm{P}^{-1}$:

$$\begin{array}{ccc} \mathbb{R}^4 & \rightarrow & \mathbb{R}^4 \\ \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} & \mapsto & \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} = \frac{1}{s}\varphi_L(\mathbf{P})\varphi_R(\mathbf{P}^*)\begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \end{array} \qquad s := \|\mathbf{P}\|^2$$

Quaternions for Rotations

The awesome quaternion sandwich: $\mathbf{PXQ} \longleftrightarrow \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x}$

Rotations in $\mathbb{E}^3$ where P is some quaternion and $\mathrm{Q} = \mathrm{P}^{-1}$:

$$\begin{aligned} \mathbb{R}^4 &\to \mathbb{R}^4 \\ \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} &\mapsto \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} = \frac{1}{s}\varphi_L(\mathbf{P})\varphi_R(\mathbf{P}^*)\begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \end{aligned} \qquad s := \|\mathbf{P}\|^2$$

Quaternions for Rotations

The awesome quaternion sandwich: $\mathbf{P}\mathbf{X}\mathbf{Q} \longleftrightarrow \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x}$

Rotations in $\mathbb{E}^3$ where P is some quaternion and $\mathrm{Q} = \mathbf{P}^{-1}$:

$$\begin{array}{ccc} \mathbb{R}^4 & \rightarrow & \mathbb{R}^4 \\ \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} & \mapsto & \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} = \frac{1}{s}\boxed{\varphi_L(\mathbf{P})\varphi_R(\mathbf{P}^*)}\begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \end{array} \qquad s := \|\mathbf{P}\|^2$$

scalable body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{scal}} = \boxed{s\ {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})}\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

rigid body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = \boxed{{}_{\mathrm{I}}\mathbf{R}(\mathbf{p})}\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

constraint: $\|\mathbf{p}\| = 1 \quad \forall t$

Quaternions for Rotations

The awesome quaternion sandwich: $\mathbf{PXQ} \longleftrightarrow \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x}$

Rotations in $\mathbb{E}^3$ where P is some quaternion and $\mathrm{Q} = \mathrm{P}^{-1}$:

$$\begin{array}{ccc} \mathbb{R}^4 & \to & \mathbb{R}^4 \\ \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} & \mapsto & \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} = \frac{1}{s}\boxed{\varphi_L(\mathbf{P})\varphi_R(\mathbf{P}^*)} \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \end{array} \qquad s := \|\mathbf{P}\|^2$$

scalable body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{scal}} = \boxed{s\ {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})}\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

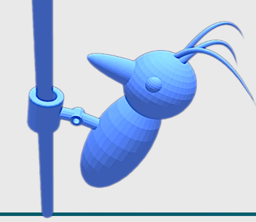
rigid body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = \boxed{{}_{\mathrm{I}}\mathbf{R}(\mathbf{p})}\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

constraint: $\|\mathbf{p}\| = 1 \quad \forall t$



Quaternions for Rotations

The awesome quaternion sandwich: $\mathbf{PXQ} \longleftrightarrow \varphi_L(\mathbf{P})\varphi_R(\mathbf{Q})\mathbf{x}$

Rotations in $\mathbb{E}^3$ where $\mathbf{P}$ is some quaternion and $\mathbf{Q} = \mathbf{P}^{-1}$:

$$\begin{array}{ccc} \mathbb{R}^4 & \rightarrow & \mathbb{R}^4 \\ \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} & \mapsto & \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} = \frac{1}{s}\varphi_L(\mathbf{P})\varphi_R(\mathbf{P}^*)\begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix} \end{array} \qquad s := \|\mathbf{P}\|^2$$

scalable body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{scal}} = s\ {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

rigid body motion:

$${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

generalized coordinates:

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

constraint: $\|\mathbf{p}\| = 1 \quad \forall t$

equation of motions
by principles

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$\boxed{0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W_I^{\mathrm{int}} + \delta W_C^{\mathrm{int}} + \delta W^{\mathrm{ext}} \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t}$$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $$\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\;{}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$$

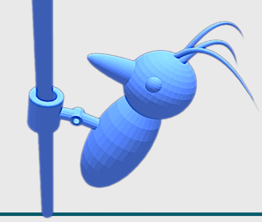
$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$\boxed{0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \boxed{\delta W^{\mathrm{dyn}}} + \delta W_I^{\mathrm{int}} + \delta W_C^{\mathrm{int}} + \delta W^{\mathrm{ext}} \qquad \forall_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\; t}$$

$$0 = \boxed{-\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\, \boxed{{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\mathrm{d}m}}$$

constitutive force law:
Newton's second law



Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\quad {}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

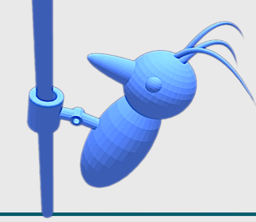
$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W_I^{\mathrm{int}} + \delta W_C^{\mathrm{int}} + \delta W^{\mathrm{ext}} \qquad \forall_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\, t$$

law of interaction
(S. Eugster 2014)

$$0 = -\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\, {}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\,\mathrm{d}m \quad + \quad 0$$

constitutive force law:
Newton's second law



Applying the Fundamental Axioms

Equations of motion for the rigid body:

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

rigid body motion: $\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\;{}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W_I^{\mathrm{int}} + \delta W_C^{\mathrm{int}} + \delta W^{\mathrm{ext}} \qquad \forall\, {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\; t$$

law of interaction
(S. Eugster 2014)

perfect constraint forces $(\infty \to 6)$ by d'Alembert-Lagrange

$$0 = \boxed{-\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}{}^{\top}\; \boxed{{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\,\mathrm{d}m}} + 0 + 0$$

constitutive force law:
Newton's second law

Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\boxed{{}_I\boldsymbol{\xi}_{\text{rig}} = {}_I\mathbf{R}(\mathbf{p})\ {}_I\boldsymbol{\rho} + {}_I\mathbf{r}_R}$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_I\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

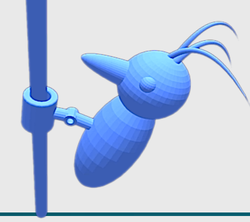
$$\boxed{0 = \delta W({}_I\delta\boldsymbol{\xi}_{\text{rig}}) = \delta W^{\text{dyn}} + \delta W_I^{\text{int}} + \delta W_C^{\text{int}} + \delta W^{\text{ext}} \qquad \forall {}_I\delta\boldsymbol{\xi}_{\text{rig}},\ t}$$

law of interaction
(S. Eugster 2014)

perfect constraint forces $(\infty \to 6)$ by d'Alembert-Lagrange

$$0 = \boxed{-\int_B {}_I\delta\boldsymbol{\xi}_{\text{rig}}^\top \boxed{{}_I\ddot{\boldsymbol{\xi}}_{\text{rig}}\mathrm{d}m}} + 0 + 0 + \int_B {}_I\delta\boldsymbol{\xi}_{\text{rig}}^\top\ {}_I\mathrm{d}\mathbf{F}^{\text{e}}$$

constitutive force law:
Newton's second law



Applying the Fundamental Axioms

Equations of motion for the rigid body:

rigid body motion: $\boxed{{}_\mathrm{I}\boldsymbol{\xi}_\mathrm{rig} = {}_\mathrm{I}\mathbf{R}(\mathbf{p})\ {}_\mathrm{I}\boldsymbol{\rho} + {}_\mathrm{I}\mathbf{r}_R}$

$$\|\mathbf{p}\| = 1 \quad \forall t$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_\mathrm{I}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$\boxed{0 = \delta W({}_\mathrm{I}\delta\boldsymbol{\xi}_\mathrm{rig}) = \delta W^\mathrm{dyn} + \delta W_I^\mathrm{int} + \delta W_C^\mathrm{int} + \delta W^\mathrm{ext} \qquad \forall_\mathrm{I}\delta\boldsymbol{\xi}_\mathrm{rig},\ t}$$

law of interaction
(S. Eugster 2014)

perfect constraint forces ($\infty \to 6$) by d'Alembert-Lagrange

$$0 = \boxed{-\int_B {}_\mathrm{I}\delta\boldsymbol{\xi}_\mathrm{rig}{}^\top \boxed{{}_\mathrm{I}\ddot{\boldsymbol{\xi}}_\mathrm{rig}\mathrm{d}m}} + 0 + 0 + \boxed{\int_B {}_\mathrm{I}\delta\boldsymbol{\xi}_\mathrm{rig}{}^\top\ {}_\mathrm{I}\mathrm{d}\mathbf{F}^\mathrm{e}}$$

constitutive force law:
Newton's second law

velocities: ${}_\mathrm{I}\dot{\boldsymbol{\xi}}_\mathrm{rig} = {}_\mathrm{K}\mathbf{R}\ {}_\mathrm{K}\tilde{\boldsymbol{\Omega}}\ {}_\mathrm{K}\boldsymbol{\rho}_t + {}_\mathrm{I}\dot{\mathbf{r}}_R$

Applying the Fundamental Axioms

Equations of motion for the rigid body:

$$\|\mathbf{p}\| = 1 \quad \forall t$$

rigid body motion: $${}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{rig}} = {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R$$

$$\mathbf{q}(t) := \begin{bmatrix} {}_{\mathrm{I}}\mathbf{r}_R \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^7$$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}) = \delta W^{\mathrm{dyn}} + \delta W_I^{\mathrm{int}} + \delta W_C^{\mathrm{int}} + \delta W^{\mathrm{ext}} \qquad \forall {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}},\ t$$

law of interaction (S. Eugster 2014)

perfect constraint forces ($\infty \rightarrow 6$) by d'Alembert-Lagrange

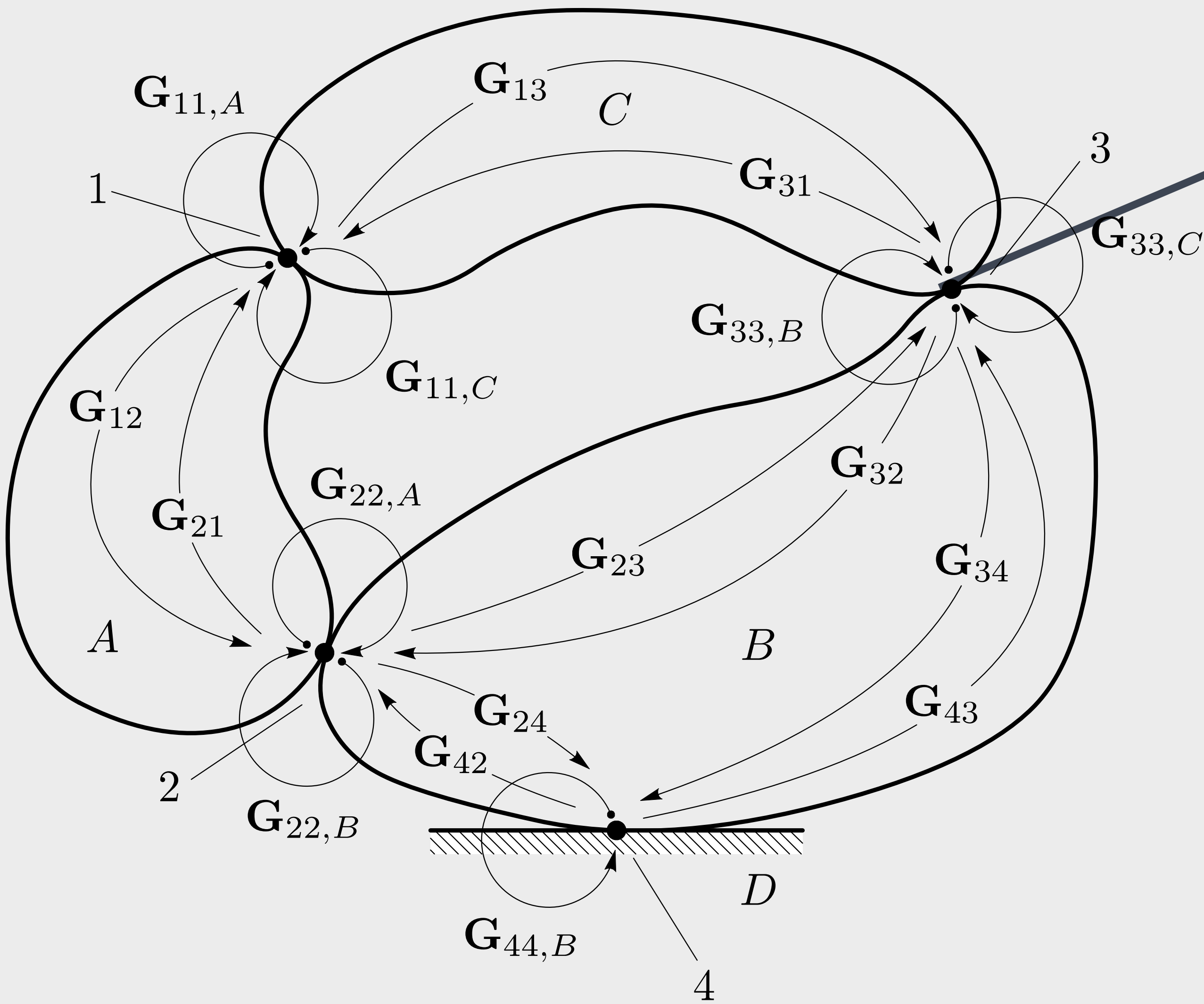
$$0 = -\int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}^{\top}\ \boxed{{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{rig}}\mathrm{d}m} + 0 + 0 + \int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{rig}}^{\top}\ {}_{\mathrm{I}}\mathrm{d}\mathbf{F}^{\mathrm{e}}$$

constitutive force law:
Newton's second law

velocities: $${}_{\mathrm{I}}\dot{\boldsymbol{\xi}}_{\mathrm{rig}} = {}_{\mathrm{K}}\mathbf{R}\ {}_{\mathrm{K}}\tilde{\boldsymbol{\Omega}}\ {}_{\mathrm{K}}\boldsymbol{\rho}_t + {}_{\mathrm{I}}\dot{\mathbf{r}}_R$$

generalized velocities: $$\mathbf{u} := \begin{bmatrix} {}_{\mathrm{I}}\dot{\mathbf{r}}_R \\ {}_{\mathrm{K}}\boldsymbol{\Omega} \end{bmatrix} \in \mathbb{R}^6$$

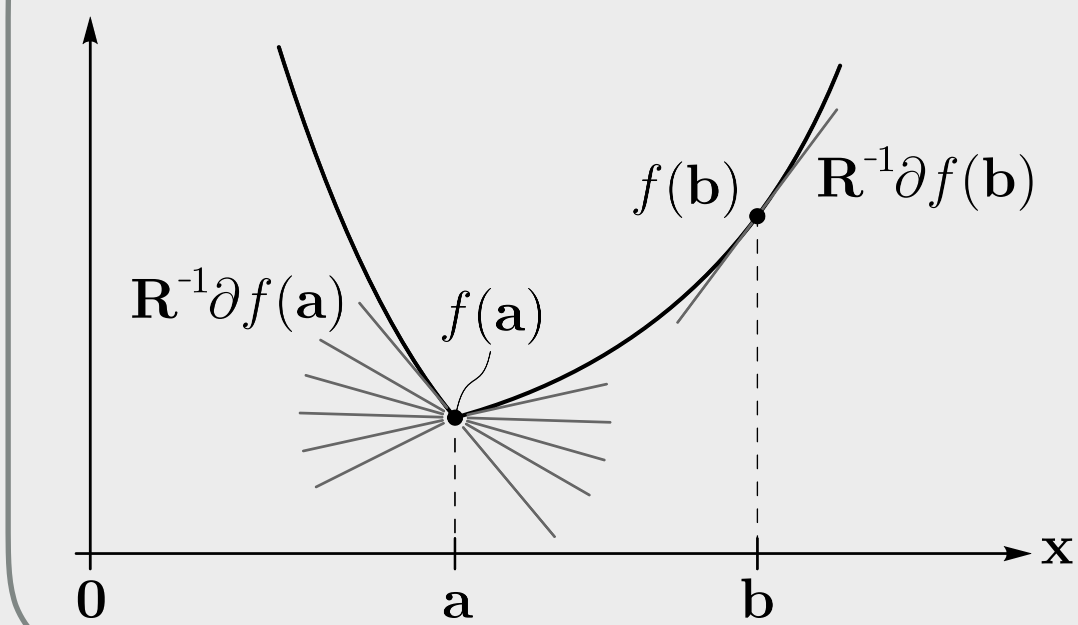
Non-Smooth Dynamics in a Nutshell



contact law

$$-\boldsymbol{\gamma}_i \in \mathcal{N}_{\mathcal{C}_i}(\boldsymbol{\lambda})$$

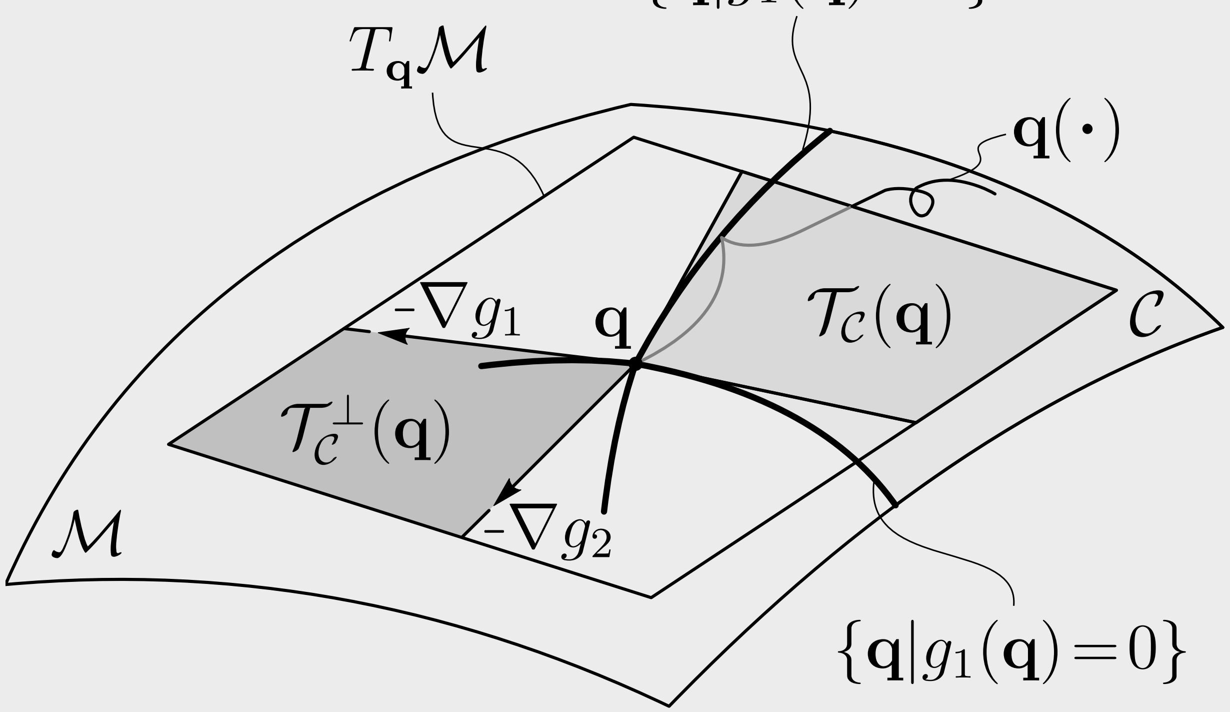
subdifferential

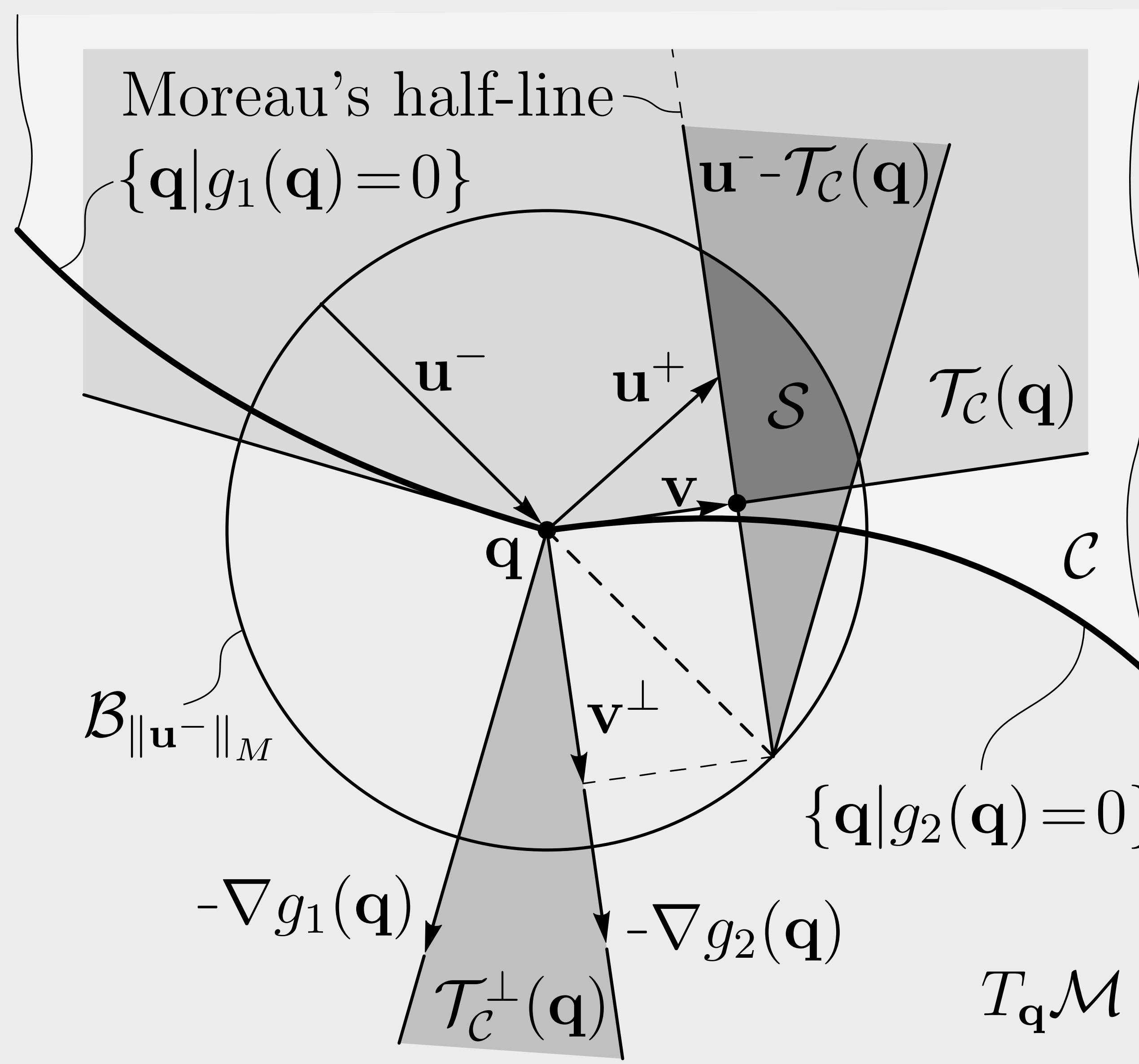


$$\min_{\boldsymbol{\lambda}\in\mathcal{C}} f(\mathbf{x}) = \min_{\boldsymbol{\lambda}\in\mathcal{C}} \frac{1}{2}\boldsymbol{\lambda}^\top \mathbf{G}\boldsymbol{\lambda} + \mathbf{c}^\top\boldsymbol{\lambda} \quad \Leftrightarrow \quad \min_{\boldsymbol{\lambda}} \frac{1}{2}\boldsymbol{\lambda}^\top \mathbf{G}\boldsymbol{\lambda} + \mathbf{c}^\top\boldsymbol{\lambda} + I_{\mathcal{C}}(\boldsymbol{\lambda})$$

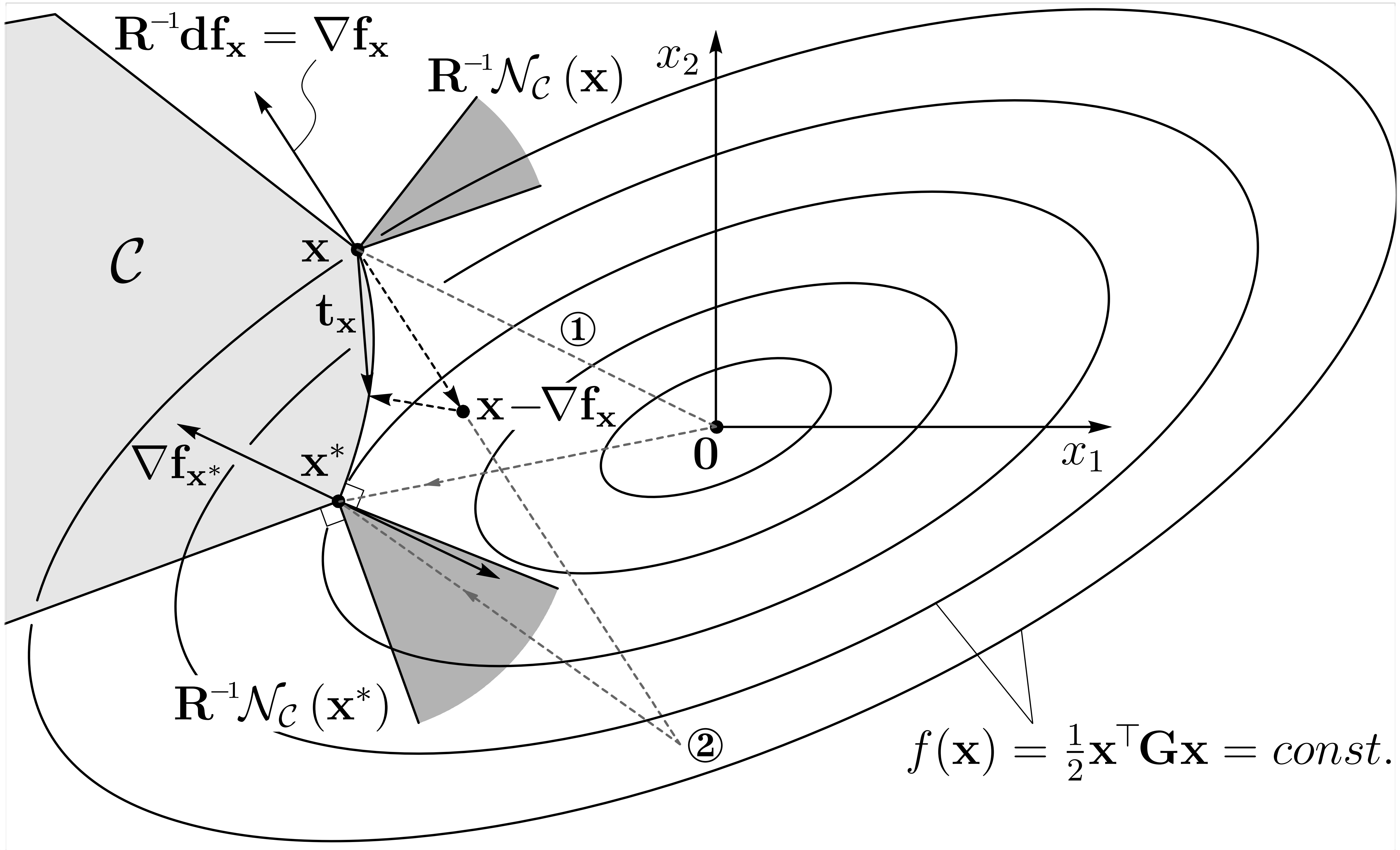
necesssary condition for minimizer: $-\mathbf{df}(\mathbf{x}^*) \in \partial I_{\mathcal{C}}(\boldsymbol{\lambda}) \equiv \mathcal{N}_{\mathcal{C}}(\mathbf{x}^*)$

Non-Smooth Dynamics in a Nutshell

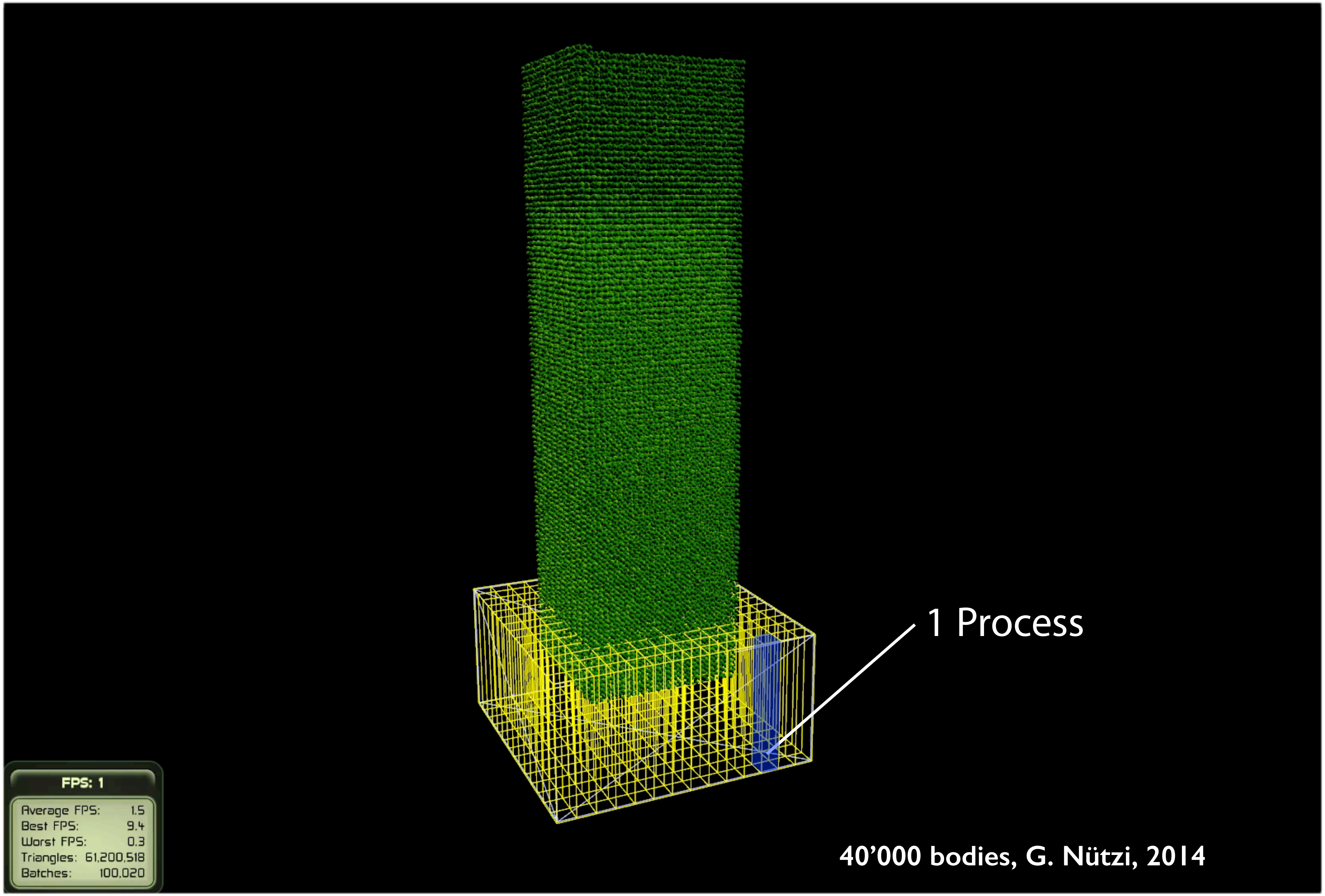




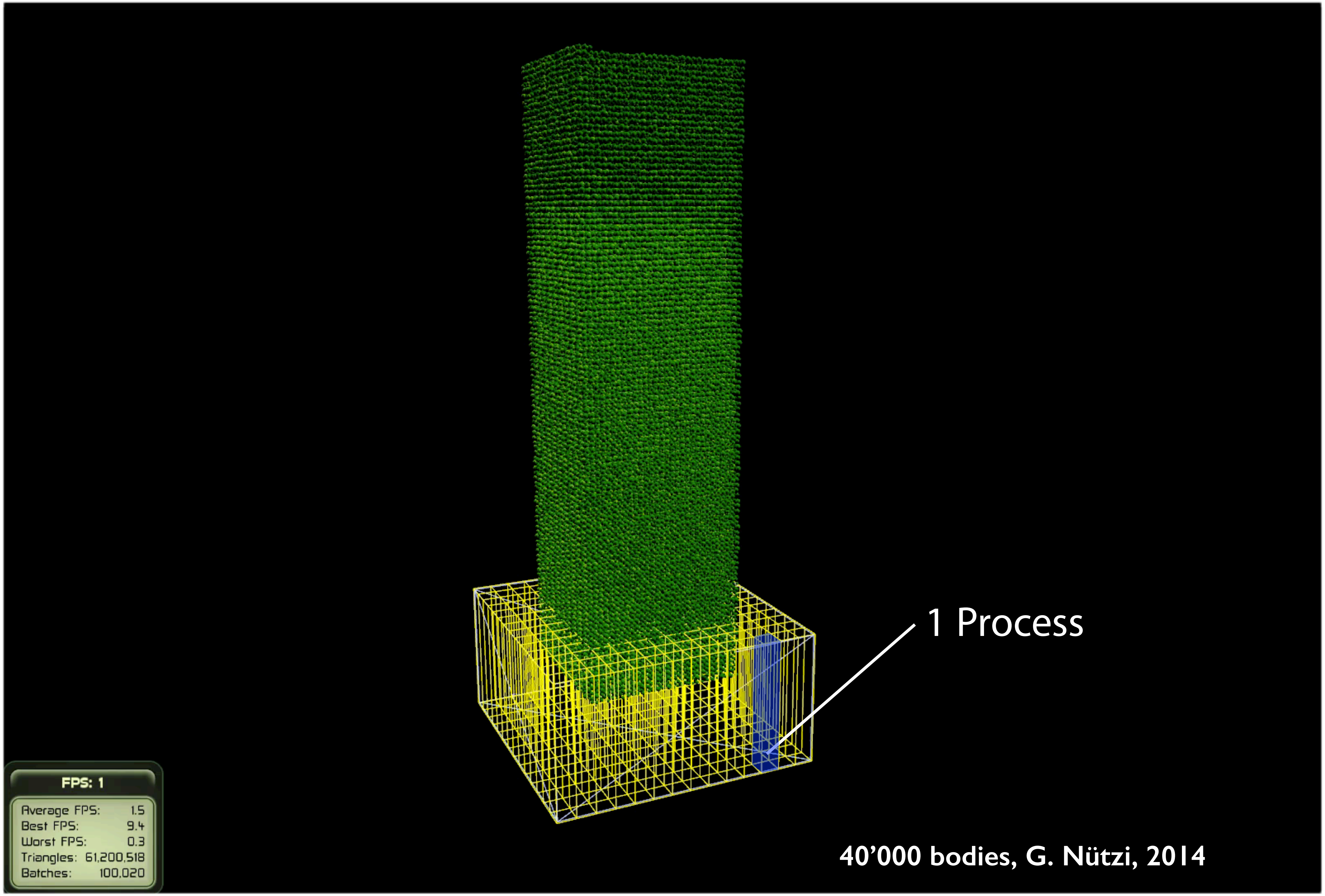
Non-Smooth Dynamics in a Nutshell



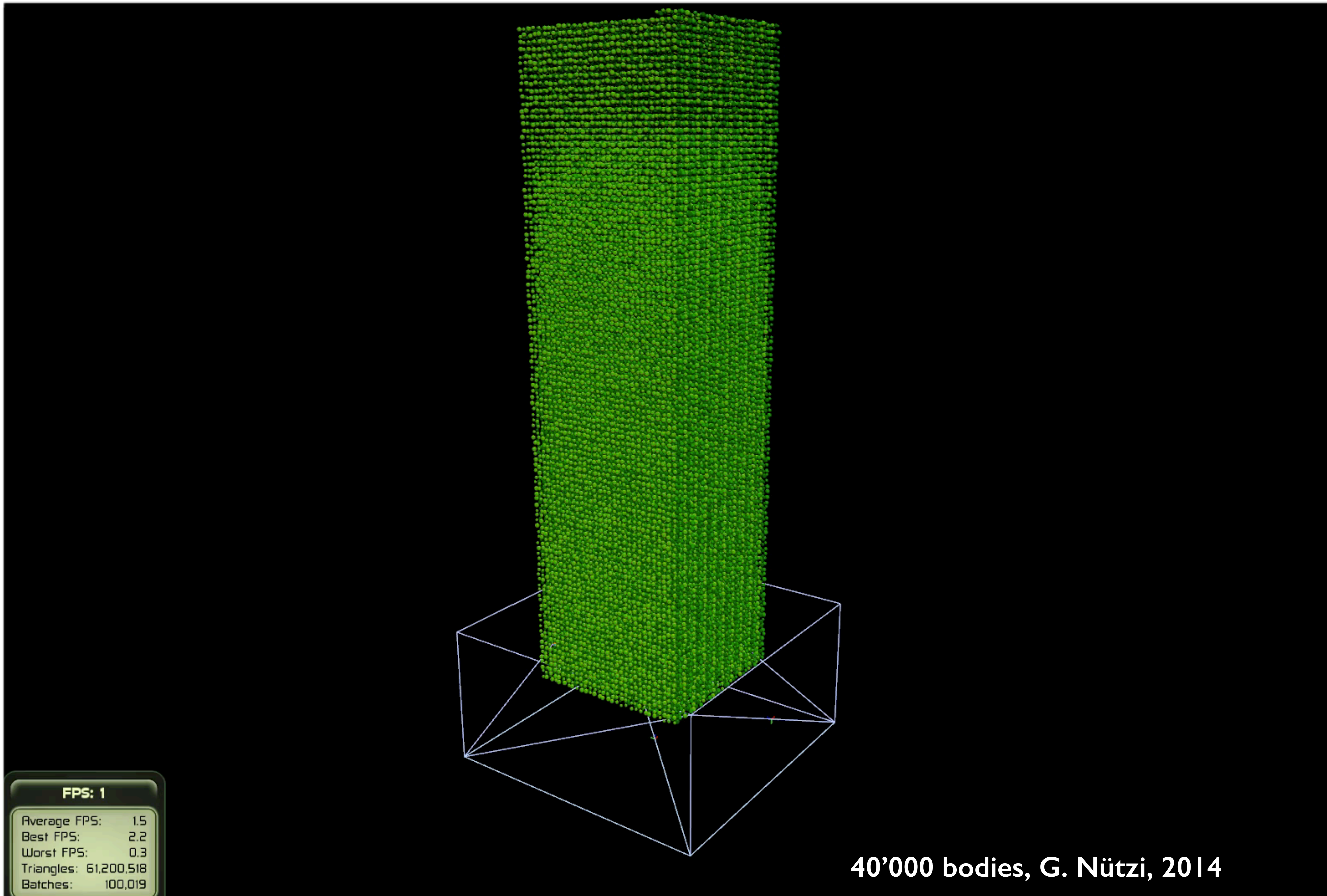
Force Laws for Contacts



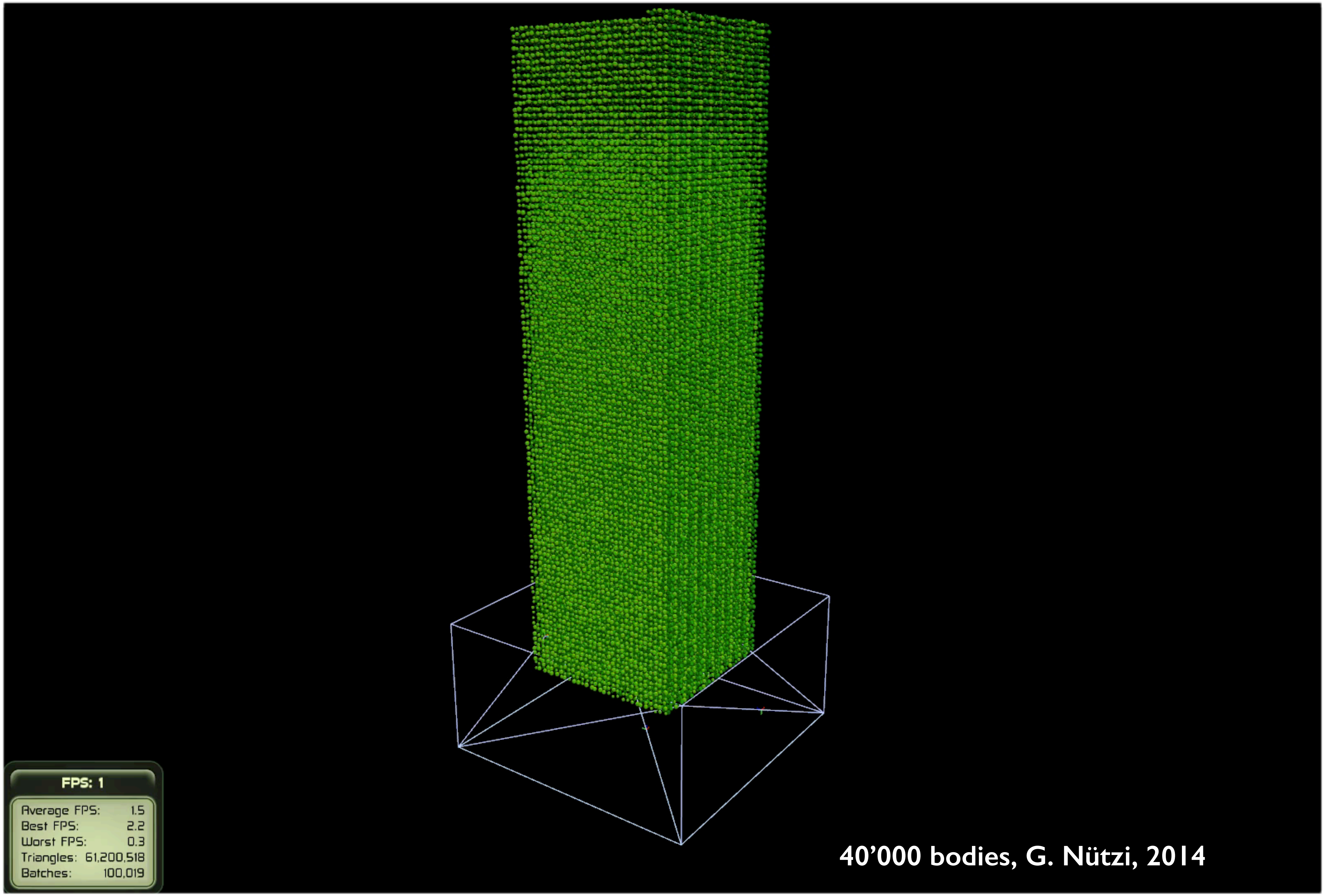
Force Laws for Contacts



Force Laws for Contacts

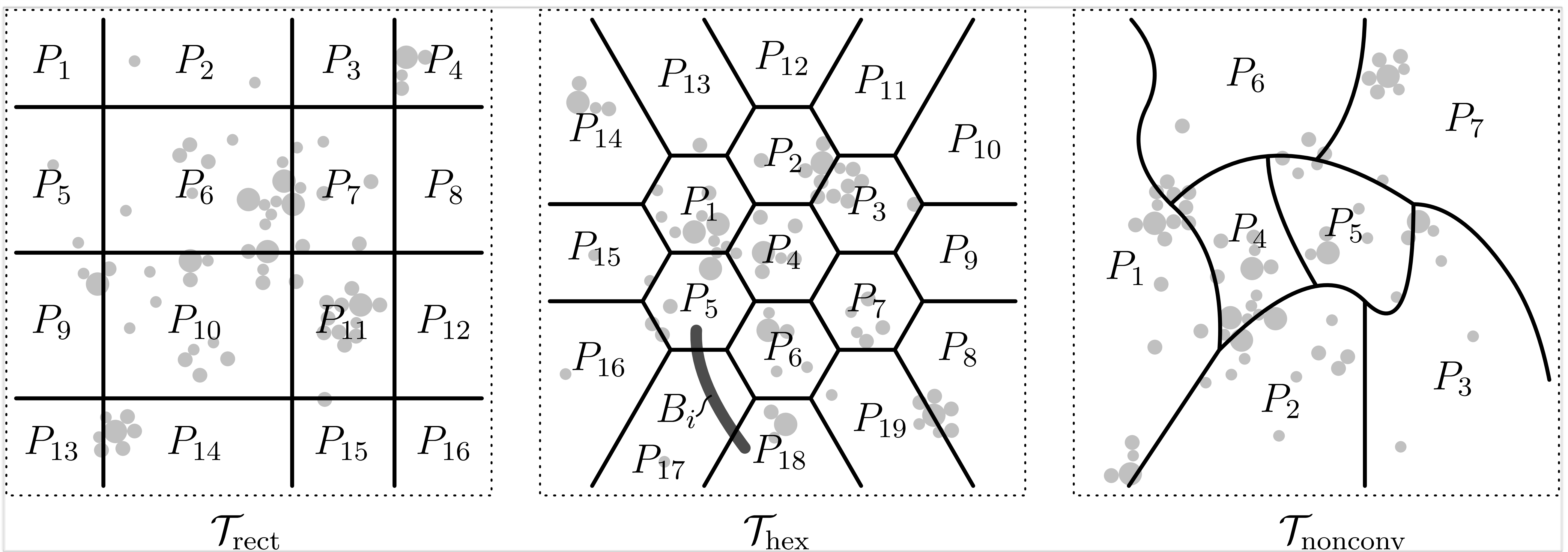


Force Laws for Contacts



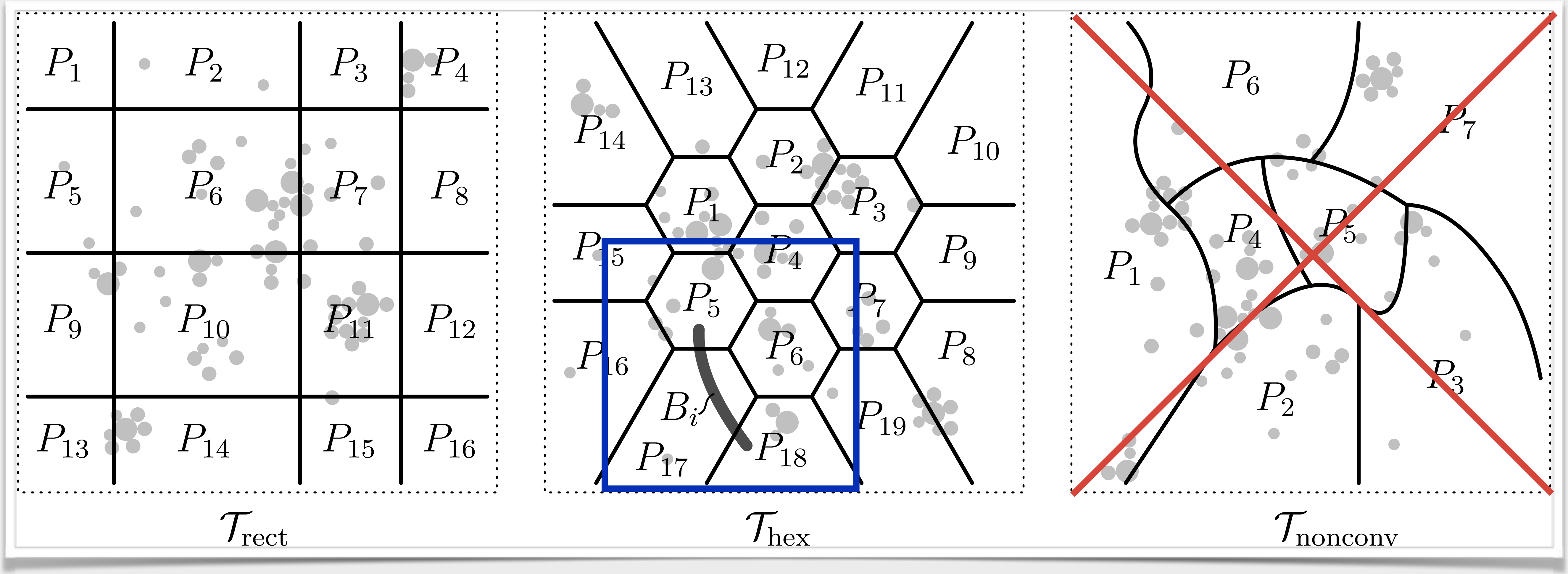
The Granular Rigid Body Simulation Framework

Parallel Implementation: Domain Decomposition



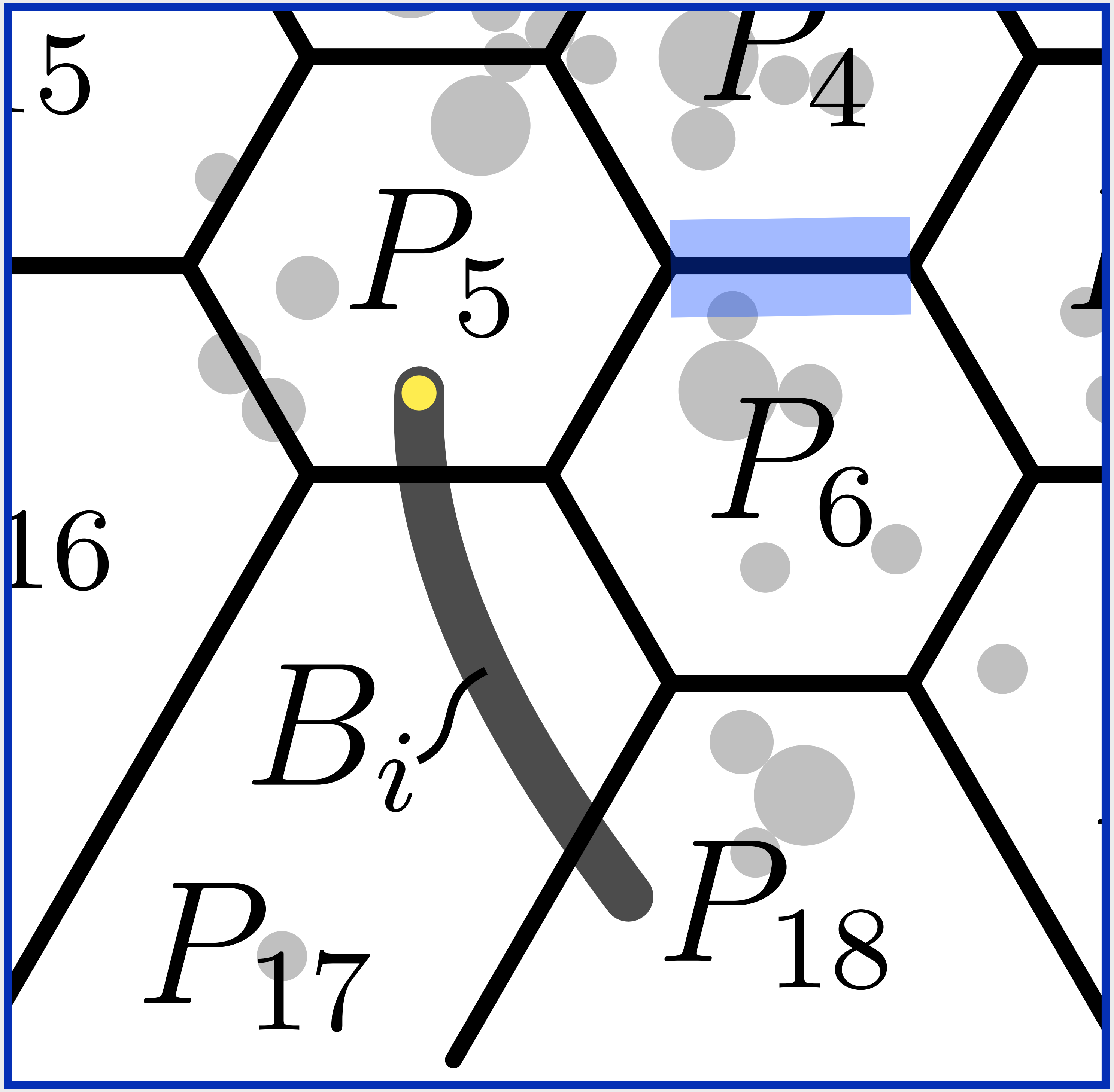
The Granular Rigid Body Simulation Framework

Parallel Implementation: Domain Decomposition



The Granular Rigid Body Simulation Framework

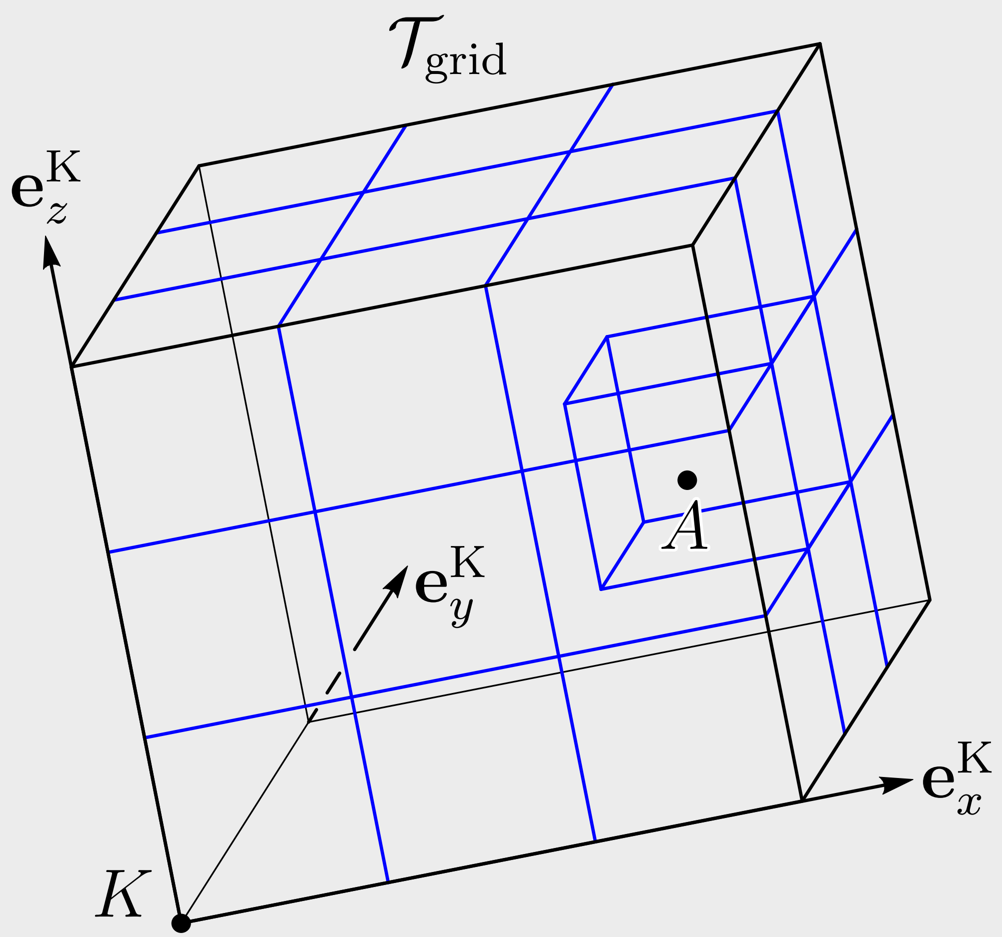
Parallel Implementation: Domain Decomposition



- communicate only with direct neighbourhood
- **restriction** on body size
- **no** epsilon boundary region as in **pe** engine or **chrono**

The Granular Rigid Body Simulation Framework

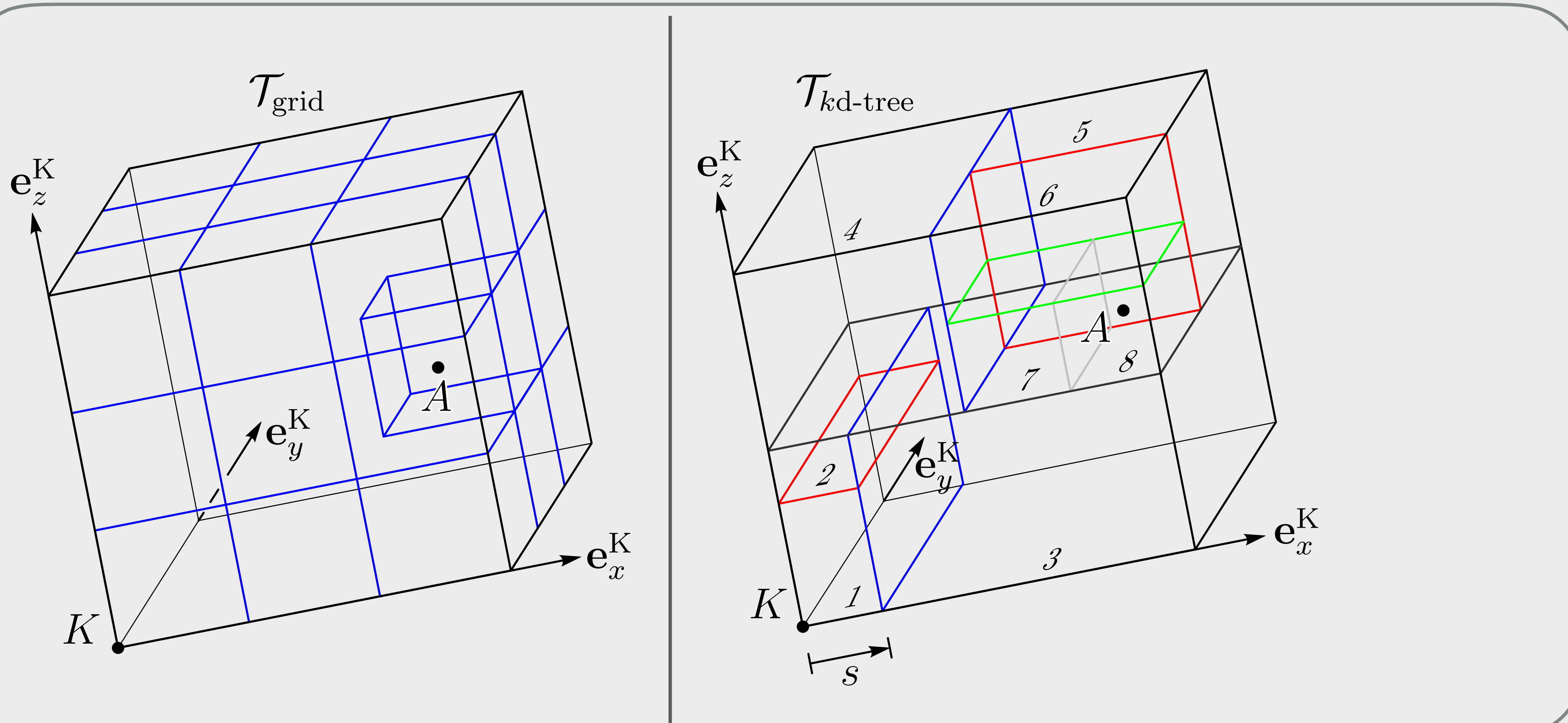
Parallel Implementation: Domain Decomposition



- **unbalanced** workload
- simple data structure
- 26 neighbors

The Granular Rigid Body Simulation Framework

Parallel Implementation: Domain Decomposition

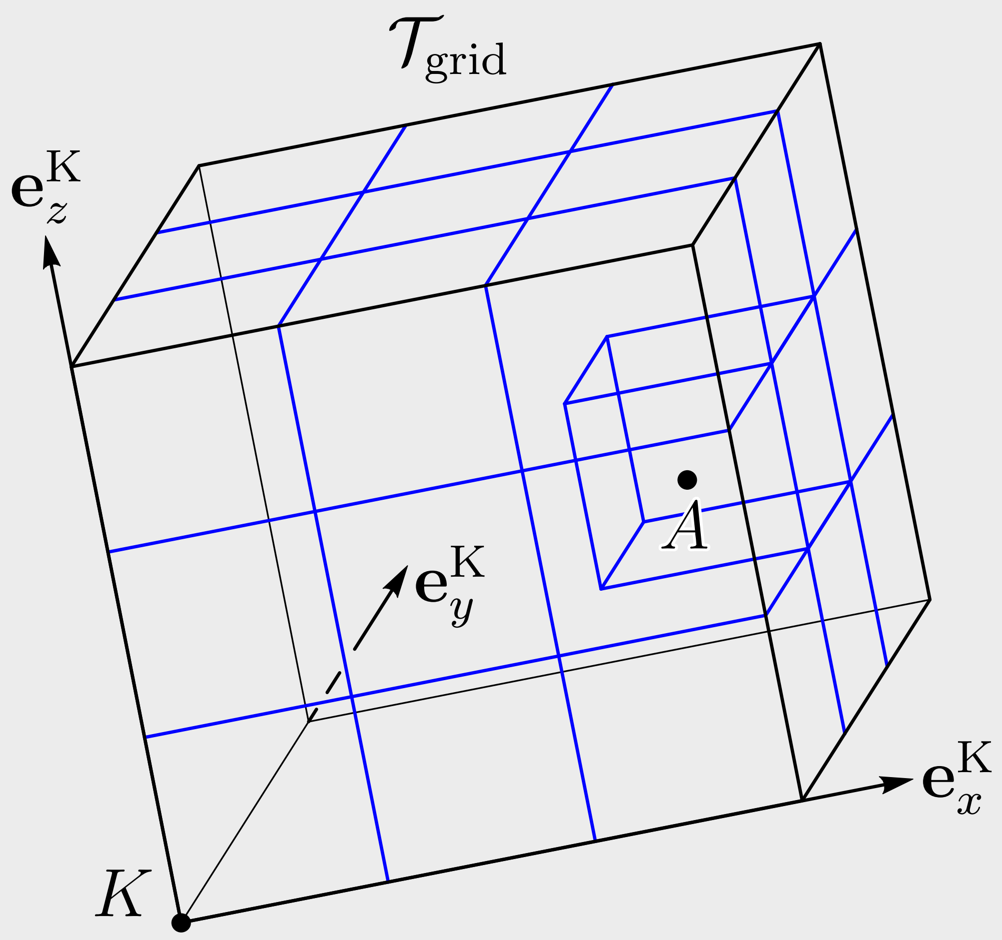


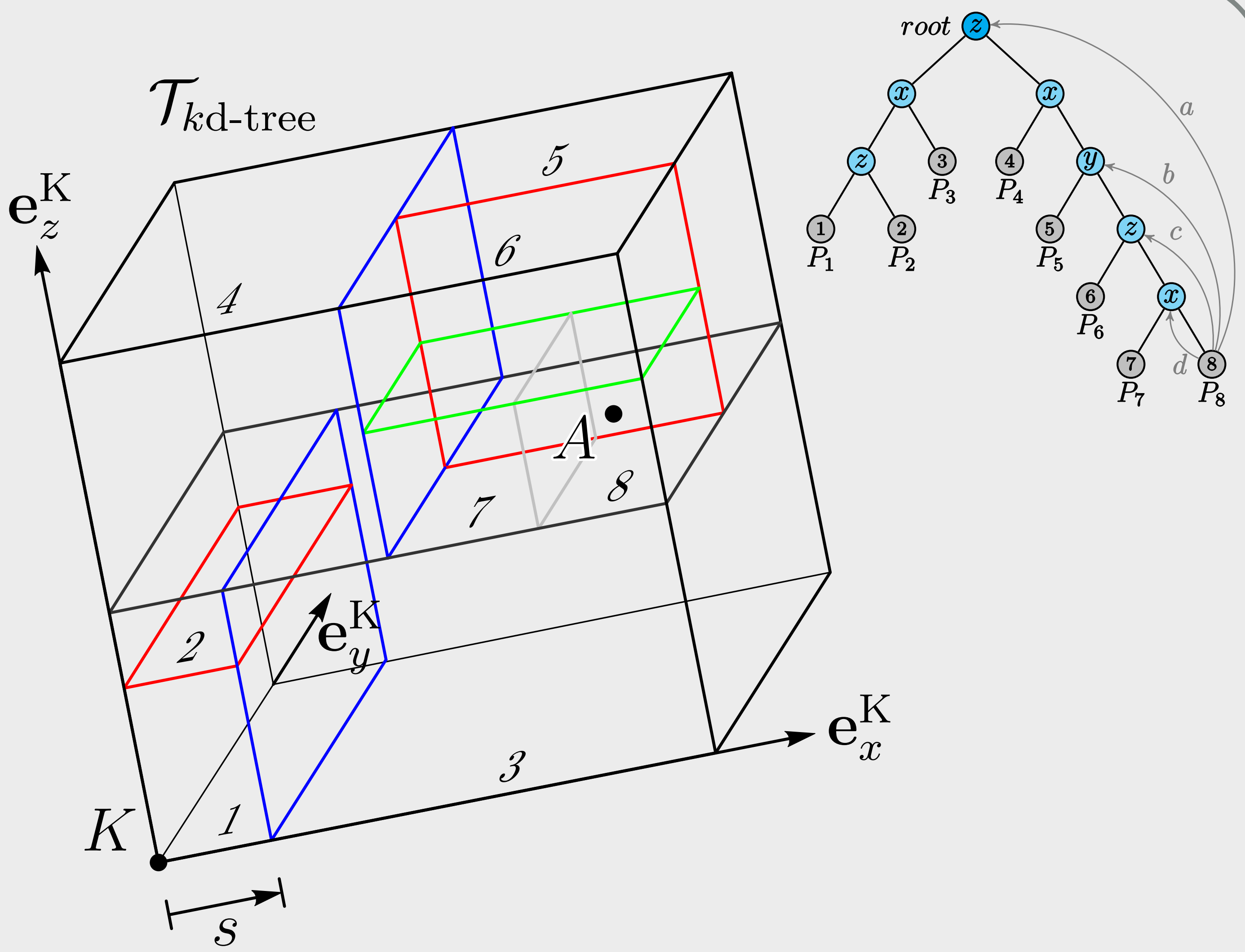
- **unbalanced** workload
- simple data structure
- 26 neighbors

- **better** balanced workload
- tree data structure
- **<26 neighbors** (mostly)

The Granular Rigid Body Simulation Framework

Parallel Implementation: Domain Decomposition

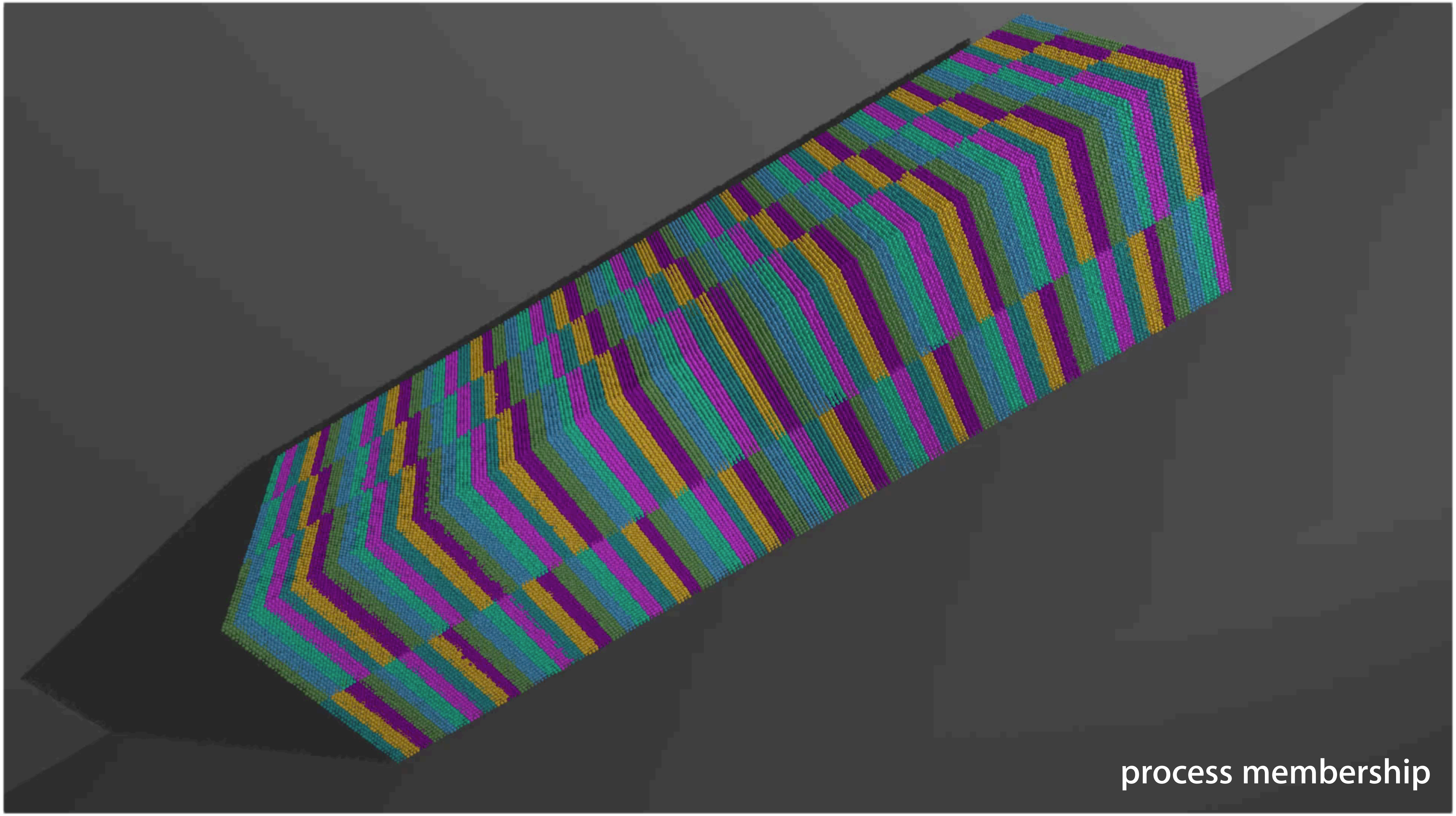




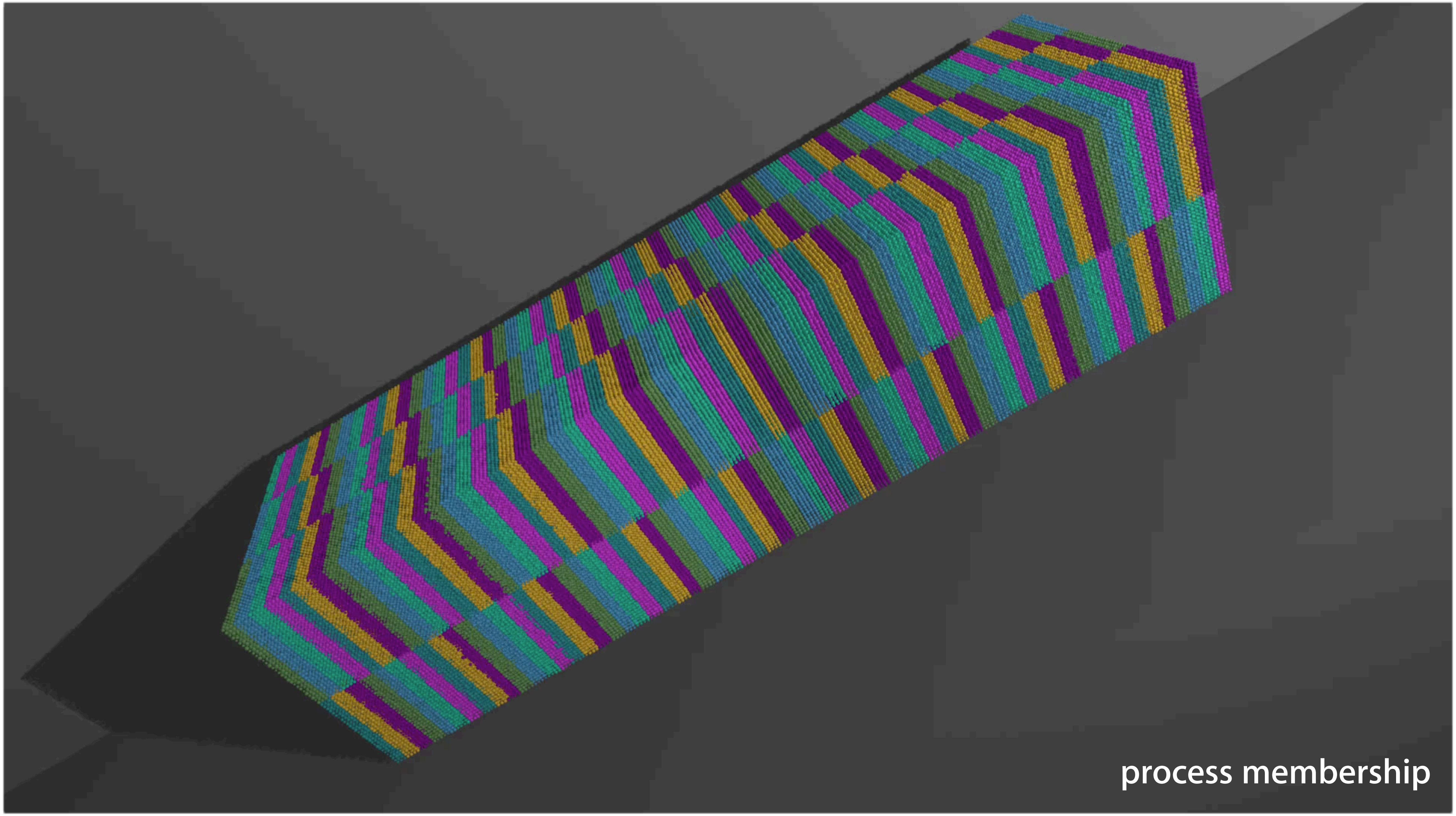
- **unbalanced** workload
- simple data structure
- 26 neighbors

- **better** balanced workload
- tree data structure
- **<26 neighbors** (mostly)

Why Domain Decomposition / Load Balancing?



Why Domain Decomposition / Load Balancing?



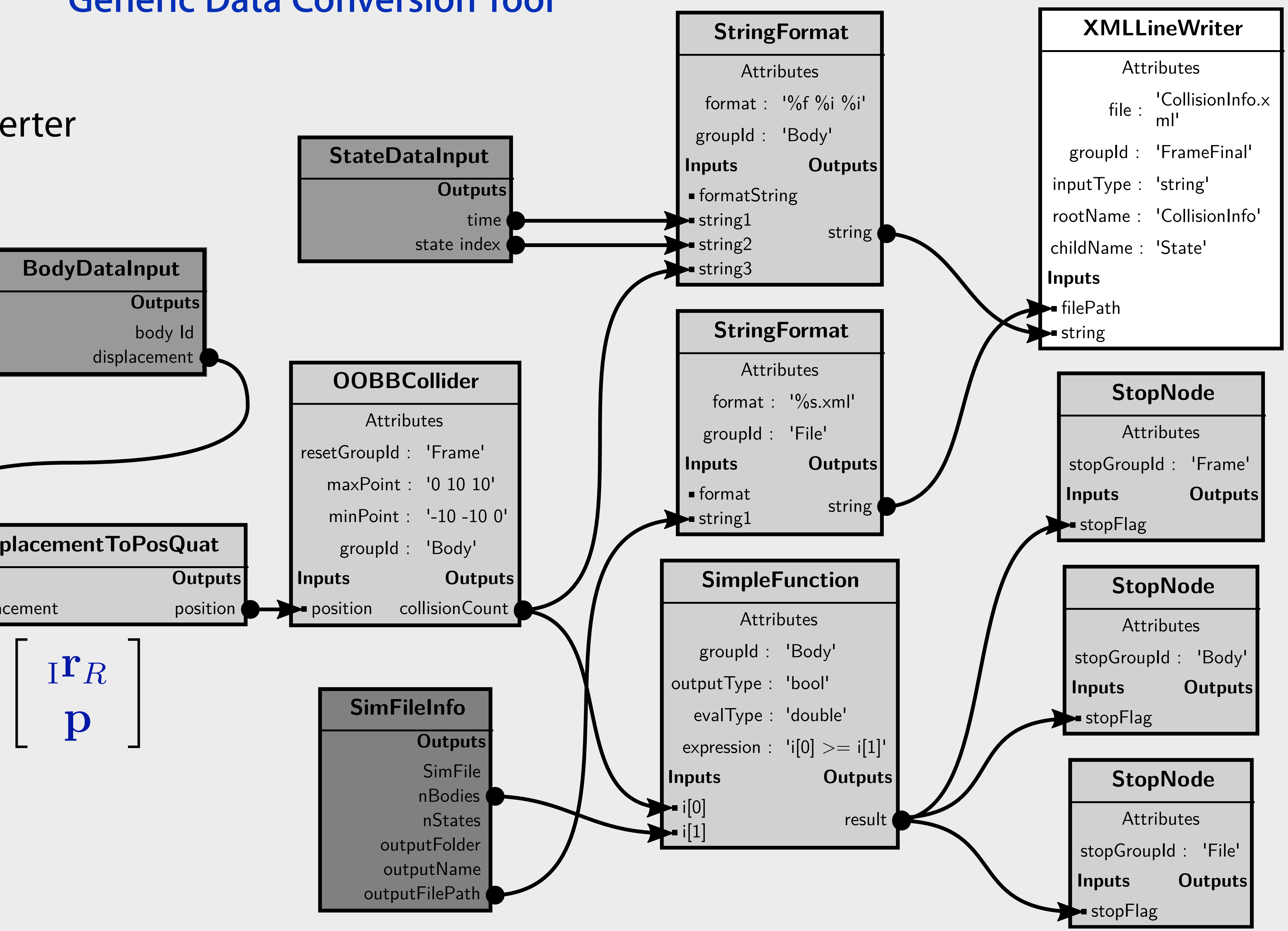
The Granular Rigid Body Simulation Framework

GRSF Converter

Generic Data Conversion Tool

The Granular Rigid Body Simulation Framework

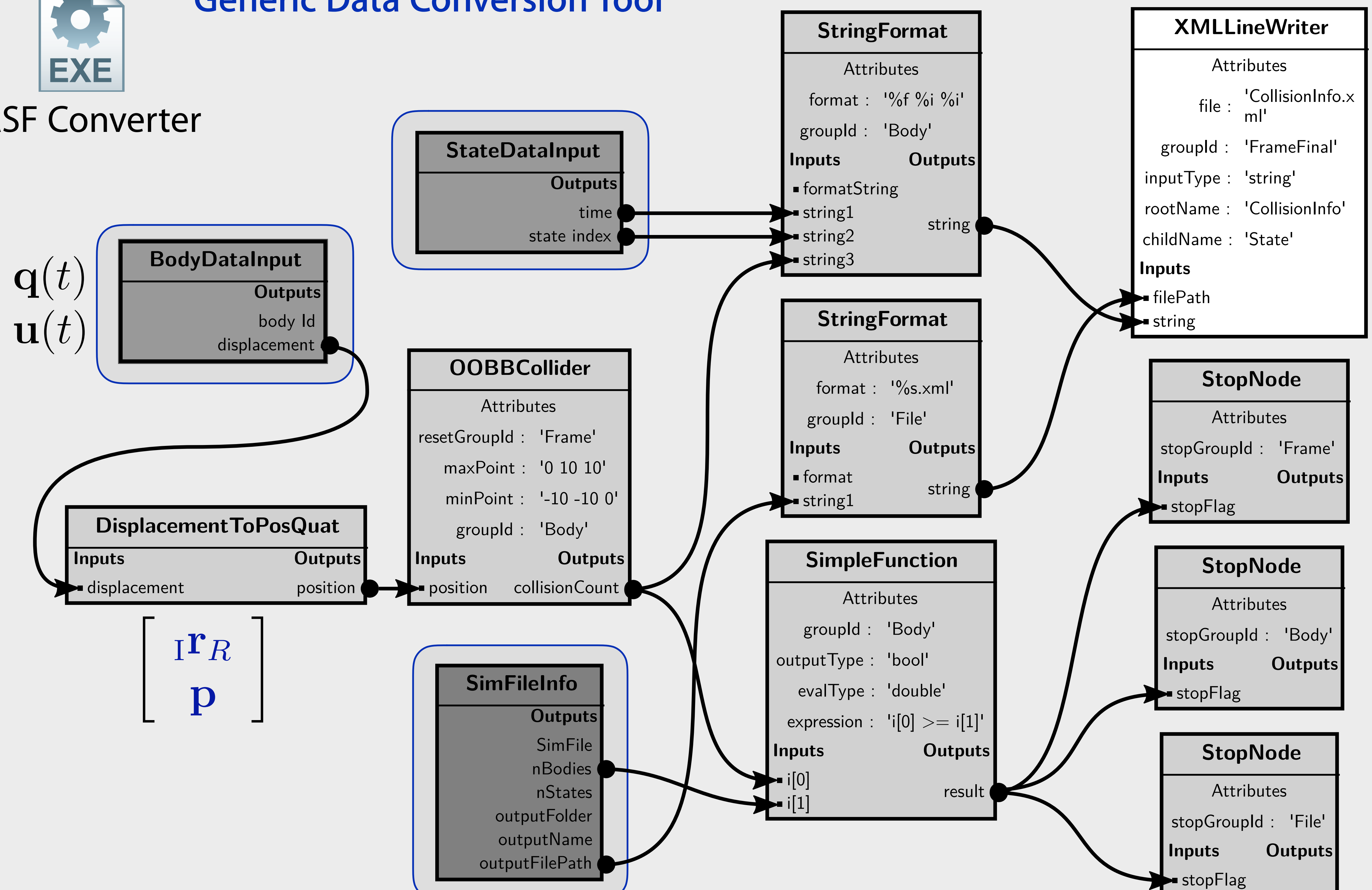
Generic Data Conversion Tool



The Granular Rigid Body Simulation Framework

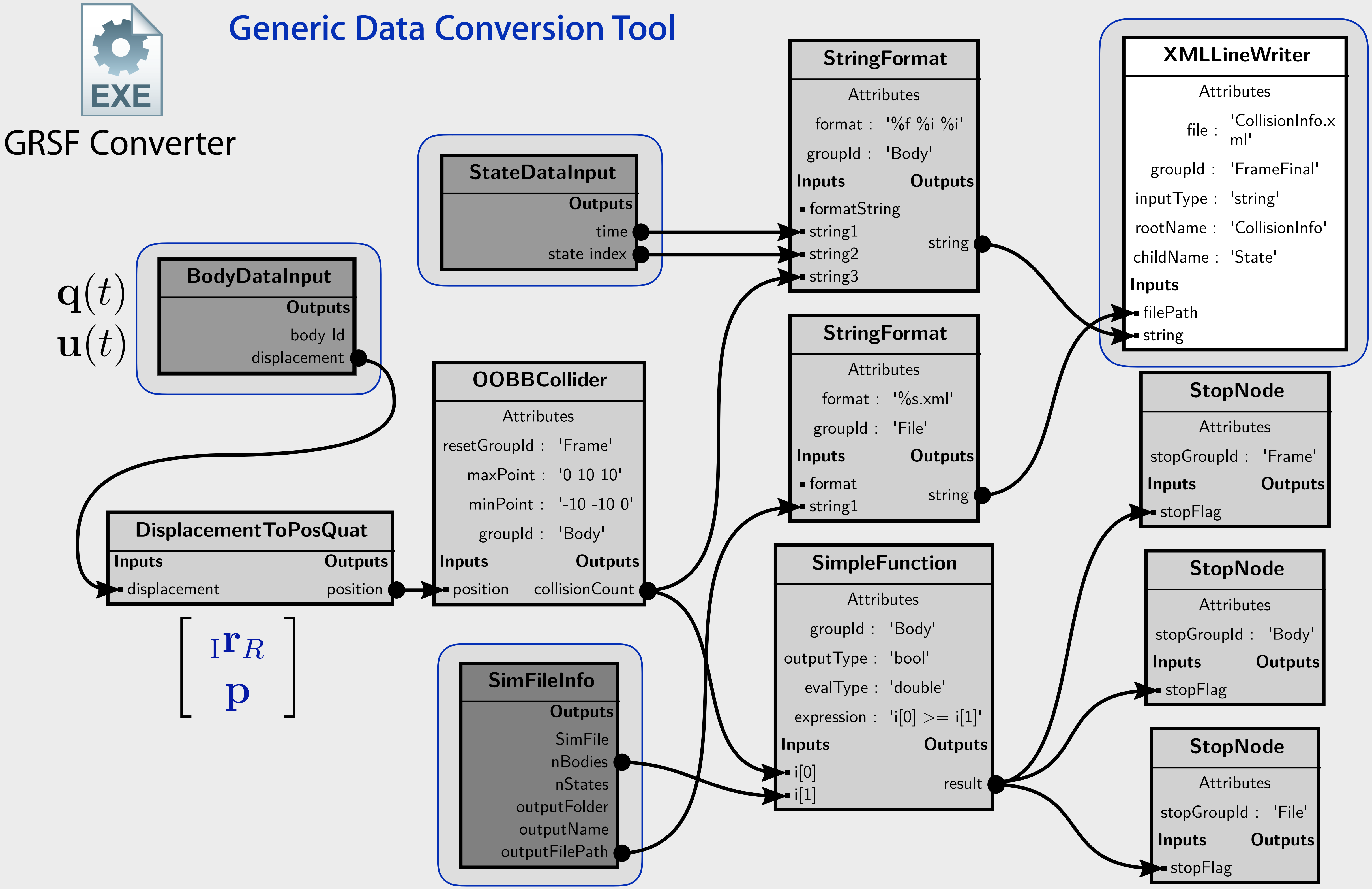
GRSF Converter

Generic Data Conversion Tool



The Granular Rigid Body Simulation Framework

Generic Data Conversion Tool



The Granular Rigid Body Simulation Framework

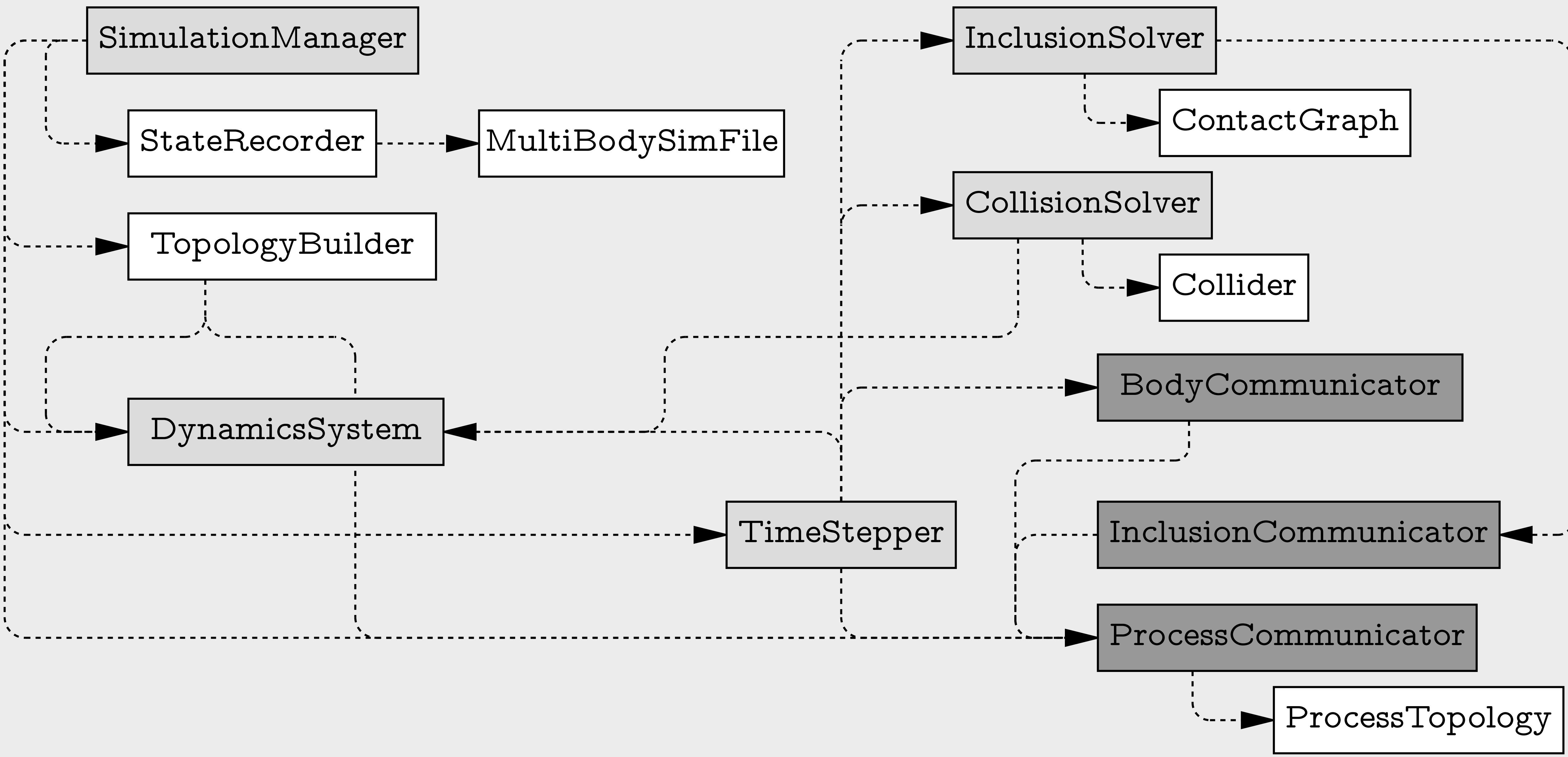
EXE

GRSF Converter

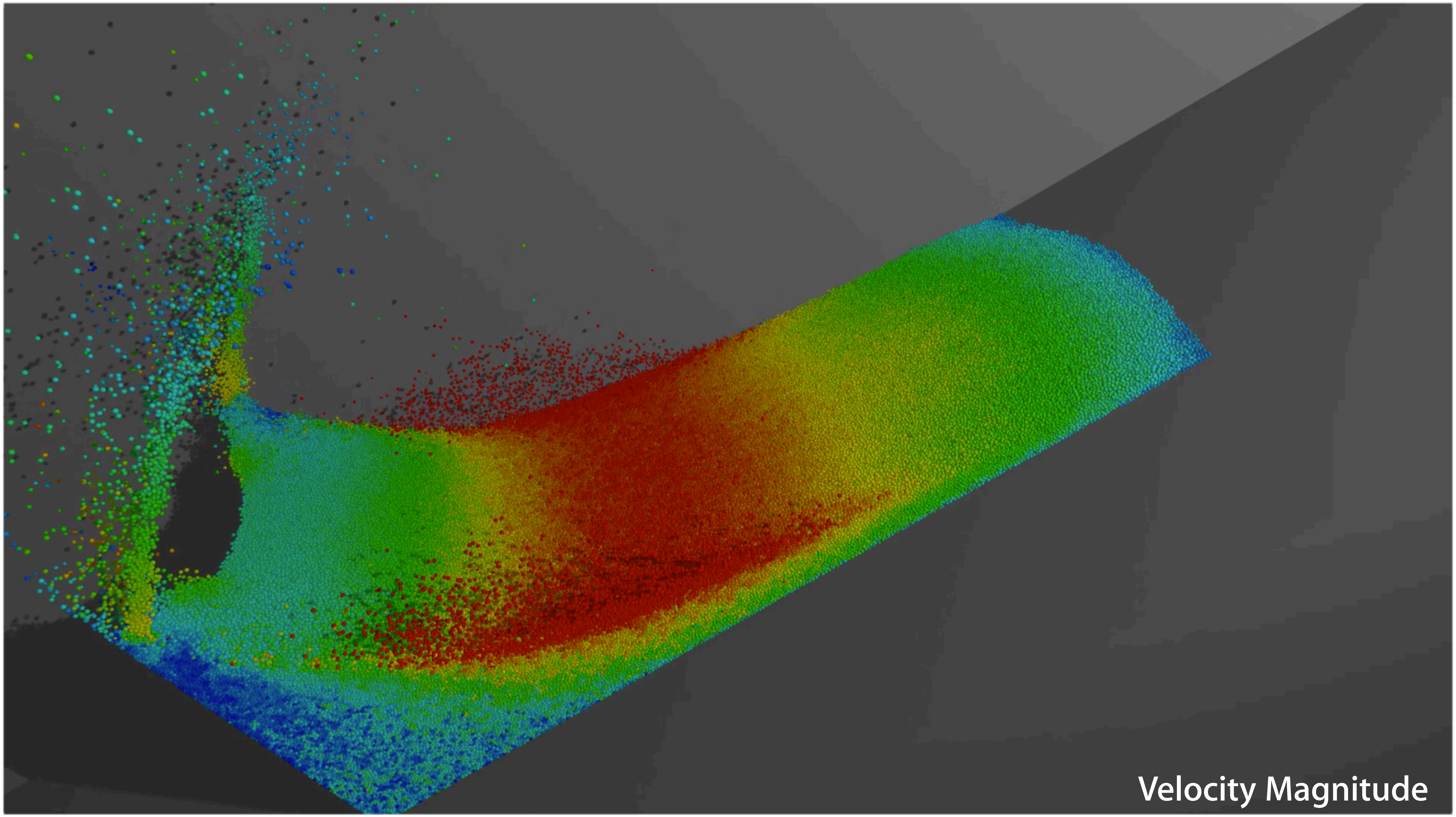
Generic Data Conversion Tool

Google TensorFlow™
2015
tensorflow.org

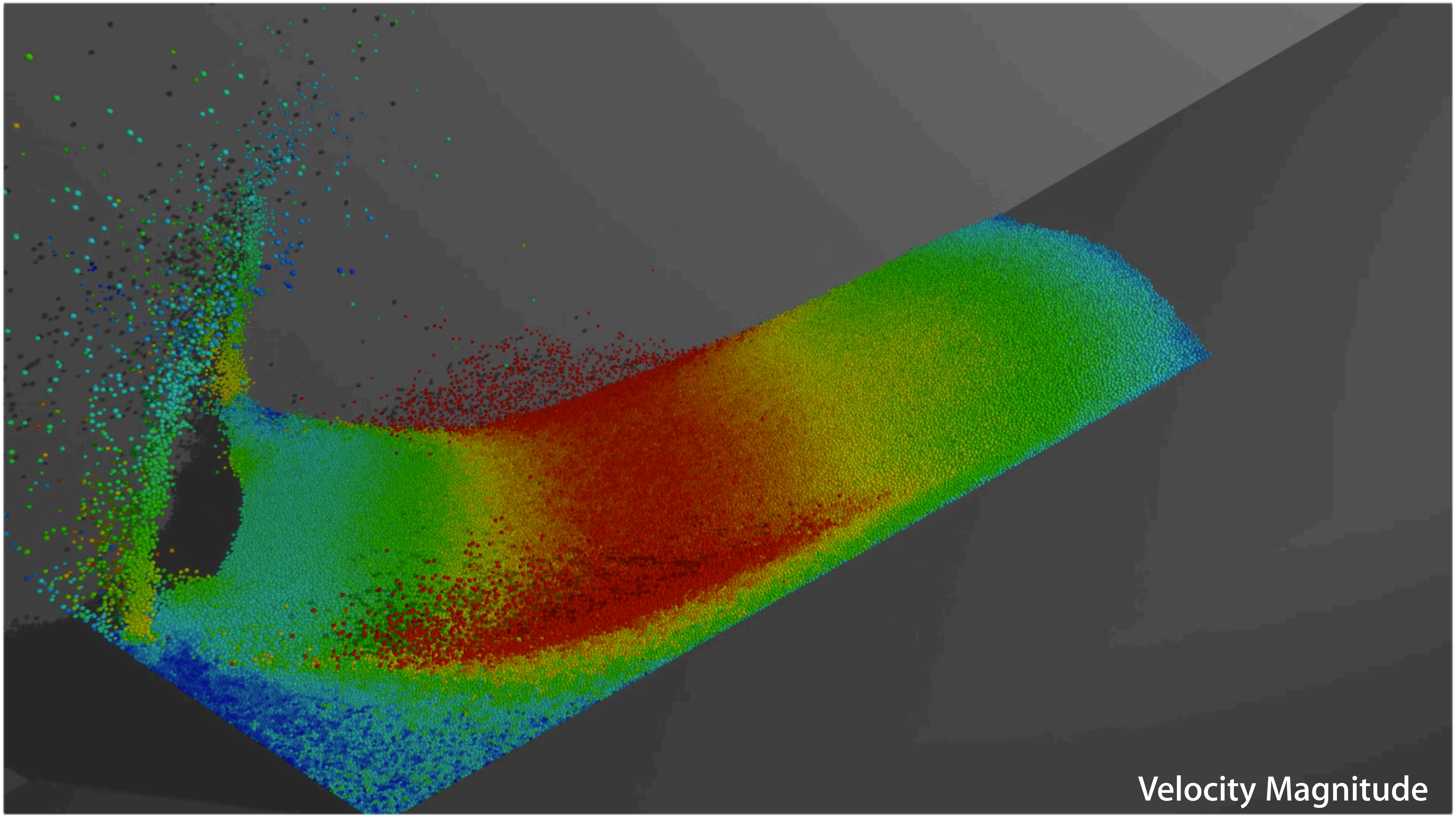
The Granular Rigid Body Simulation Framework



Drift Correction (cheap on velocity level)



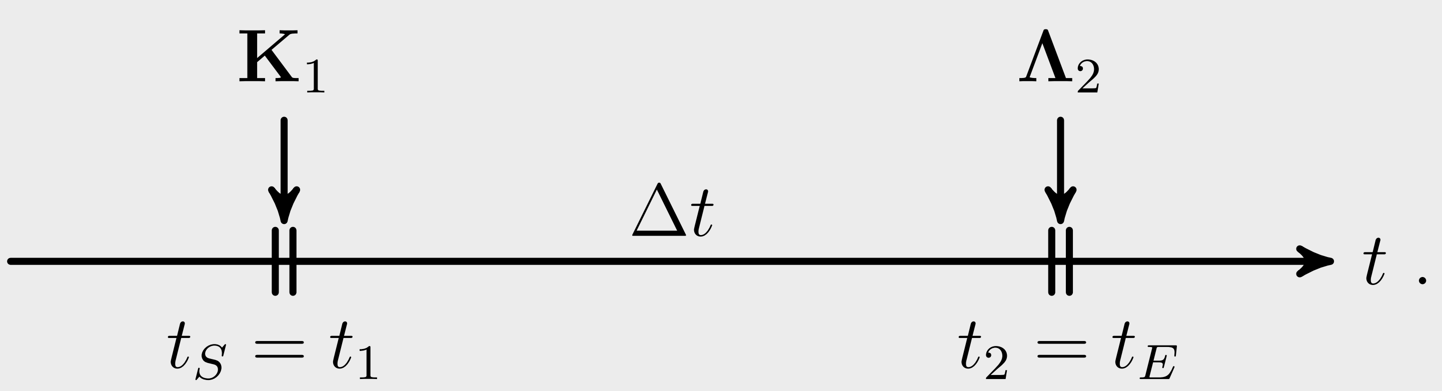
Drift Correction (cheap on velocity level)

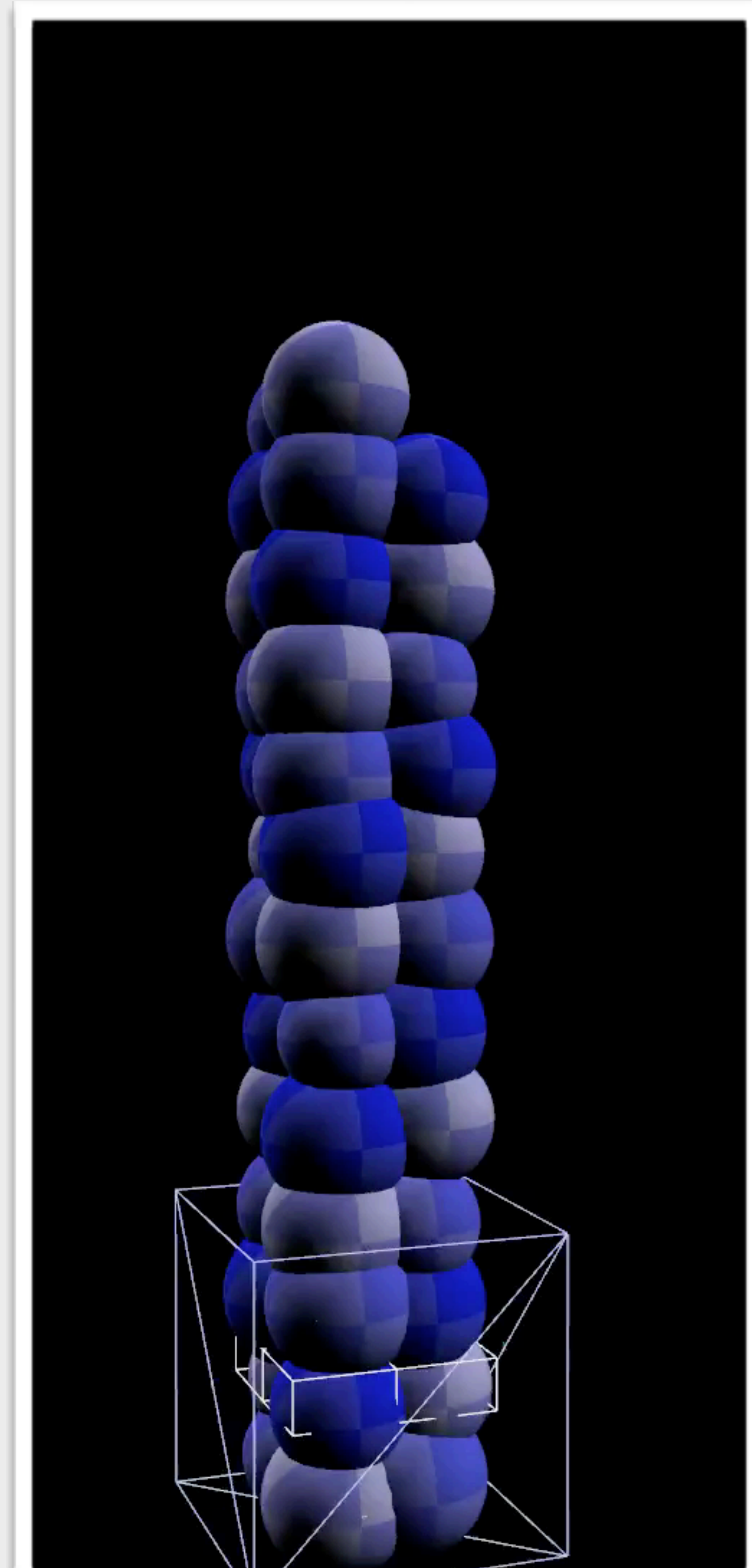


Drift Correction (on displacement level)

$$\mathrm{d}\mathbf{u} = \mathbf{M}(\mathbf{q},t)^{-1}\left(\mathbf{h}(\mathbf{q},\mathbf{u},t)\mathrm{d}t - \mathbf{W}(\mathbf{q},t)\mathrm{d}\mathbf{P}\right)$$
$$\mathrm{d}\mathbf{q} = \mathbf{F}(\mathbf{q},t)\mathbf{u}\mathrm{d}t + \mathbf{b}(\mathbf{q},t)\mathrm{d}t + \mathbf{Q}(\mathbf{q},t)\boxed{\mathrm{d}\boldsymbol{\Sigma}}$$

gradually correcting the penetration by using impulse-impulses acting on displacement level

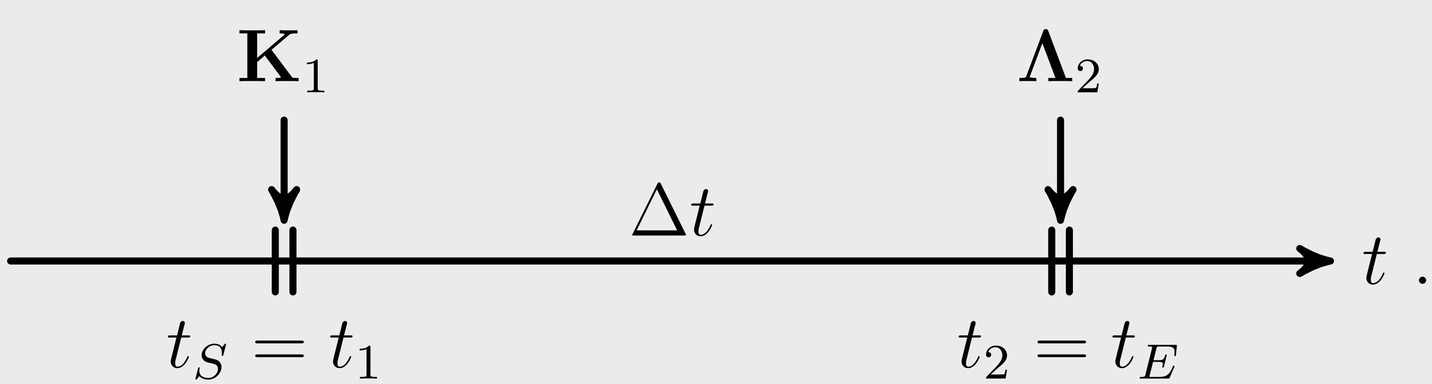


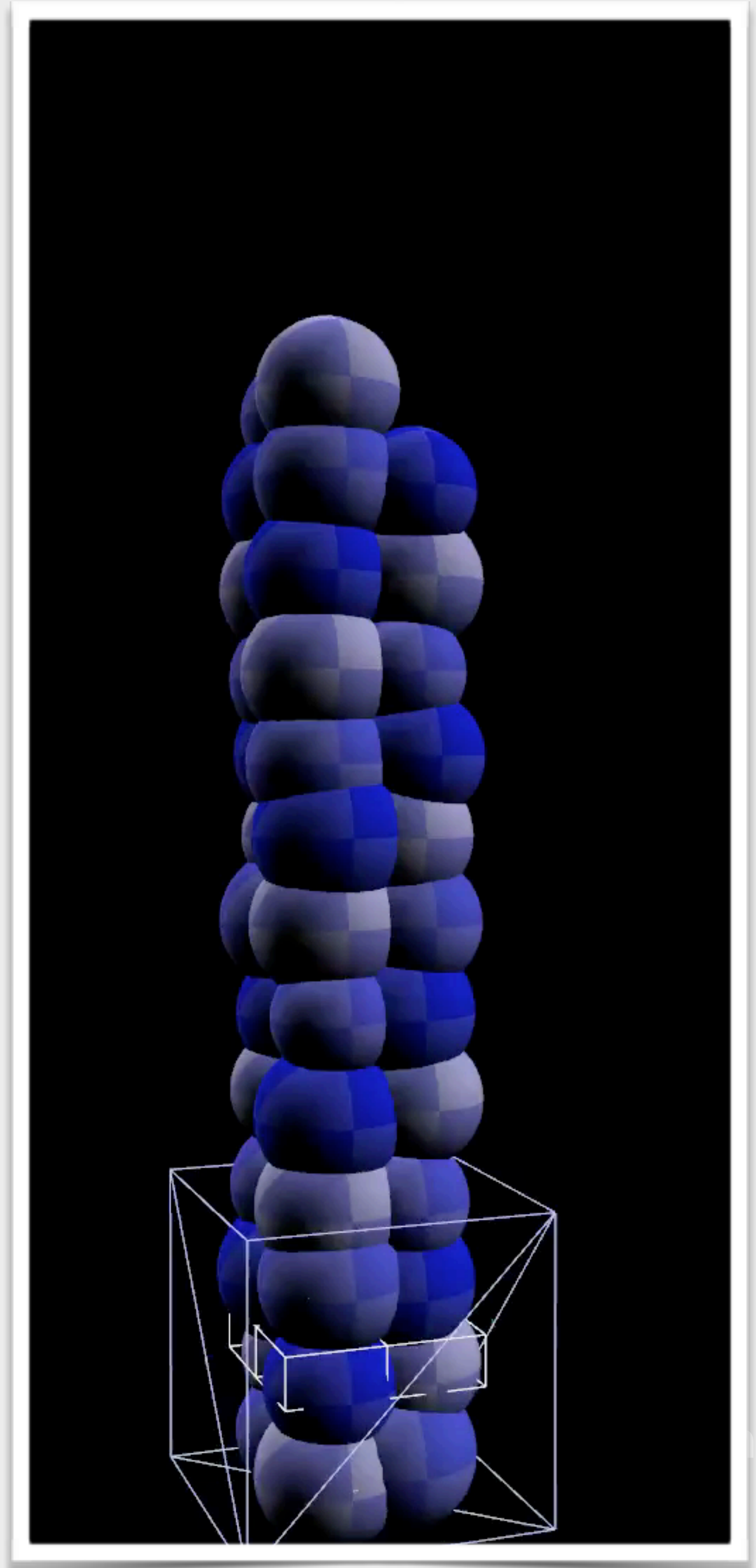


Drift Correction (on displacement level)

$$\mathrm{d}\mathbf{u} = \mathbf{M}(\mathbf{q},t)^{-1}\left(\mathbf{h}(\mathbf{q},\mathbf{u},t)\mathrm{d}t - \mathbf{W}(\mathbf{q},t)\mathrm{d}\mathbf{P}\right)$$
$$\mathrm{d}\mathbf{q} = \mathbf{F}(\mathbf{q},t)\mathbf{u}\mathrm{d}t + \mathbf{b}(\mathbf{q},t)\mathrm{d}t + \mathbf{Q}(\mathbf{q},t)\boxed{\mathrm{d}\boldsymbol{\Sigma}}$$

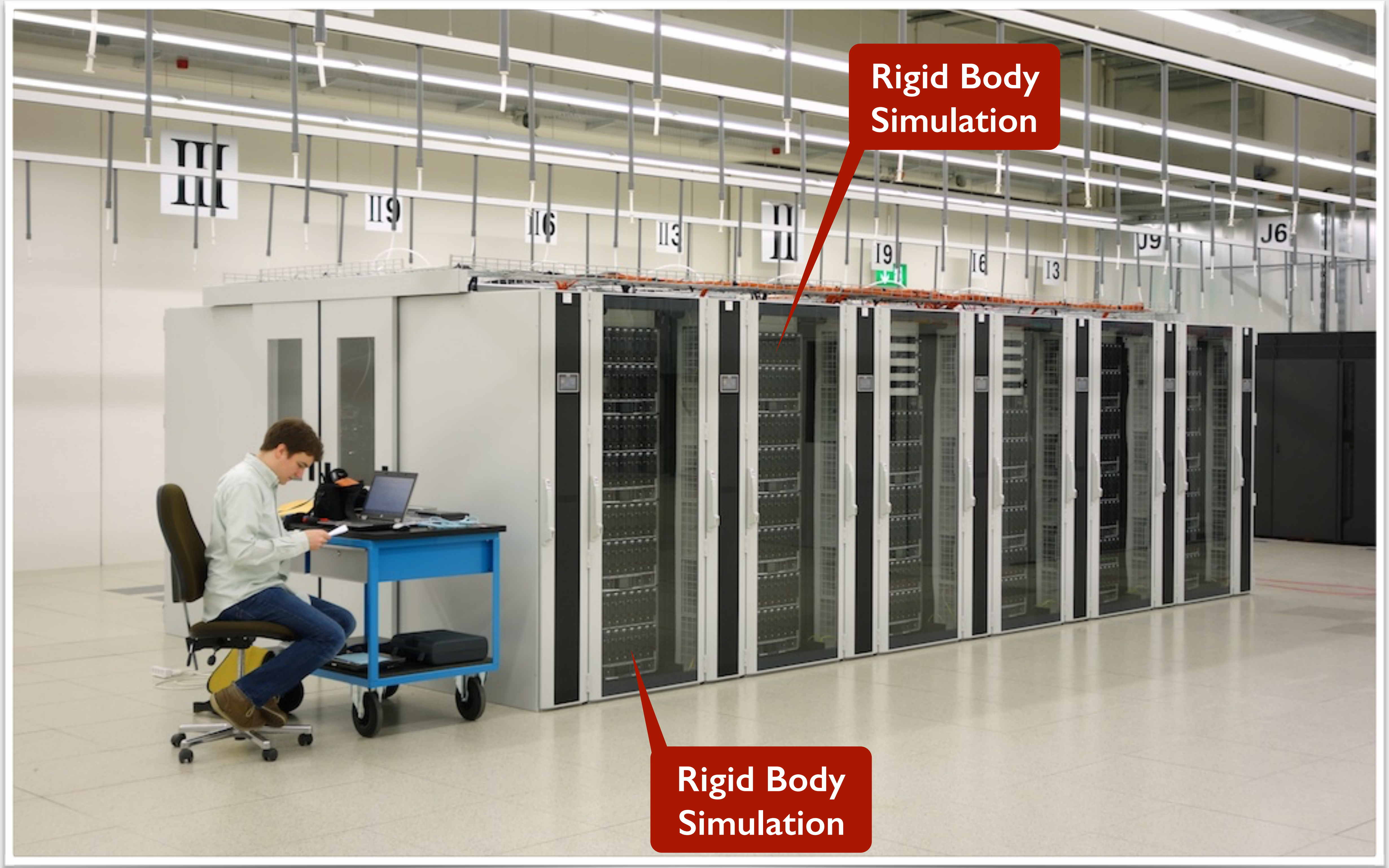
gradually correcting the penetration by using **impulse-impulses** acting on displacement level



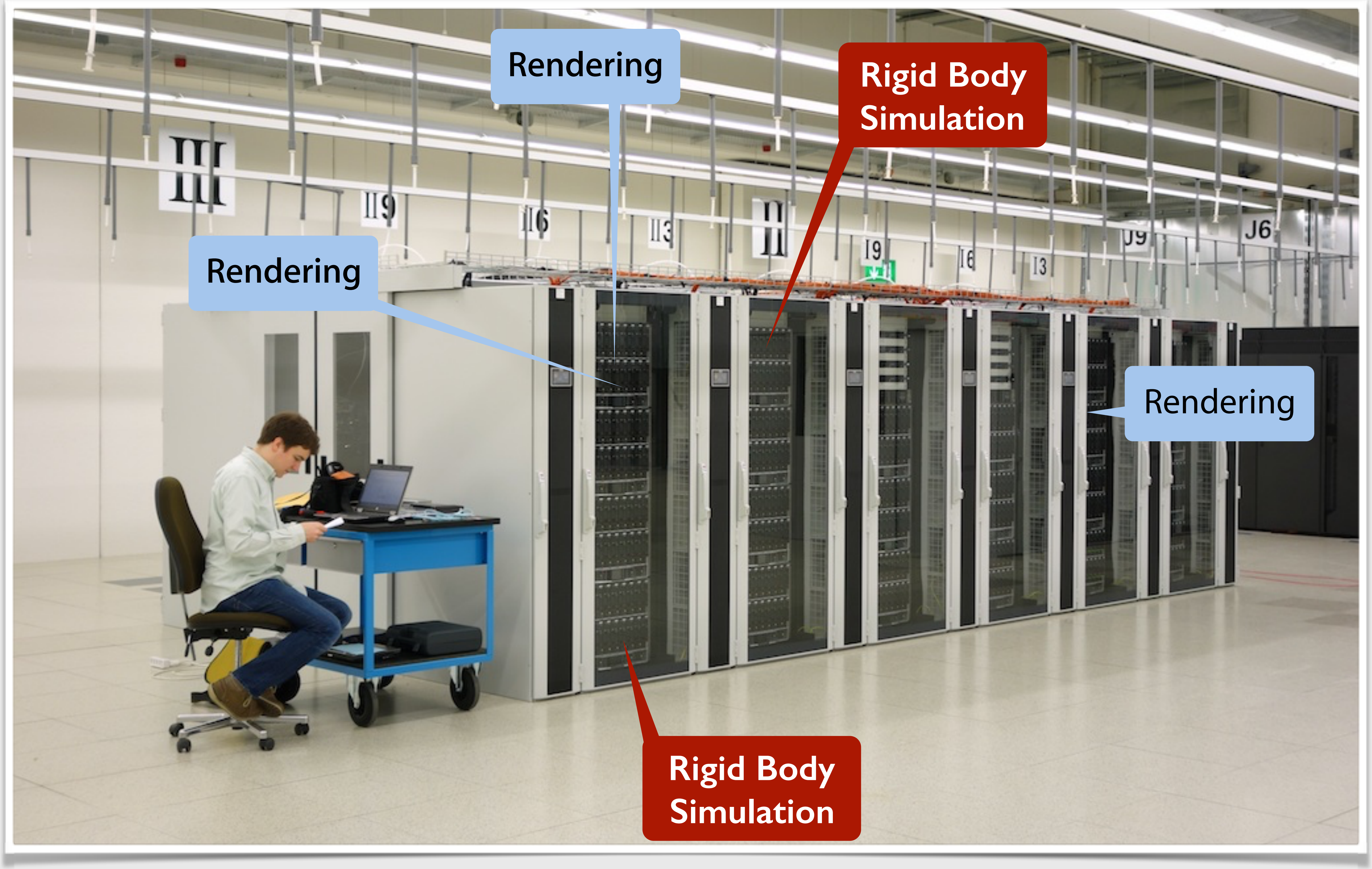


What is Cluster Computing?

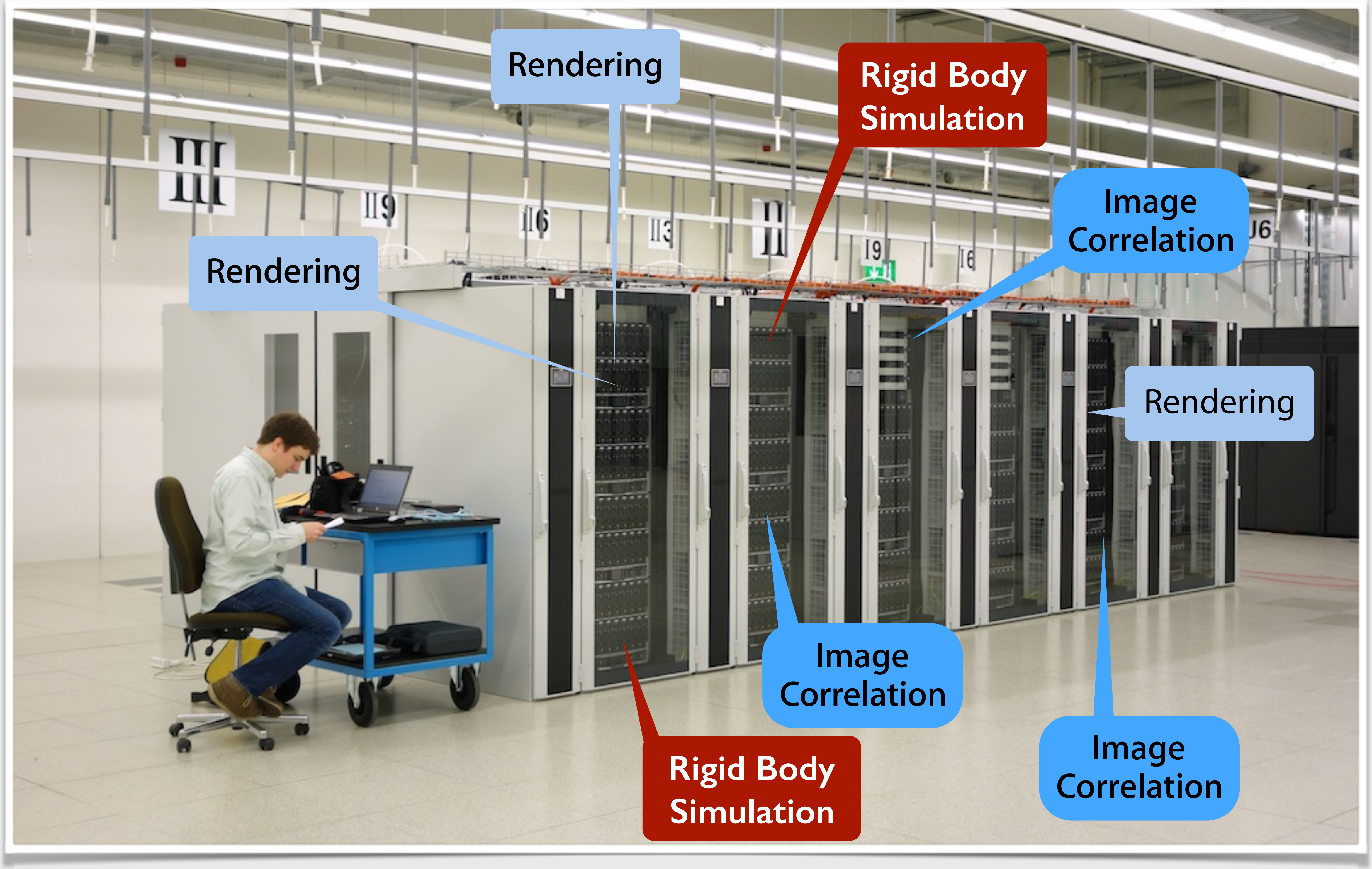
What is Cluster Computing?



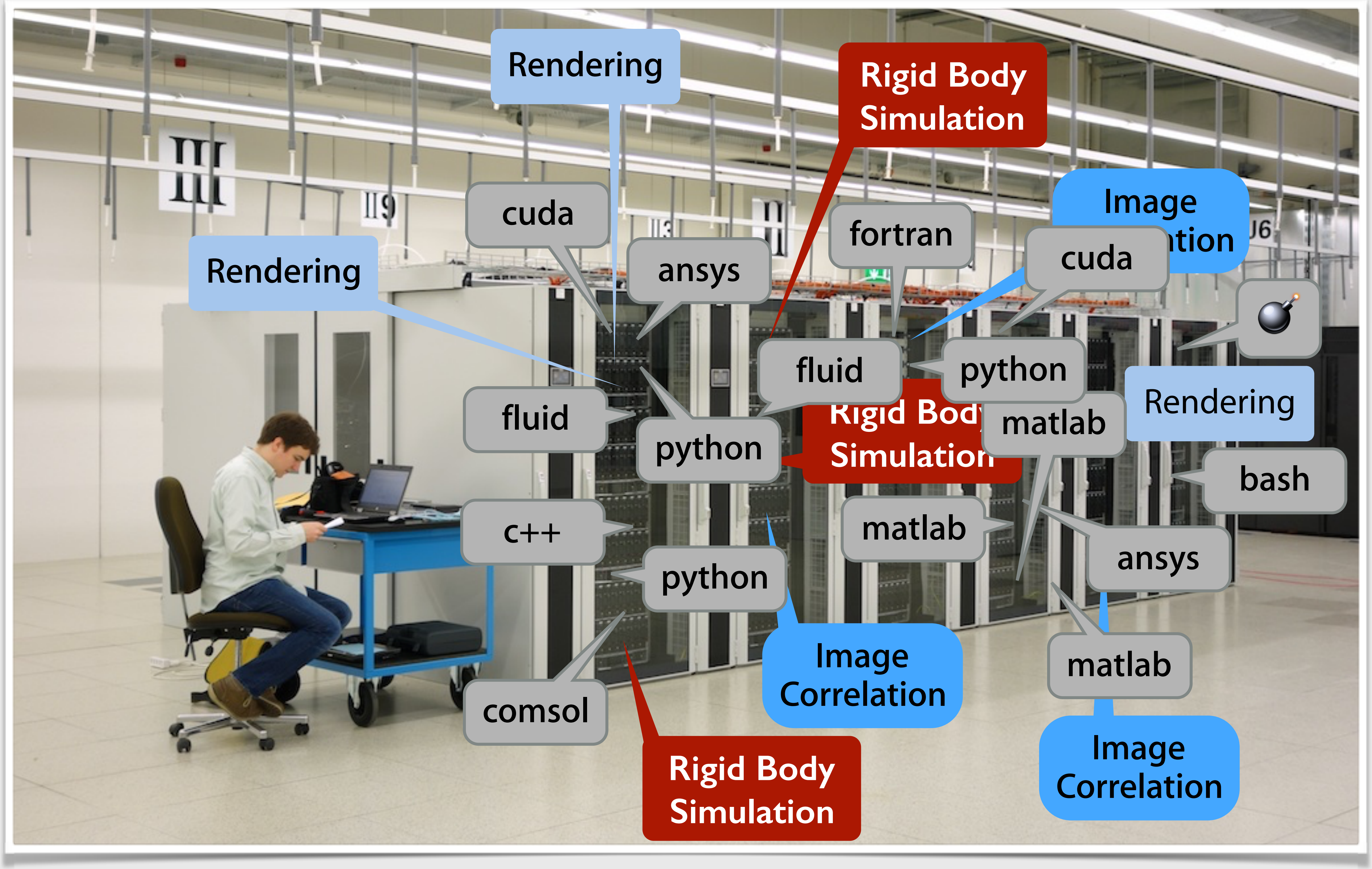
What is Cluster Computing?



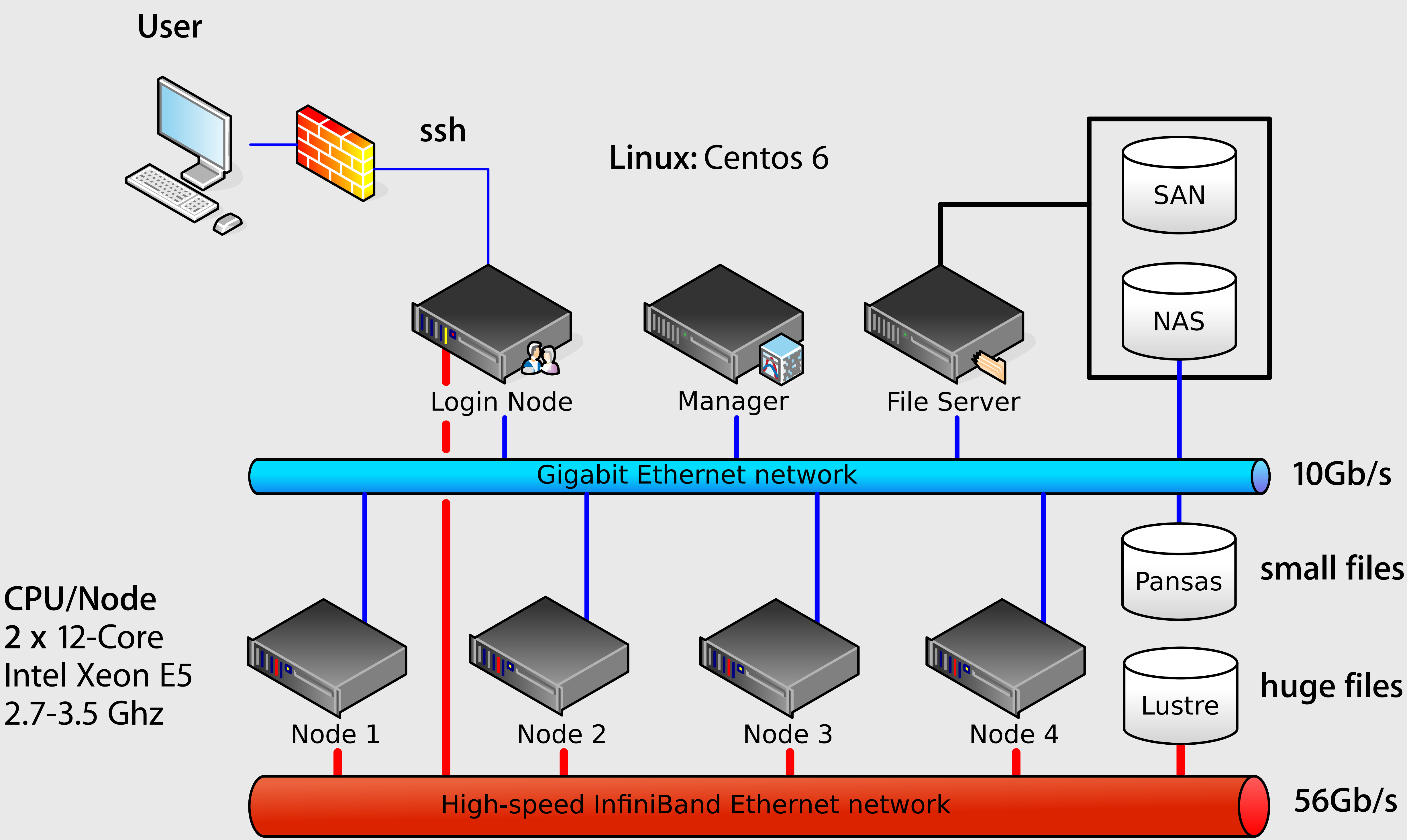
What is Cluster Computing?



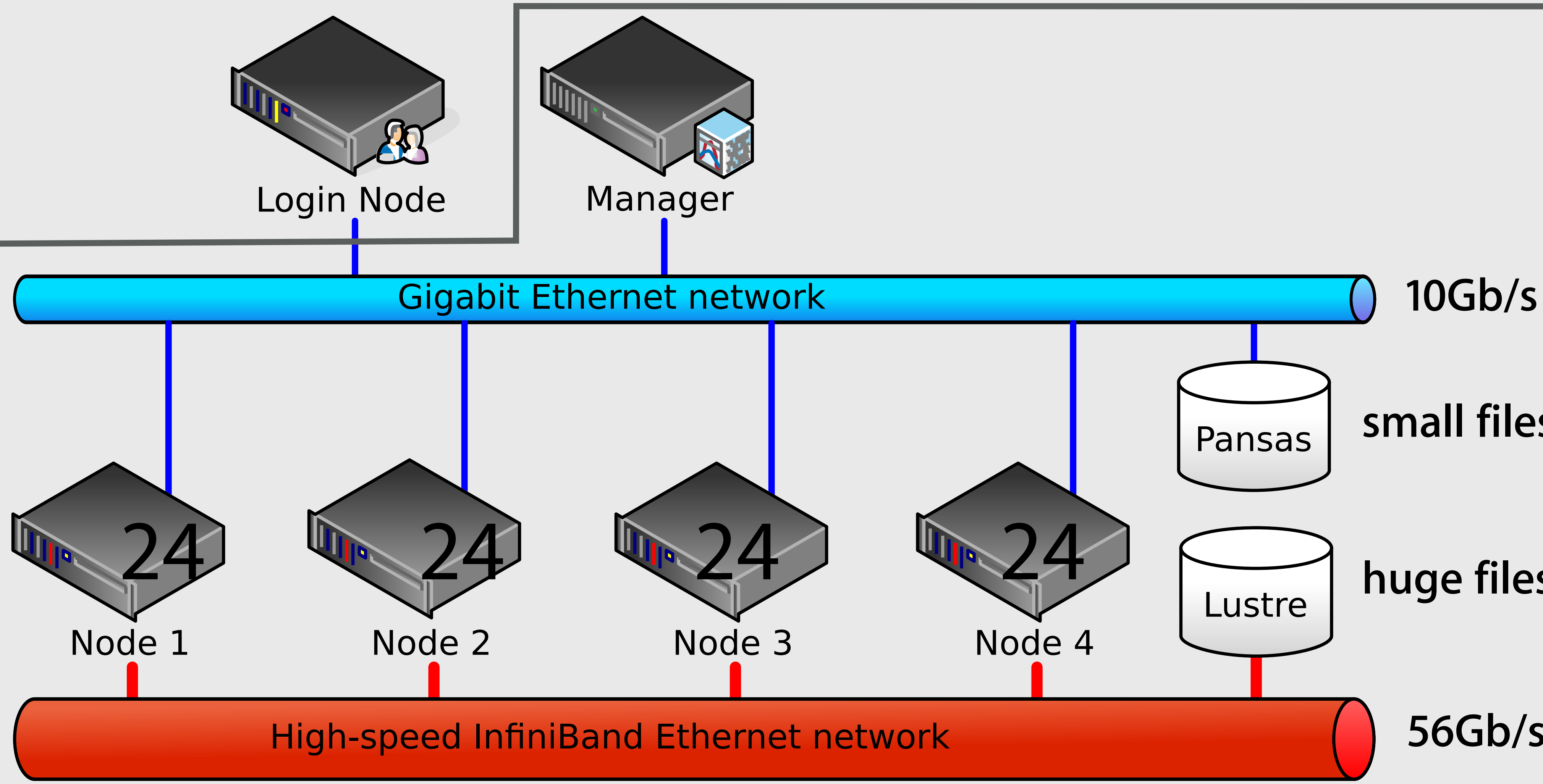
What is Cluster Computing?



Cluster Architecture



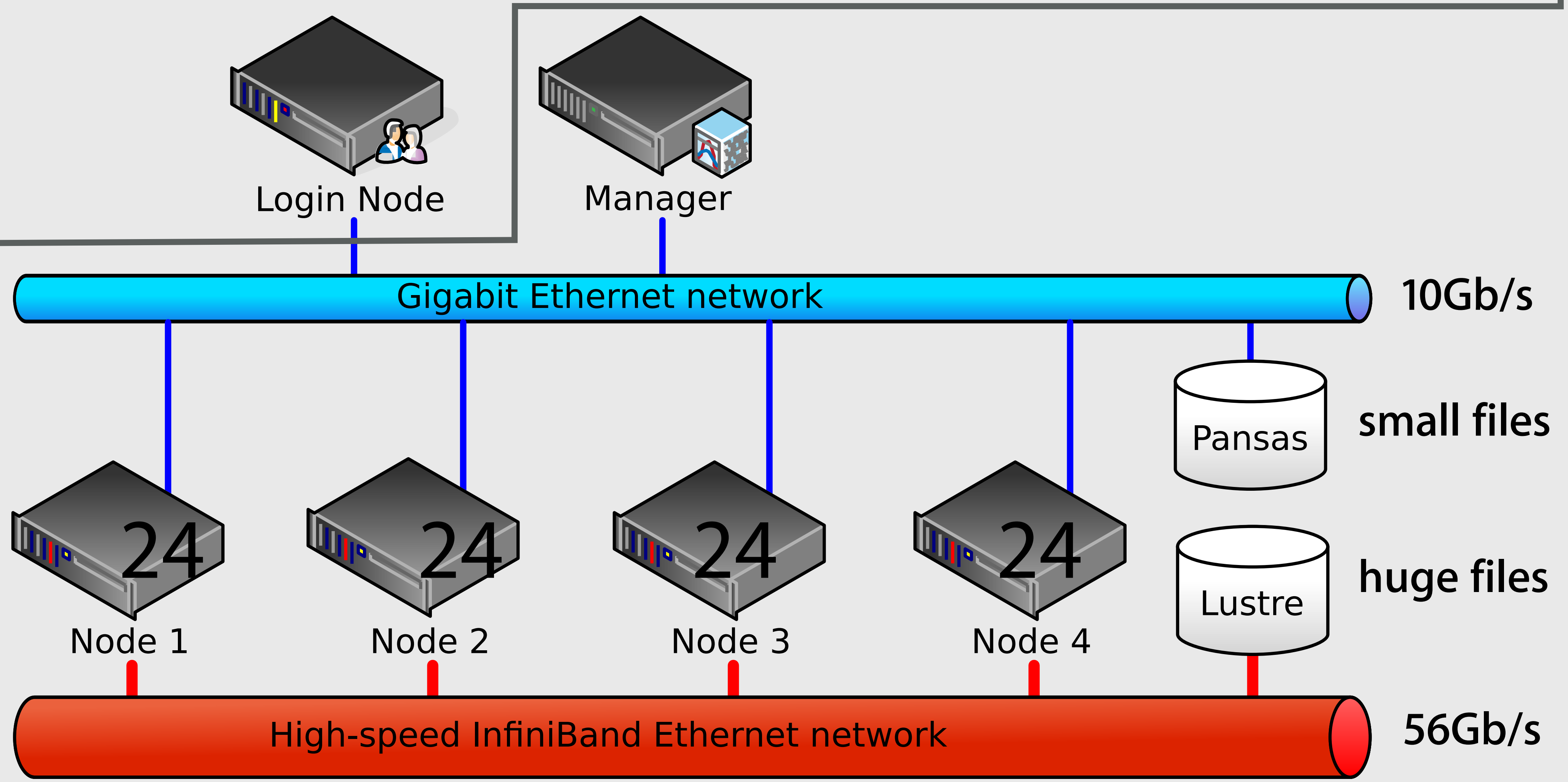
Clean Job Babysitting



Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:

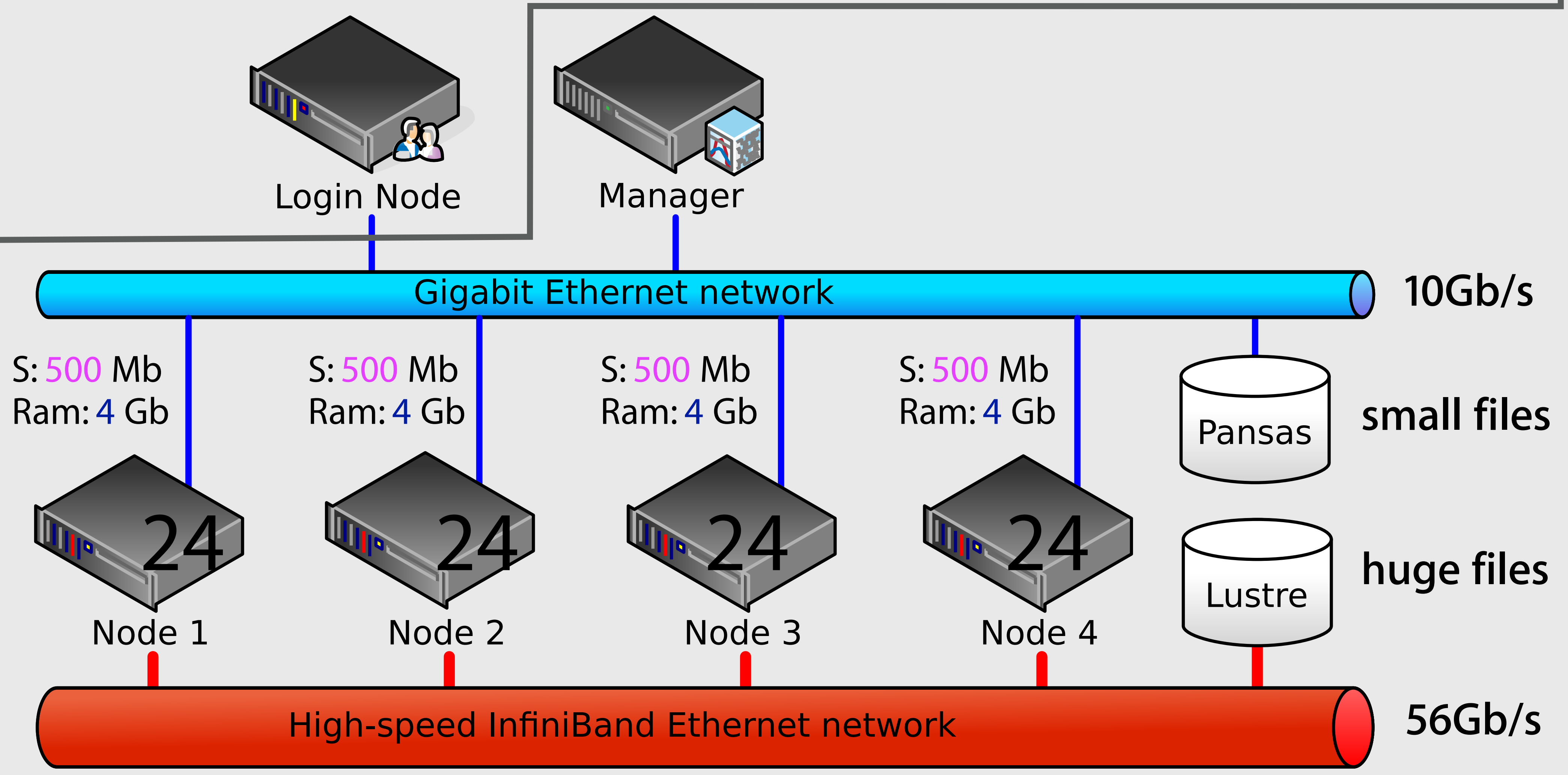
```
python configureJob.py -x  Job.ini
-p 92 -l 500 -r 4000 -t 720 -n 2
```



Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:

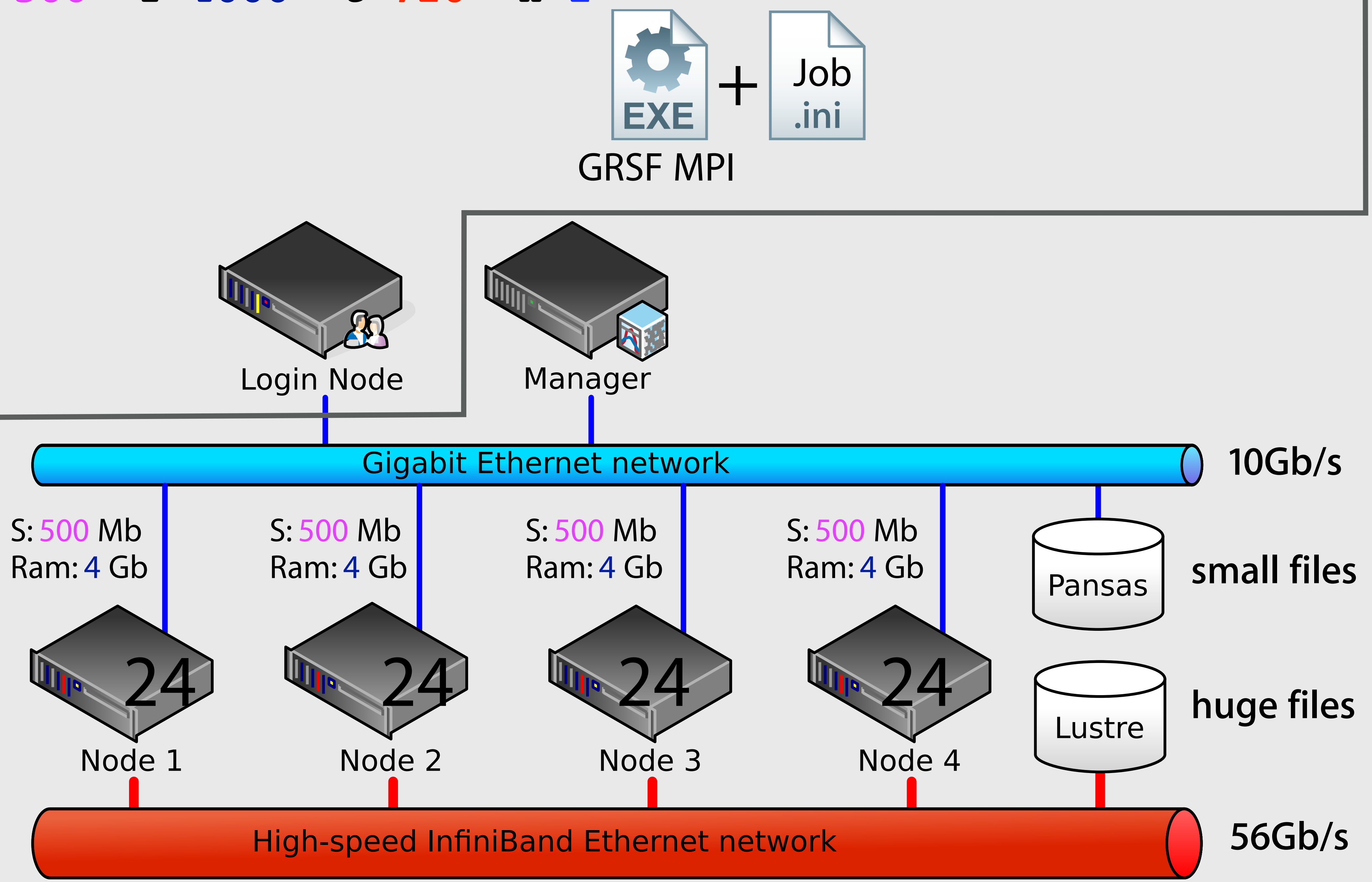
```
python configureJob.py -x  Job.ini
-p 92 -l 500 -r 4000 -t 720 -n 2
```



Clean Job Babysitting

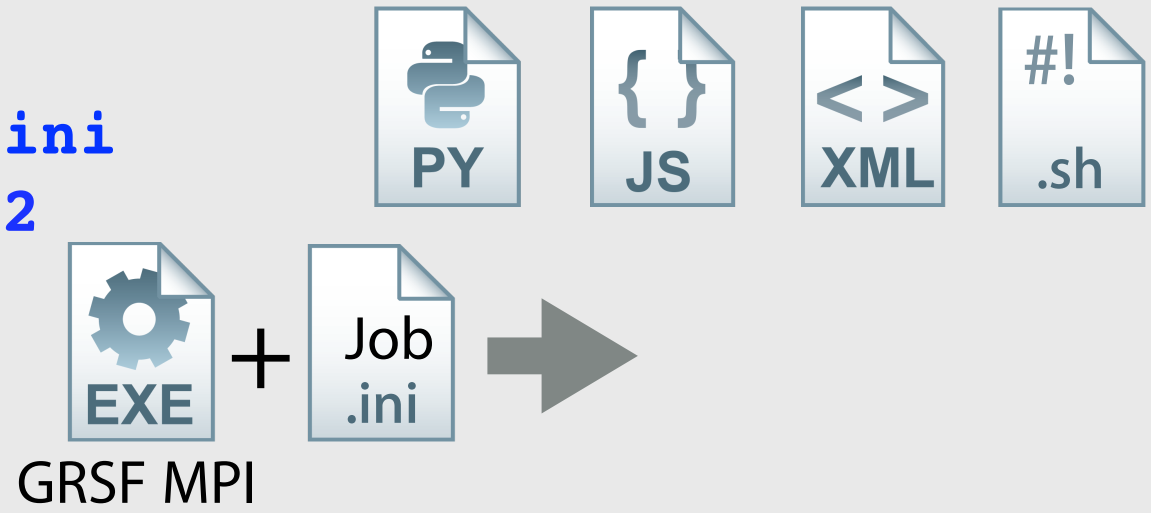
1. Configure Job with HPCJobConfigurator:

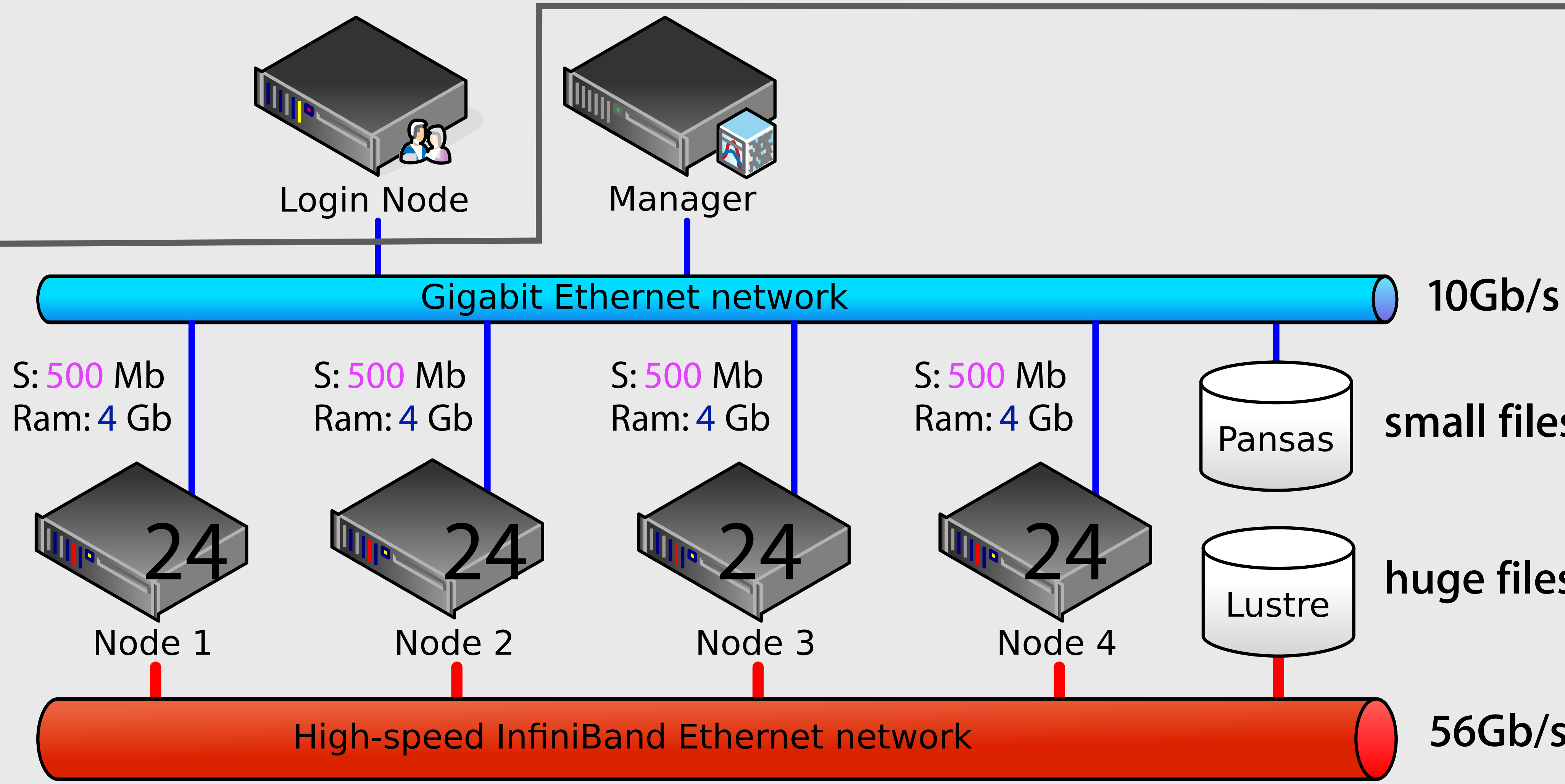
```
python configureJob.py -x  Job.ini
-p 92 -l 500 -r 4000 -t 720 -n 2
```



Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini
   -p 92 -l 500 -r 4000 -t 720 -n 2

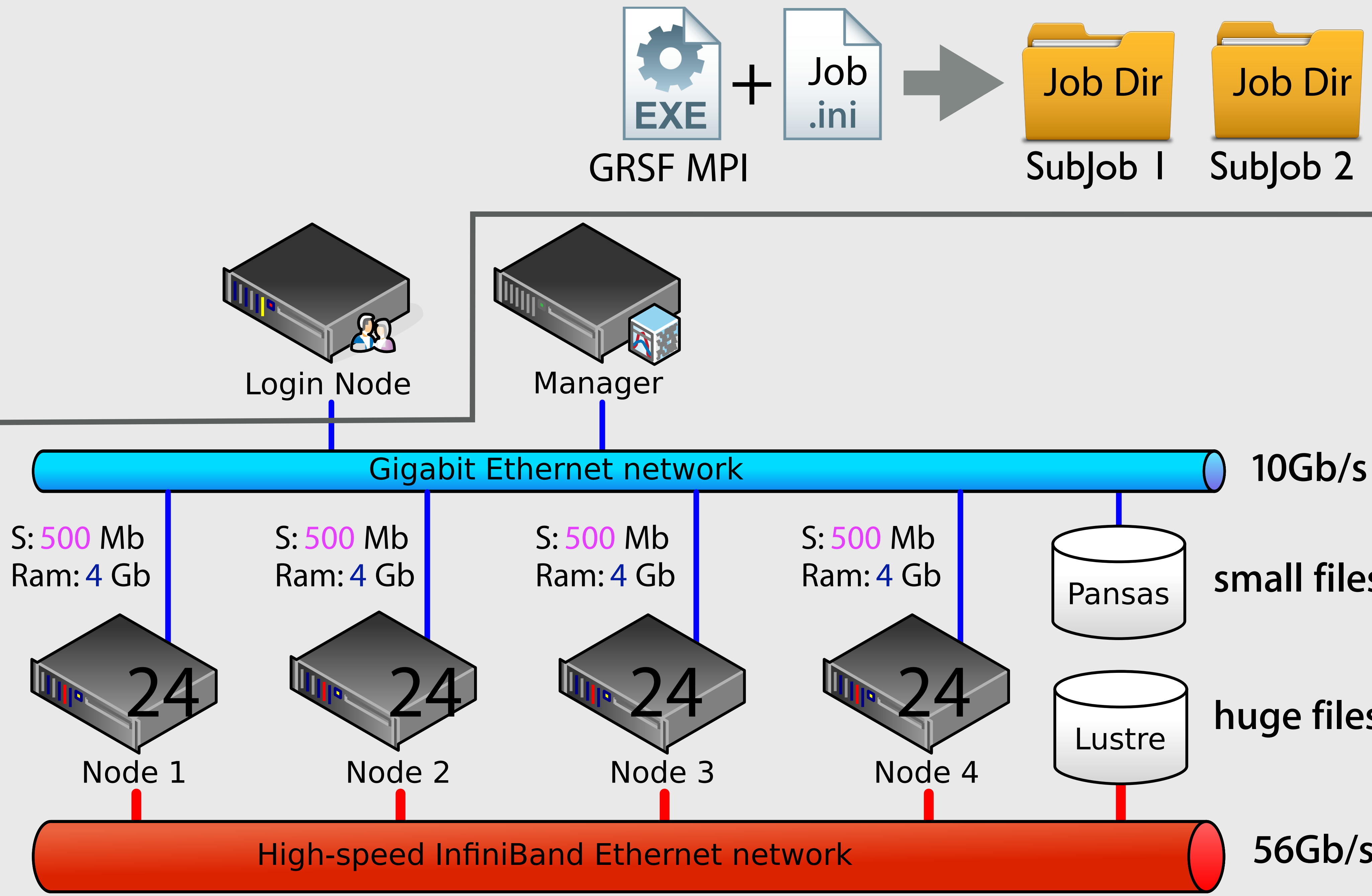




Clean Job Babysitting

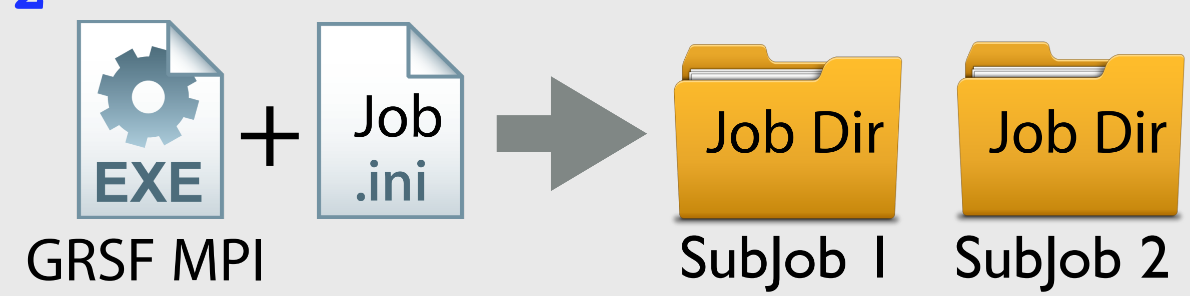
1. Configure Job with HPCJobConfigurator:

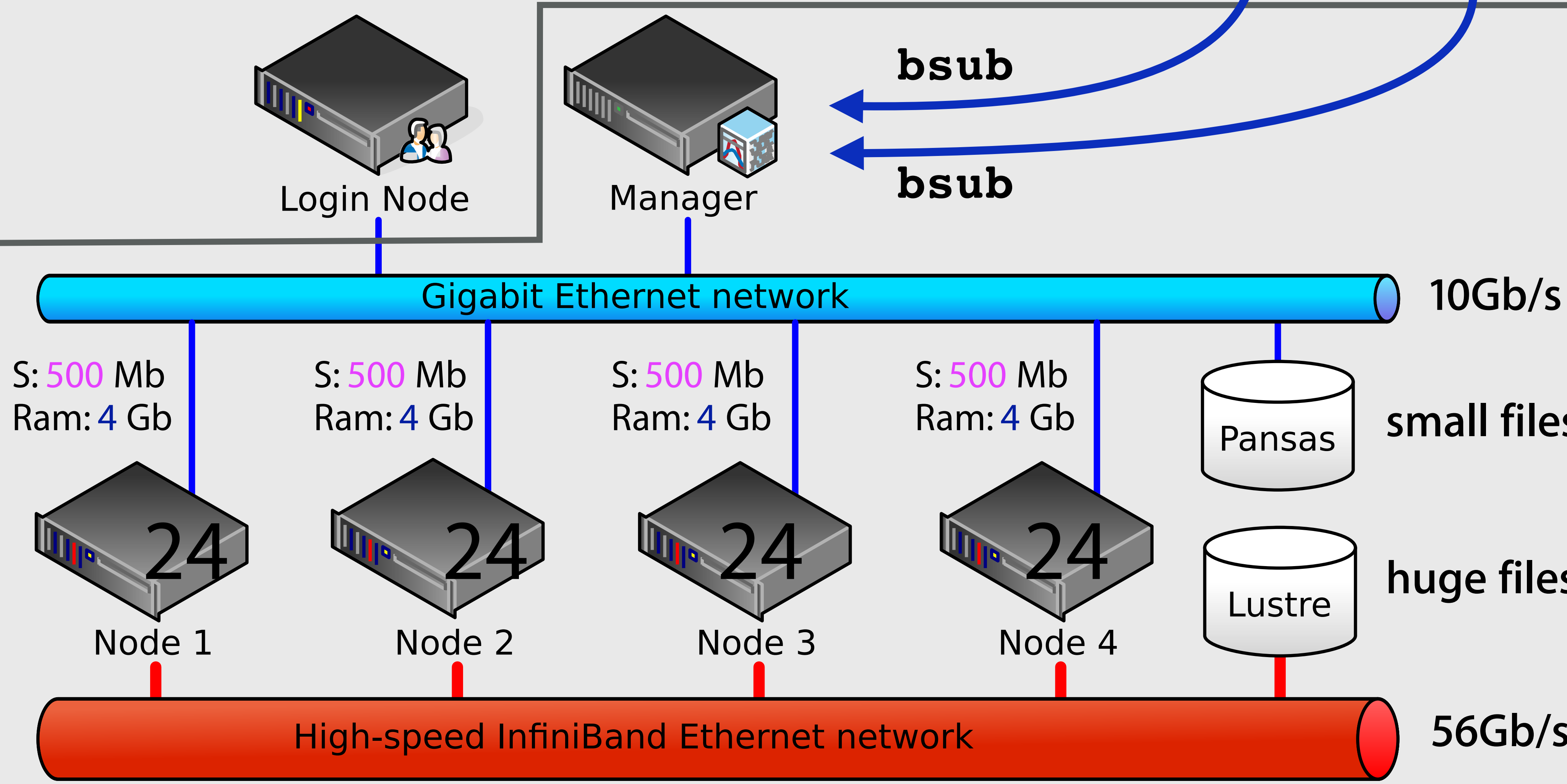
```
python configureJob.py -x  Job.ini
-p 92 -l 500 -r 4000 -t 720 -n 2
```



Clean Job Babysitting

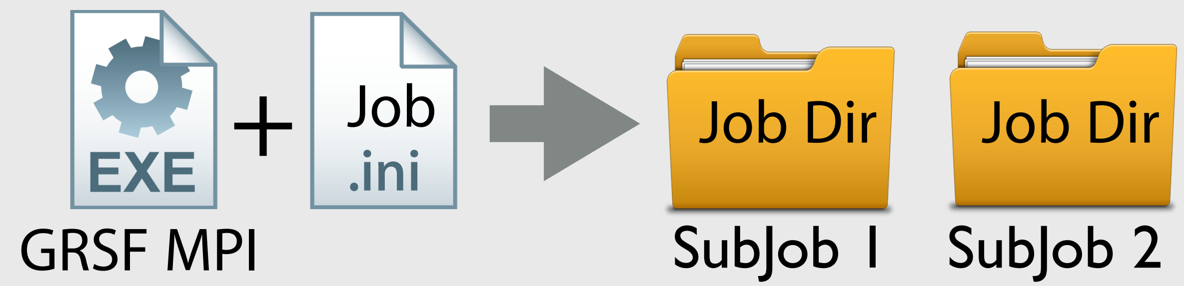
1. Configure Job with HPCJobConfigurator:
   `python configureJob.py -x Job.ini`
   `-p 92 -l 500 -r 4000 -t 720 -n 2`
2. Submit Job with **`bsub`**

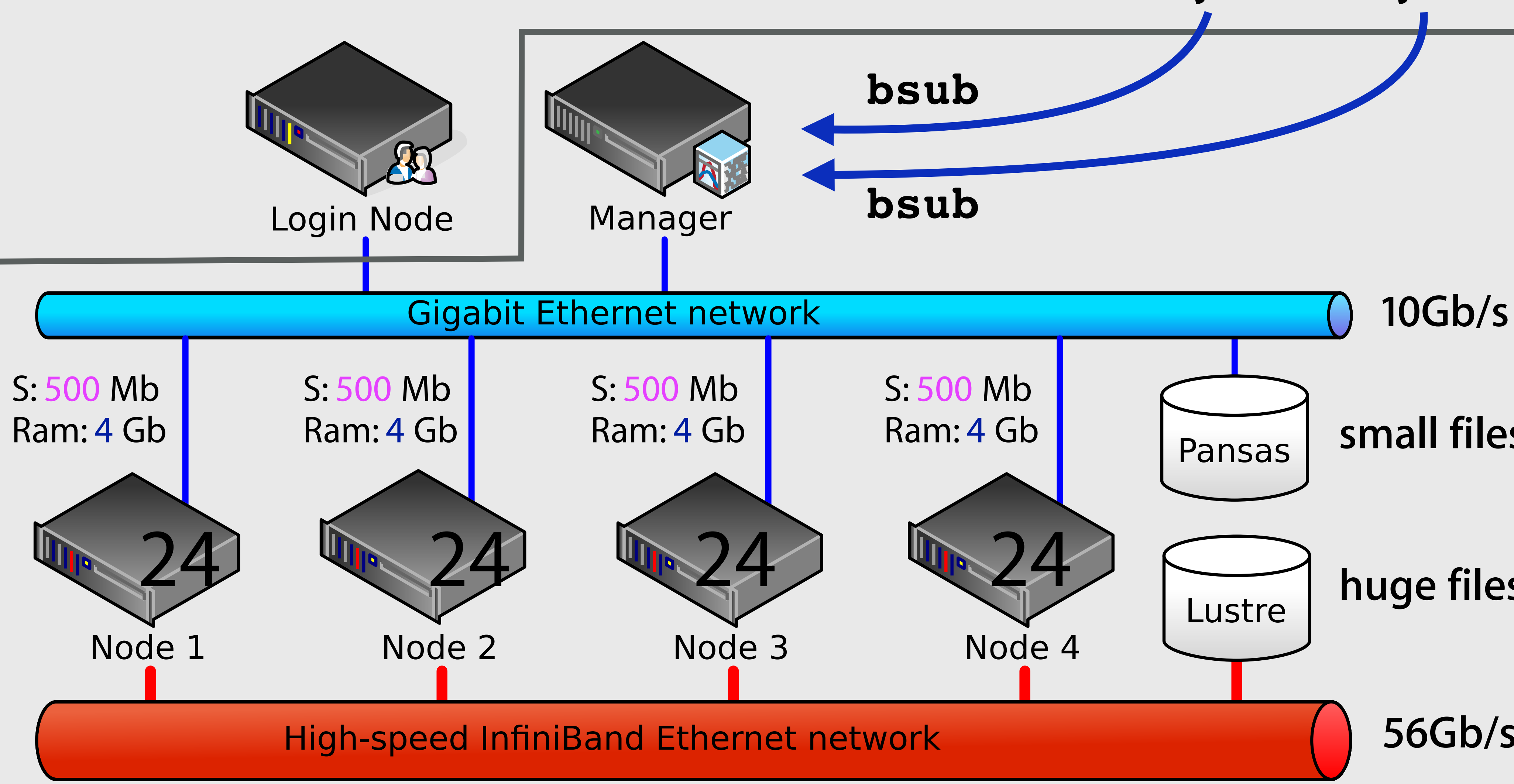




Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini
   -p 92 -l 500 -r 4000 -t 720 -n 2
2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**





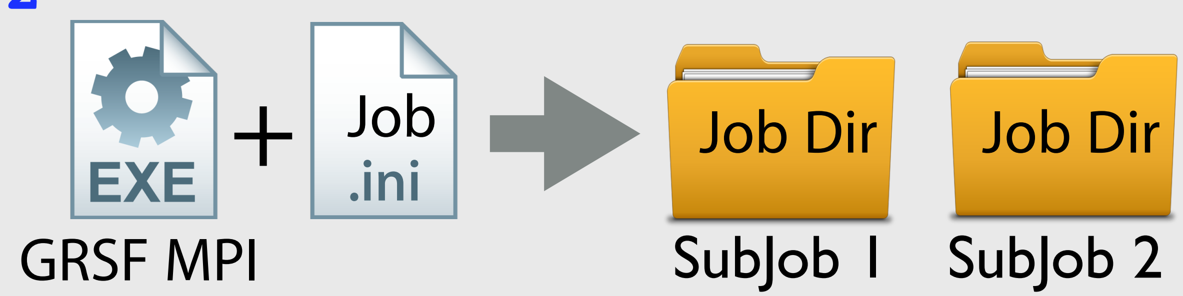
Clean Job Babysitting

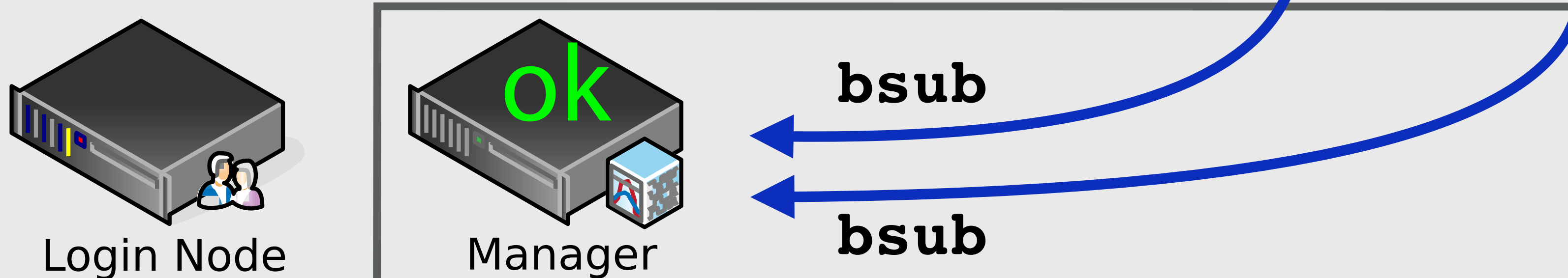
1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini
   -p 92 -l 500 -r 4000 -t 720 -n 2

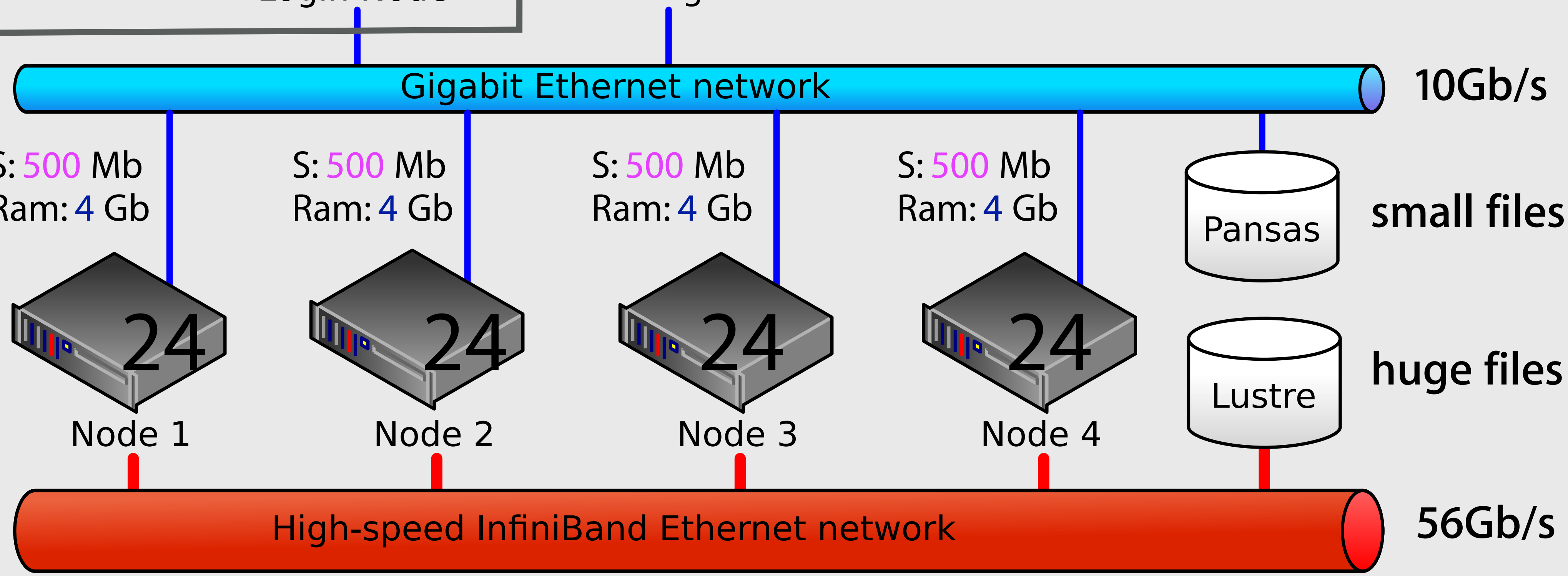
2. Submit Job with **bsub**

3. Trink coffee and **hope for the best!**

4. Job launched: woo!

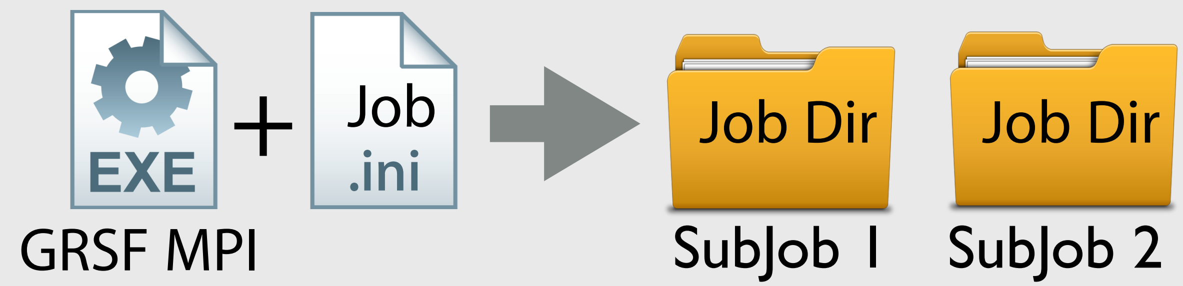


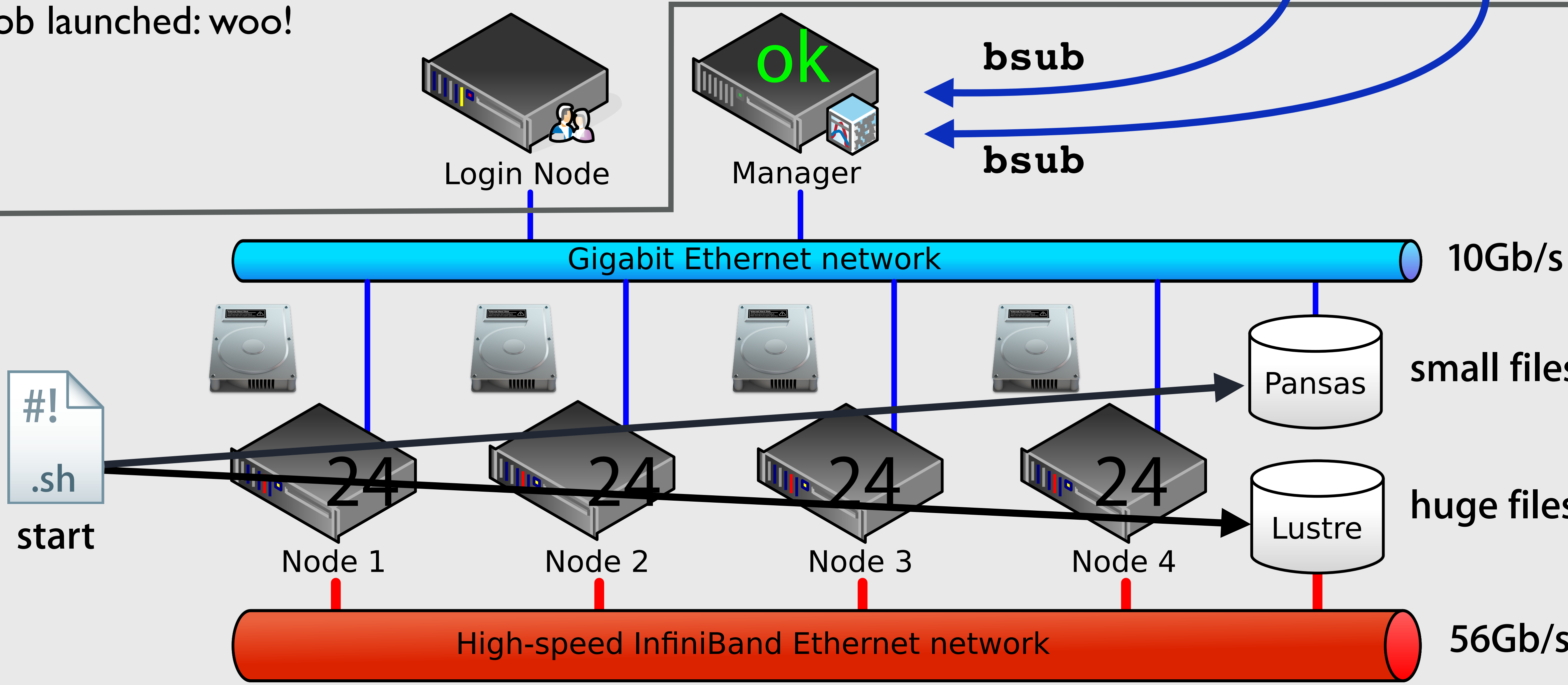




Clean Job Babysitting

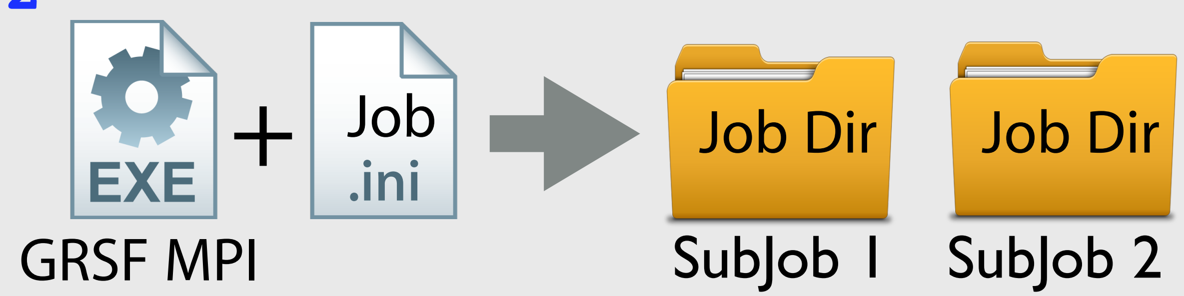
1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini
   -p 92 -l 500 -r 4000 -t 720 -n 2
2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**
4. Job launched: woo!

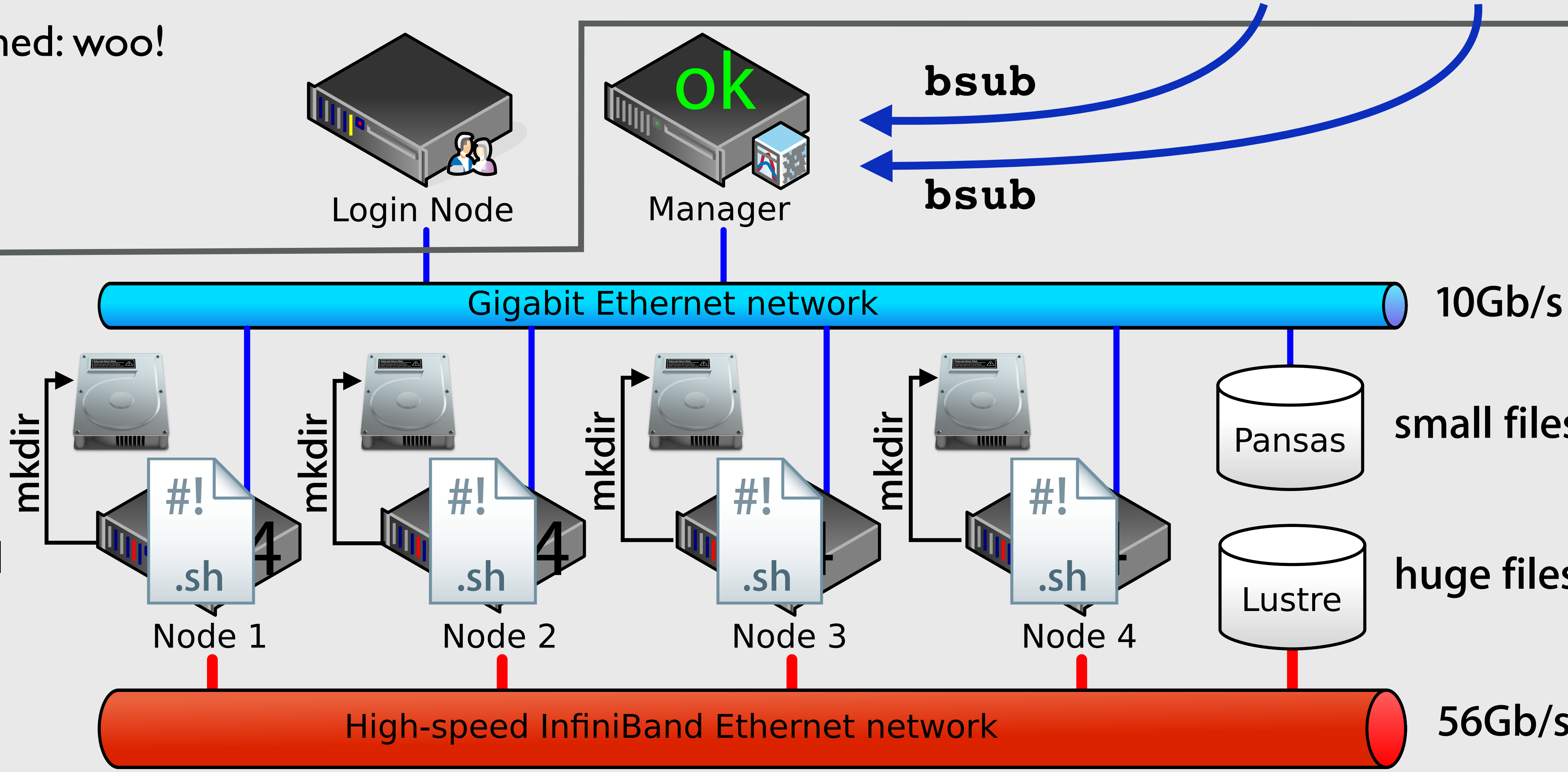




Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini
   -p 92 -l 500 -r 4000 -t 720 -n 2
2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**
4. Job launched: woo!

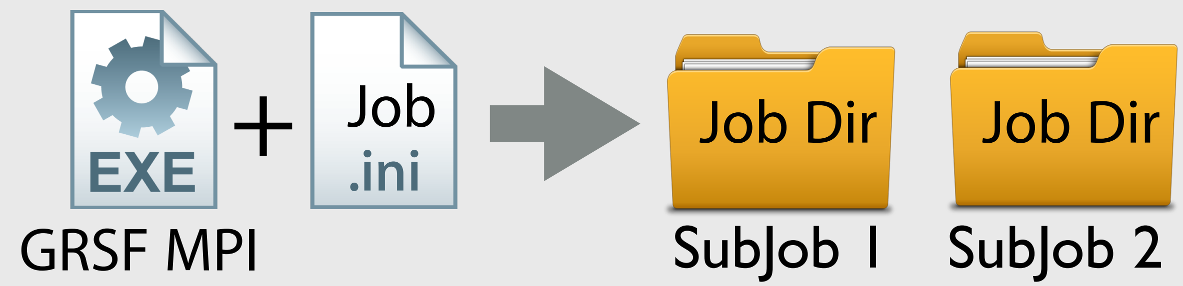


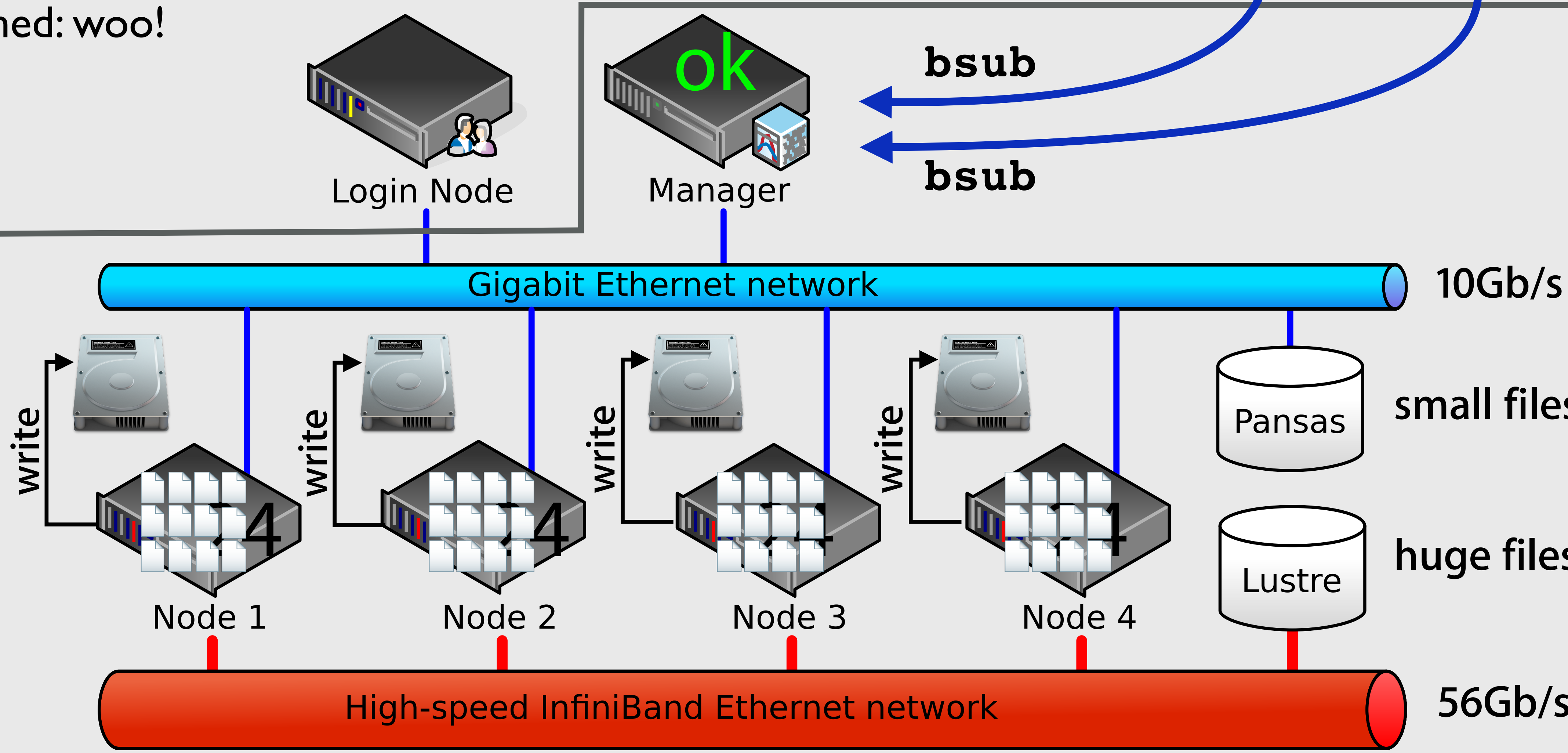


preProcess-PerNode.sh:
I/O from global to scratch

Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini
   -p 92 -l 500 -r 4000 -t 720 -n 2
2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**
4. Job launched: woo!

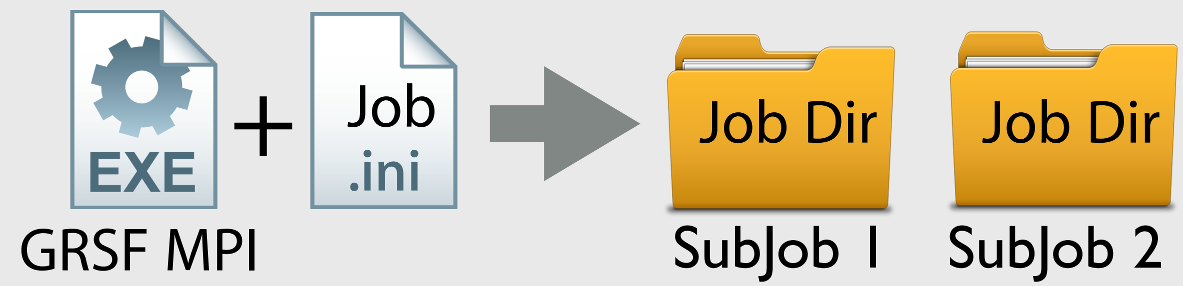




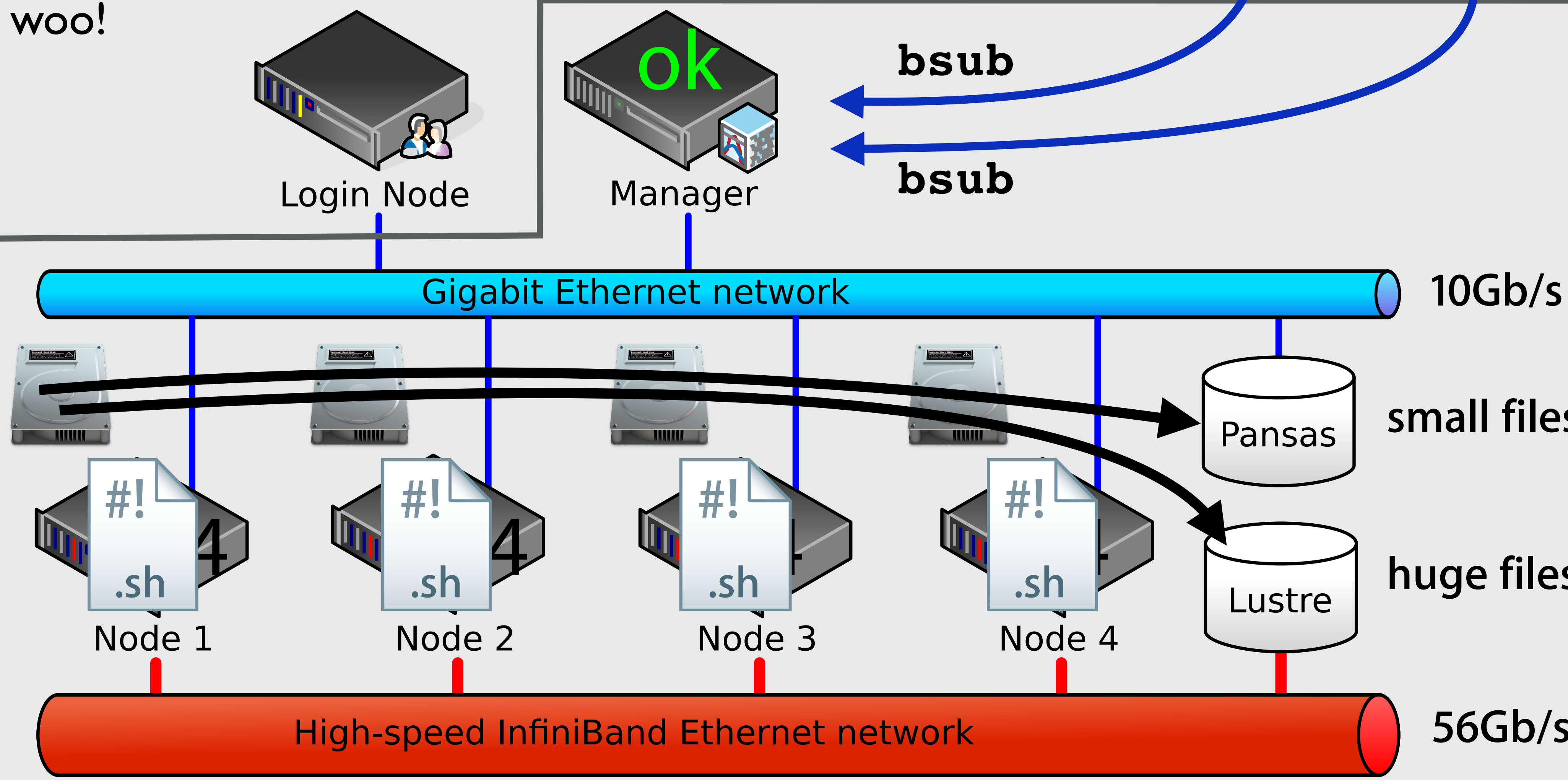
process.sh:
- compute
- do I/O

Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini
   -p 92 -l 500 -r 4000 -t 720 -n 2
2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**



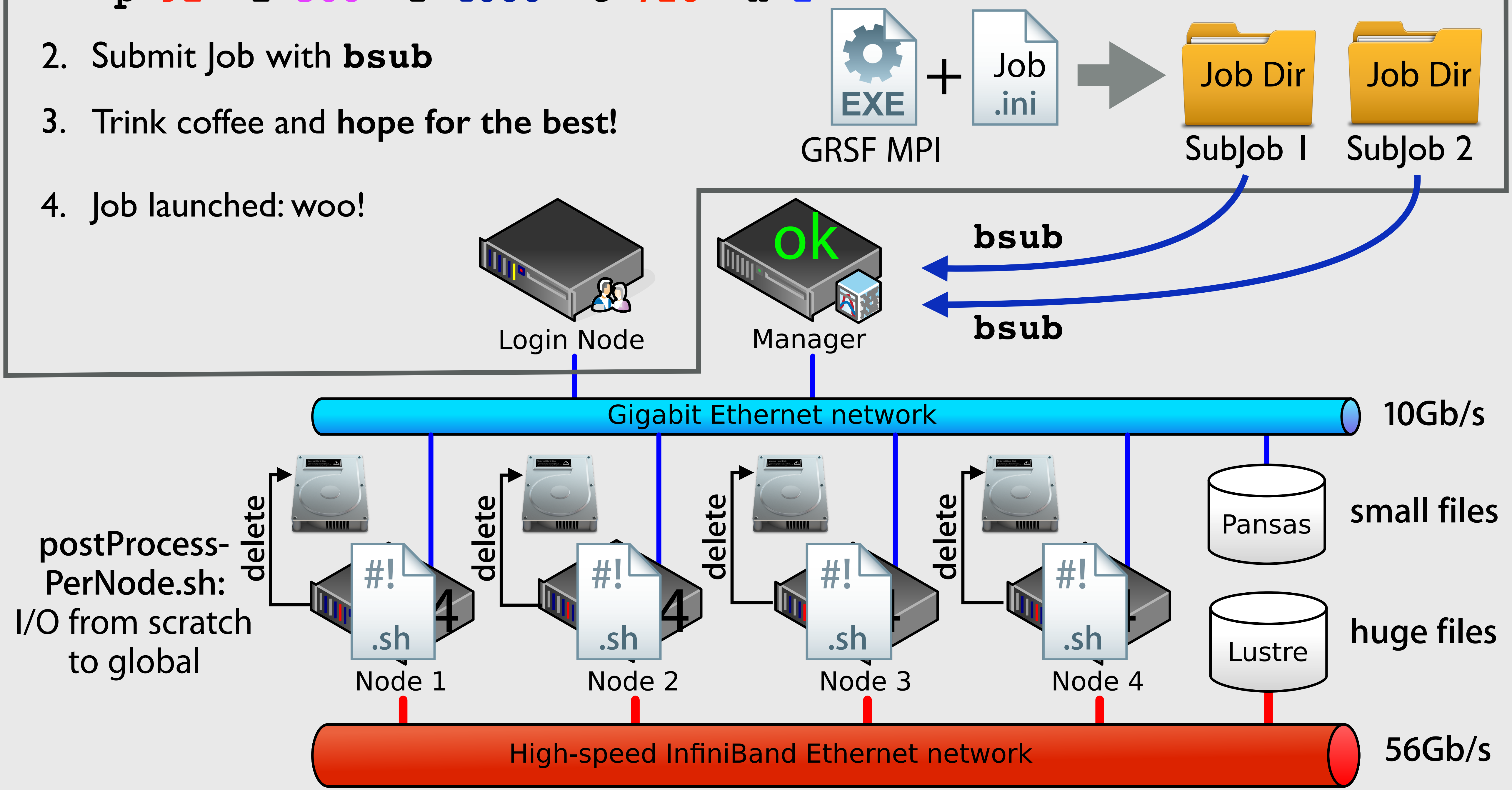
4. Job launched: woo!



postProcess-PerNode.sh:
I/O from scratch to global

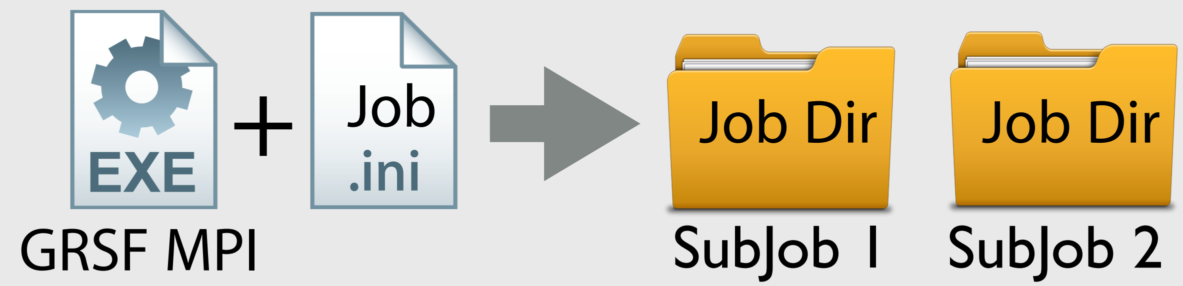
Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:
   python configureJob.py -x Job.ini -p 92 -l 500 -r 4000 -t 720 -n 2
2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**
4. Job launched: woo!

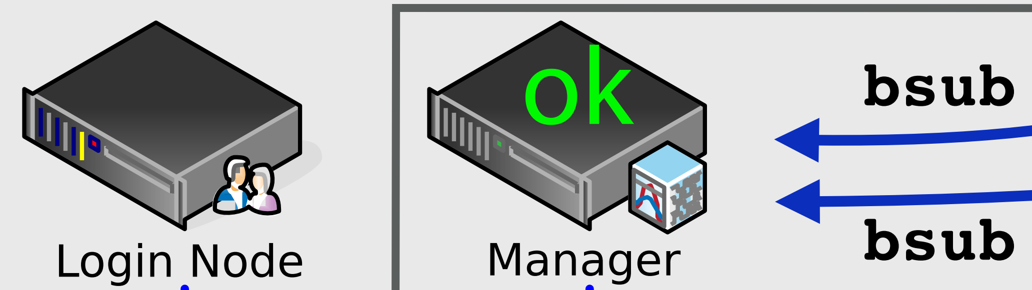


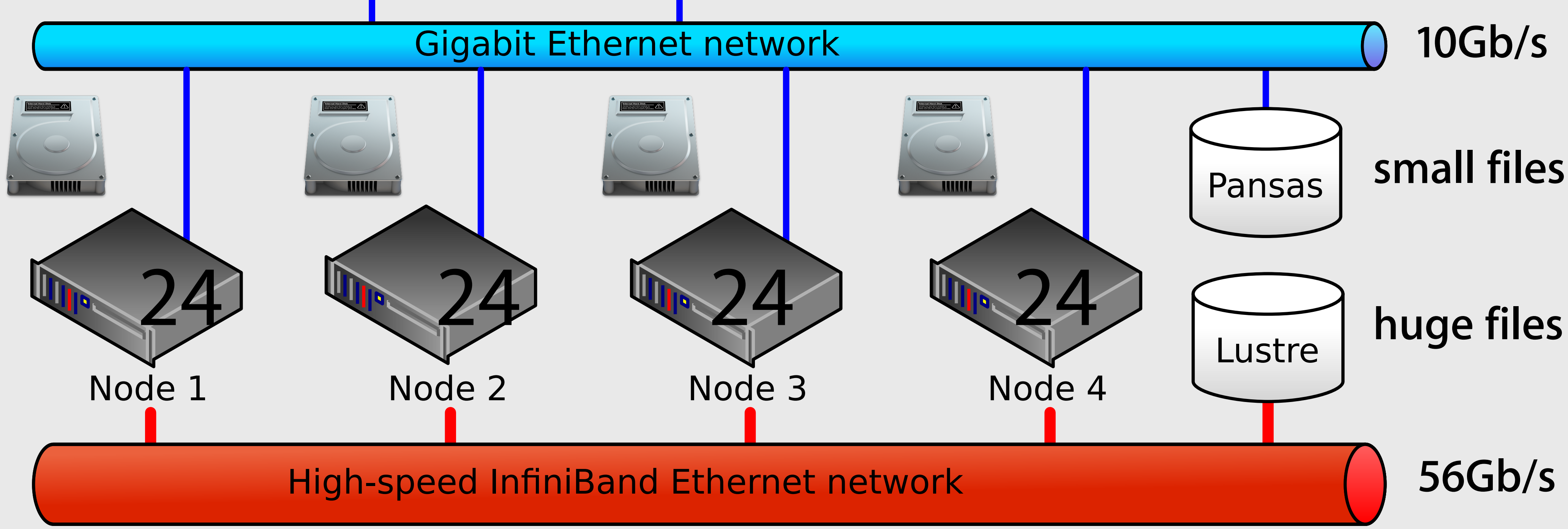
Clean Job Babysitting

1. Configure Job with HPCJobConfigurator:
   `python configureJob.py -x Job.ini`
   `-p 92 -l 500 -r 4000 -t 720 -n 2`



2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**
4. Job launched: woo!

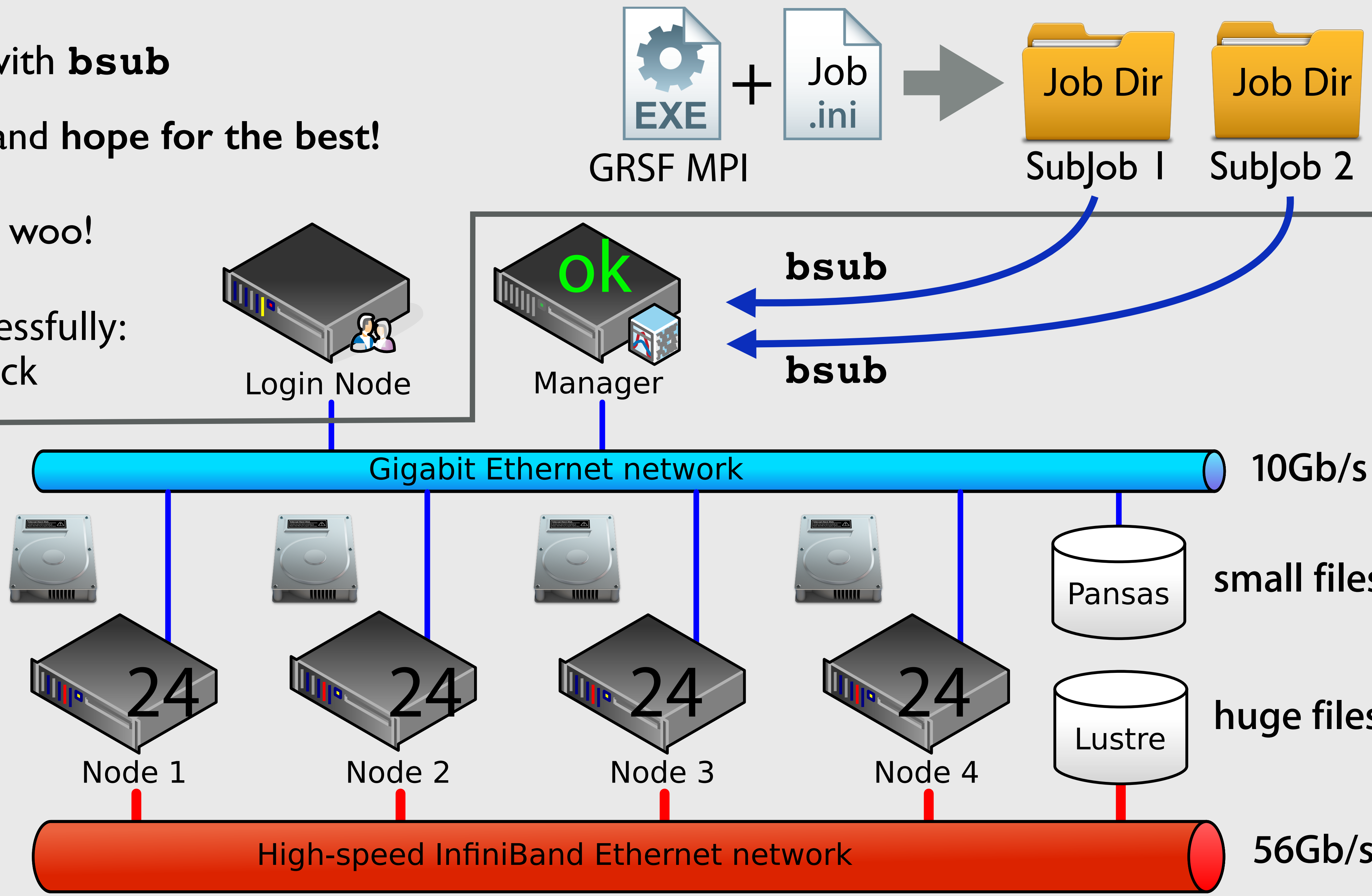




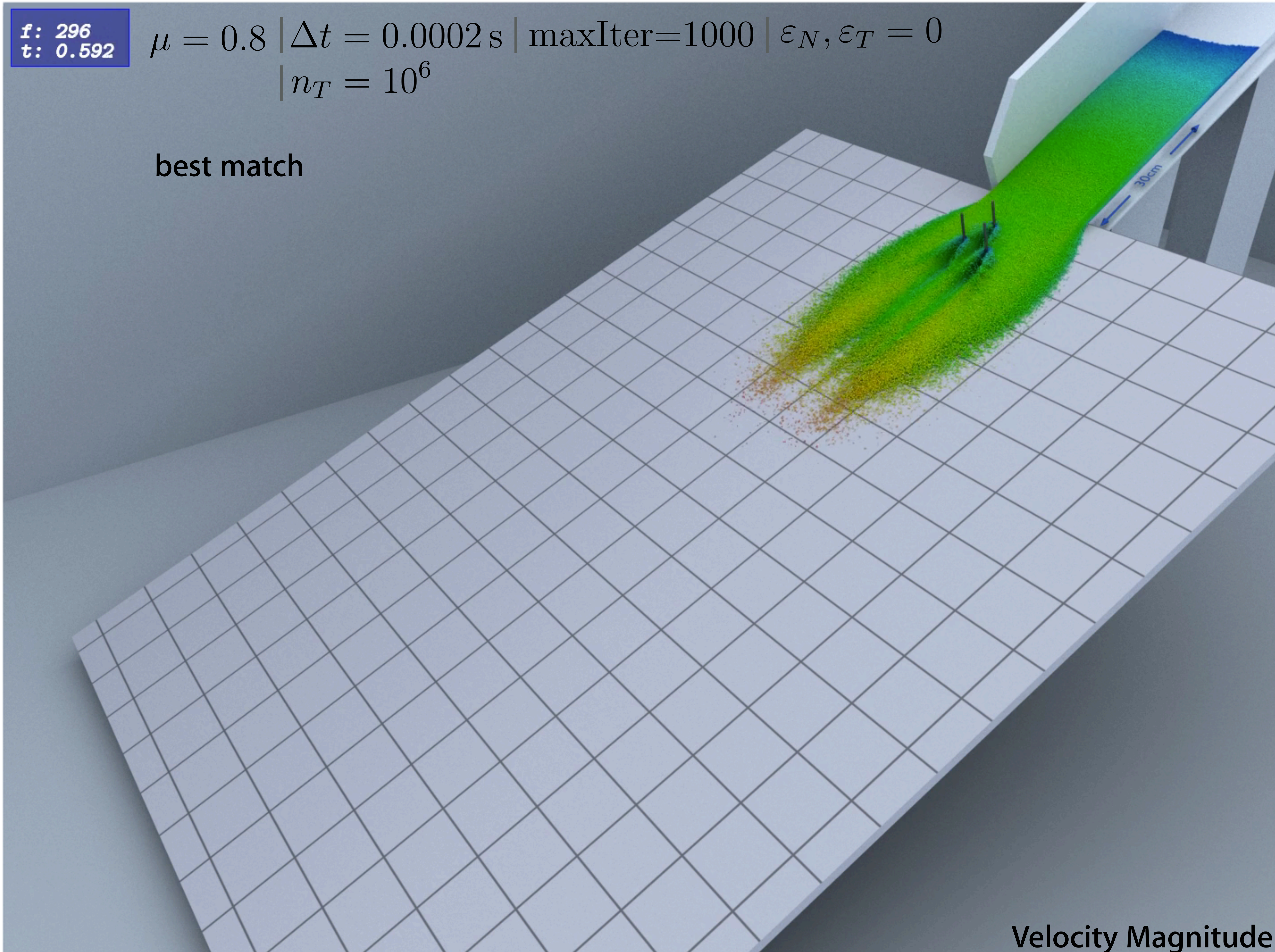
end:
- (file validation)
- notify user

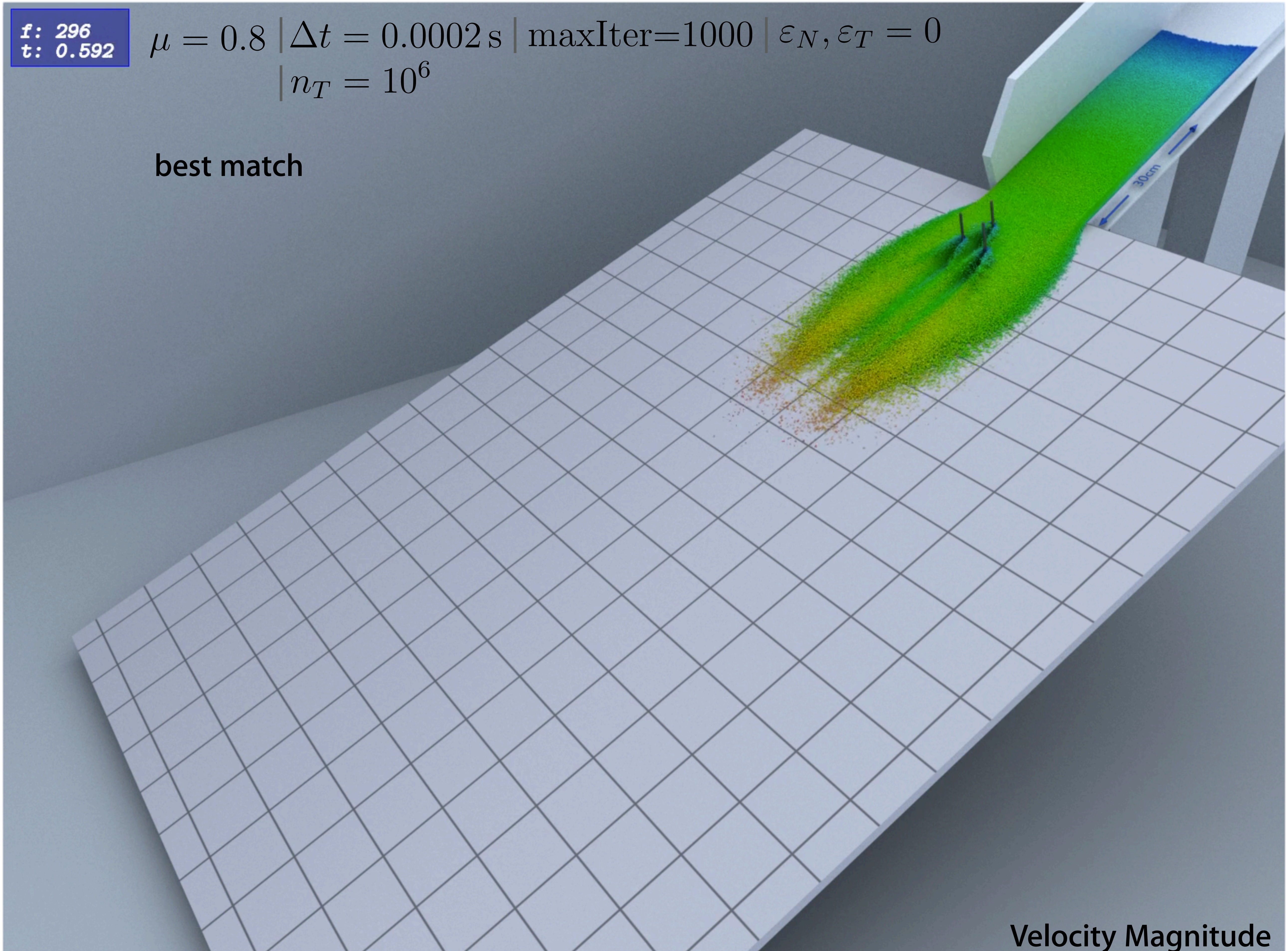
Clean Job Babysitting

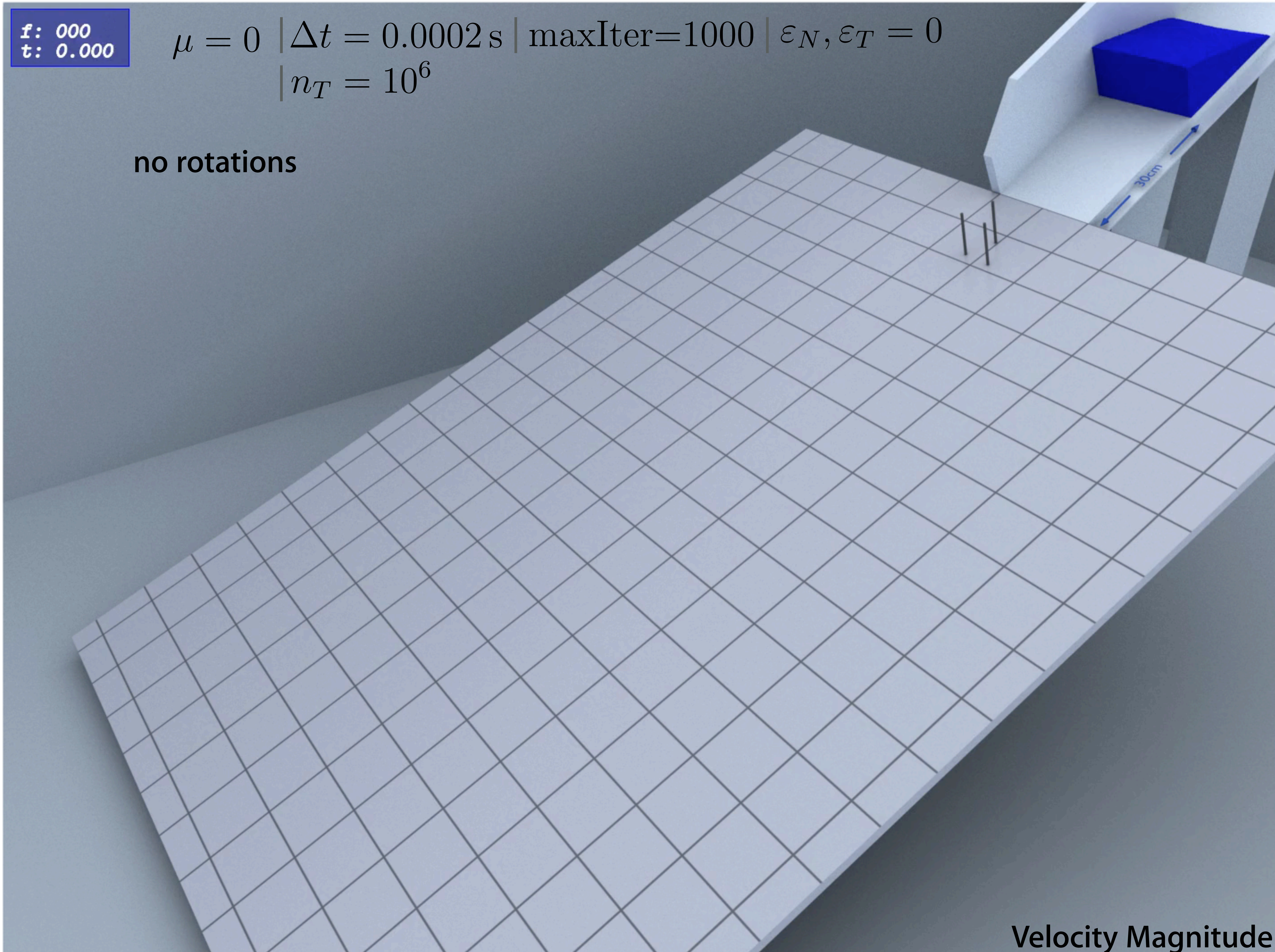
1. Configure Job with HPCJobConfigurator:
   `python configureJob.py -x Job.ini`
   `-p 92 -l 500 -r 4000 -t 720 -n 2`
2. Submit Job with **bsub**
3. Trink coffee and **hope for the best!**
4. Job launched: woo!
5. Job run successfully:
   copy data back

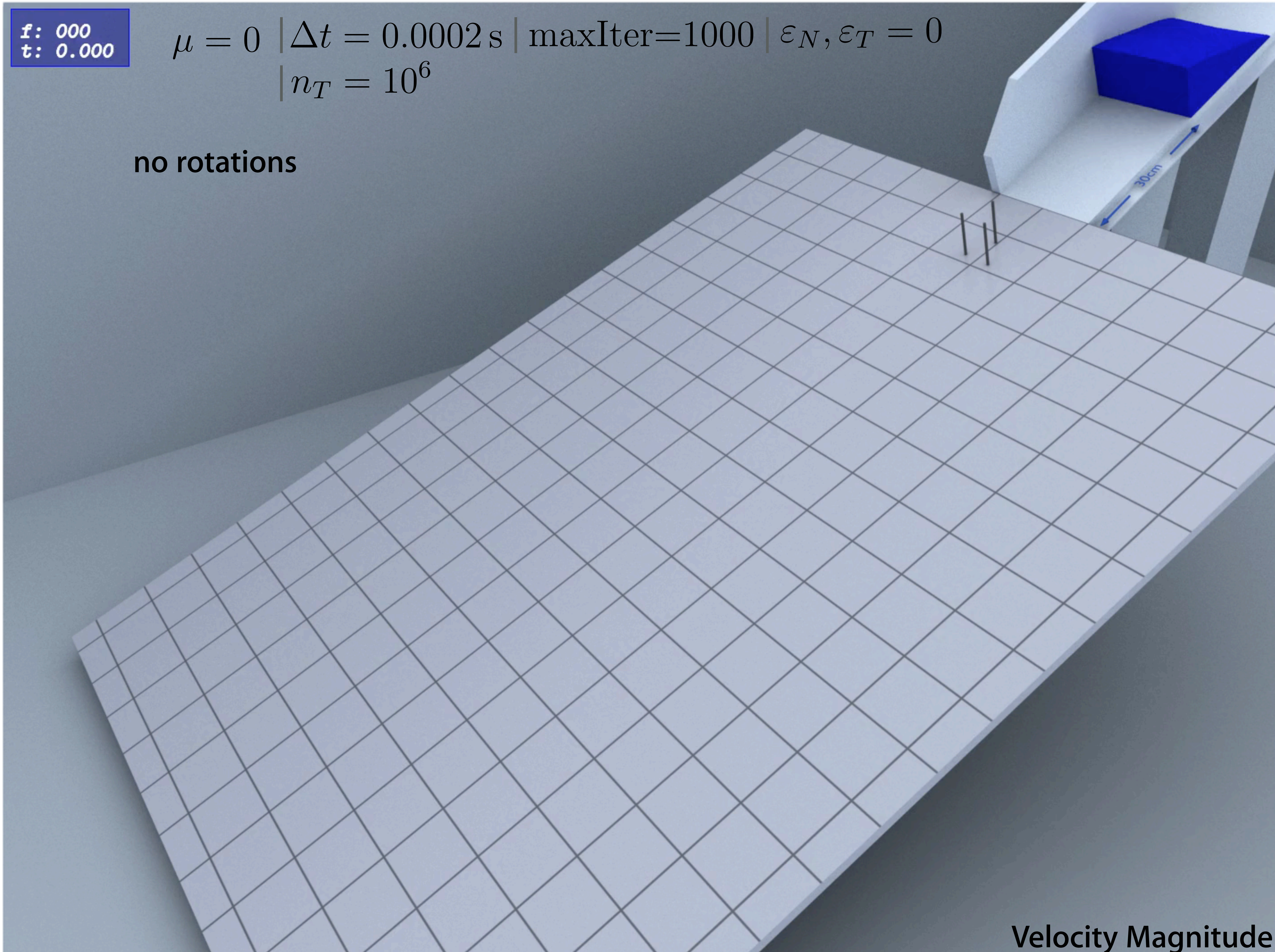


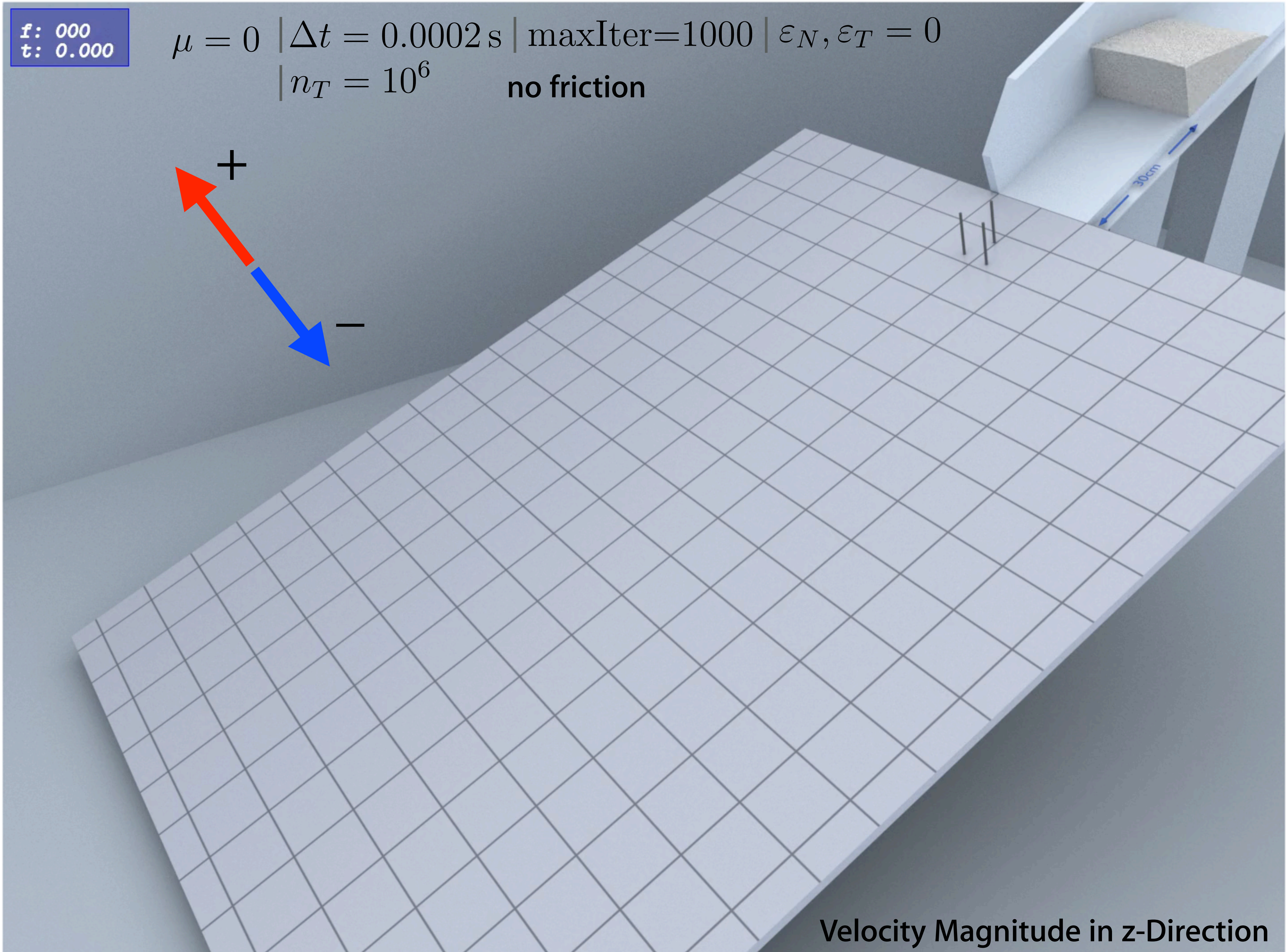
end:
- (file validation)
- notify user

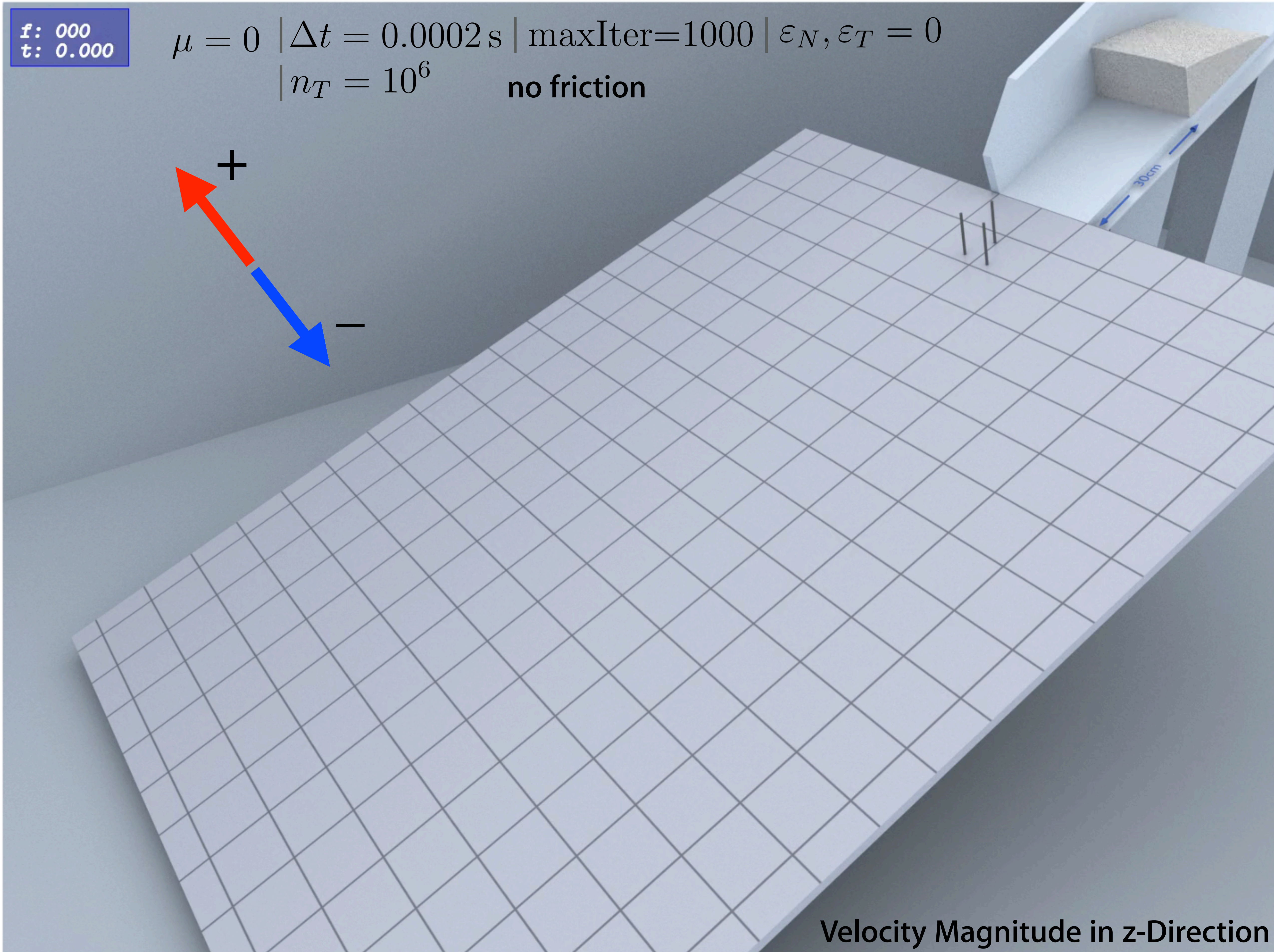

f: 296
t: 0.592
$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter{=}1000} \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$
best match
30cm
Velocity Magnitude


f: 296
t: 0.592
$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0$
$\mid n_T = 10^6$
best match
30cm
Velocity Magnitude

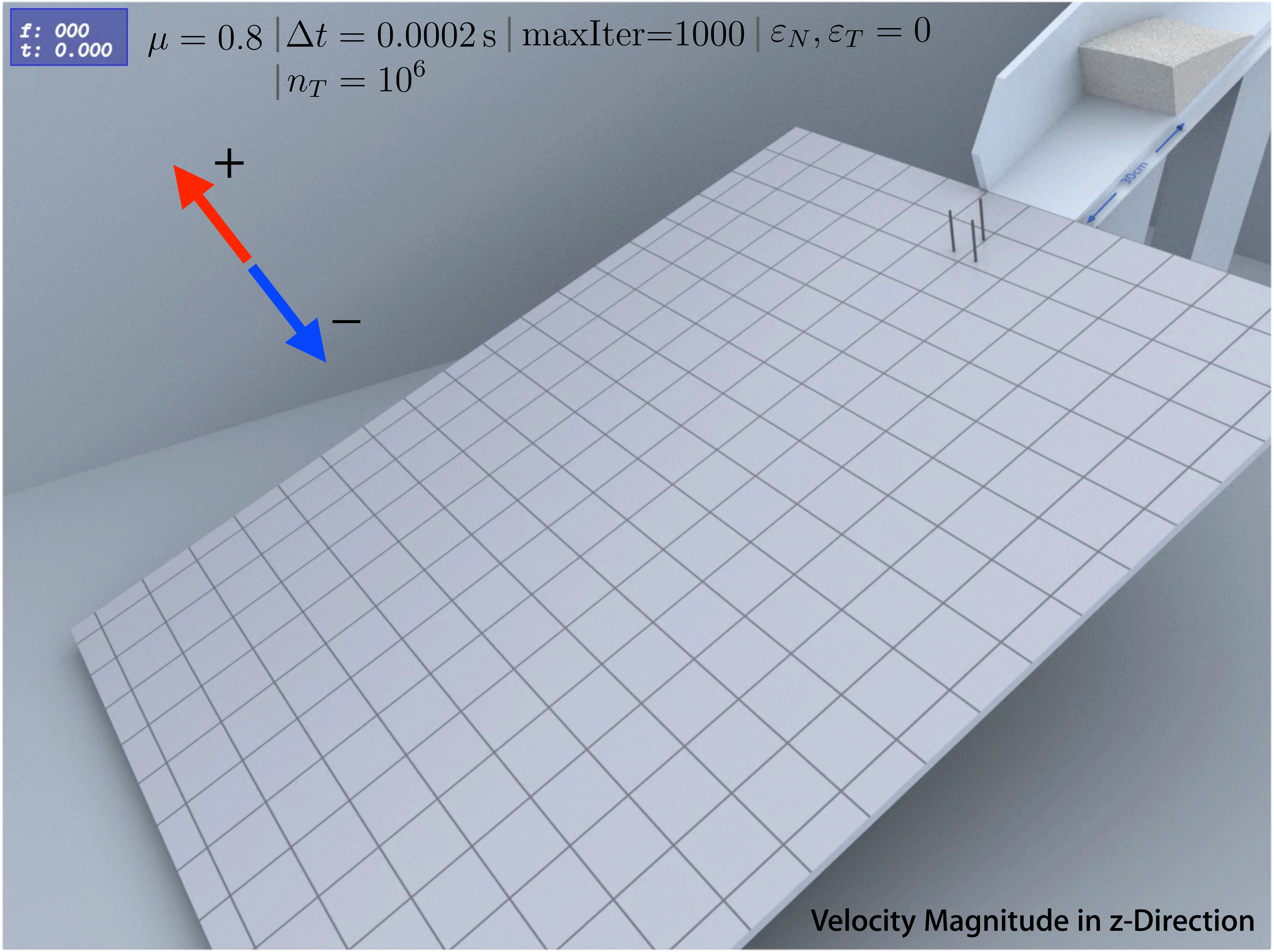

f: 000
t: 0.000
$\mu = 0 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter{=}1000} \mid \varepsilon_N, \varepsilon_T = 0$
$\mid n_T = 10^6$
no rotations
30cm
Velocity Magnitude

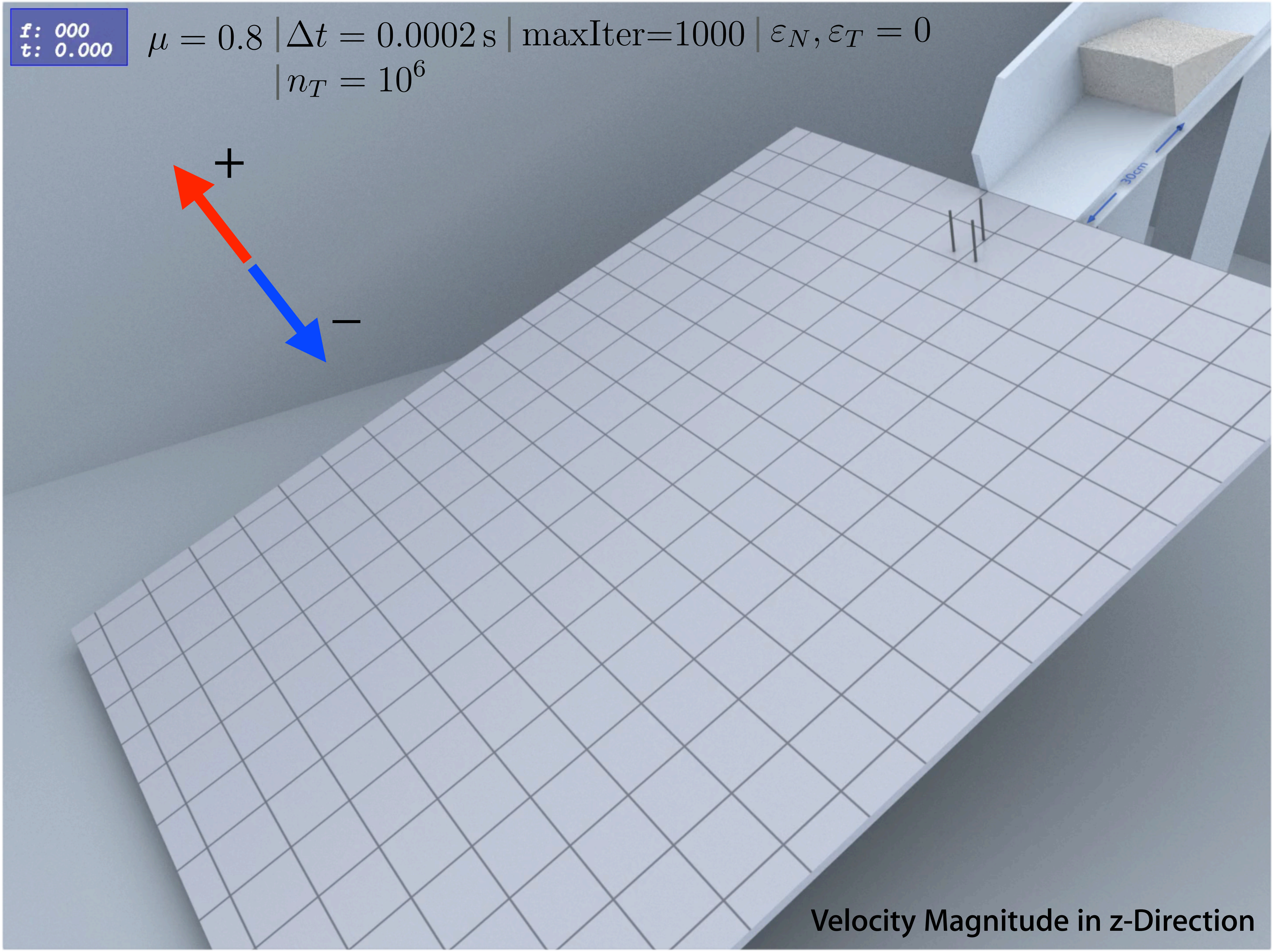

f: 000
t: 0.000
$\mu = 0 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$
no rotations
30cm
Velocity Magnitude


f: 000
t: 0.000
$\mu = 0 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0$
$\mid n_T = 10^6$
no friction
+
−
30cm
Velocity Magnitude in z-Direction

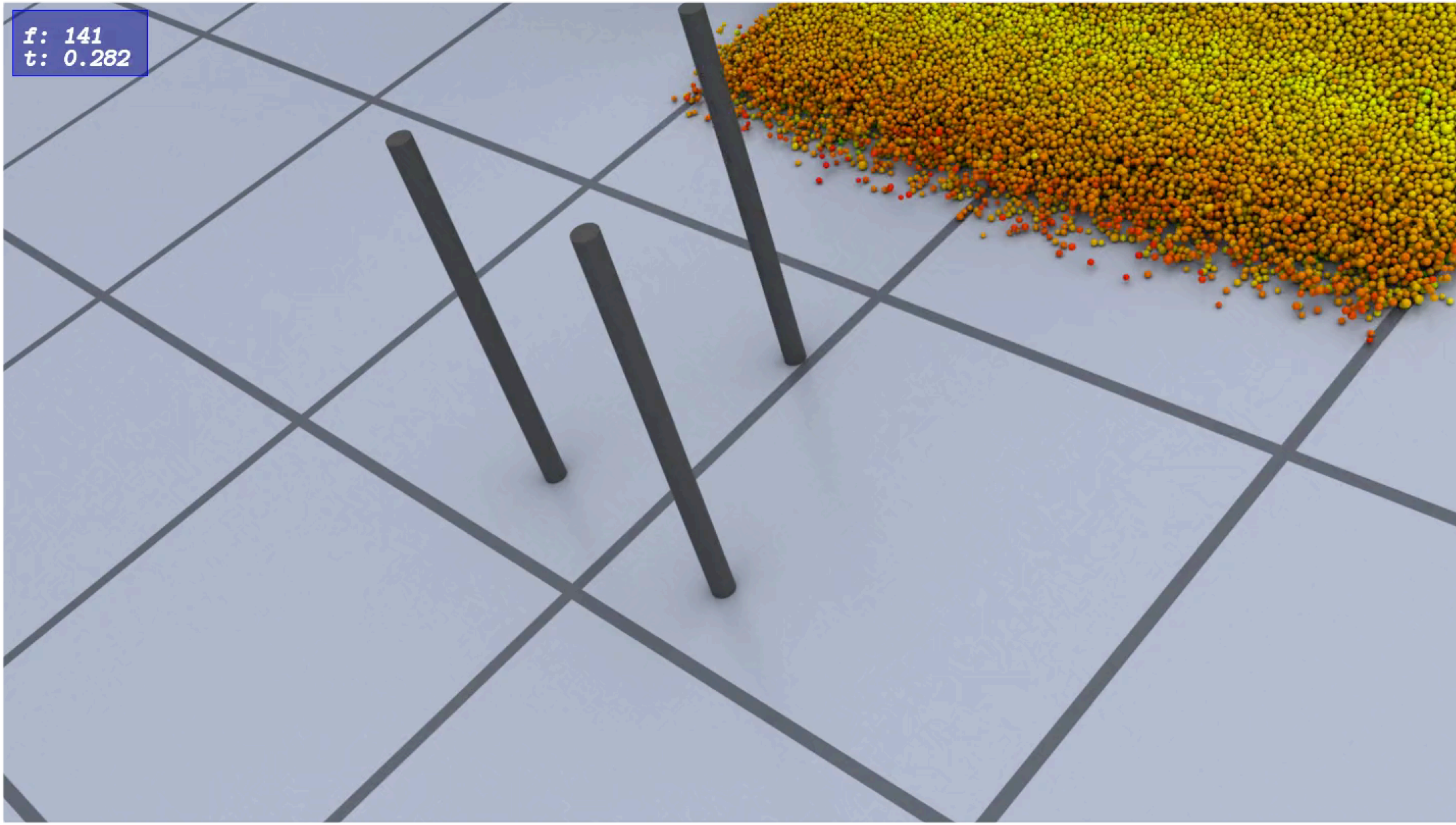


f: 000
t: 0.000
$\mu = 0 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0$
$\mid n_T = 10^6$
no friction
+
−
30cm
Velocity Magnitude in z-Direction


f: 000
t: 0.000
$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0$
$\mid n_T = 10^6$
+
−
30cm
Velocity Magnitude in z-Direction


f: 000
t: 0.000
$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter}{=}1000 \mid \varepsilon_N, \varepsilon_T = 0$
$\mid n_T = 10^6$
+
−
30cm
Velocity Magnitude in z-Direction

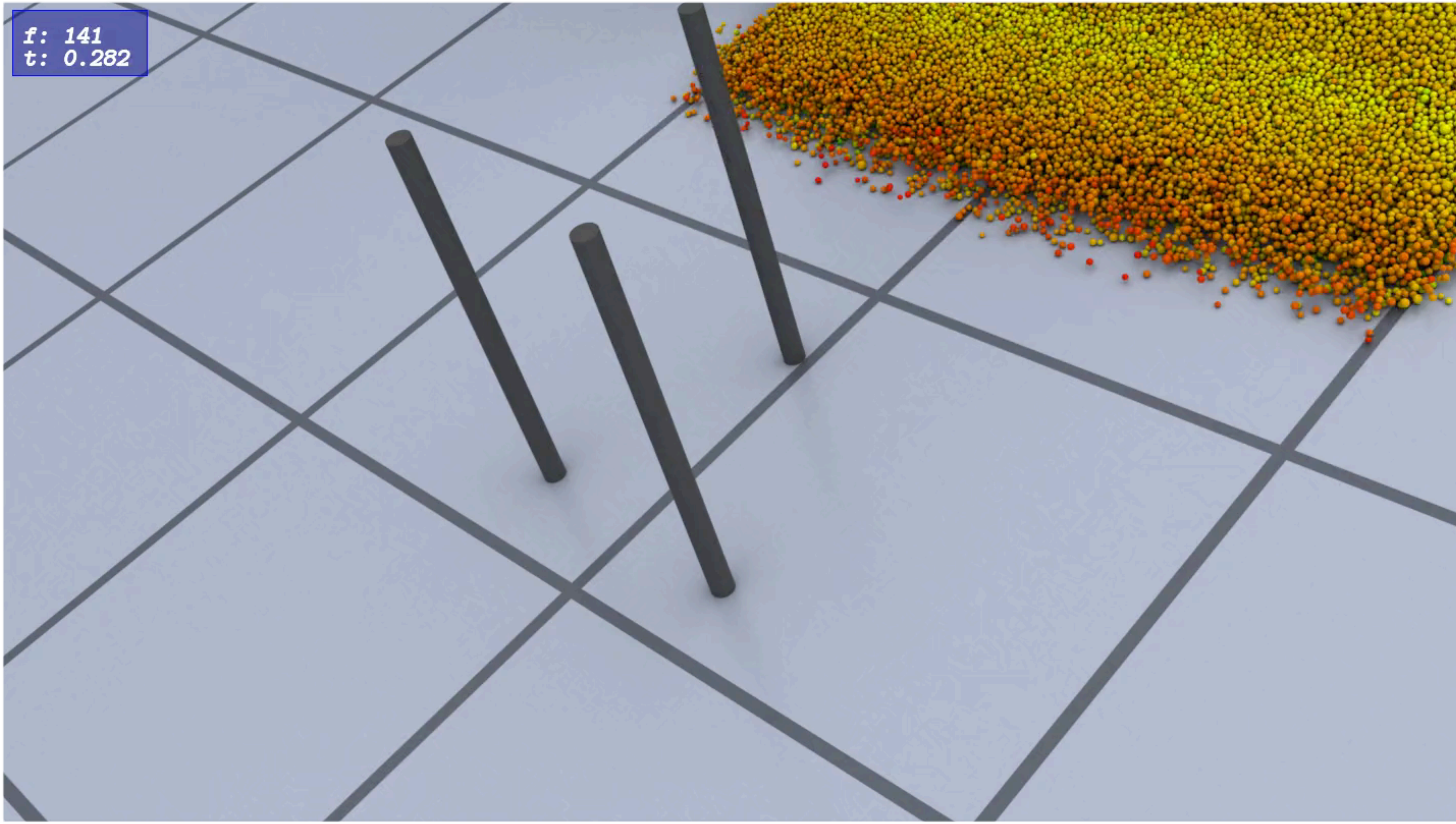
Velocity Magnitude



$\mu = 0.8 \,|\, \Delta t = 0.0002\,\mathrm{s} \,|\, \mathrm{maxIter}{=}1000 \,|\, \varepsilon_N, \varepsilon_T = 0 \,|\, n_T = 10^6$

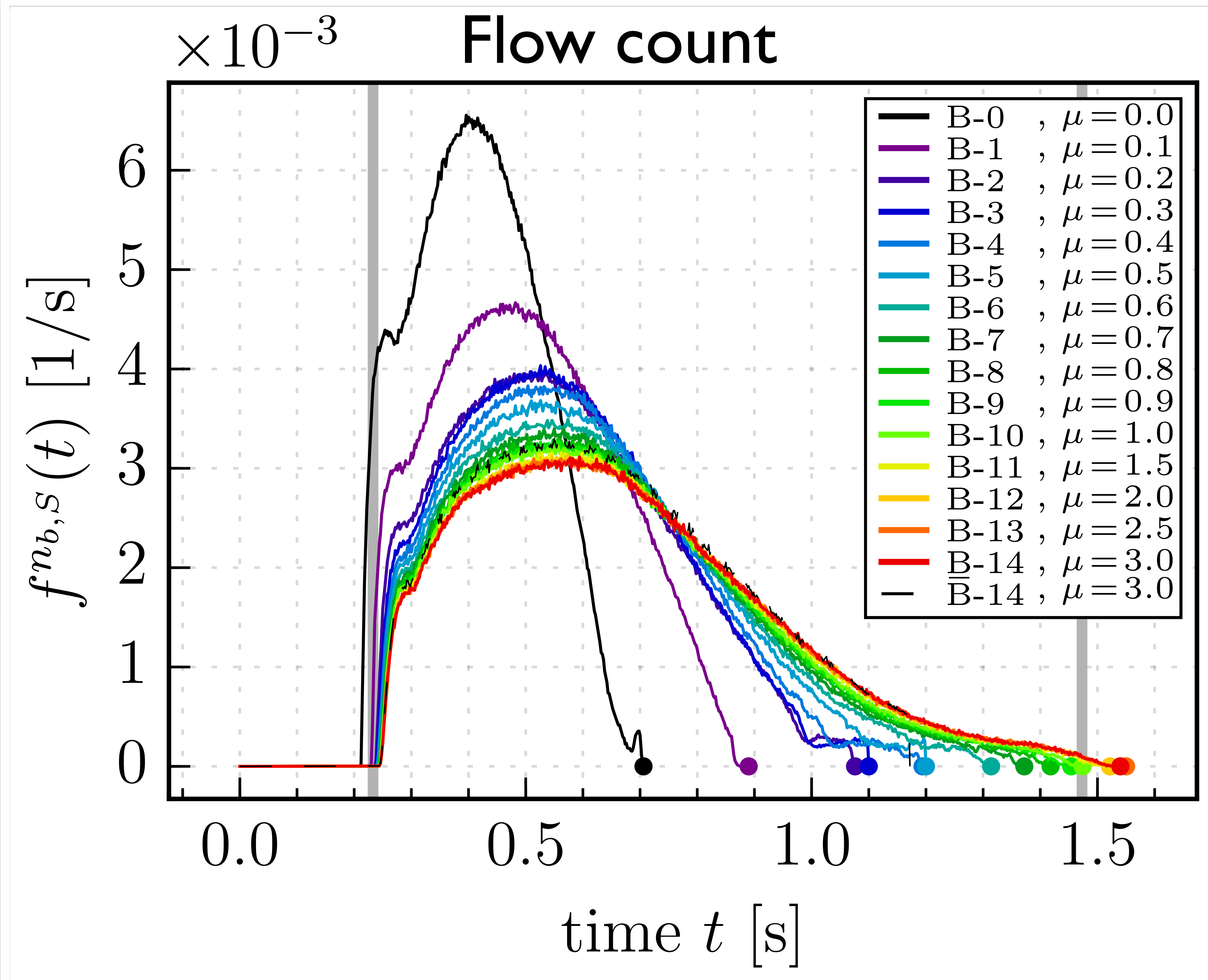
best match

Velocity Magnitude

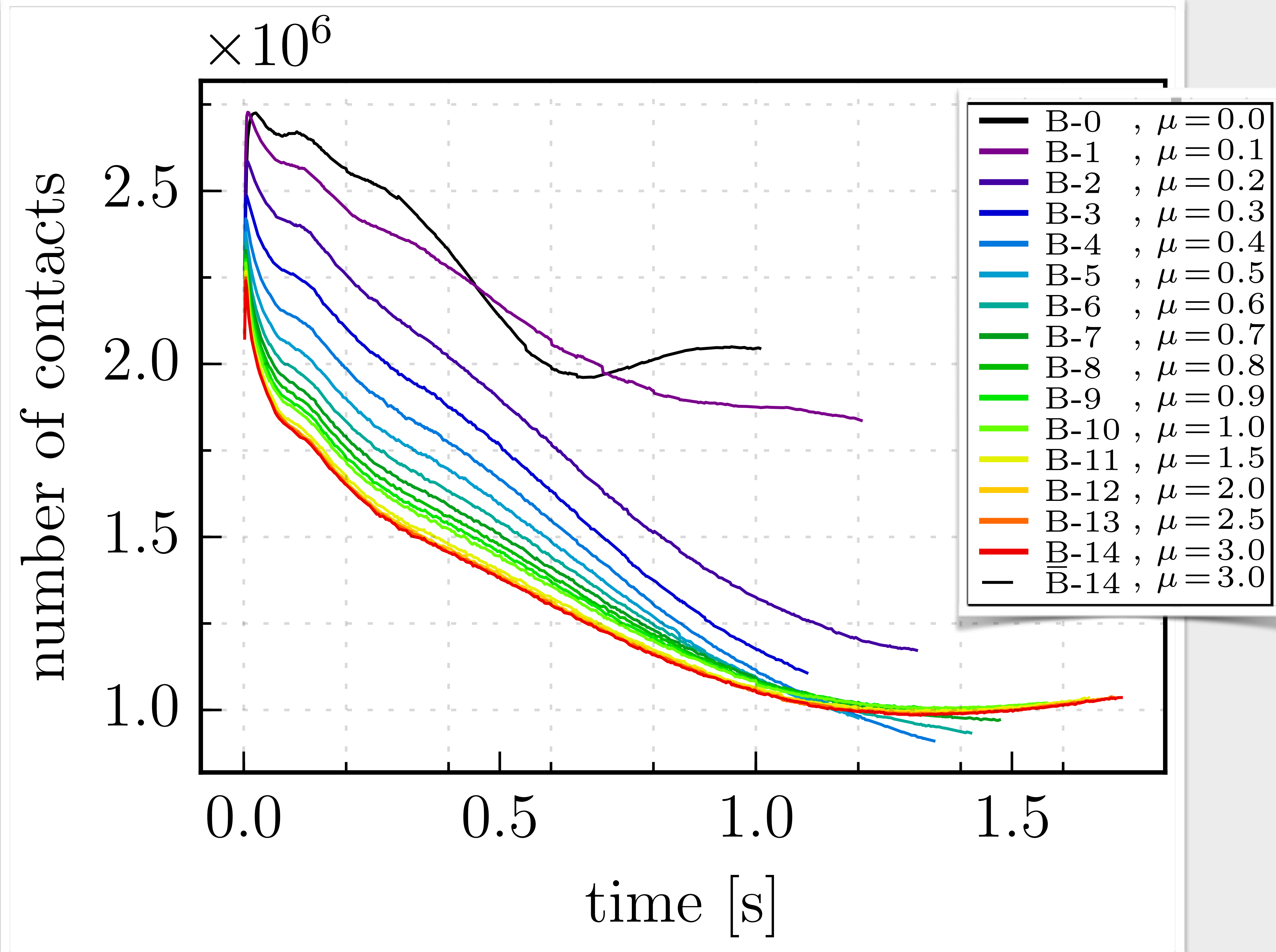


$\mu = 0.8 \mid \Delta t = 0.0002\,\mathrm{s} \mid \mathrm{maxIter{=}1000} \mid \varepsilon_N, \varepsilon_T = 0 \mid n_T = 10^6$ **best match**

Comparison

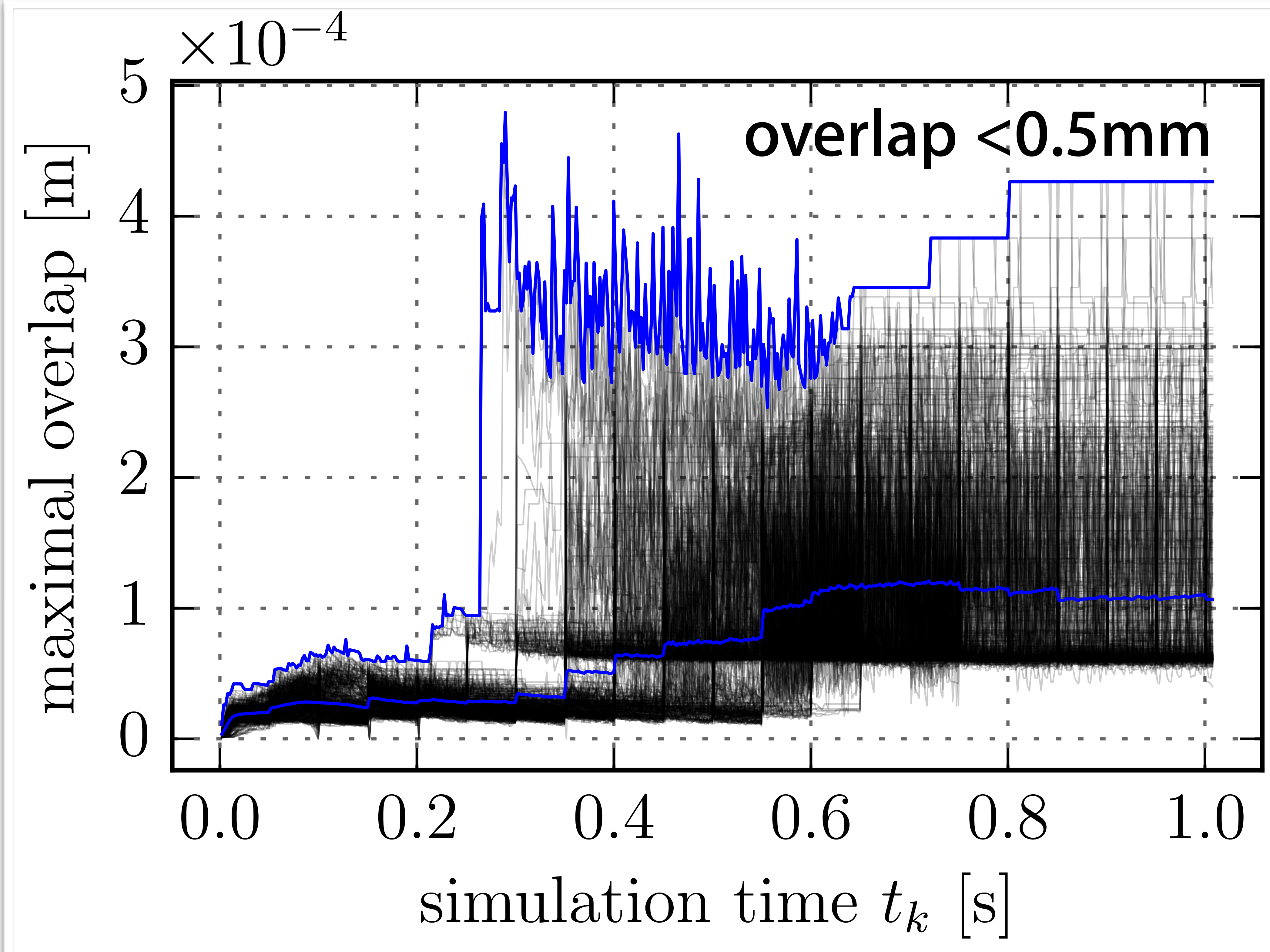

Flow count
×10^−3
$f n_{b,S}(t)$ [1/s]
time t [s]
0
1
2
3
4
5
6
0.0
0.5
1.0
1.5
B-0 , μ = 0.0
B-1 , μ = 0.1
B-2 , μ = 0.2
B-3 , μ = 0.3
B-4 , μ = 0.4
B-5 , μ = 0.5
B-6 , μ = 0.6
B-7 , μ = 0.7
B-8 , μ = 0.8
B-9 , μ = 0.9
B-10 , μ = 1.0
B-11 , μ = 1.5
B-12 , μ = 2.0
B-13 , μ = 2.5
B-14 , μ = 3.0
B-14 , μ = 3.0

Comparison


$\times 10^6$
number of contacts
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
time [s]
B-0 , $\mu = 0.0$
B-1 , $\mu = 0.1$
B-2 , $\mu = 0.2$
B-3 , $\mu = 0.3$
B-4 , $\mu = 0.4$
B-5 , $\mu = 0.5$
B-6 , $\mu = 0.6$
B-7 , $\mu = 0.7$
B-8 , $\mu = 0.8$
B-9 , $\mu = 0.9$
B-10 , $\mu = 1.0$
B-11 , $\mu = 1.5$
B-12 , $\mu = 2.0$
B-13 , $\mu = 2.5$
B-14 , $\mu = 3.0$
$\overline{\text{B}}$-14 , $\mu = 3.0$

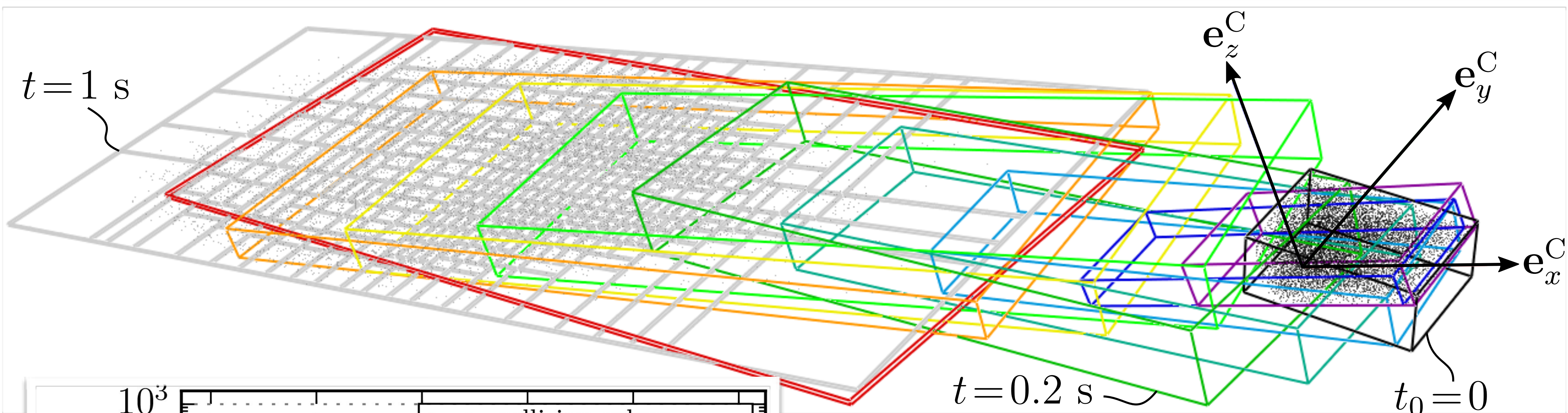
Penetration Depth

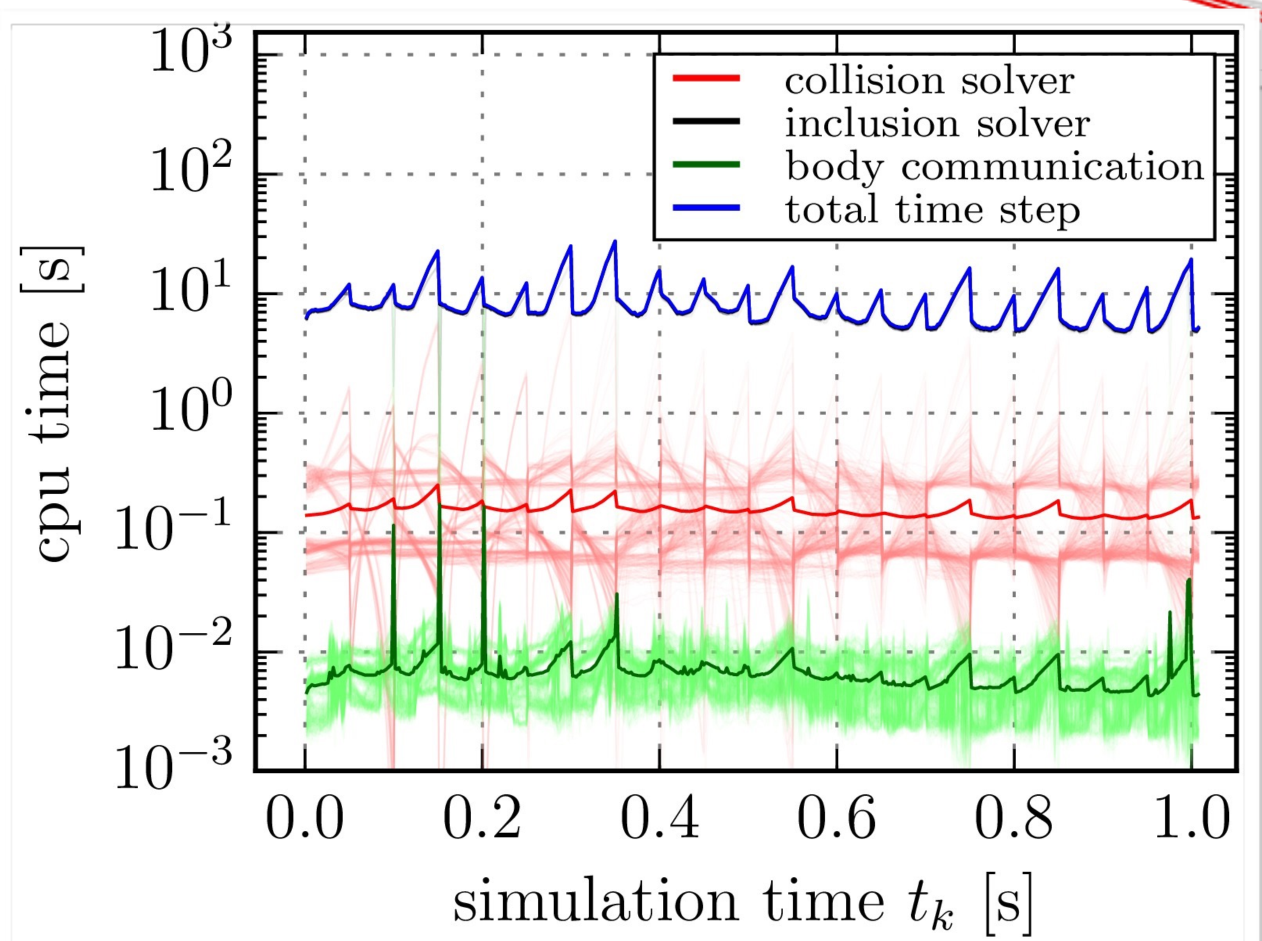
$\mu = 0$ less overlap for $\mu > 0$



Simulation Topologies

$\mu = 0$





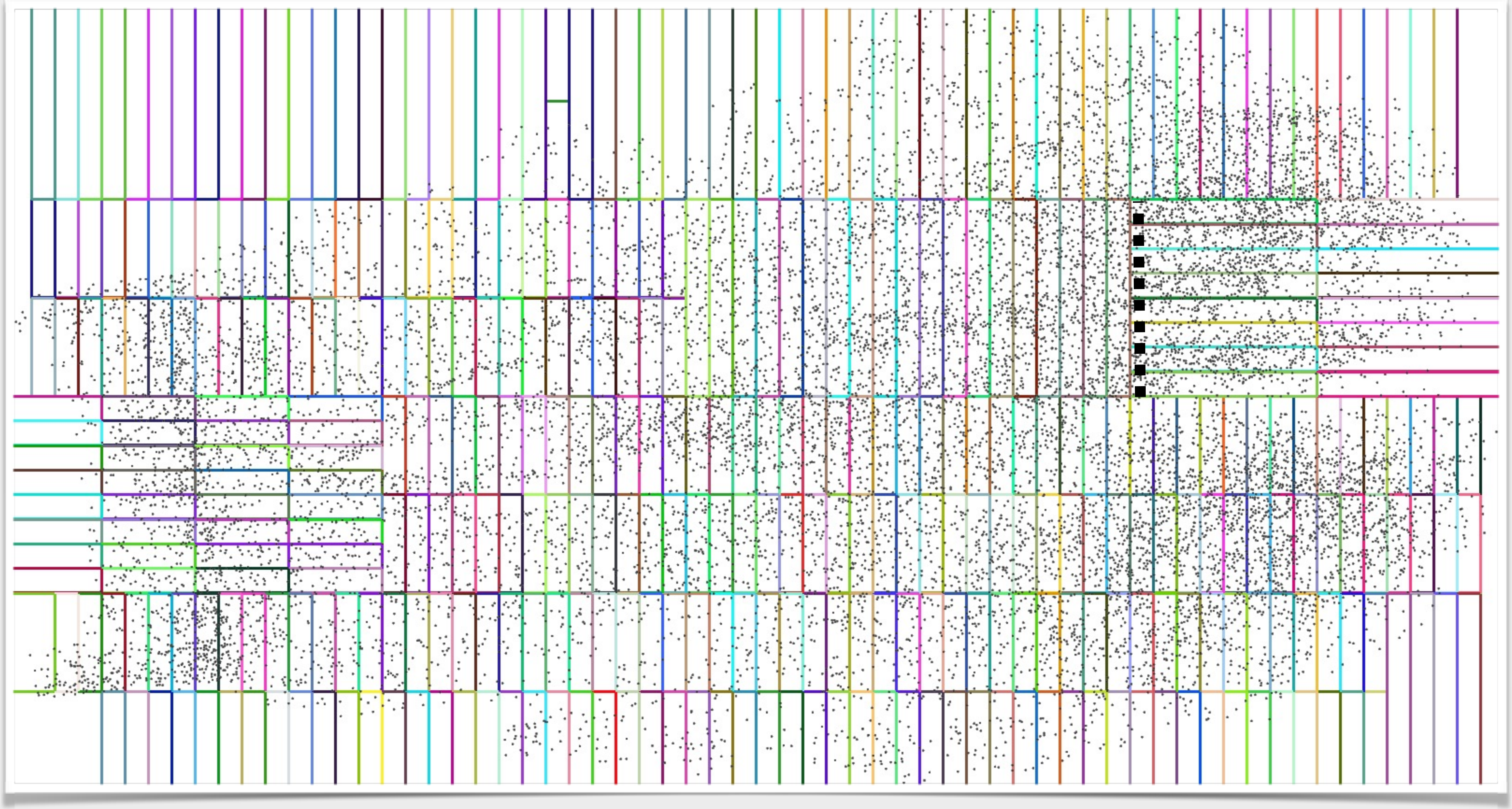
Communication: ~ 0.02 s

Collision: ~ 0.15 s

Time Step: 10 - 30 s

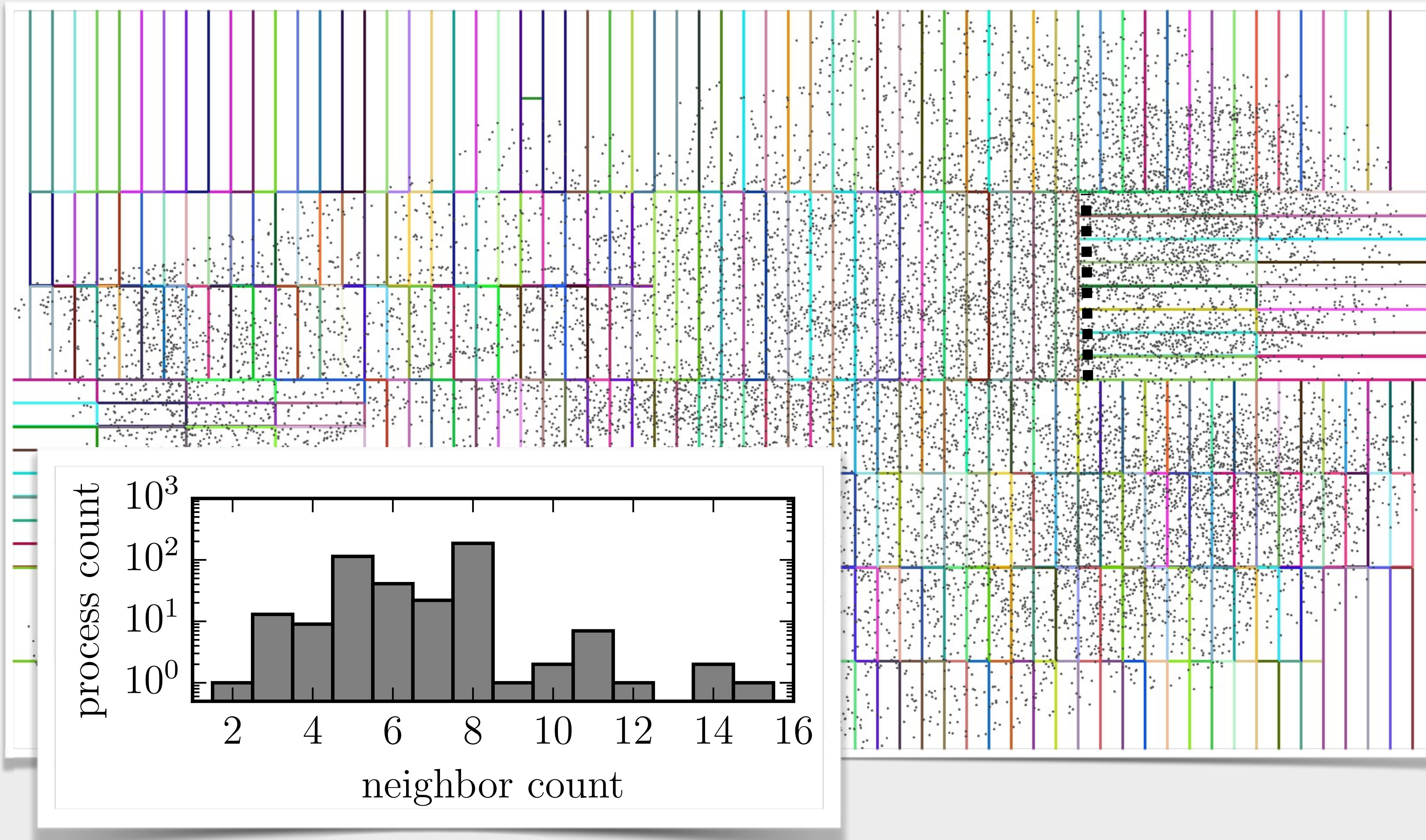
KdTree Topology

worse splitting



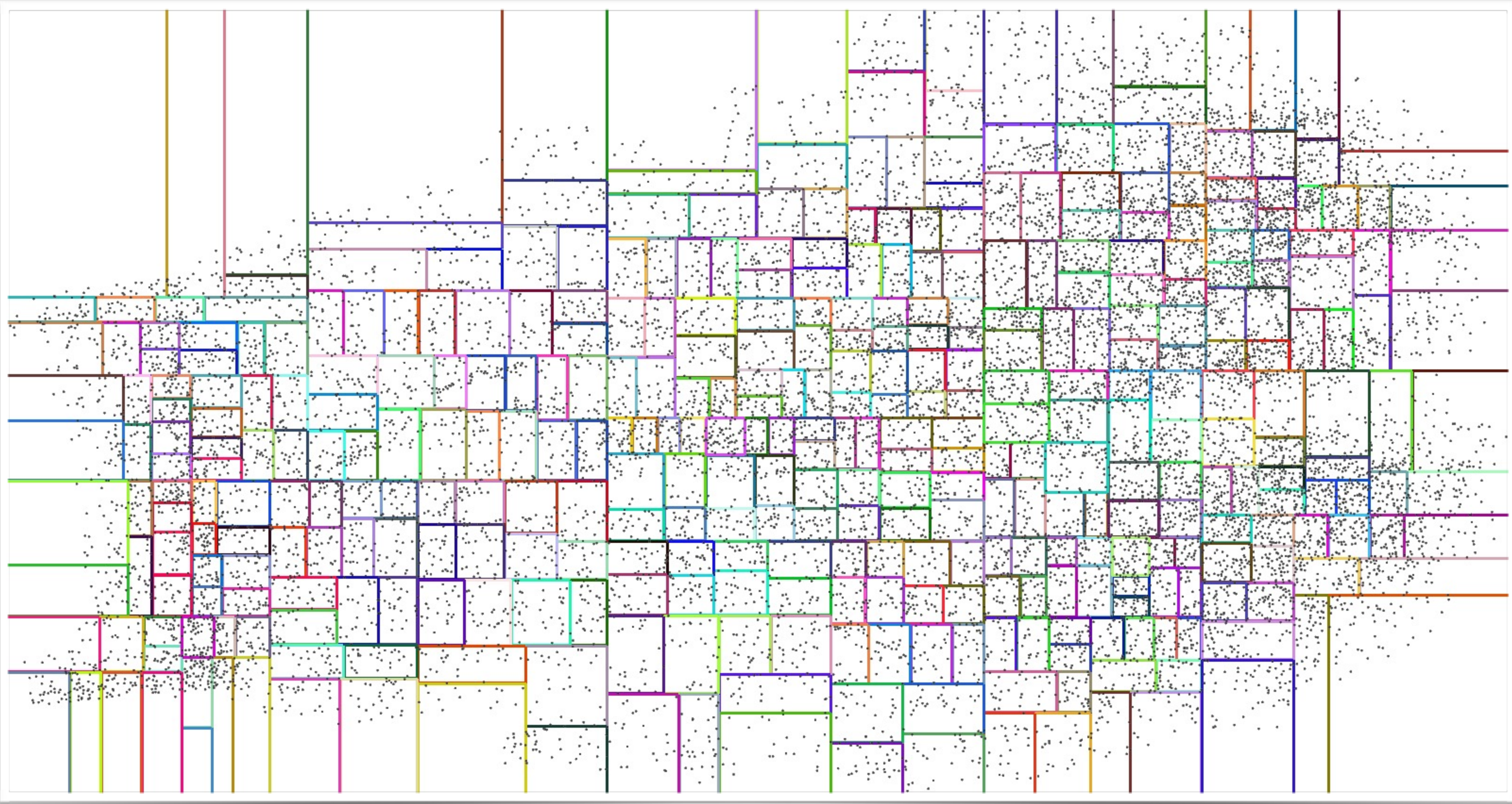
KdTree Topology

worse splitting



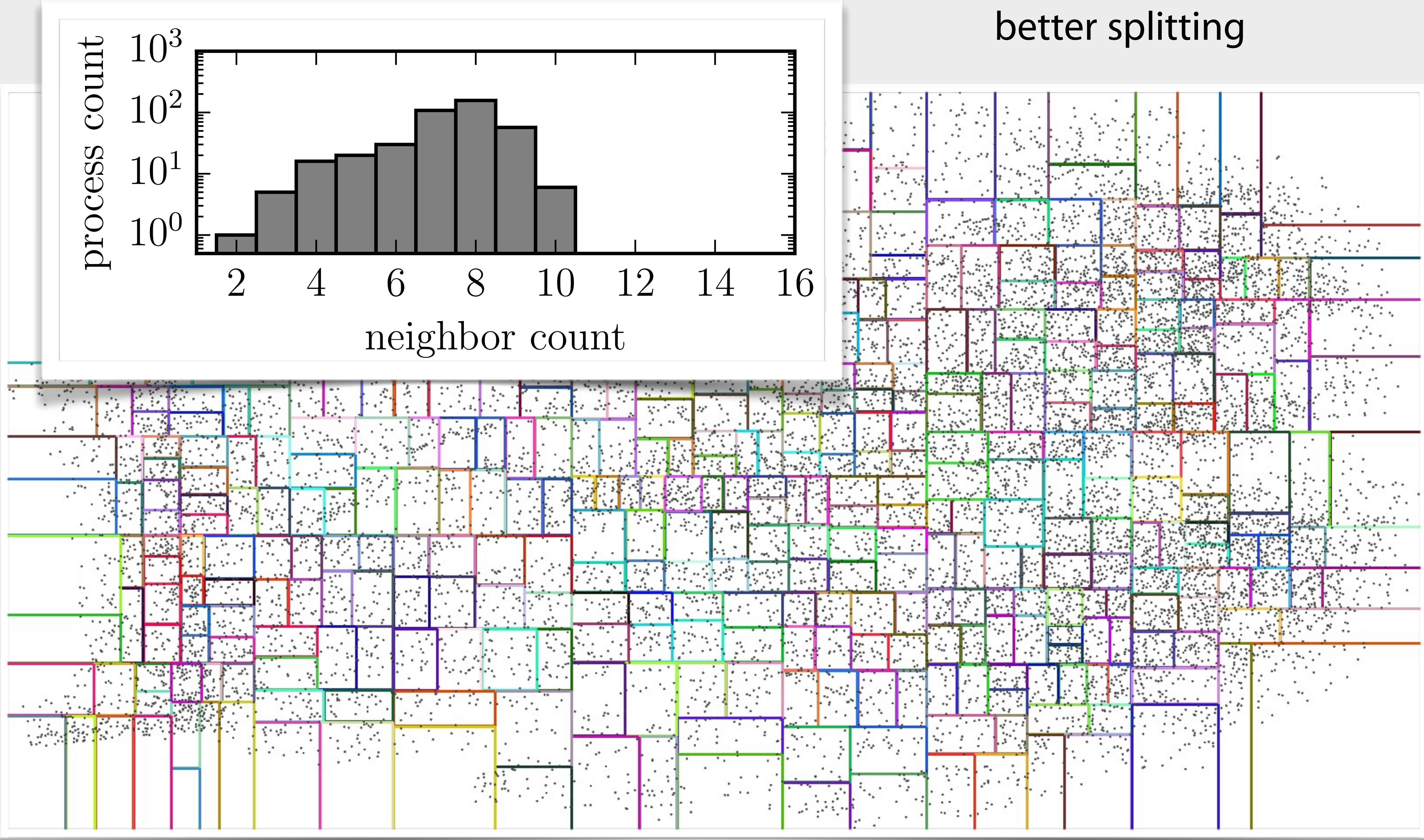
KdTree Topology

better splitting

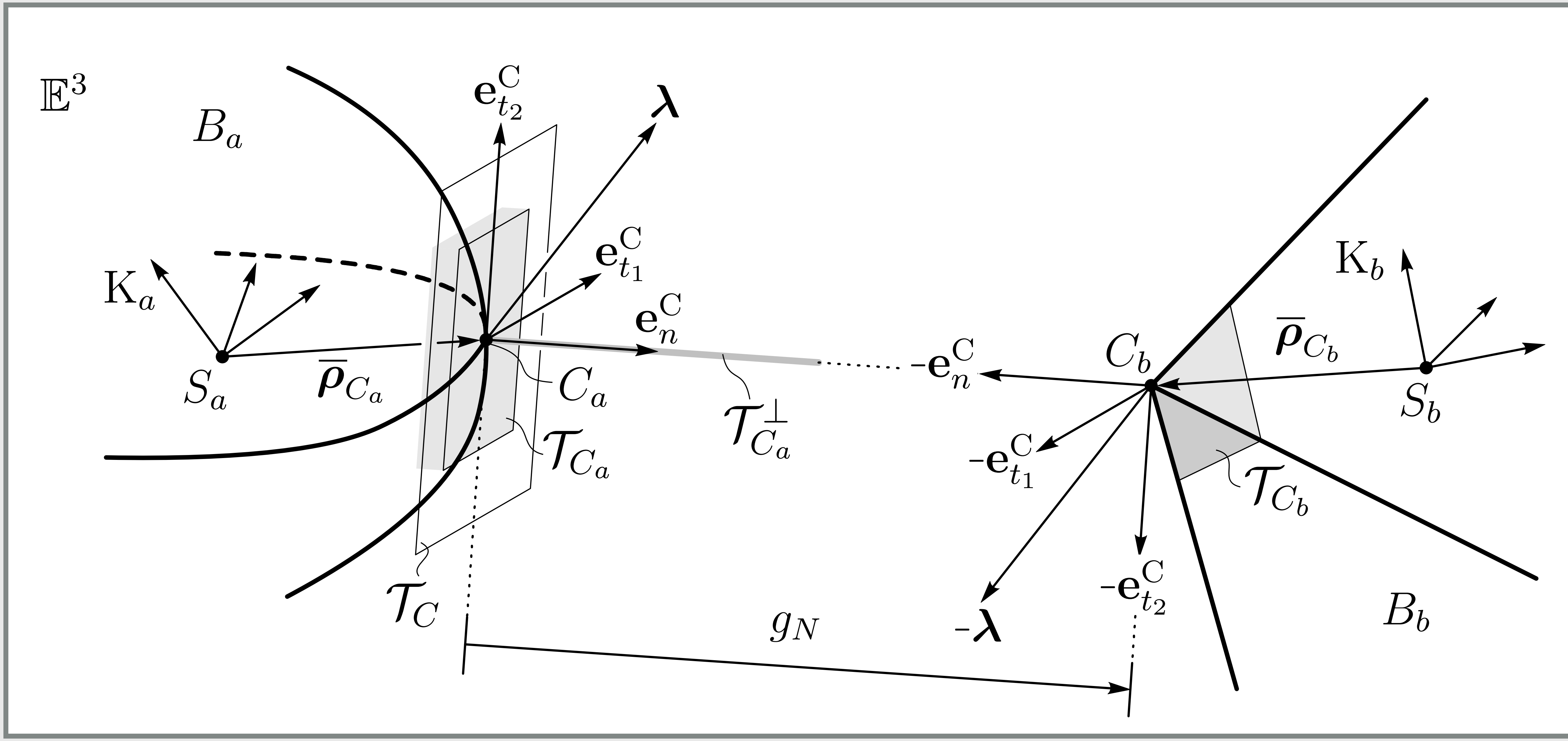


KdTree Topology

better splitting



Contact Configuration



Applying the Fundamental Axioms

Principle of Virtual Work for the Scalable Body:

$$ {}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{scal}} = s\ {}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\ {}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R $$

$$ 0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}}) = \delta W^{\mathrm{dyn}} + \delta W_I^{\mathrm{int}} + \underbrace{\delta W_C^{\mathrm{int}}}_{=0 \text{ by d'Alembert-Lagrange}} + \delta W^{\mathrm{ext}} \qquad \forall_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}},\ t $$

$$ = \int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}}^{\top}\ {}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{scal}}\mathrm{d}m - \delta s\lambda_s - {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}}^{\top}\ {}_{\mathrm{I}}\mathrm{d}\mathbf{F}^{\mathrm{e}} \qquad \forall_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}},\ t\ . $$

$$ \begin{bmatrix} {}_{\mathrm{I}}\delta\mathbf{r}_S \\ \delta s \\ \delta\mathbf{a} \end{bmatrix}^{\top} \left(\begin{bmatrix} m\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\mathrm{Tr}({}_{\mathrm{K}}\boldsymbol{\Theta}_S) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}_{\mathrm{K}}\boldsymbol{\Theta}_S \end{bmatrix} \dot{\mathbf{u}}_p + \frac{1}{s} \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\alpha}^{\top}{}_{\mathrm{K}}\boldsymbol{\Theta}_S\boldsymbol{\alpha} \\ (\nu\mathbf{I} + \tilde{\boldsymbol{\alpha}})_{\mathrm{K}}\boldsymbol{\Theta}_S\boldsymbol{\alpha} \end{bmatrix} - \begin{bmatrix} {}_{\mathrm{I}}\mathbf{F}^{\mathrm{e}} \\ E_S^{\mathrm{e}} + \lambda_s \\ {}_{\mathrm{K}}\mathbf{M}_S^{\mathrm{e}} \end{bmatrix} \right) = 0 $$

$$ \forall\ {}_{\mathrm{I}}\delta\mathbf{r}_S, \delta s, \delta\mathbf{a} $$

Applying the Fundamental Axioms

Principle of Virtual Work for the Scalable Body:

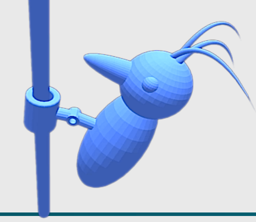
$$\boxed{{}_{\mathrm{I}}\boldsymbol{\xi}_{\mathrm{scal}} = s\;{}_{\mathrm{I}}\mathbf{R}(\mathbf{p})\;{}_{\mathrm{I}}\boldsymbol{\rho} + {}_{\mathrm{I}}\mathbf{r}_R}$$

$$0 = \delta W({}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}}) = \delta W^{\mathrm{dyn}} + \delta W_I^{\mathrm{int}} + \underbrace{\delta W_C^{\mathrm{int}}}_{=0 \text{ by d'Alembert-Lagrange}} + \delta W^{\mathrm{ext}} \qquad \forall_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}},\, t$$

$$= \int_B {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}}{}^{\top}\,{}_{\mathrm{I}}\ddot{\boldsymbol{\xi}}_{\mathrm{scal}}\mathrm{d}m - \delta s\lambda_s - {}_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}}{}^{\top}\,{}_{\mathrm{I}}\mathrm{d}\mathbf{F}^{\mathrm{e}} \qquad \forall_{\mathrm{I}}\delta\boldsymbol{\xi}_{\mathrm{scal}},\, t\;.$$

$$\begin{bmatrix} {}_{\mathrm{I}}\delta\mathbf{r}_S \\ \delta s \\ \delta\mathbf{a} \end{bmatrix}^{\top} \left(\begin{bmatrix} m\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\,\mathrm{Tr}({}_{\mathrm{K}}\boldsymbol{\Theta}_S) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}_{\mathrm{K}}\boldsymbol{\Theta}_S \end{bmatrix} \dot{\mathbf{u}}_p + \frac{1}{s} \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\alpha}^{\top}{}_{\mathrm{K}}\boldsymbol{\Theta}_S\boldsymbol{\alpha} \\ (\nu\mathbf{I} + \tilde{\boldsymbol{\alpha}})_{\mathrm{K}}\boldsymbol{\Theta}_S\boldsymbol{\alpha} \end{bmatrix} - \begin{bmatrix} {}_{\mathrm{I}}\mathbf{F}^{\mathrm{e}} \\ E_S^{\mathrm{e}} + \lambda_s \\ {}_{\mathrm{K}}\mathbf{M}_S^{\mathrm{e}} \end{bmatrix} \right) = 0$$

$$\forall\; {}_{\mathrm{I}}\delta\mathbf{r}_S, \delta s, \delta\mathbf{a}$$



Applying the Fundamental Axioms

Principle of Virtual Work for the Scalable Body:

scalable body motion: $$\boxed{{}_I\boldsymbol{\xi}_{\mathrm{scal}} = s\ {}_I\mathbf{R}(\mathbf{p})\ {}_I\boldsymbol{\rho} + {}_I\mathbf{r}_R}$$

$$0 = \delta W({}_I\delta\boldsymbol{\xi}_{\mathrm{scal}}) = \delta W^{\mathrm{dyn}} + \delta W_I^{\mathrm{int}} + \underbrace{\delta W_C^{\mathrm{int}}}_{=0 \text{ by d'Alembert-Lagrange}} + \delta W^{\mathrm{ext}} \qquad \forall_I\delta\boldsymbol{\xi}_{\mathrm{scal}},\ t$$

$$= \int_B {}_I\delta\boldsymbol{\xi}_{\mathrm{scal}}^{\top}\ {}_I\ddot{\boldsymbol{\xi}}_{\mathrm{scal}}\mathrm{d}m - \delta s\lambda_s - {}_I\delta\boldsymbol{\xi}_{\mathrm{scal}}^{\top}\ {}_I\mathrm{d}\mathbf{F}^{\mathrm{e}} \qquad \forall_I\delta\boldsymbol{\xi}_{\mathrm{scal}},\ t\ .$$

$$\begin{bmatrix} {}_I\delta\mathbf{r}_S \\ \delta s \\ \delta\mathbf{a} \end{bmatrix}^{\top} \left(\begin{bmatrix} m\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\mathrm{Tr}({}_K\boldsymbol{\Theta}_S) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & {}_K\boldsymbol{\Theta}_S \end{bmatrix} \dot{\mathbf{u}}_p + \frac{1}{s} \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{\alpha}^{\top}\,{}_K\boldsymbol{\Theta}_S\boldsymbol{\alpha} \\ (\nu\mathbf{I} + \tilde{\boldsymbol{\alpha}})\,{}_K\boldsymbol{\Theta}_S\boldsymbol{\alpha} \end{bmatrix} - \begin{bmatrix} {}_I\mathbf{F}^{\mathrm{e}} \\ E_S^{\mathrm{e}} + \lambda_s \\ {}_K\mathbf{M}_S^{\mathrm{e}} \end{bmatrix} \right) = 0$$

$$\forall\ {}_I\delta\mathbf{r}_S, \delta s, \delta\mathbf{a}$$

Fundamental Axioms in Mechanics

Principle of Virtual Work:

At any instant of time t, the virtual work δW of a body $B \in \mathbb{E}^3$ vanishes for all virtual displacements $\delta\boldsymbol{\xi}$, that is,

$$\delta W(\delta\boldsymbol{\xi}) = \int_B \langle \mathrm{d}\mathbf{F}\,, \delta\boldsymbol{\xi}(\mathbf{x})\rangle = \delta W^{\mathrm{dyn}}(\delta\boldsymbol{\xi}) + \delta W^{\mathrm{int}}(\delta\boldsymbol{\xi}) + \delta W^{\mathrm{ext}}(\delta\boldsymbol{\xi}) = 0 \quad \forall \delta\boldsymbol{\xi}\ .$$

Fundamental Axioms in Mechanics

Principle of Virtual Work:

At any instant of time t, the virtual work δW of a body $B \in \mathbb{E}^3$ vanishes for all virtual displacements $\delta\boldsymbol{\xi}$, that is,

$$\delta W(\delta\boldsymbol{\xi}) = \int_B \langle \mathrm{d}\mathbf{F}\,, \delta\boldsymbol{\xi}(\mathbf{x})\rangle = \boxed{\delta W^{\mathrm{dyn}}(\delta\boldsymbol{\xi})} + \boxed{\delta W^{\mathrm{int}}(\delta\boldsymbol{\xi})} + \delta \boxed{W^{\mathrm{ext}}(\delta\boldsymbol{\xi})} = 0 \quad \forall \delta\boldsymbol{\xi}\ .$$

Inertia Contribution:

$$\delta W^{\mathrm{dyn}}(\delta\boldsymbol{\xi}) := -\int_B \delta\boldsymbol{\xi}(\mathbf{x})^\top \boxed{\ddot{\boldsymbol{\xi}}(\mathbf{x})\mathrm{d}m(\mathbf{x})}$$

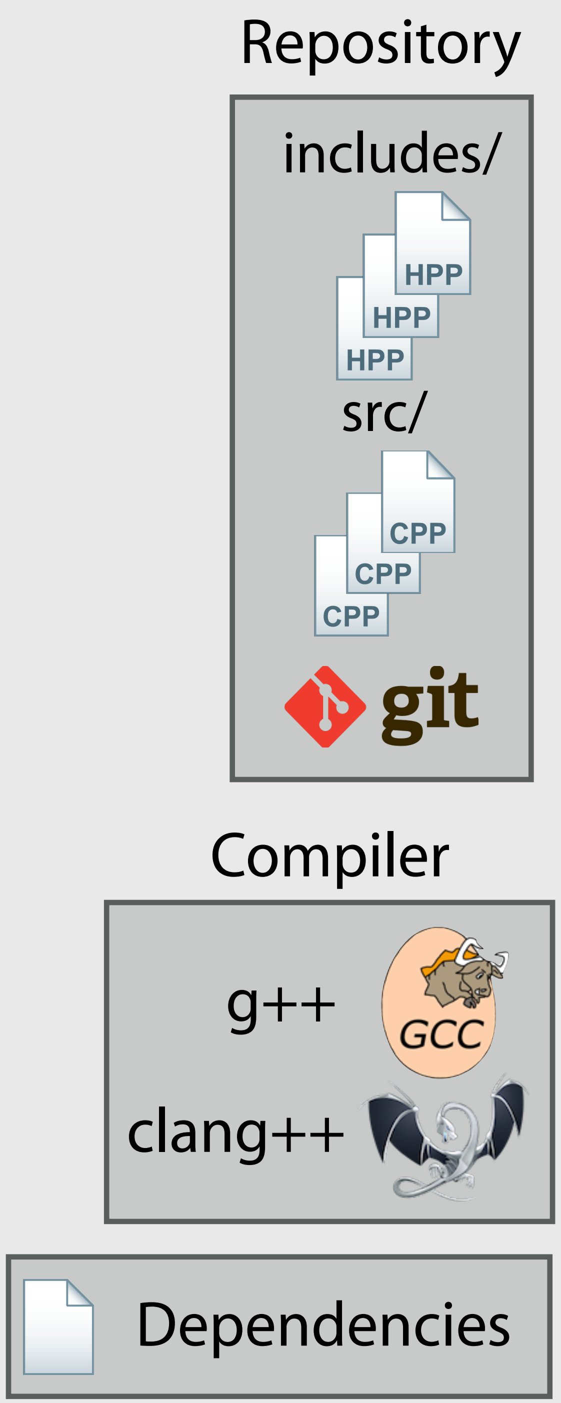
Constitutive force law:
Newton's second law

Law of Interaction as Requirement on the Internal Forces: (S. Eugster 2014)

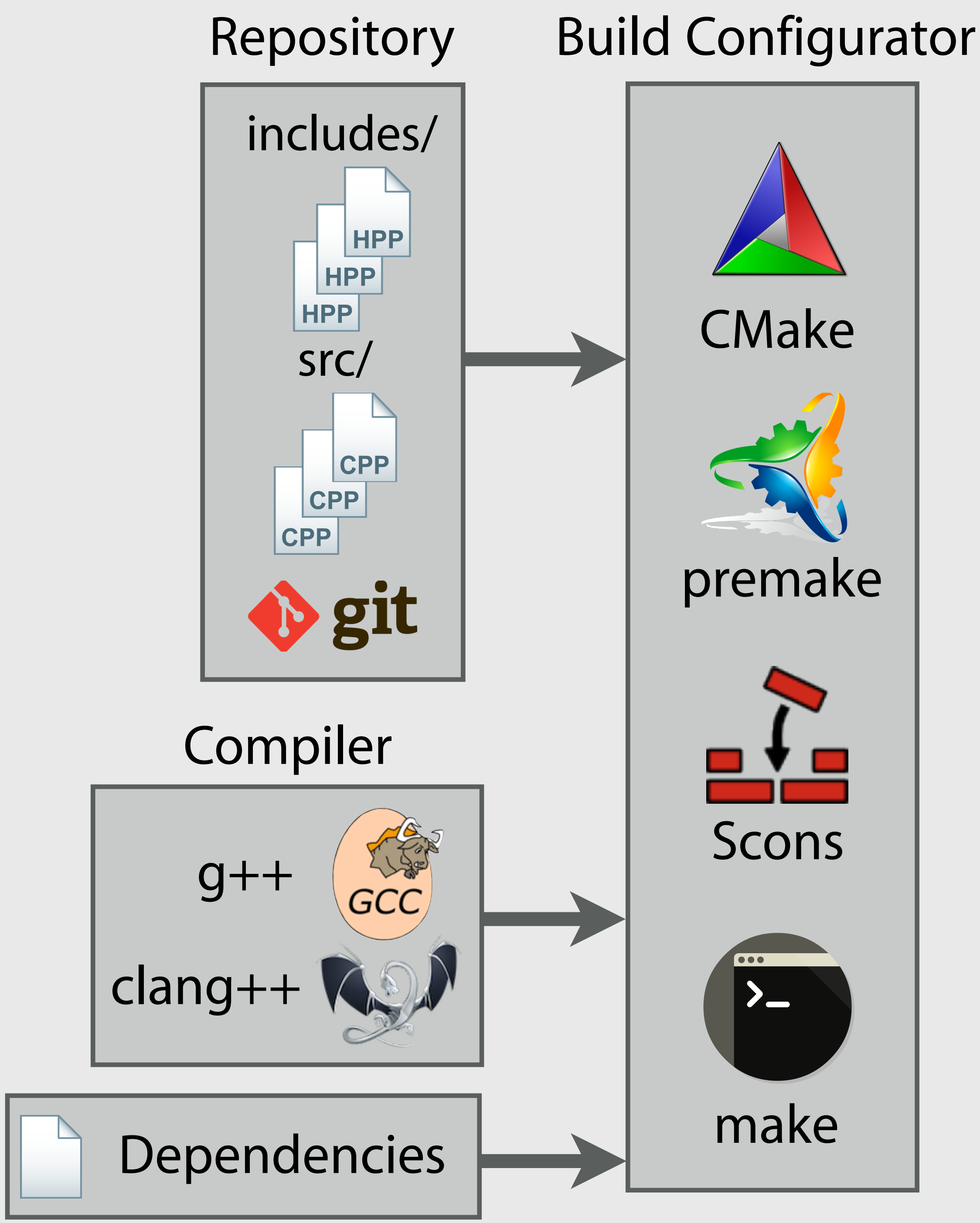
At any instant of time t, the *internal* virtual work $\delta W^{\mathrm{int}}_{B'}$ of any subsystem $B' \subseteq B$ vanishes for all *rigid* virtual displacements $\delta\boldsymbol{\xi}$, that is,

$$\delta W^{\mathrm{int}}_{B'}(\delta\boldsymbol{\xi}) = 0 \quad \forall\ \delta\boldsymbol{\xi} \text{ rigidifying}, \ \ B' \subseteq B\ .$$

Modern Software Development in C++

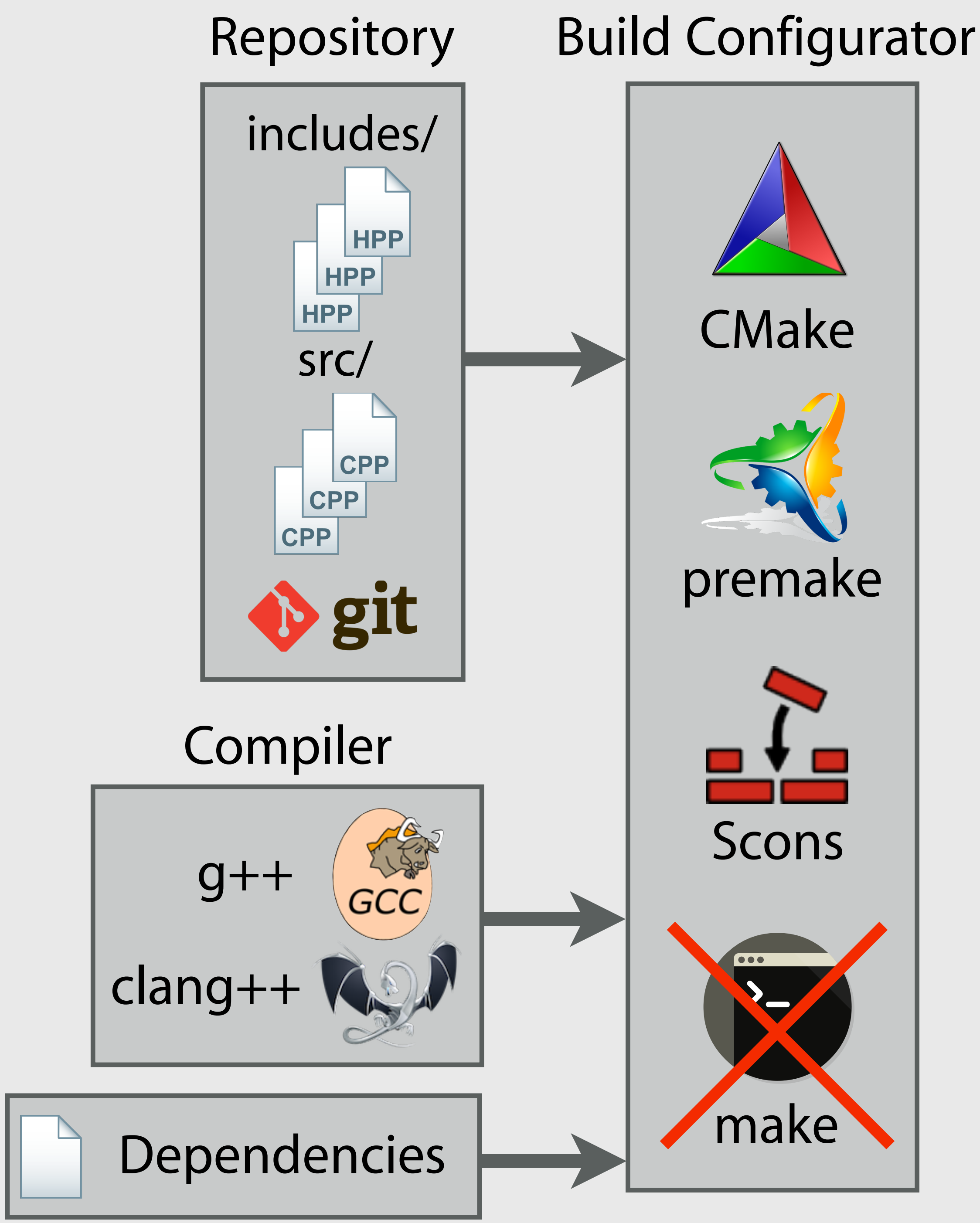


Modern Software Development in C++



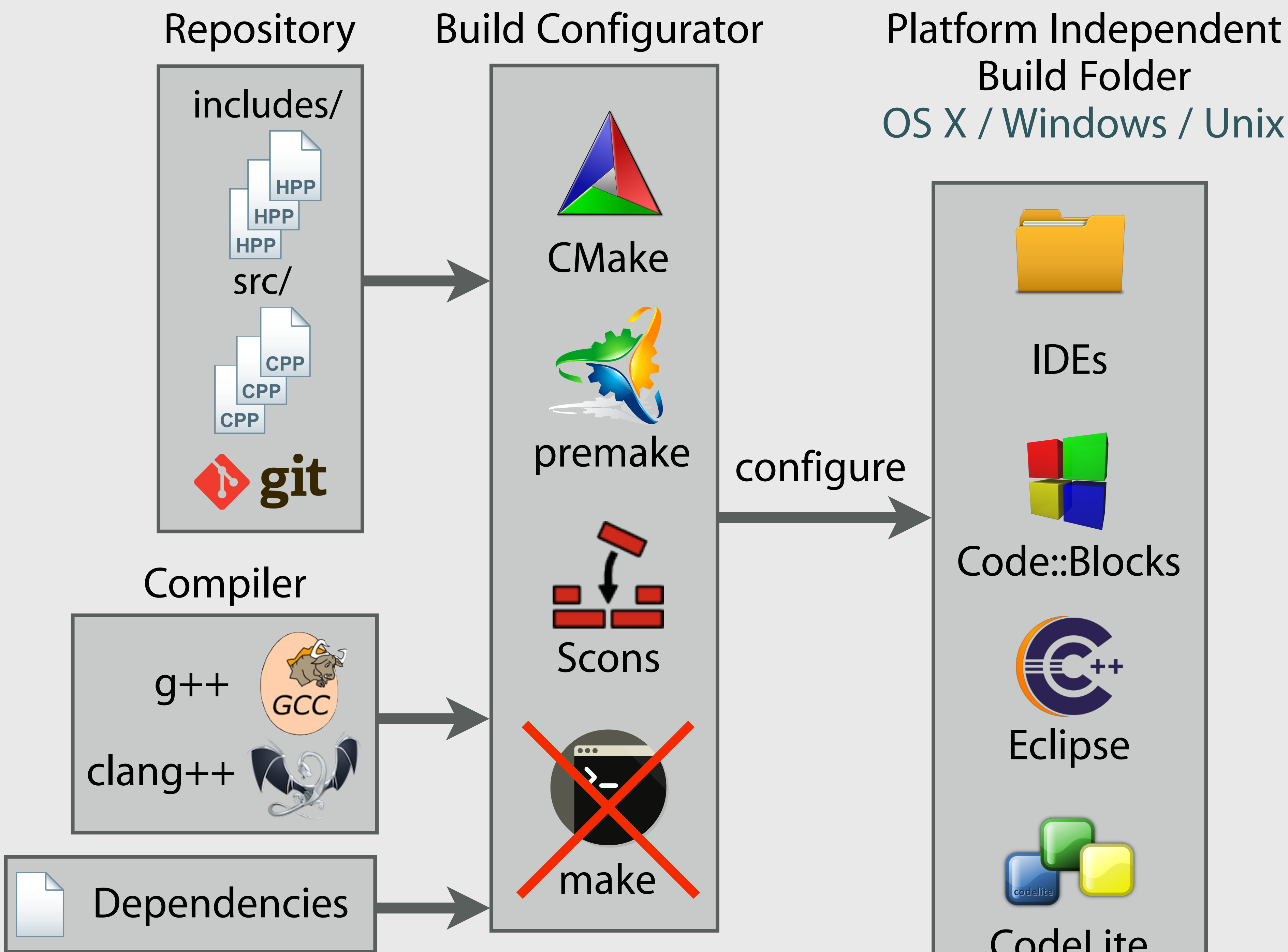
- Find Dependencies
- Setup Build Settings

Modern Software Development in C++



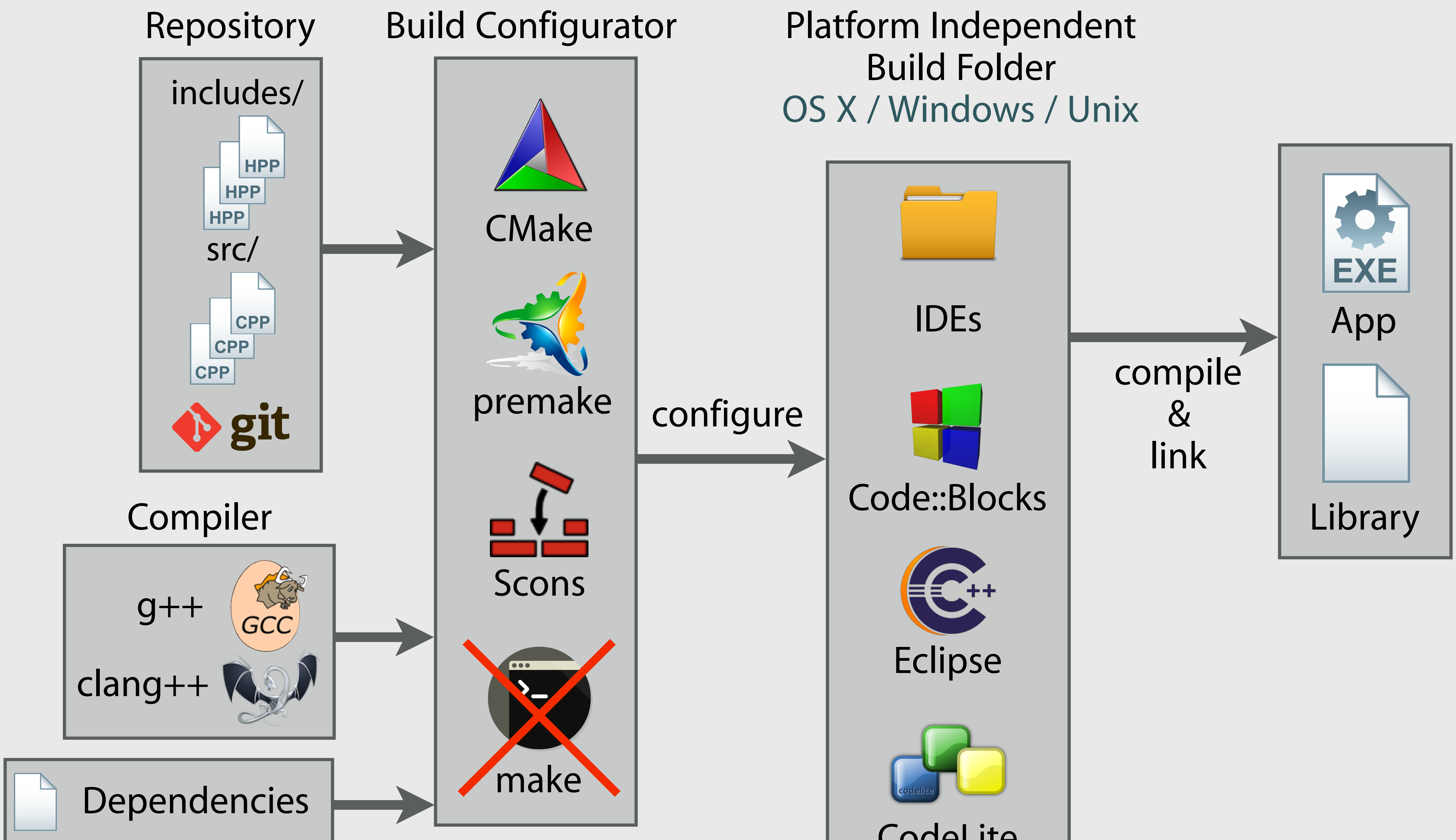
- Find Dependencies
- Setup Build Settings

Modern Software Development in C++



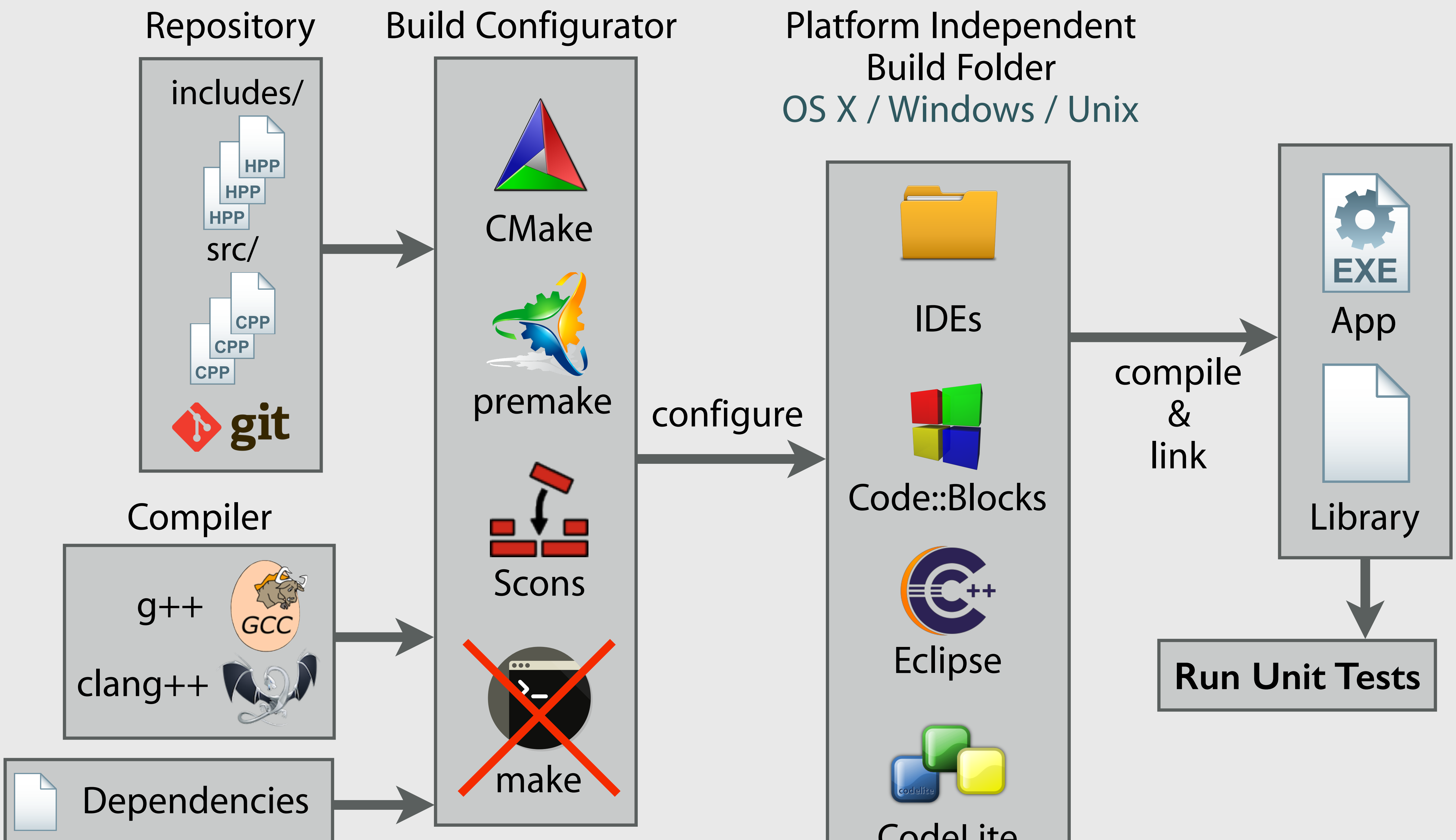
- Find Dependencies
- Setup Build Settings

Modern Software Development in C++



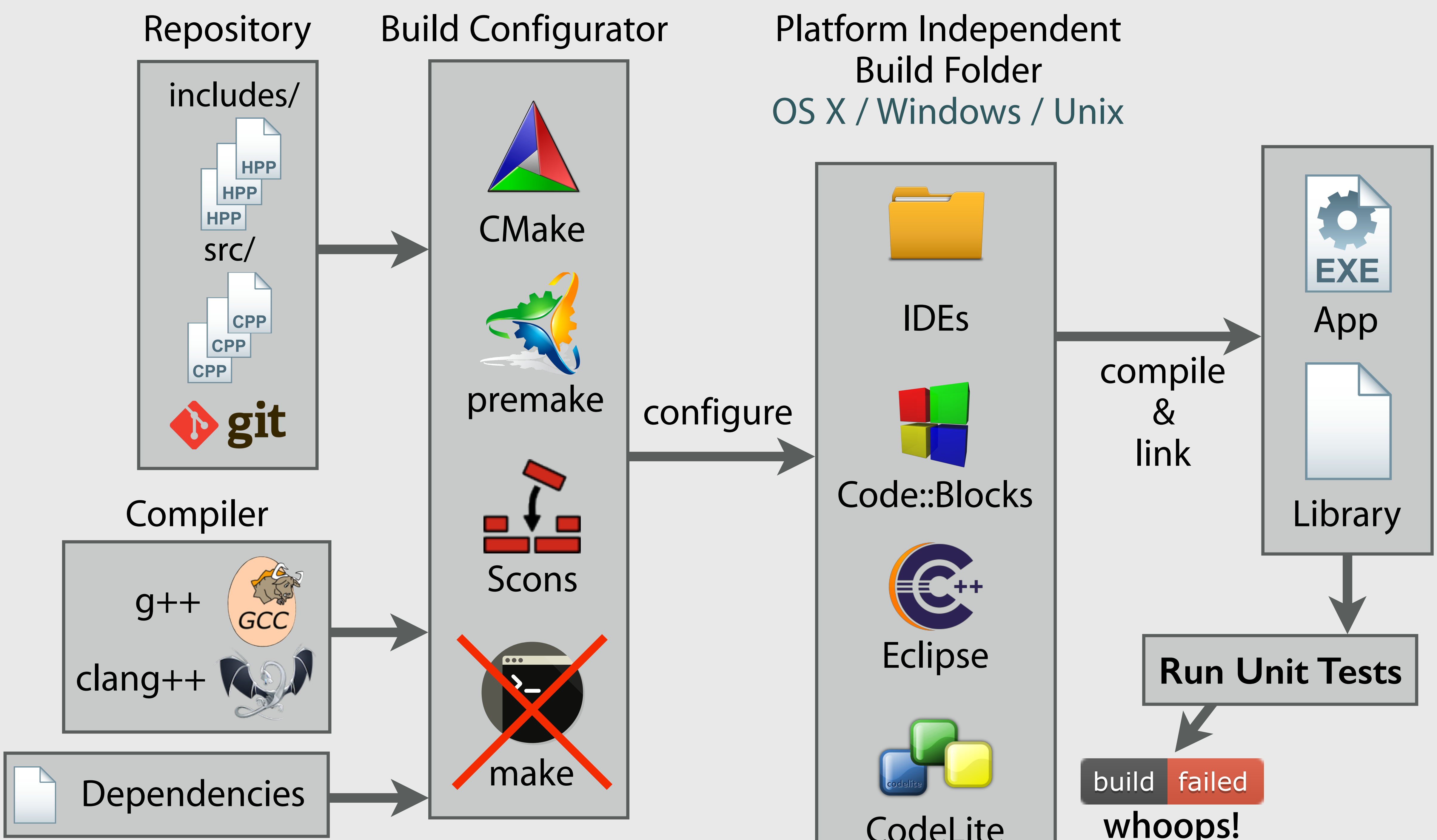
- Find Dependencies
- Setup Build Settings

Modern Software Development in C++



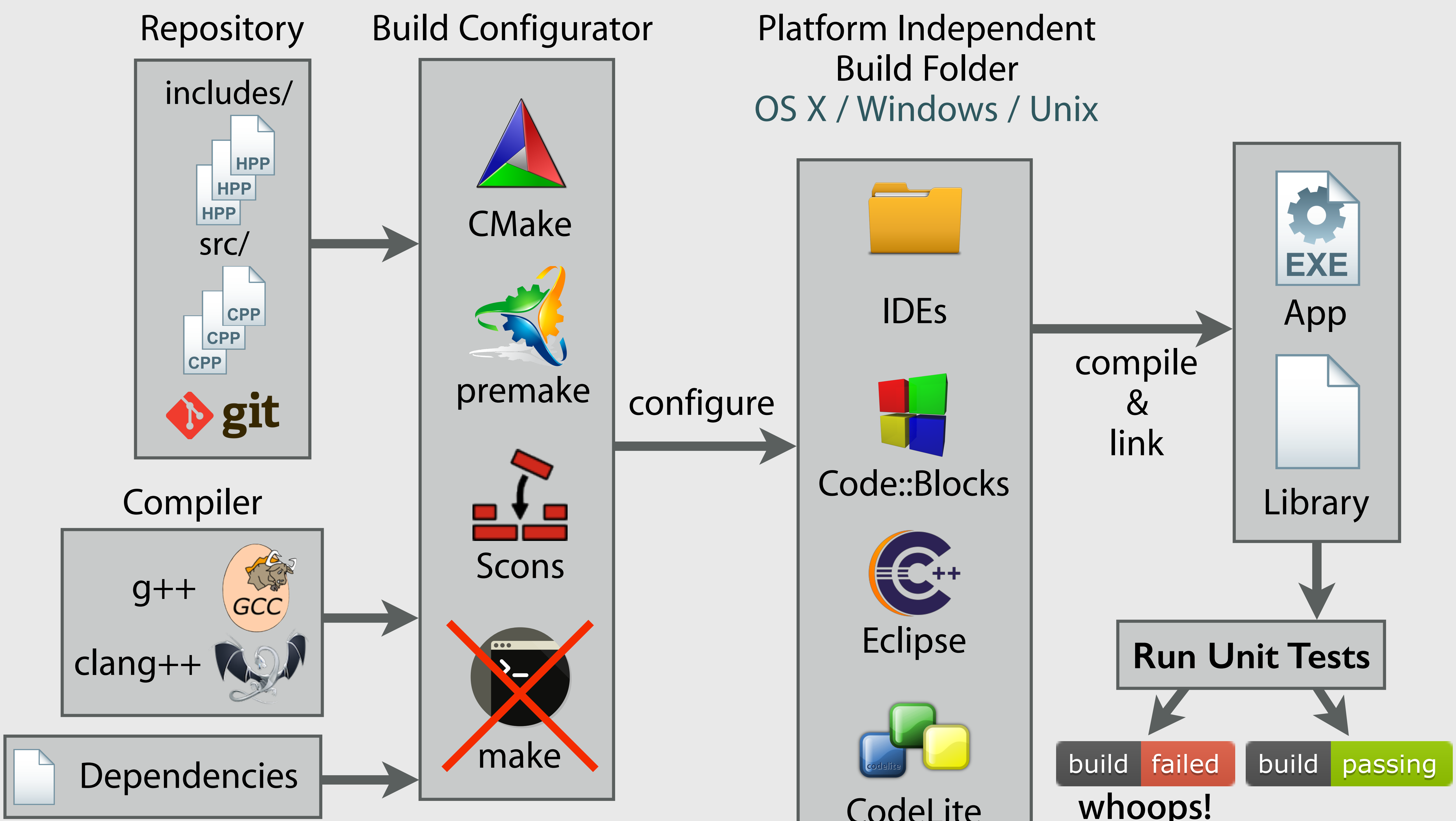
- Find Dependencies
- Setup Build Settings

Modern Software Development in C++



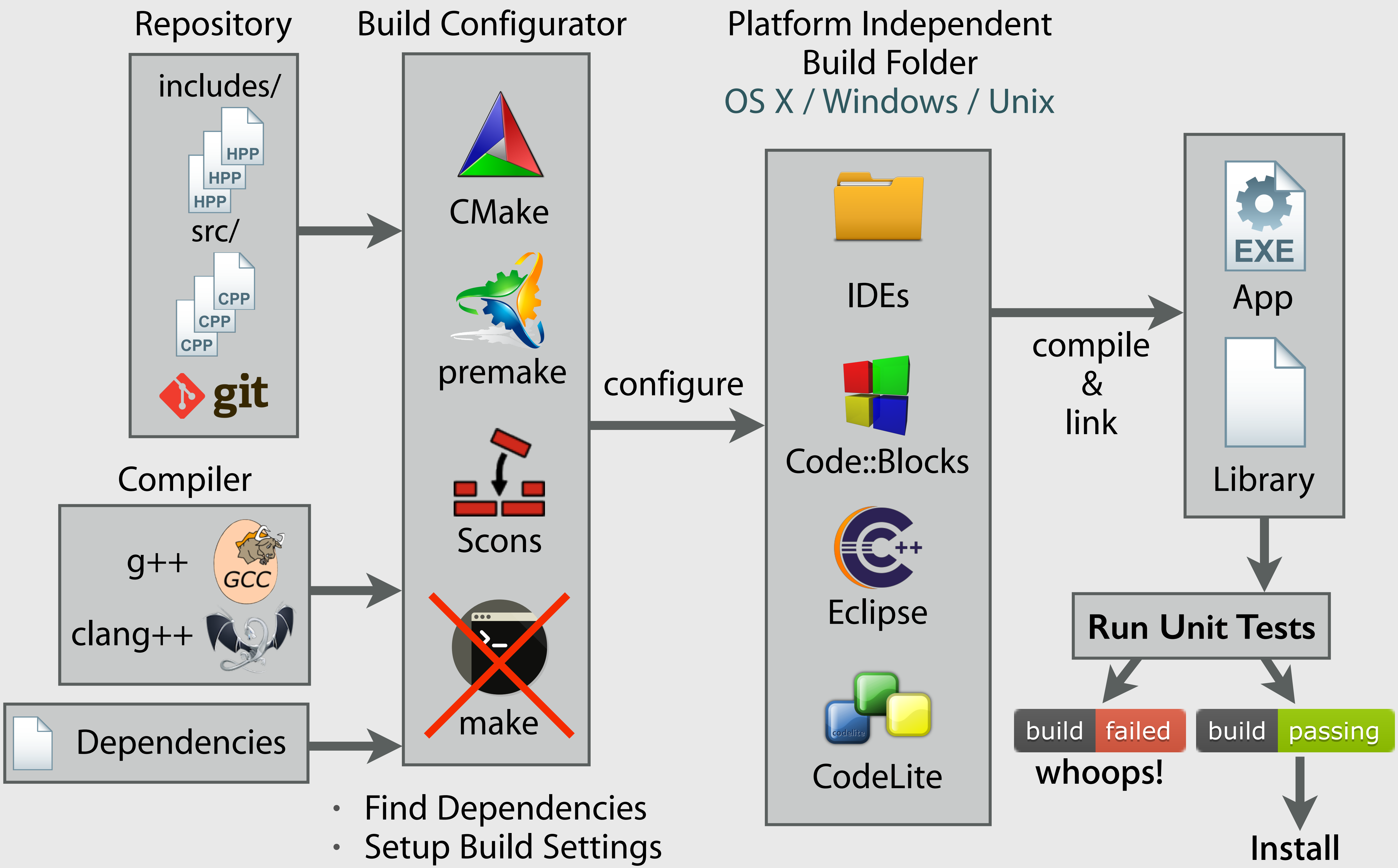
- Find Dependencies
- Setup Build Settings

Modern Software Development in C++

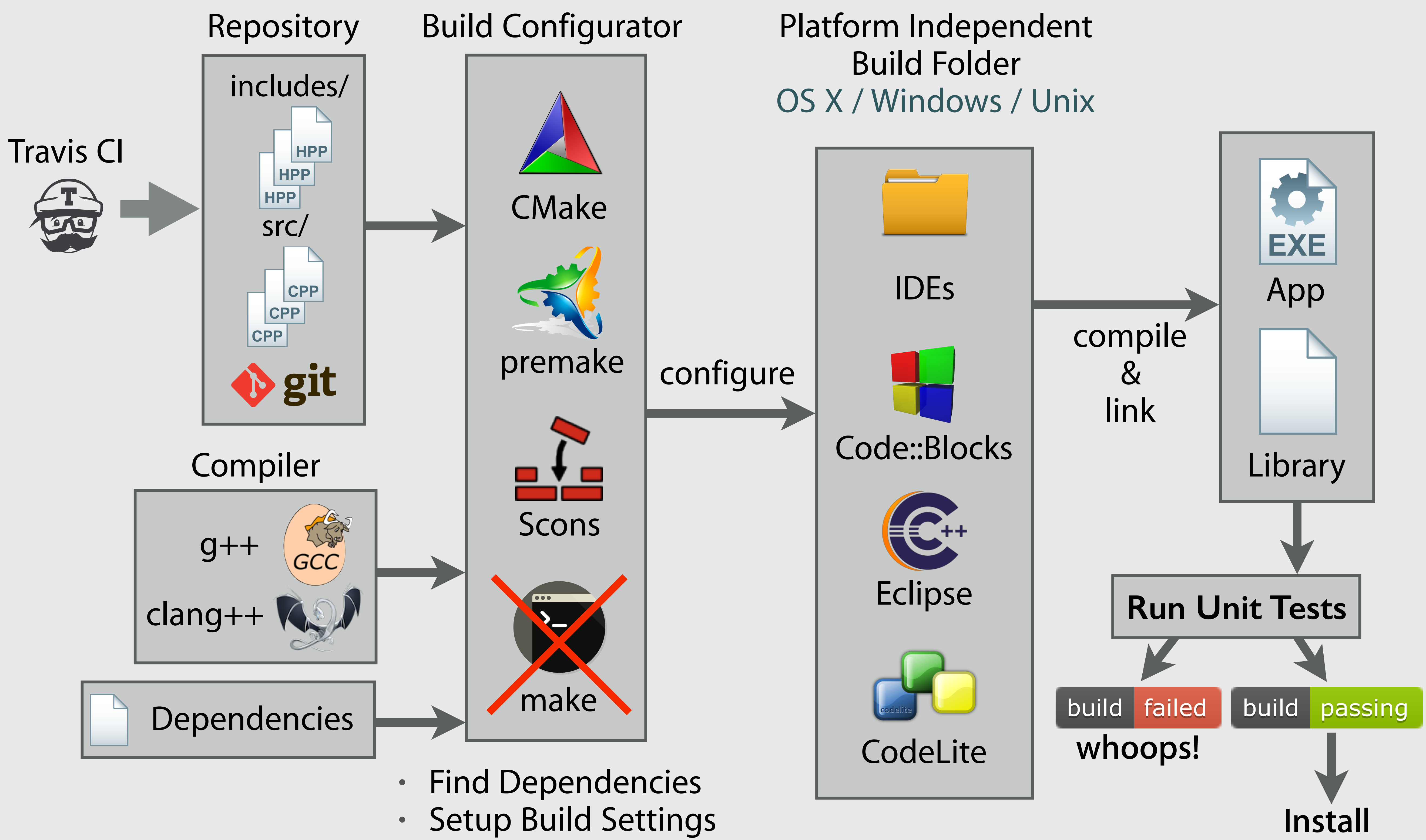


- Find Dependencies
- Setup Build Settings

Modern Software Development in C++



Modern Software Development in C++



- Find Dependencies
- Setup Build Settings