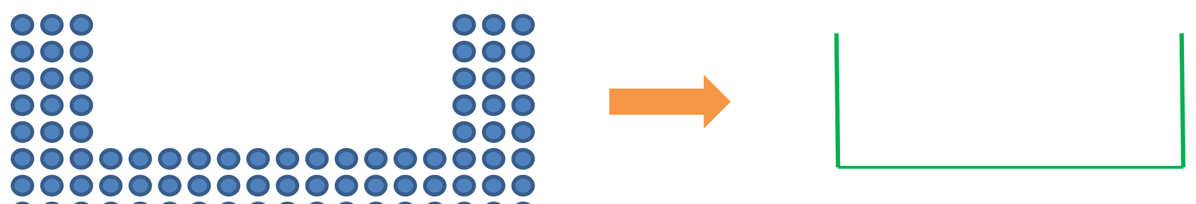
Dynamic Load Balancing Algorithms for E-MPS Method Adopting Polygon Wall Boundary Model

Yoshiki Mizuno

School of Engineering, The University of Tokyo

Introduction

- Our research groups developed the fluid analysis system, **HDDM_EMPS**[*1].
- It **formerly** adopted a wall-particles model.
- They developed load balancing algorithms for a wall-particles model[*1].
- Subsequently, they replaced it with a **polygon-walls model** [*2].
  - (a) can allocate arbitrarily-shaped triangular polygons for boundaries.
  - (b) has a potential to reduce computational cost for planar walls.
- However, the system with the model was NOT well load-balanced.

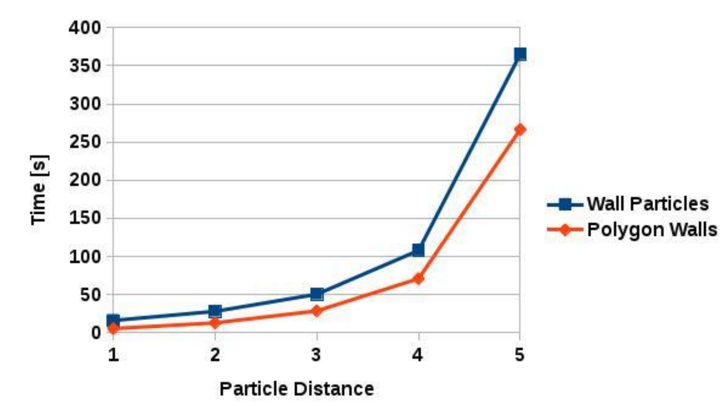


Wall-Particles Model **Polygon-Walls Model**

[*1] Murotani, K. et al., Journal of Advanced Simulation in Science and Engineering, Vol. 1, No. 1, 16-35, 2014.
[*2] Mitsume, N. et al., Computational Particle Mechanics, Vol. 2, No. 1, 73–89, 2015.

Computational Cost of Two Wall Models

- A hydrostatic pressure problem is solved to measure the computational cost.
- The number of particles in the wall-particles model equals to the total of fluid particles, wall particles, and dummy particles.
- The number of particles in the polygon-wall particles equals to fluid particles.



No.	Particle Distance	#Particles: Wall Model	#Particles: Poly. Model
1	0.030	19650	3960
2	0.025	28420	7680
3	0.020	45500	15000
4	0.015	83110	34320
5	0.010	221000	120000

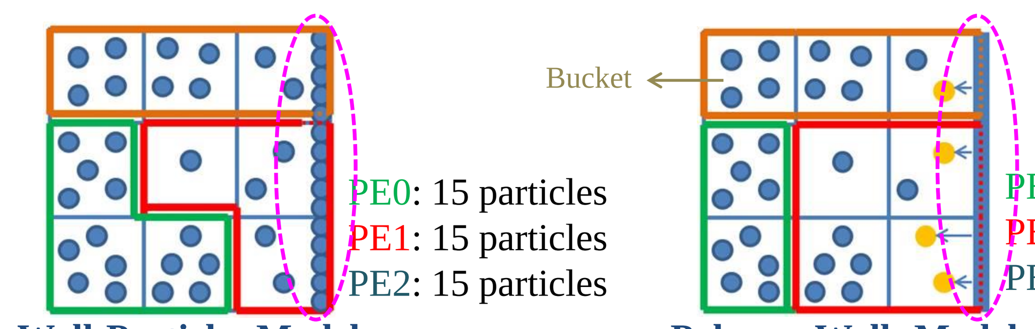
There exists a strong possibility that polygon-wall models are more appropriate for large-scale problems than wall-particles models.

Difference in Two Wall Boundary Models

The system with a polygon-walls model was not well load-balanced by the original algorithms adjusted for a wall-particles model due to the following two reasons:

(1) The amount of particles in a polygon-walls model becomes distinct from the one in a wall-particles model.

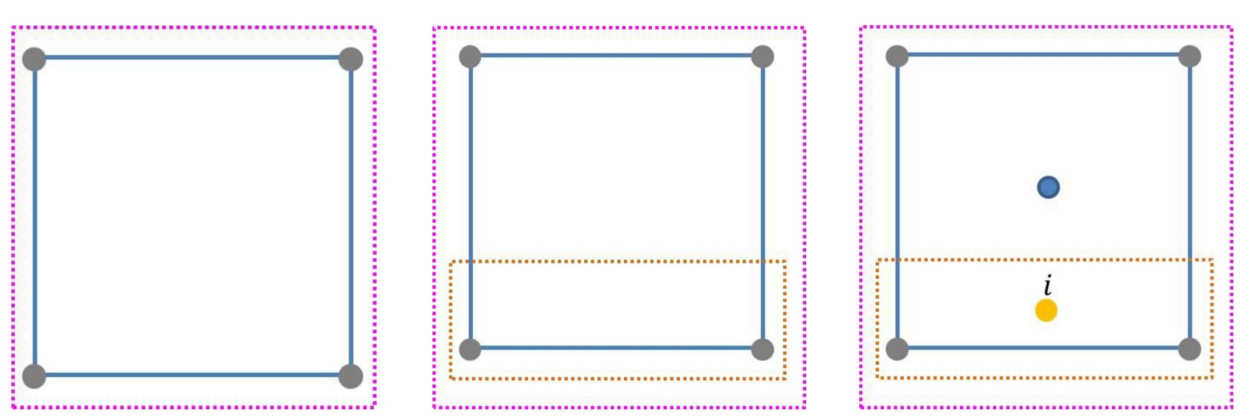
(2) The system holds two different types of particles.
- (a) Normal Particles (●) ⇒ are NOT filtered in the polygon-walls algorithms.
- (b) Polygon Particles (●) ⇒ ARE screened in them.



Wall-Particles Model **Polygon-Walls Model**

Polygon-Walls Algorithms

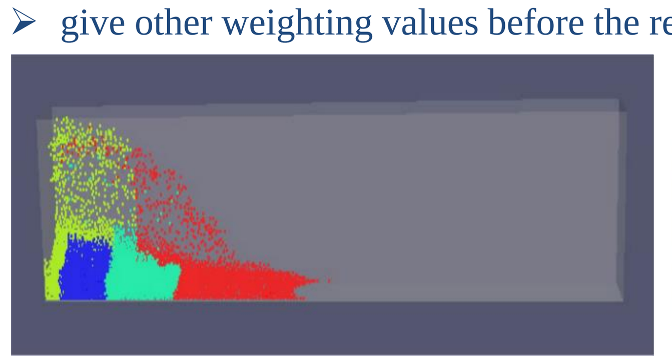
- The following steps in polygon-walls algorithms increase computational cost in a certain PE that hold particles located at near polygon-wall boundaries.
  - (a) Set up a global bounding box for an analysis domain.
  - (b) Allocate a local bounding box inside the global one.
  - (c) Filter whether the particle i is inside the local box.

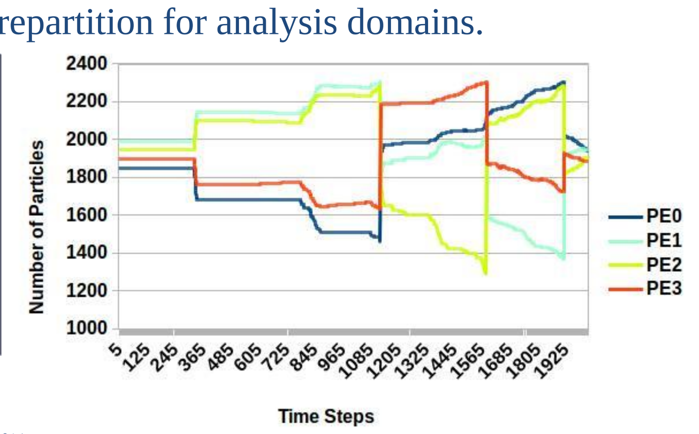


(a) (b) (c)

Original Algorithms for Wall-Particles

The original algorithms:
- utilize ParMETIS, an MPI-based library that partitions unstructured meshes and graphs[*1, 2].
- give weighting values based on the number of particles each PE hold in order to stabilize the load balancing.
- regard a particle as the weighting value of one.
- repartition an analysis domain of each PE if the balance ratio in a PE exceeds the arbitrary value.
- give other weighting values before the repartition for analysis domains.





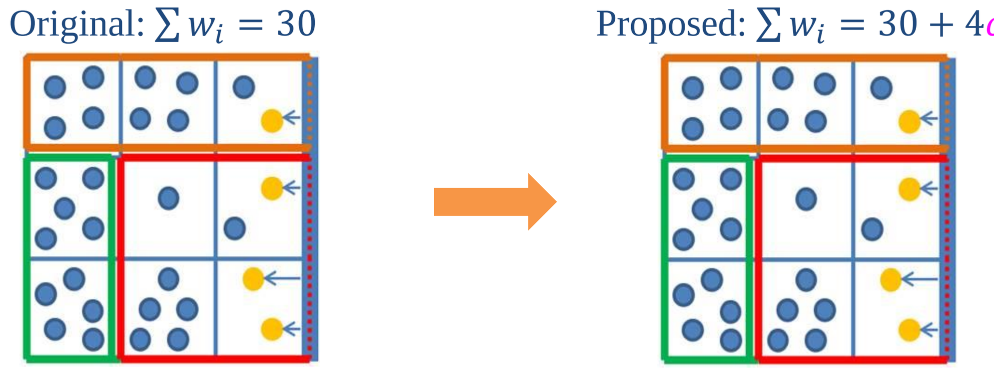
*1 Murotani, K. et al., Journal of Advanced Simulation in Science and Engineering, Vol. 1, No. 1, 16-35, 2014.
*2 Karypis, G. et al., Siam Review, Vol. 41, No.2, 278-300, 1999.

Proposed Algorithms for Polygon Walls

- The proposed algorithms practice the following steps.

(a) Identify whether a particle i is filtered by the polygon-walls algorithms.
(b) If i is a normal particle, then $w_i = 1$ where w_i is a weighting value.
(c) If i is a polygon particle, then $w_i = 1 + c$ where c is a constant value assumed to be computational cost caused by the polygon-walls algorithms.
(d) Repeat (a) through (c) for each particle.

Original: $\sum w_i = 30$ Proposed: $\sum w_i = 30 + 4c$



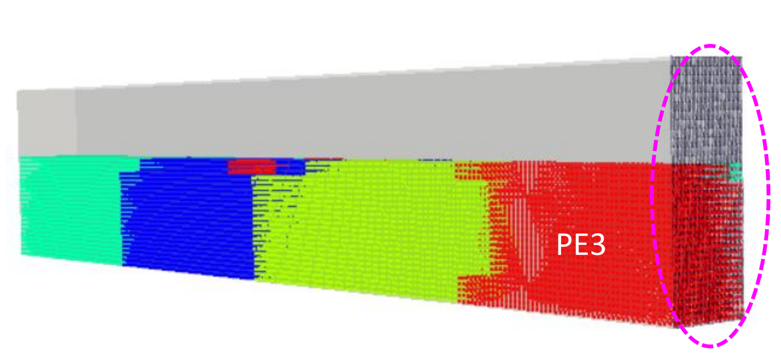
Application to Hydrostatic Pressure Problem

Based on the two factors explained in the previous page, I developed load balancing algorithms that distribute appropriate weighting values to each PE dynamically.

- Hydrostatic water ($5.0 \times 0.4 \times 0.6[m^3]$) is placed in a cuboid container ($5.0 \times 0.4 \times 1.0[m^3]$).
- Polygon walls are expressed by triangular polygons.
- Each side of this container has two triangular polygons except the specific one, which holds $2n^2$ polygons*.
- The computational cost increases in PE3 whose analysis domain is in charge of the specific side.

.

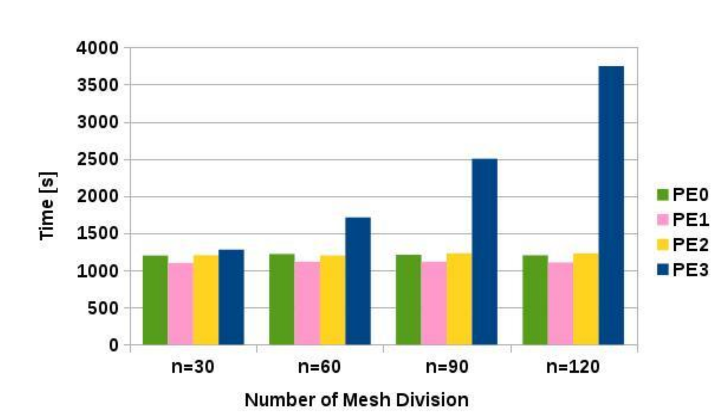
CPU	Intel Core i5
Clock Rate	3.20 GHz
Cache Size	4096 KB
#PEs	4
#Particles	76,800
Time Step	1.0×10^{-4}[s]
Particle Distance	2.5×10^{-2}[m]
Kinematic Viscosity	$1.0 \times 10^{-6}[\mathrm{m}^2/\mathrm{s}]$
Fluid Density	$1.0 \times 10^{3}[\mathrm{kg}/\mathrm{m}^3]$
Effective Radius	3.1×10^{-2}[m]
Gravity	$-9.81[\mathrm{m}/\mathrm{s}^2]$



* n indicates the number of mesh division.

Influence of Mesh Division on Each PE

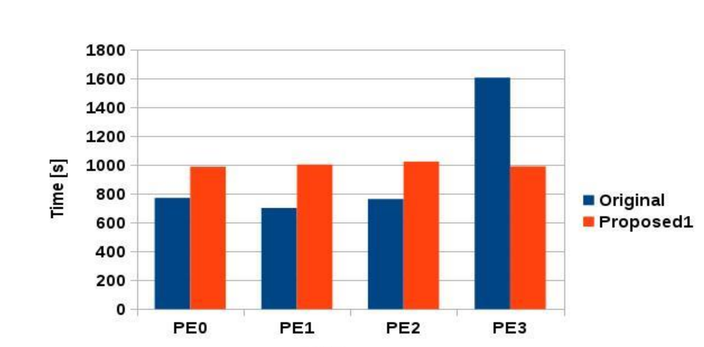
- In the original algorithms are first applied to the hydrostatic pressure problem to investigate how the number of mesh division affects computational time on each PE.
- The number of mesh division is set to 30, 60, 90, and 120.
- The simulation time is 1.5×10^{-1} [sec].



- The cost of PE3 increases as the number of mesh division becomes larger.
- The cost of other PE does not change very much because the polygon-walls algorithms are not involved in them less than PE3.

Results of Load Balancing

- The "Proposed1" indicates the proposed algorithms.
- The calculation was conducted three times to take its average.
- The simulation time is 1.0×10^{-1} [sec], and $n = 90$.



PEs	Original	Proposed1
PE0	772.72 [s]	989.14 [s]
PE1	702.41 [s]	1003.99 [s]
PE2	765.36 [s]	1025.25 [s]
PE3	1608.12 [s]	991.60 [s]
Total	1641.49 [s]	1172.84 [s]

- The "proposed1" decreased the computational cost for PE3.
- Computational time in each PE was equally distributed.
- The graph/table demonstrate the validity of the algorithms.
- The algorithms have been applied to hydrodynamic problems for future works.