UNIVERSITÉ
DE GENÈVE

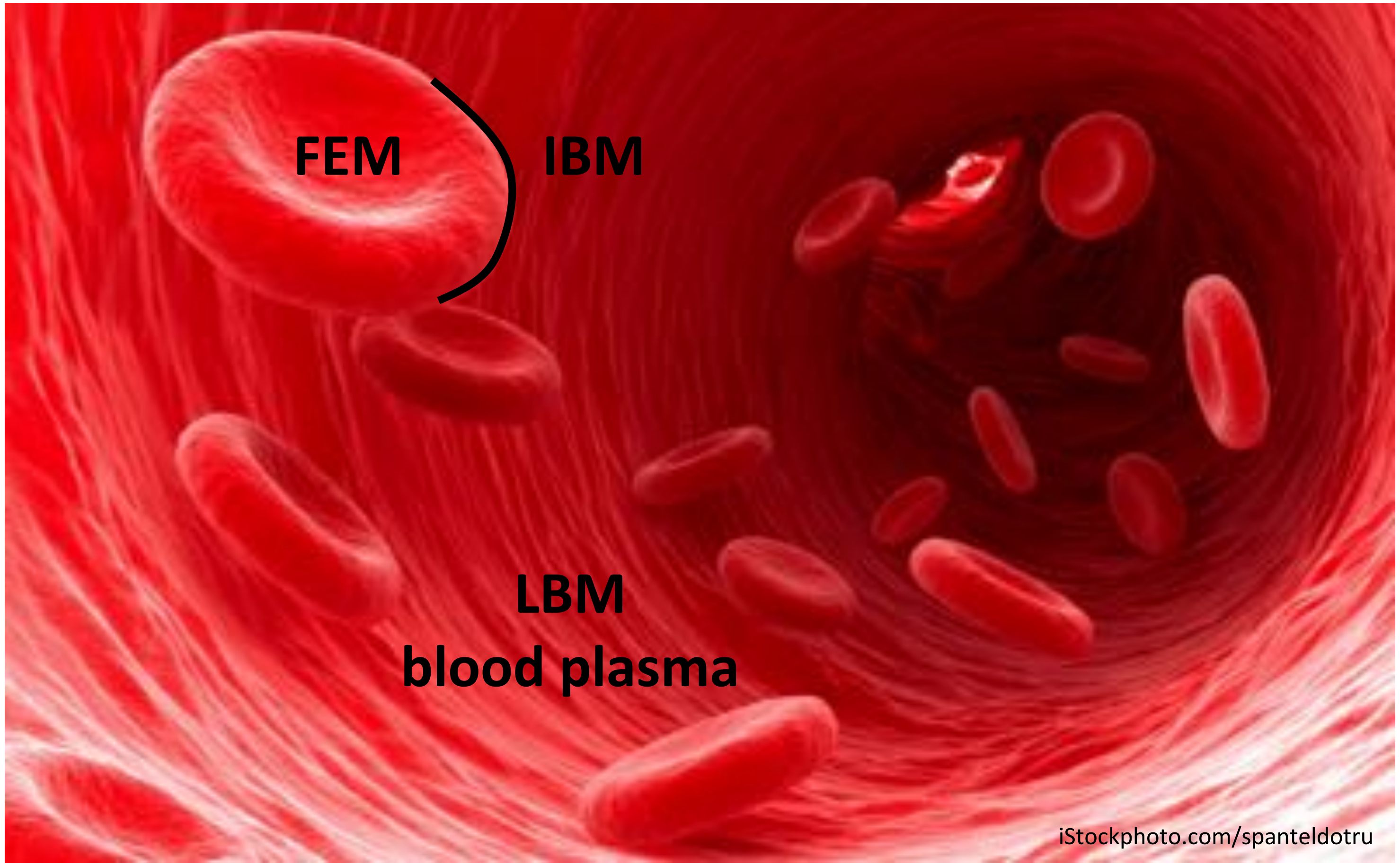
Digital Blood in Massively Parallel CPU/GPU Systems for the Study of Platelets deposition

Kotsalos Christos

Supervisors: Prof. B. Chopard
Dr. J. Latt

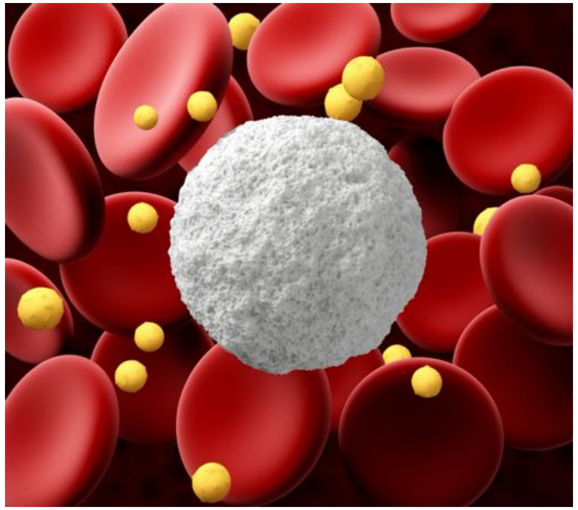
Scientific and Parallel Computing Group (SPC)

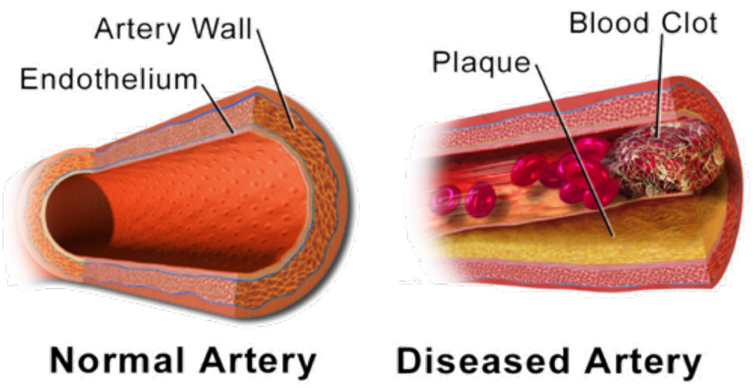
High fidelity and fast simulations of **deformable blood cells** using a combined **Finite Element Immersed Boundary Lattice Boltzmann** method (FE-IB-LBM)



- FEM for the blood cells
- LBM for the blood plasma
- IBM for the Fluid-Solid Interaction

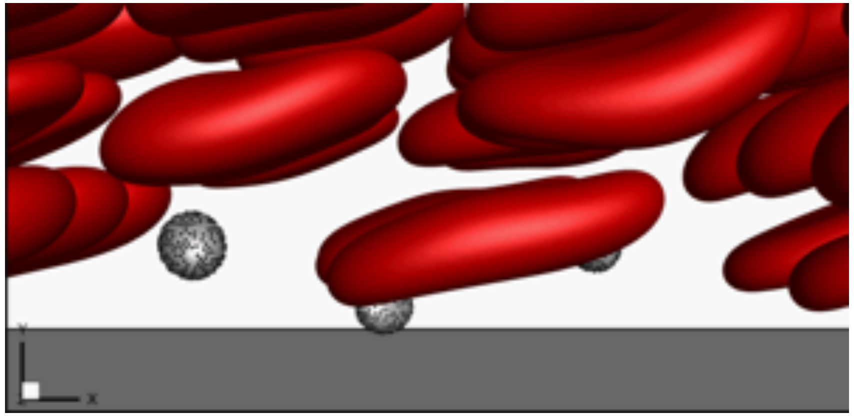
CPU/ GPU images from flaticon.com



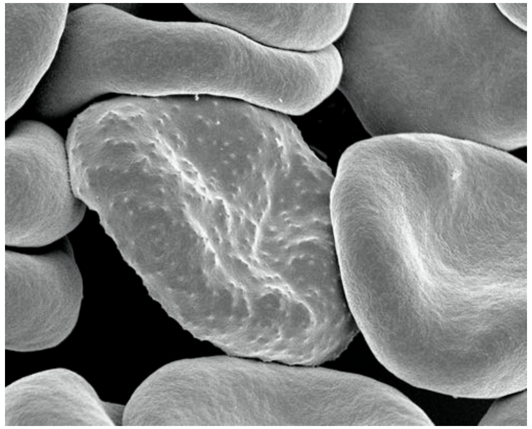


Thrombus formation

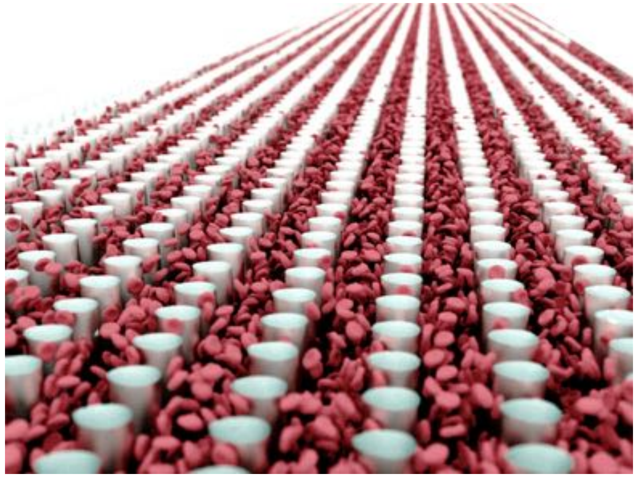
Why to simulate at this level?



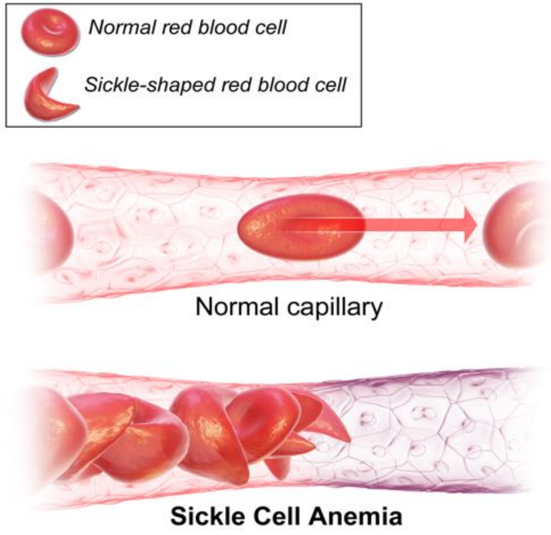
Prof. Bagchi Group | **Drug delivery**



Malaria



Prof. Koumoutsakos Group | **Lab-on-a-chip Cancer Cell Detection**



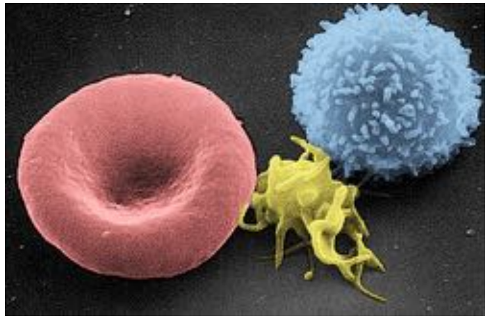
Blood in numbers

The average adult has a blood volume of roughly 5 liters = 5×10^6 mm^3

Huge amount of deformable bodies: **fast simulations are one-way**

1 mm^3 of blood contains:

- $O(10^6)$ red blood cells
- $O(10^5)$ platelets
- $O(10^3)$ white blood cells



Wikipedia: Blood

Plasma is the liquid component of blood occupying almost 55% of the whole blood. It is a Newtonian fluid **just like water**.

40-45% of the whole blood volume is occupied by red blood cells (RBCs). This volume fraction is called **hematocrit.**

Numerical methods:

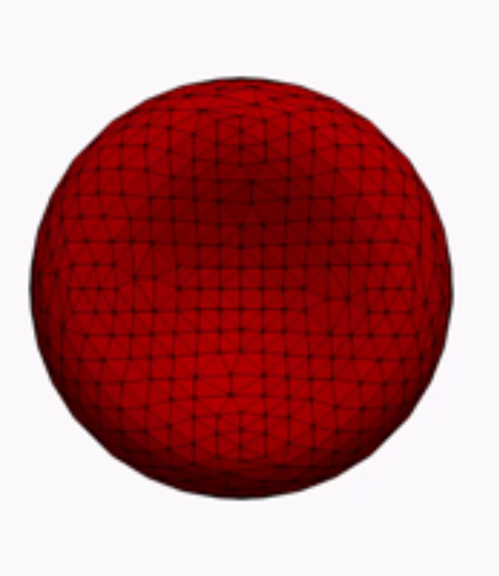
Mass-lumped **FEM (npFEM)** for the deformable bodies (developed in **C++/ CUDA**)

Lattice Boltzmann method (LBM) for the blood plasma (Palabos in **C++/ MPI**)

Immersed boundary method (IBM) for the coupling (Palabos in **C++/ MPI**)

www.palabos.org : Parallel Lattice Boltzmann Solver open-source CFD solver
(by SPC Lab, UniGe)

Mass-lumped FEM (**nodal projective FEM**)



Implicit Euler time integration:
update rule

$$\boldsymbol{F}_{int}(\boldsymbol{x}) = -\sum_i \nabla E_i(\mathbf{x}),$$

$$\boldsymbol{v}_{n+1} = \frac{\boldsymbol{x}_{n+1} - \boldsymbol{x}_n}{h},$$

$$\boldsymbol{F}_{int}(\boldsymbol{x}_{n+1}) + \boldsymbol{F}_{ext} - \mathbf{C}\boldsymbol{v}_{n+1} = \mathbf{M}\frac{\boldsymbol{v}_{n+1} - \boldsymbol{v}_n}{h},$$

$$C = \alpha_D \mathbf{M} + \beta_D \widetilde{\mathbf{H}}$$

Newton's 2nd law per vertex

Variational Implicit Euler formulation

$$\min_{\boldsymbol{x}_{n+1}} \frac{1}{2h^2} \left\| \widetilde{\mathbf{M}}^{\frac{1}{2}} (\boldsymbol{x}_{n+1} - \boldsymbol{y}_n) \right\|_F^2 + \sum_i E_i(\boldsymbol{x}_{n+1})$$

Mass-lumped FEM (**npFEM**)

$$\min_{\boldsymbol{x}_{n+1}} \; g(\boldsymbol{x}_{n+1}) = \frac{1}{2h^2} \left\| \widetilde{\mathbf{M}}^{\frac{1}{2}} (\boldsymbol{x}_{n+1} - \boldsymbol{y}_n) \right\|_F^2 + \sum_i E_i(\boldsymbol{x}_{n+1})$$

4 different kinds to describe a blood cell:

- Area Conservation
- Global Volume Conservation
- Bending rigidity
- Material model (modified Skalak)

Discretized by piecewise linear basis functions

Quasi-Newton: $\boldsymbol{x}_{n+1}^{k+1} = \boldsymbol{x}_{n+1}^{k} - \alpha \widetilde{\mathbf{H}}^{-1} \nabla g(\boldsymbol{x}_{n+1}^{k})$

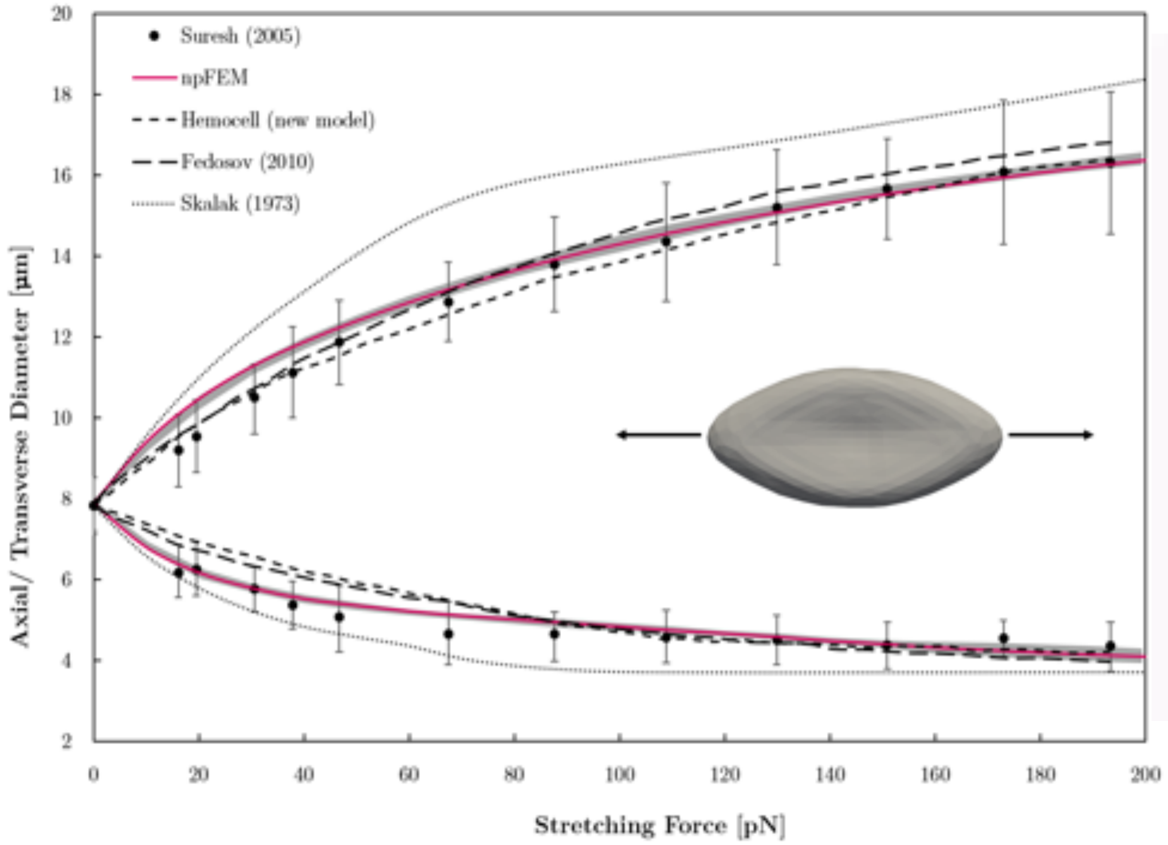
For more details:

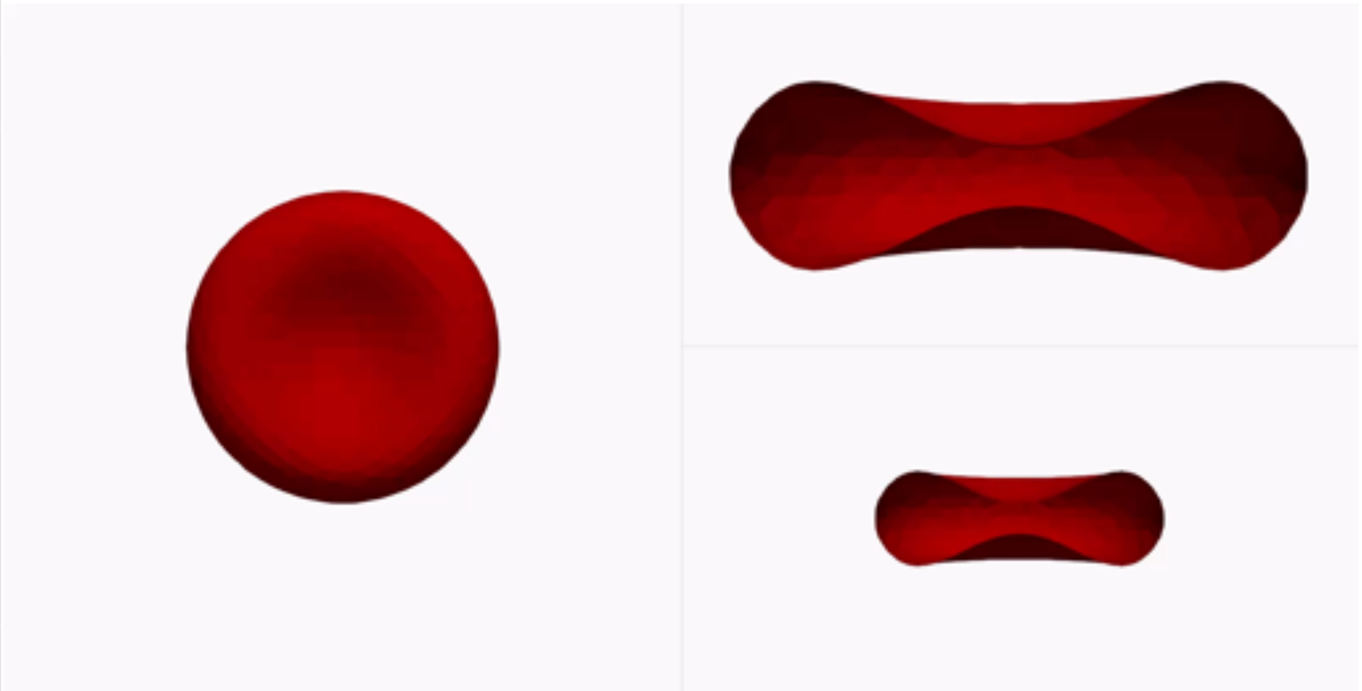
Bridging the computational gap between mesoscopic and continuum modeling of red blood cells for fully resolved blood flow

Journal of Computational Physics 2019

https://doi.org/10.1016/j.jcp.2019.108905

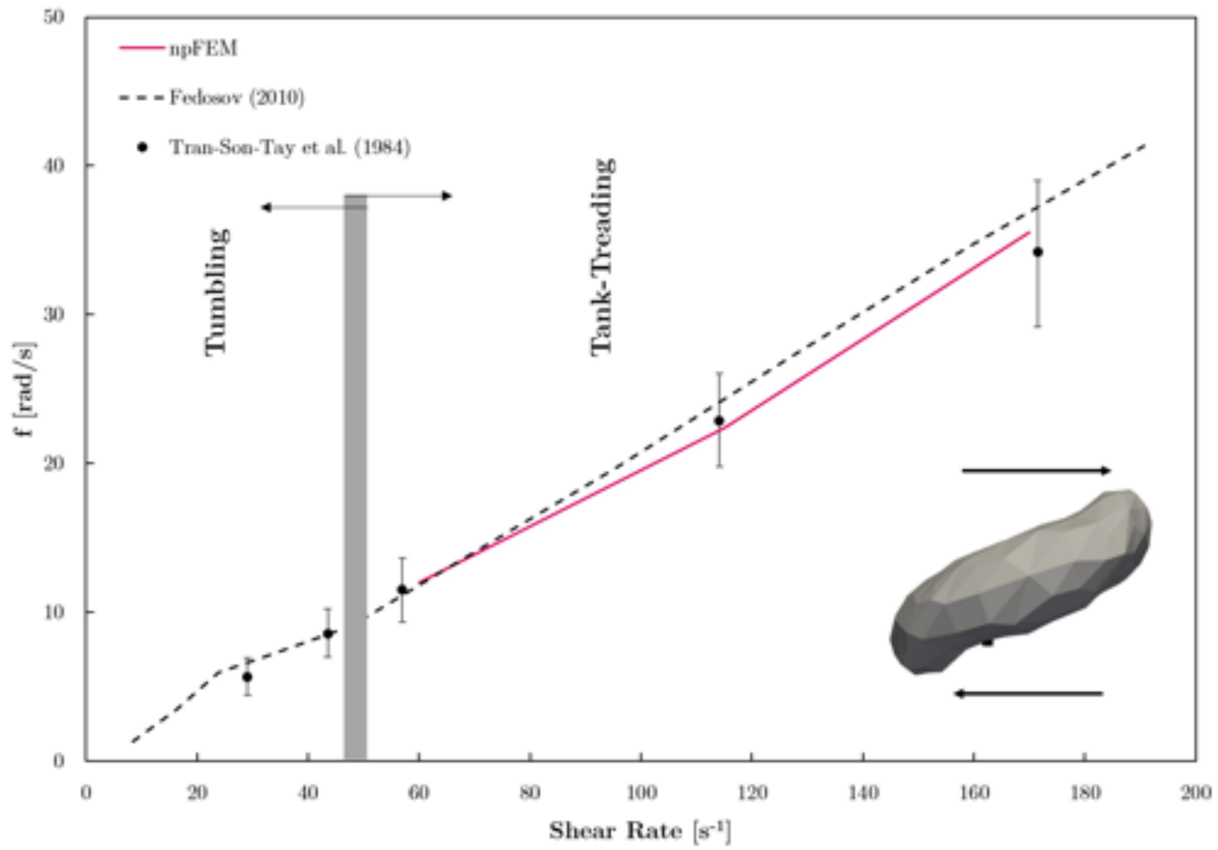
Stretching experiment

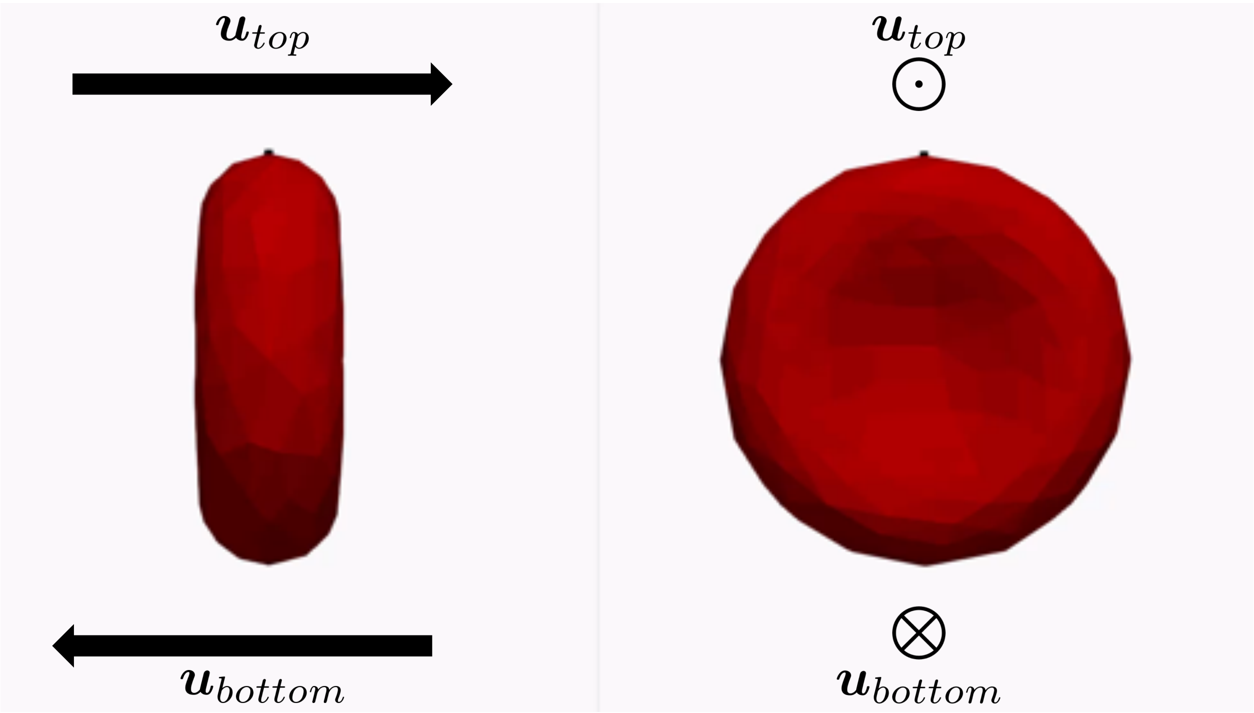




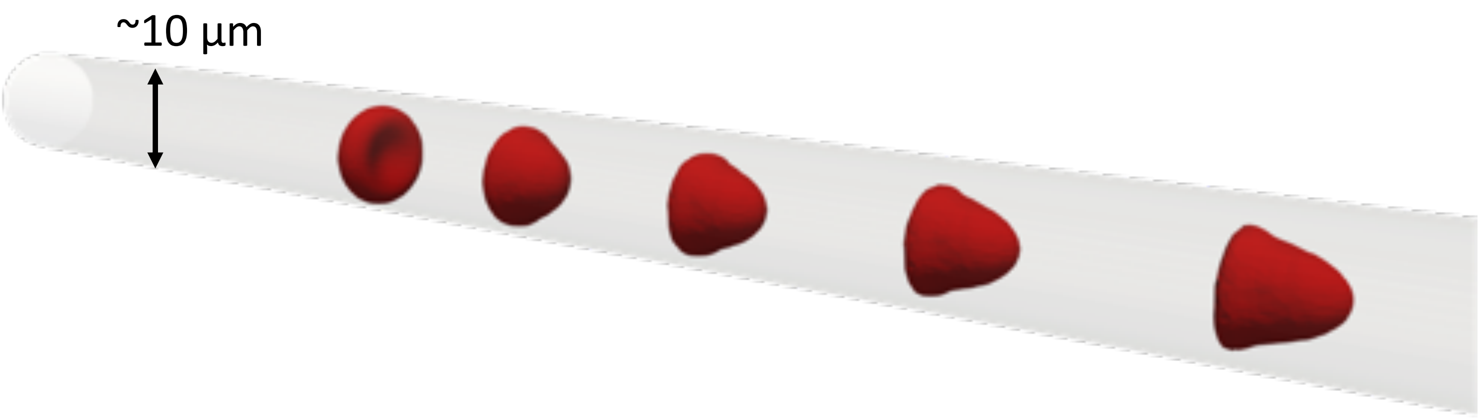
Optical tweezers experiment
by Dao, Li & Suresh

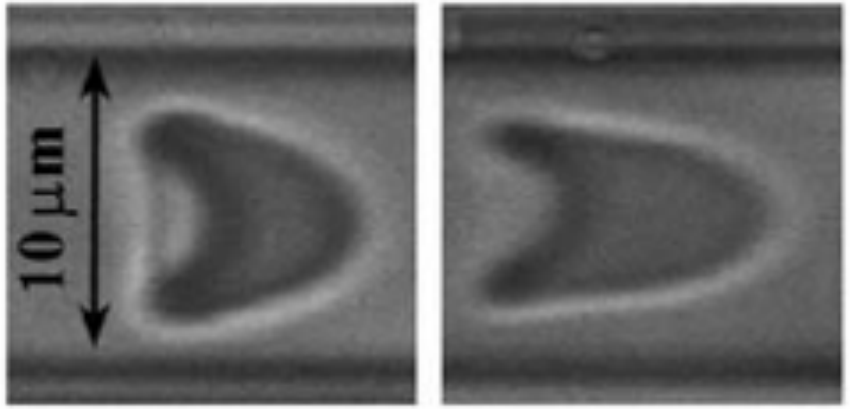
Shear flow experiment: Tank-treading



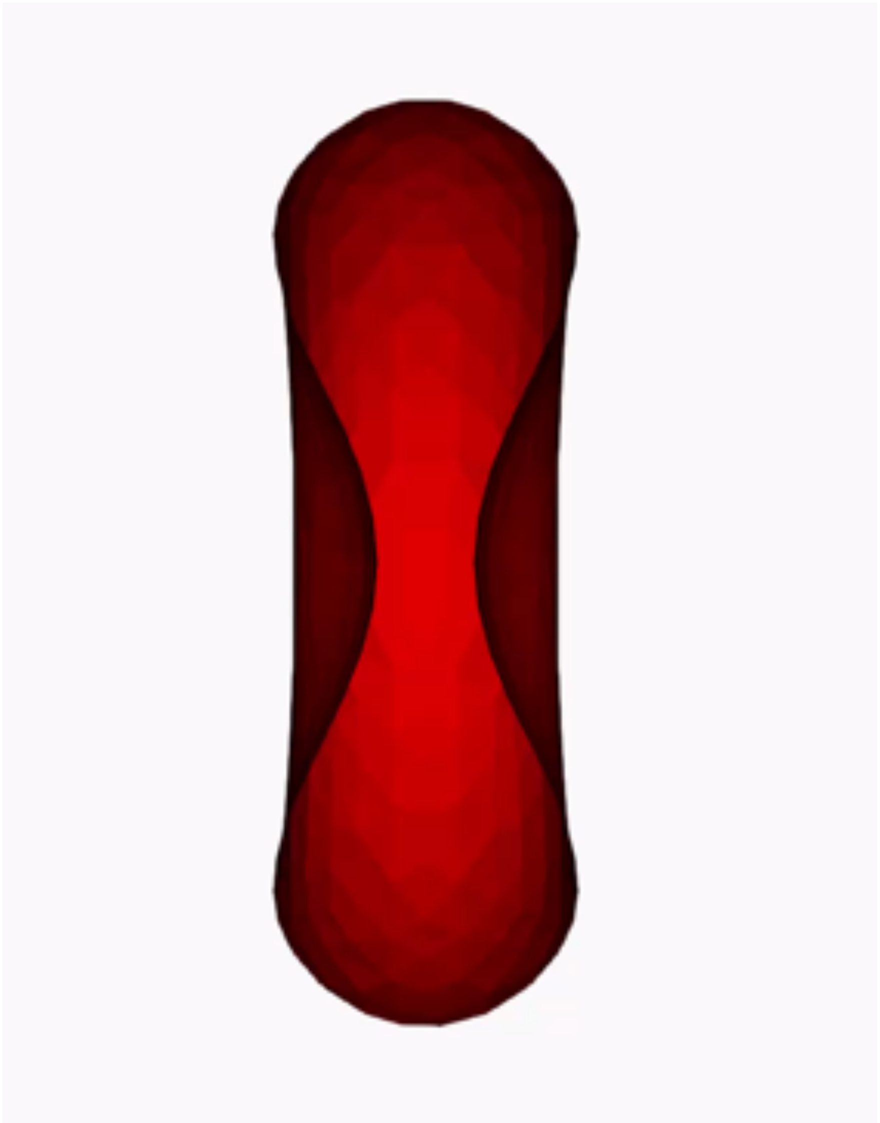


Poiseuille flow: Parachute-like shape

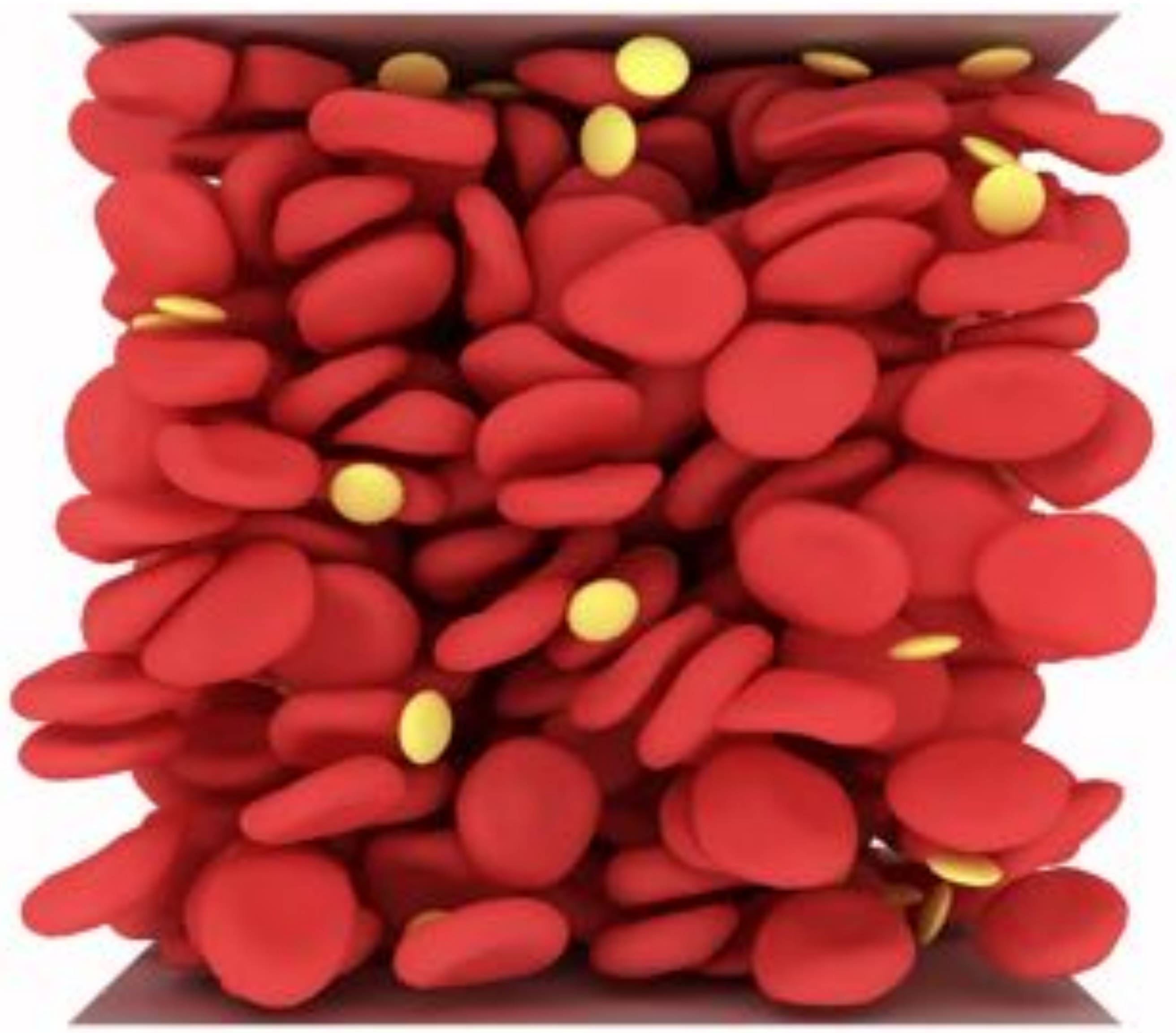




Tomaiuolo 2011



Multiple blood cells simulations

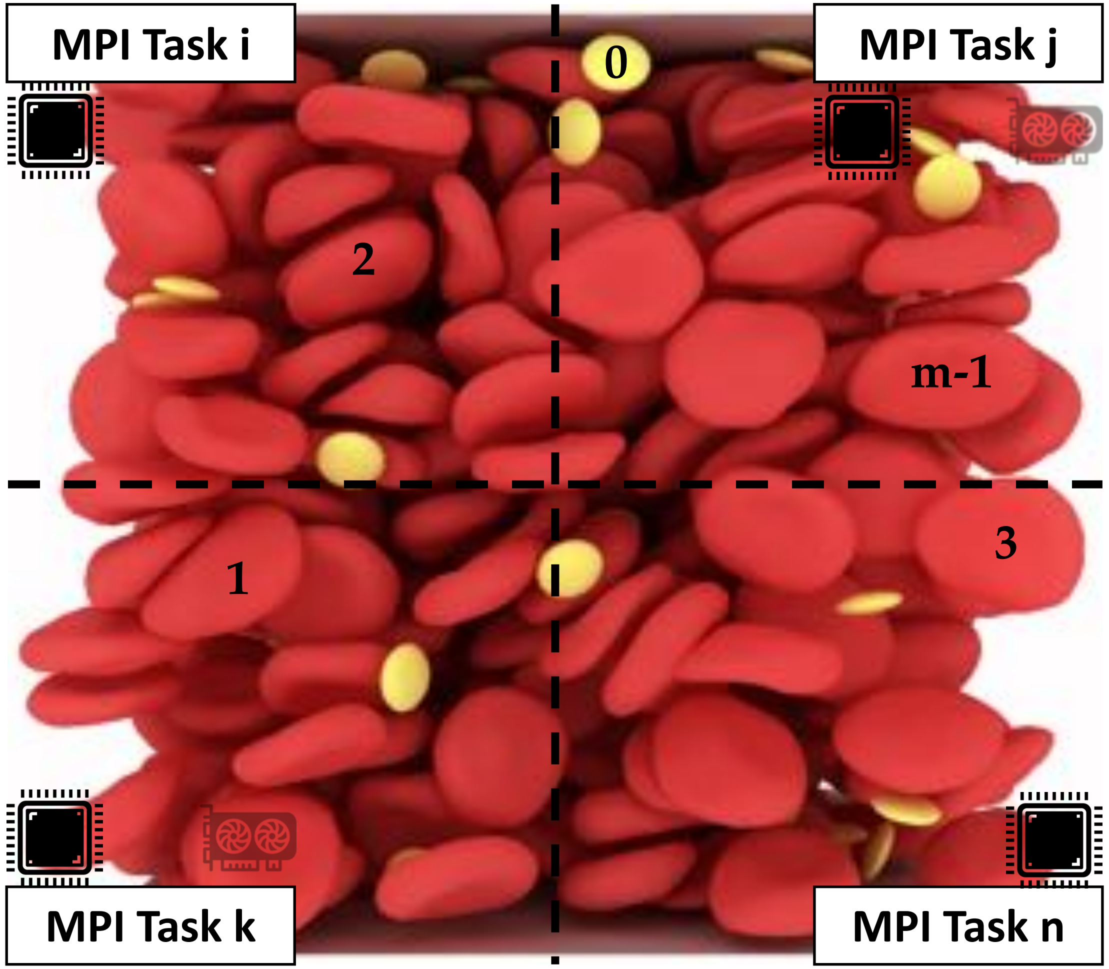


How to split the load to the available CPUs & GPUs ?

Presentation of the Hybrid Version of our computational framework

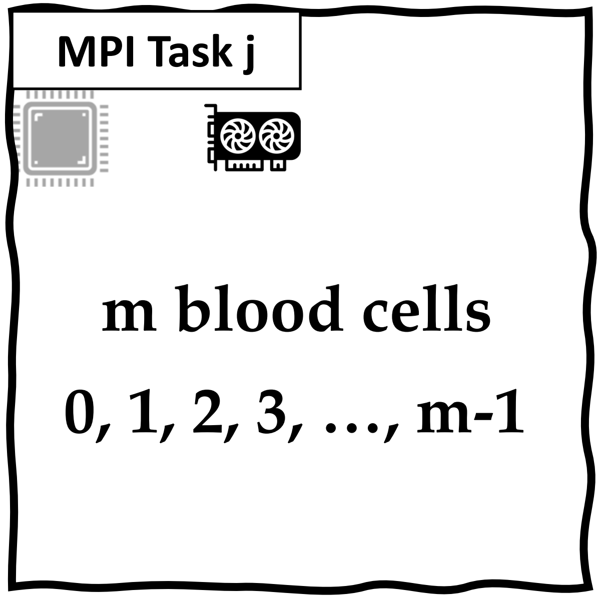
Load balancing

LBM &IBM



Straightforward to partition a static homogeneous grid
CPUs deal with grid points (LBM) & Lagrangian points (IBM)

npFEM



The blood cells are distributed once at the beginning to the available GPUs

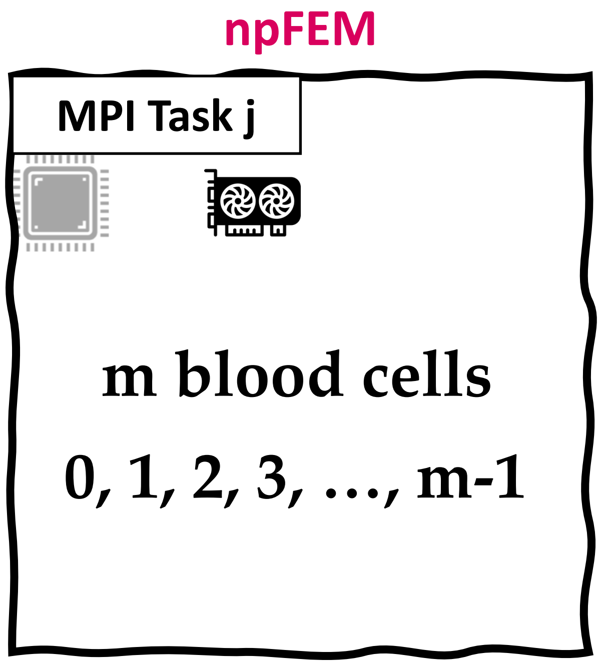
They can be spatially everywhere

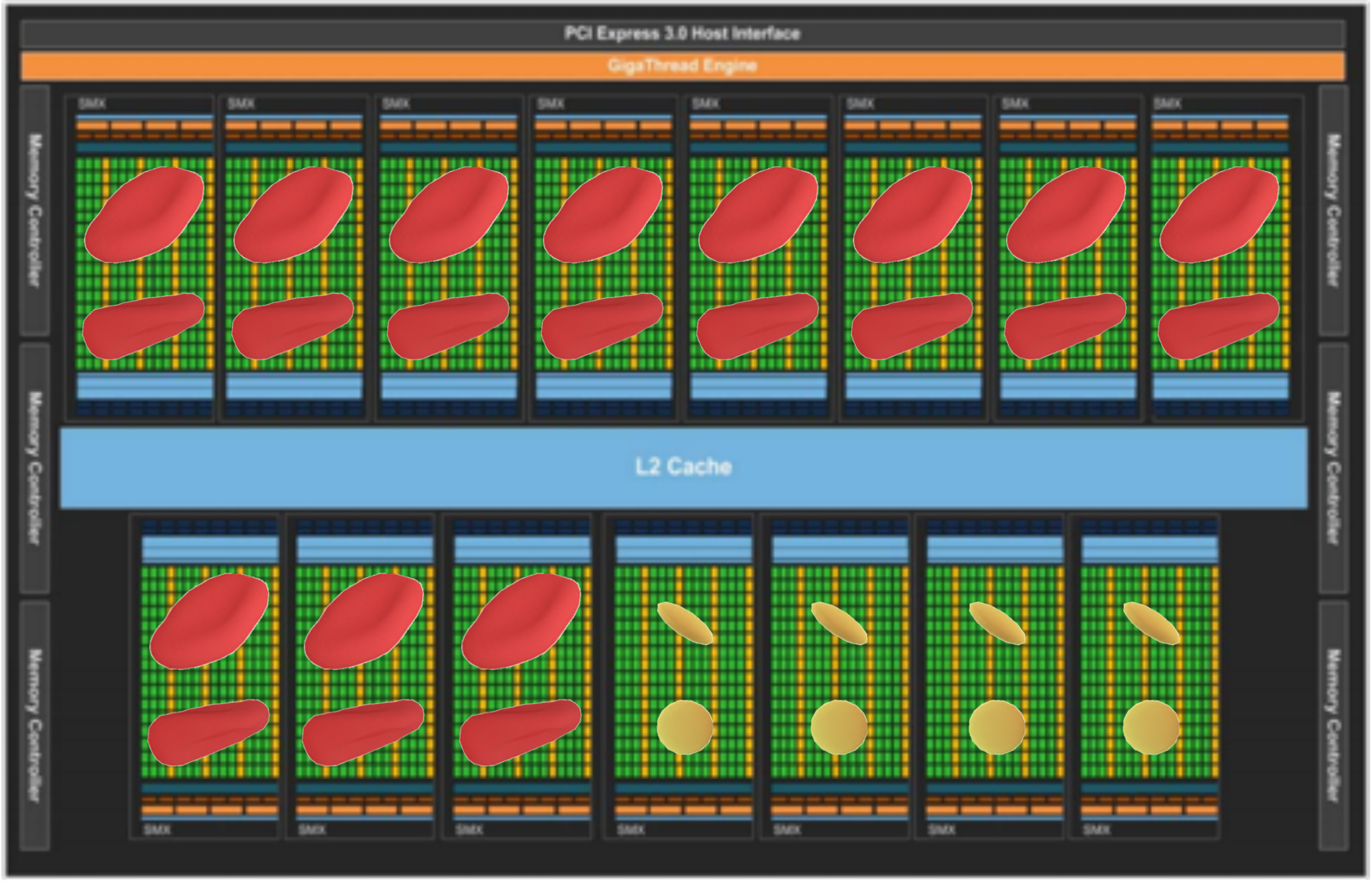
MPI point-to-point communication

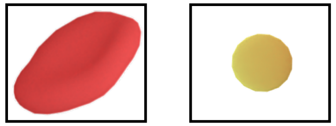
- The fluid solver sends forces & collision data to the solid solver
- The solid solver communicates the state at t+1

GPU acceleration

Fit bodies in the Streaming Multiprocessors to gain from the fast shared-memory and thread synchronization

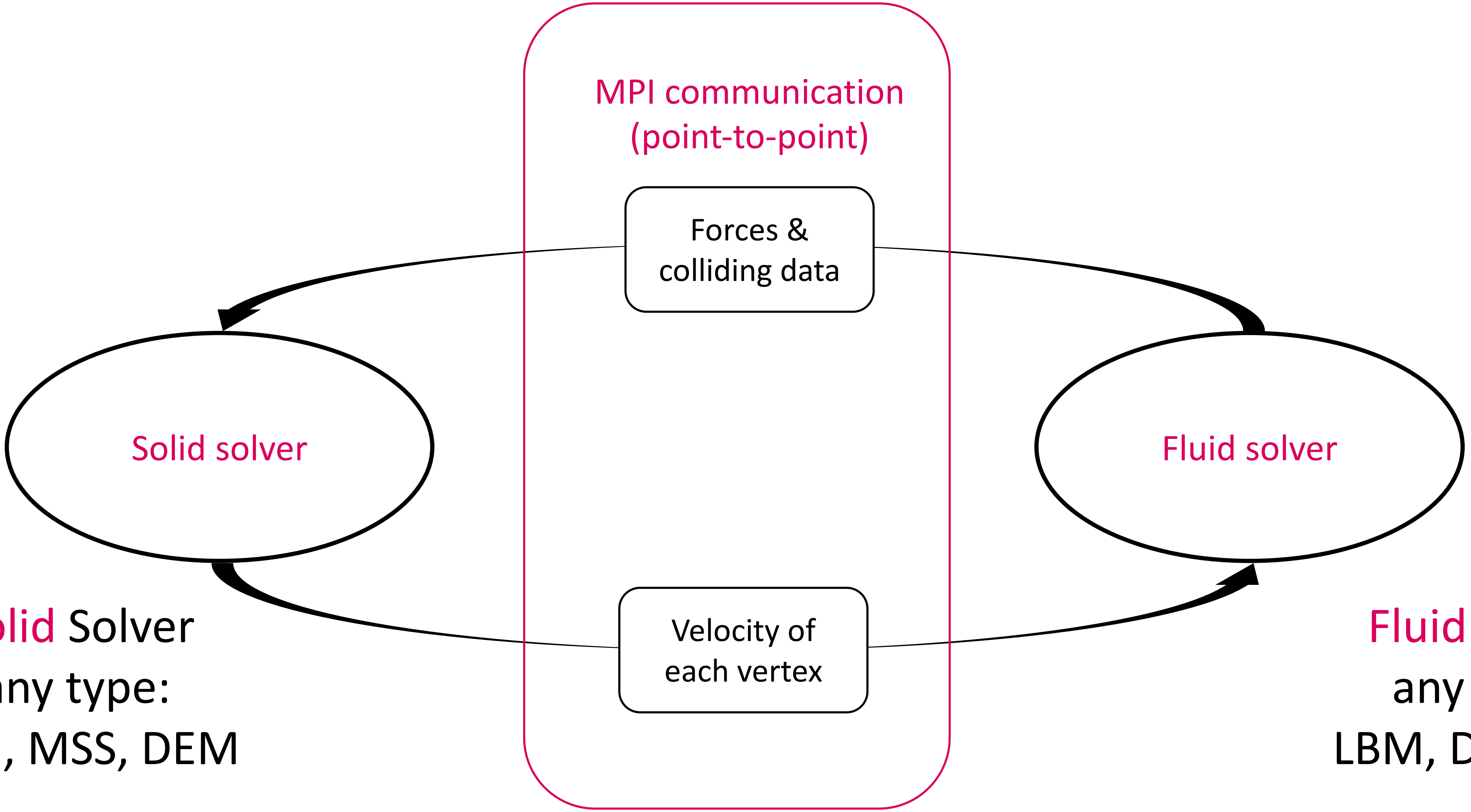






1 CUDA Block per blood cell
1 SMX deals with 1 CUDA block per time

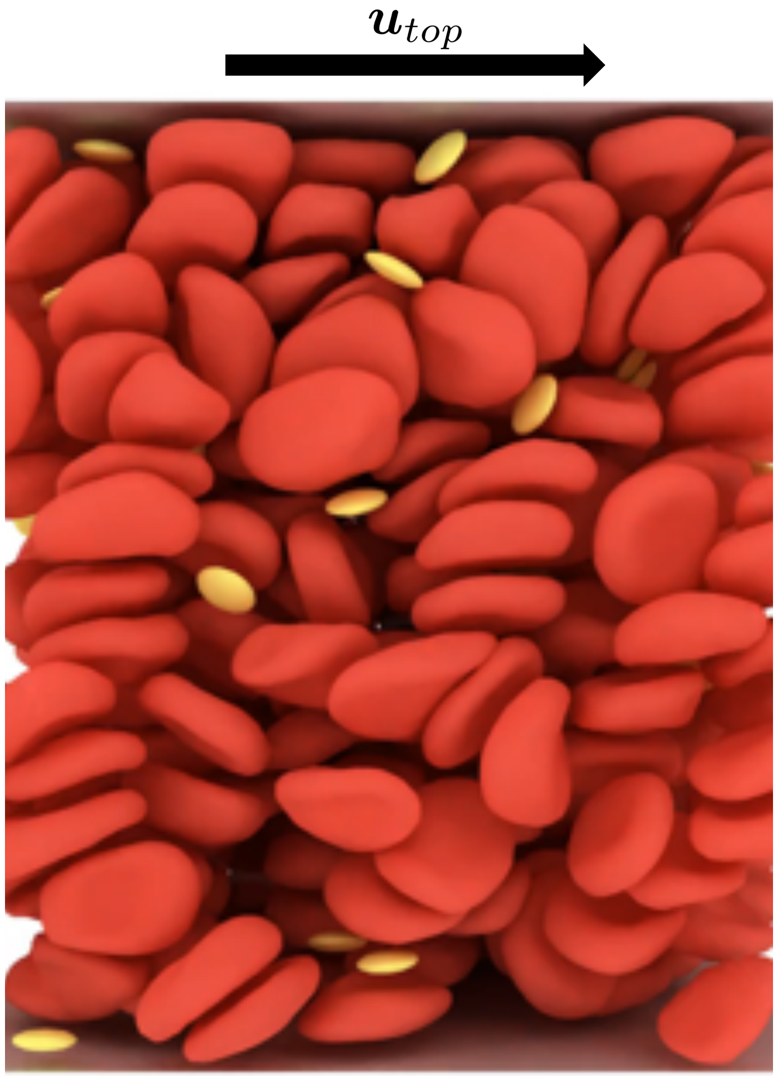
Genericity & modularity of our framework

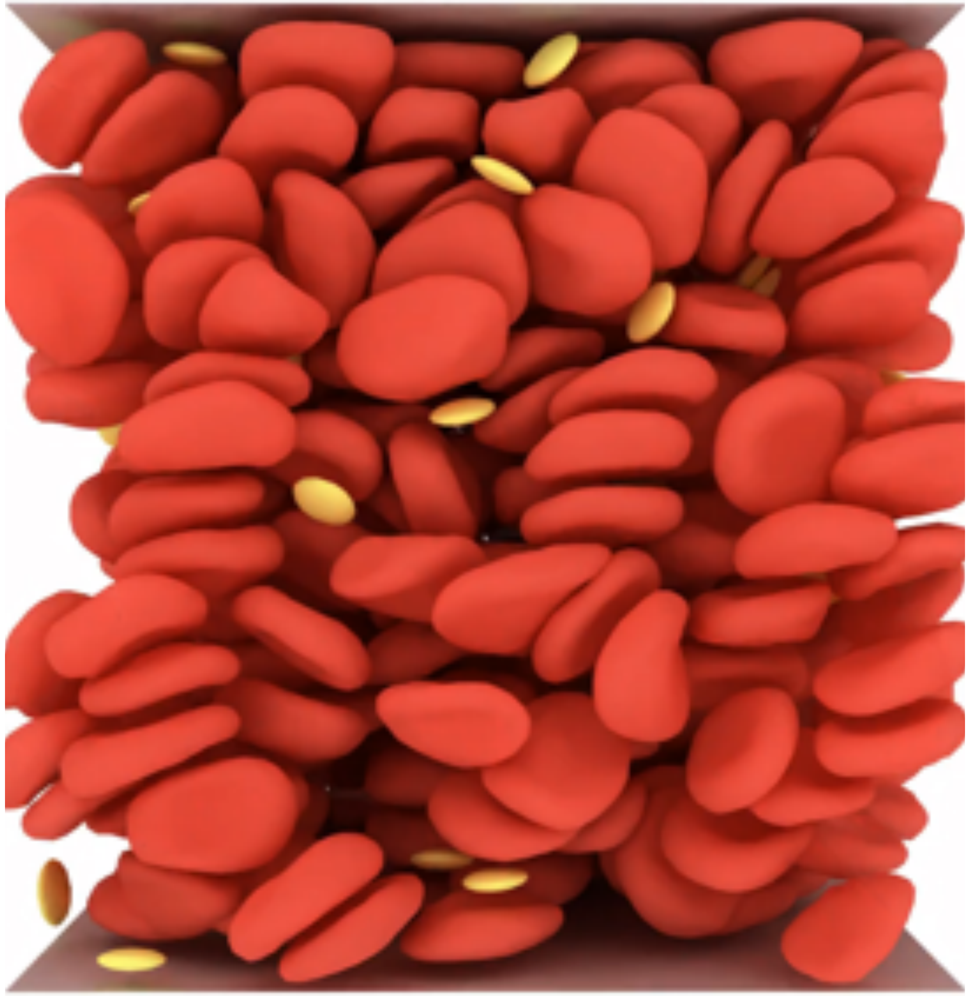


Solid Solver
any type:
FEM, MSS, DEM

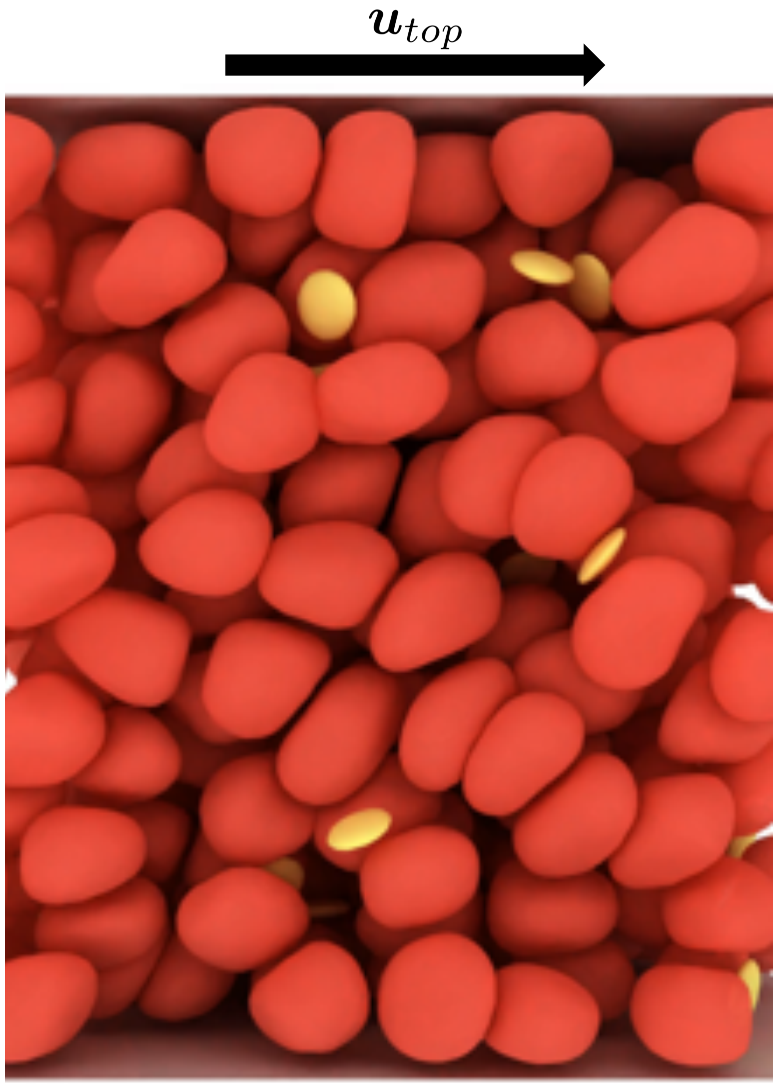
Fluid Solver
any type:
LBM, DPD, SPH

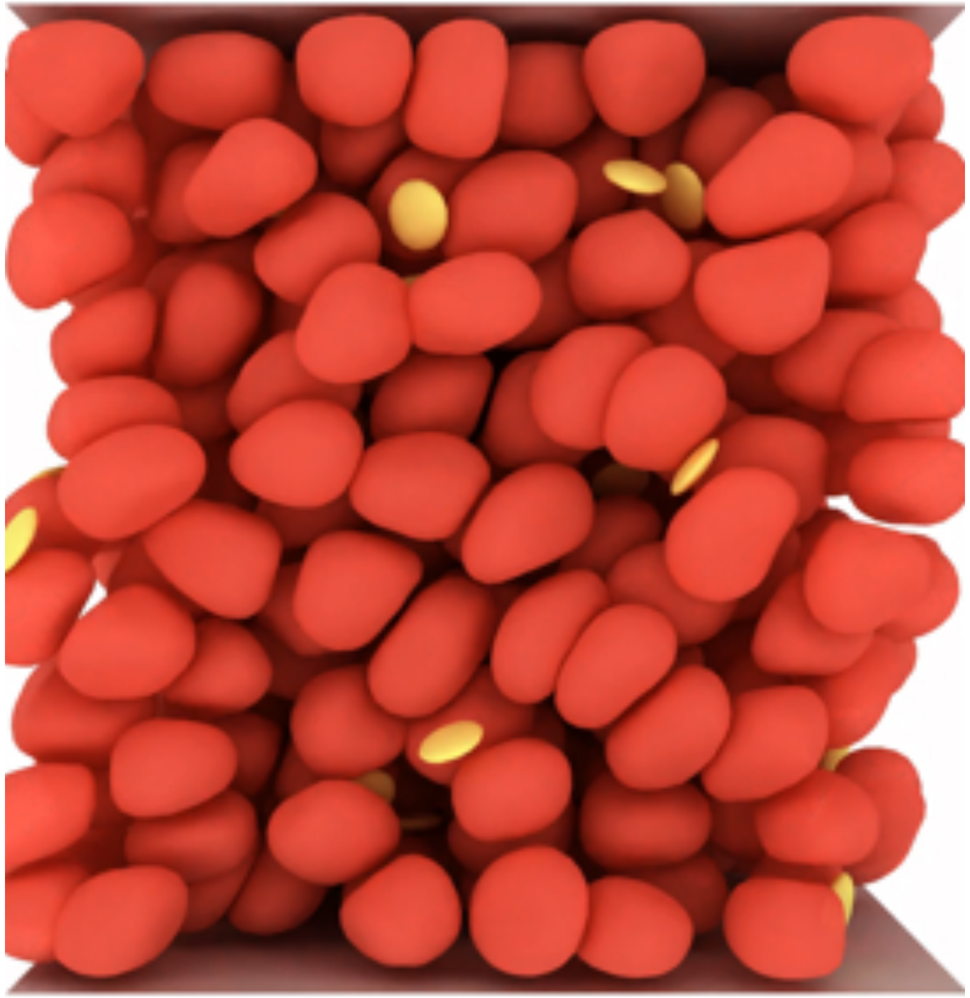
Shear flow: >500 blood cells | Hematocrit 35% | Box 50x50x50 μm^3





Focus on Pathological conditions, like Diabetes (swollen RBCs)





Performance measures

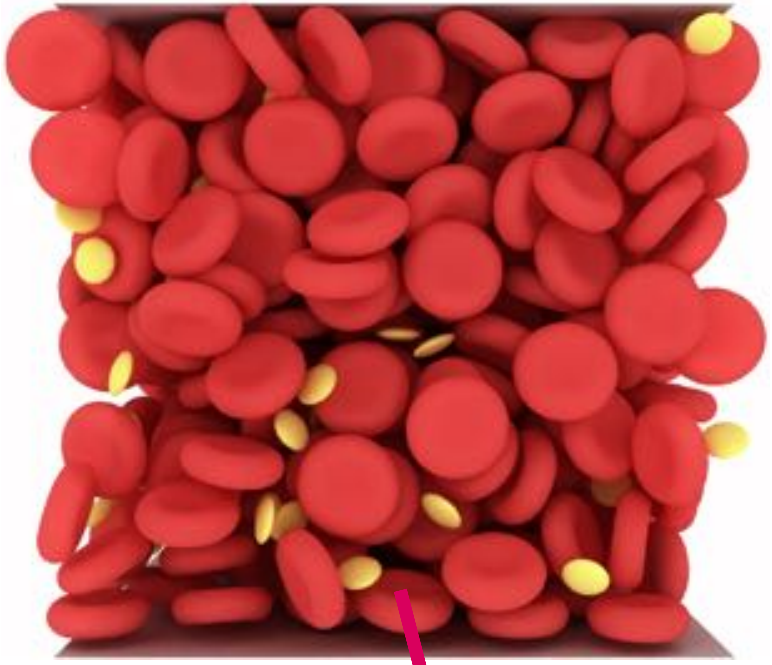
Piz Daint @CSCS

No. 6 Worldwide
No. 1 Europe

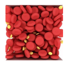
GPU Node (5704 Nodes)	CPU Node (1813 Nodes)
Hybrid Node (CPU+GPU)	CPU-only Node (**MC**)
CRAY XC50 12 cores – 64GB RAM (Intel Xeon E5-2690 v3 @ 2.60GHz) + 1 NVIDIA Tesla P100 16GB	CRAY XC40 36 cores – 128GB RAM (Two Intel Xeon E5-2695 v4 @ 2.10GHz)

Weak Scaling

Reference case study
Ht 35%, Box 50x50x50 μm^3



RBCs 258 surface vertices
PLTs 66 surface vertices



box50x50x50 : **1**
on 5 GPUs
RBCs: 476
PLTs: 95

box50x100x50 : **2**
on 10 GPUs
RBCs: 953
PLTs: 190

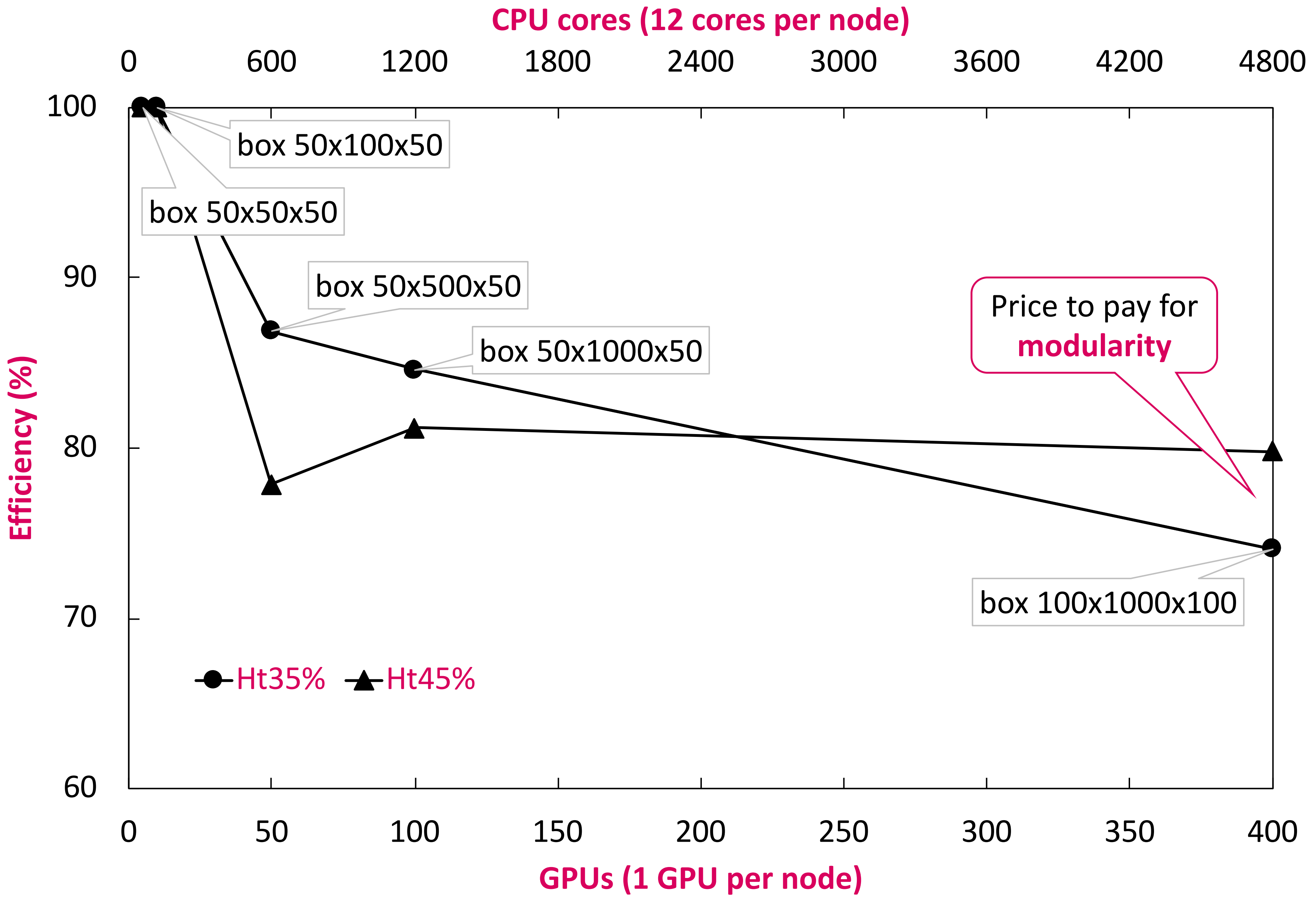
box50x500x50 : **10**
on 50 GPUs
RBCs: 4765
PLTs: 953

box50x1000x50 : **20**
on 100 GPUs
RBCs: 9531
PLTs: 1906

box100x1000x100 : **80**
on 400 GPUs
RBCs: 38126
PLTs: 7625

Weak Scaling
Hybrid Version

$$\text{Efficiency}_{weak} = \frac{t_{N_0}}{t_N} \times 100\ \%$$



Profiling of npFEM

Reference case study
Ht 35%, Box 50x50x50 μm^3

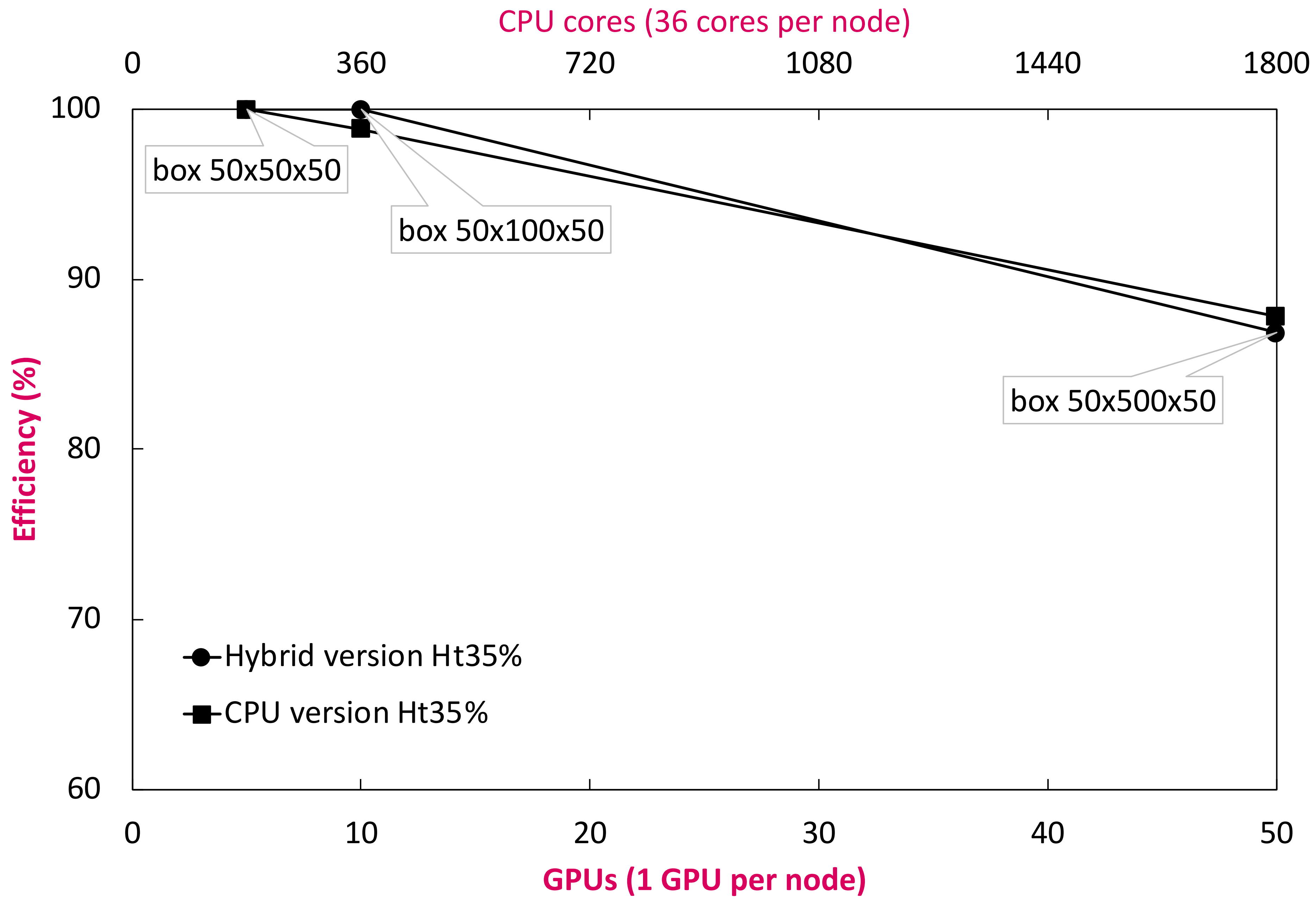
RBCs: 476
PLTs: 95

1 GPU node - npFEM [msec]	11
1 CPU node - npFEM [msec]	70
Speedup of npFEM	6.4
5 CPU node - npFEM [msec]	14

Weak Scaling
Hybrid / CPU

Solve the same problem with the same number of nodes (Hybrid & MC)

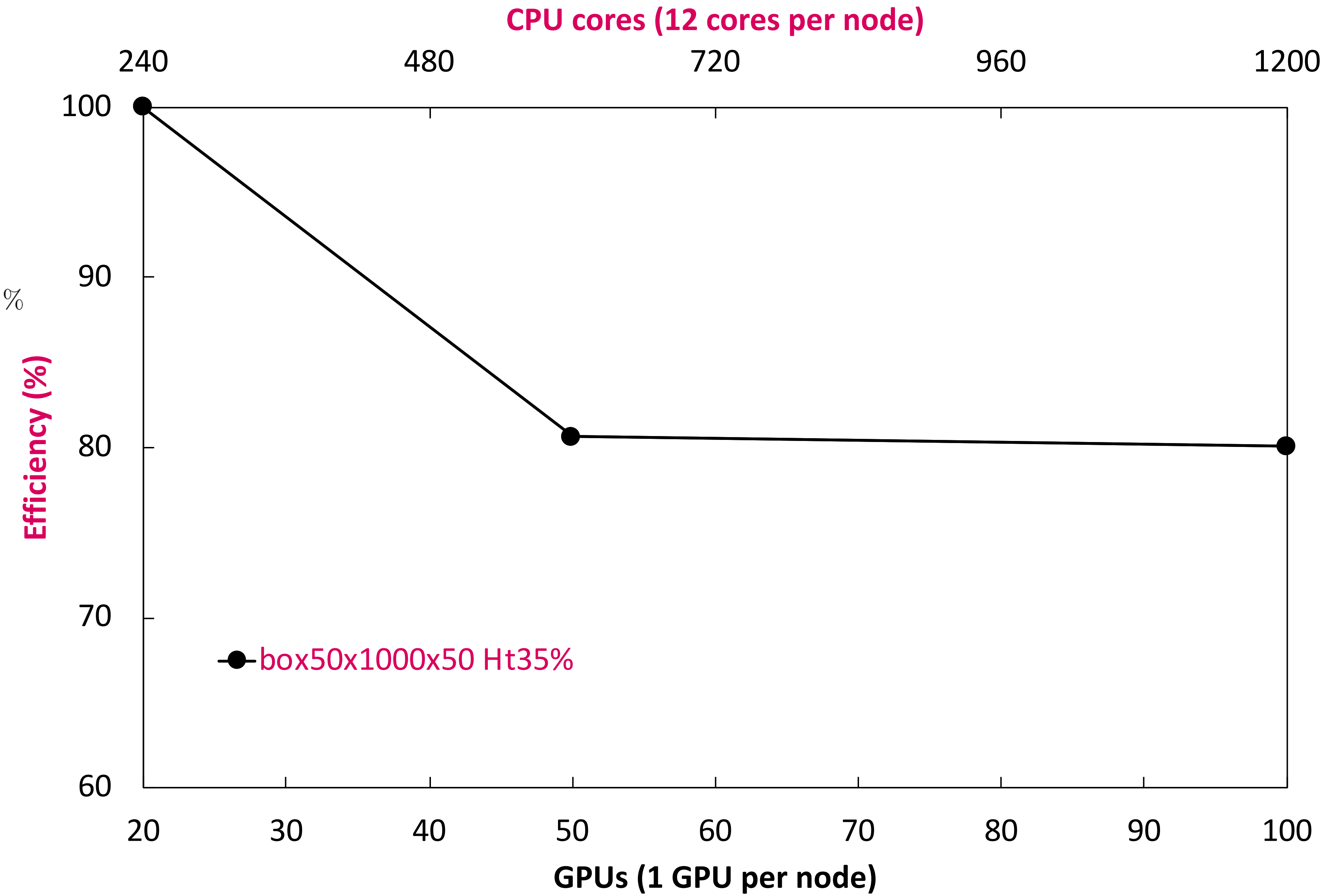
Example:
Ref problem
5 Hybrid Nodes
To
5 MC Nodes



Strong Scaling Hybrid Version

$$\text{Efficiency}_{strong} = \frac{t_{N_0}}{t_N}\frac{N_0}{N} \times 100\ \%$$

box50x1000x50
RBCs: 9531
PLTs: 1906



Future plans:

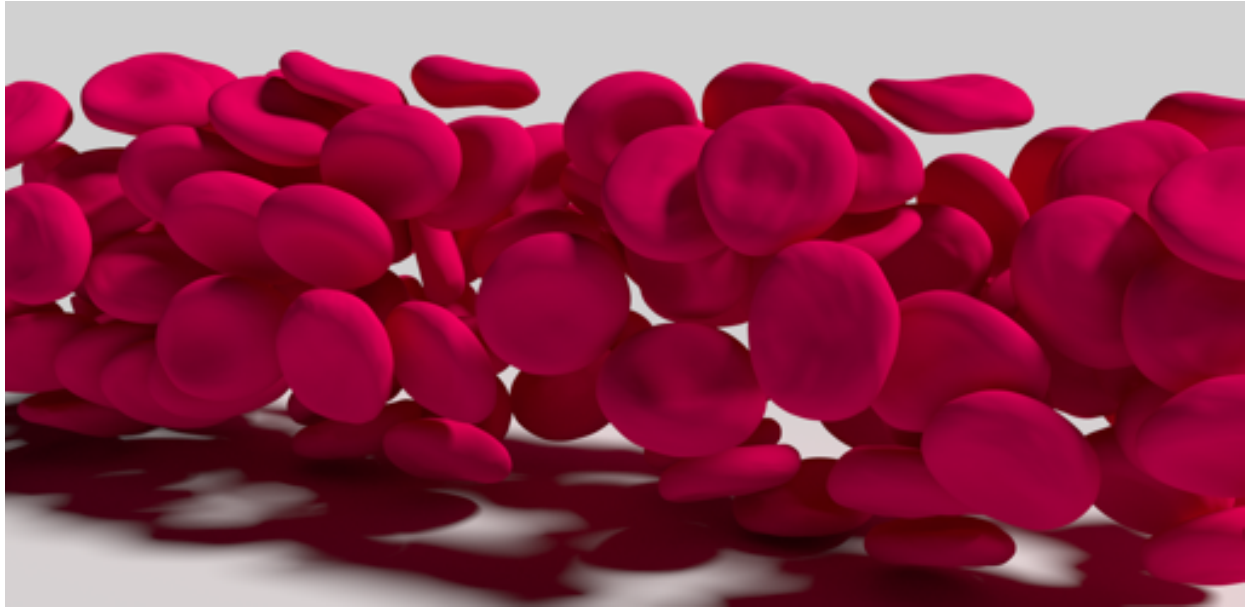
- Port the Fluid solver (and/or FSI) on GPUs using CUDA

1 mm^3 of blood contains:
O(10^6) RBCs

- Check scalability of our solver for multiple deformable bodies **O(10^6 - 10^9)**

- More investigation of platelet deposition through in-vitro and in-silico experiments

UNIVERSITÉ DE GENÈVE



Thank you very much