

Rapid Variable Resolution Particle Initialization for Complex Geometries

Navaneet Villodi and Prabhu Ramachandran
 Department of Aerospace Engineering
 Indian Institute of Technology Bombay
 Mumbai, India
 navaneet@iitb.ac.in, prabhu@aero.iitb.ac.in

Abstract - This paper presents a novel method for variable resolution particle initialization for complex geometries in 2D/3D. The literature on variable resolution particle initialization with SPH is sparse. The presented algorithm is demonstrated to be much faster than some state-of-the-art methods. The results also indicate that the generated particle distributions are of comparable quality. By enabling improved simulation of flow in and around complex geometries, this method aims to improve the applicability of SPH in a wide range of industrial and engineering simulations.

I. INTRODUCTION

Boundary treatment methods in SPH usually involve the use of ghost particles. Proper initialization of these ghost particles representing the body is crucial for accurate results. A uniform distribution of non-ghost particles is also important for accurate simulation with the SPH. The following are the common pain points that one encounters when dealing with the present methods, (a) complexity: the present state-of-the-art methods [1], [2] are quite complex to implement, (b) efficiency and speed: complexity often brings with it performance overhead and (c) adaptive resolution: which is crucial for resolving intricate features while still keeping the number of particles and thereby the computational cost low. Some of the existing methods [1], [2] do not provide a way to generate particle initialization with adaptive resolution. Other methods have tried to address variable resolution [3], [4], [7] but are still quite complex to implement.

We propose a method that is relatively easy to implement, very efficient, and allows one to generate variable resolution particle initializations with accuracy that is comparable to existing methods. Moreover, since the method consists of pure SPH building blocks, it can be reproduced within any SPH framework by an SPH practitioner. We describe the method in the following sections.

II. OBJECTIVES

The initialized particles are meant to be used both for the main particles (say the fluid) as well as ghost particles to enforce boundary conditions. The ghost particles complete the support of the nearby fluid particles near the interface. Properties from the fluid particles are interpolated onto the ghost particles

after appropriate modifications. This is done to ensure that the boundary conditions are satisfied. To this end, the main objectives can be summarized as follows:

- 1) Density should be uniform, i.e,

$$\rho_0 - \rho_i = \rho_0 - \sum_{j \in \mathcal{N}} W_{ij} m_j \rightarrow 0, \quad (1)$$

where ρ_0 denotes the target density, ρ the density and m the mass. i and j represent particle indices. \mathcal{N} represents the set of particles in the neighbourhood of the i^{th} particle. W is the kernel function.

- 2) Partition of unity should be satisfied, i.e,

$$1 - \sum_{j=1}^N W_{ij} \frac{m_j}{\rho_j} \rightarrow 0 \quad (2)$$

This ensures zero-order consistency and curbs the approximation errors that creep in as a fluid particle interacts with the ghost particles.

- 3) The resolution should not change abruptly. Abrupt changes in number density may be useful in applications like image dithering [5], but this does not support objectives 1 and 2.
- 4) We are not aiming for a body-fitted distribution of particles, i.e., we do not want particles to be on the interface. In SPH simulations, the fluid and ghost particles interact on either side of the interface. So, particles should ideally be $\sqrt{3}\Delta s/2$ away from the interface in 2D and $\sqrt{2}\Delta s/\sqrt{3}$ in 3D. $\Delta s_i \propto (m_i/\rho_i)^{(1/d)}$ where d is the dimensionality of the problem. These numbers denote the minimum distance adjacent close-packed layers in 2D and 3D, respectively.

III. ALGORITHMIC COMPONENTS

Having clearly defined the objectives, we can now describe the existing components from SPH that can be used to achieve these.

- 1) Restoring force: We assume that the geometry to be packed with particles is basically a shell in a pressurized fluid container. Thus, we can make use of the familiar equations of motion for fluids, commonly discretized in SPH as

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho_i} \sum_{j \in \mathcal{N}} (p_i + p_j) \nabla W_{ij} \frac{m_j}{\rho_j}, \quad (3)$$

where \mathbf{v} is the velocity and p is the pressure computed as $p = p_0(\rho/\rho_0)^\gamma$. p_0 and ρ_0 are constants.

- 2) Particle shifting: A particle shifting technique commonly used in SPH [6] to regularize the distribution of particles. The particle displacement, $\delta\mathbf{r}_i$ is given as

$$\delta\mathbf{r}_i = \begin{cases} -0.5h^2\widehat{\nabla}C_i & \text{if } |0.5h^2\widehat{\nabla}C_i| < 0.2h \\ -0.2h\frac{\widehat{\nabla}C_i}{|\widehat{\nabla}C_i|} & \text{otherwise} \end{cases}, \quad (4)$$

where

$$\widehat{\nabla}C_i = \sum_{j \in \mathcal{N}} \left[1 + 0.2 \left(\frac{W_{ij}}{W(\Delta x_i)} \right)^4 \right] \frac{m_j}{\rho_j} \nabla W_{ij}. \quad (5)$$

Here, h is the smoothing length, Δx is taken as the smoothing length times the inflection point of the kernel function, W .

- 3) a) Volume based adaptivity: The algorithm from [8] is borrowed and implemented within the framework of [12]. In a nutshell, the volume of a particle exceeds a certain threshold, it is split into four equal mass equal volume particles, if the volume is below a certain threshold, it is merged with its nearest that also satisfies the same condition. These upper and lower thresholds are derived from a reference volume, V_{ref} as $8V_{ref}/5$ and $2V_{ref}/3$, respectively. The volume is derived from local spacing, Δs .
- b) Refinement guide: In order to capture some local detail, we set the local spacing of adjacent particles to be smaller than the rest. The local spacing needs to vary from coarse to fine as we move away from the region of interest. This is achieved by using the technique from [10], [12]. This technique results in stratified spacing bands around the region of interest.
- 4) Mass dissipation: Any irregularities arising out of stratification or other issues with adaptive refinement, say an eligible particle not being able to find a merge partner, are effectively mitigated with the mass dissipation from [11]. In a nutshell, interacting particles with different masses exchange a small portion of their mass as

$$\frac{dm}{dt} = \sum_{j \in \mathcal{N}} \frac{m_j + m_i}{\rho_j + \rho_i} \psi_{ij} v_{ij}^{sig,m} m_{ij} \hat{\mathbf{r}}_{ij} \cdot \nabla_i W_{ij}, \quad (6)$$

where the signal velocity, $v_{ij}^{sig,m}$ is given as

$$v_{ij}^{sig,m} = \begin{cases} \sqrt{\frac{|p_{ij}|}{\rho_i + \rho_j}} & \text{if } \hat{\mathbf{r}}_{ij} \cdot \mathbf{v}_{ij} < 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

and the limiter function, ψ_{ij} is given as

$$\psi_{ij} = \frac{(\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij})^2}{\mathbf{v}_{ij} \cdot \mathbf{v}_{ij} + (\epsilon_\psi v_{ij}^{sig,m})^2 + (0.0005(c_i + c_j))^2}, \quad (8)$$

where ϵ_ψ is set as 0.1. $m_{ij} = m_i - m_j$ and similarly \mathbf{v}_{ij} and p_{ij} . c the speed of sound and \mathbf{r} denotes the position vector. $\hat{\mathbf{r}}_{ij}$ is the unit vector from j to i .

- 5) Boundary handling: Given a set of points representing the interface, particles near the interface are displaced normal to the interface. The displacement is just enough to ensure that there are no particles at a distance less than $\sqrt{3}\Delta s/2$ from the interface in 2D and $\sqrt{2}\Delta s/\sqrt{3}$ in 3D. Although this displacement may seem crude, it is pretty effective in practice.

To summarise, item 1 above automatically ensures the satisfaction of objective 1 by virtue of Pascal's law and the inherent diffusion. Item 2 ensures the satisfaction of objective 2. Items 3 and 4 ensure satisfaction of objective 3. Finally, item 5 ensures the satisfaction of objective 4. Additionally, for the initial 10 iterations, particle positions are perturbed randomly with Gaussian noise with mean zero and standard deviation equal to 0.4 times the smoothing length. For time integration, Euler integration is used with a time step, $\Delta t = 0.4h/(c + 10|\mathbf{v}|)$ and $c = \sqrt{\gamma p/\rho}$.

IV. RESULTS

Figure 1 shows the result of packing a unit circle with variable resolution. The resolution is refined around the point (0.0, 1.0) by setting $\Delta s = 0.025$ and the ratio of the spacing between adjacent bands is set to 1.2.

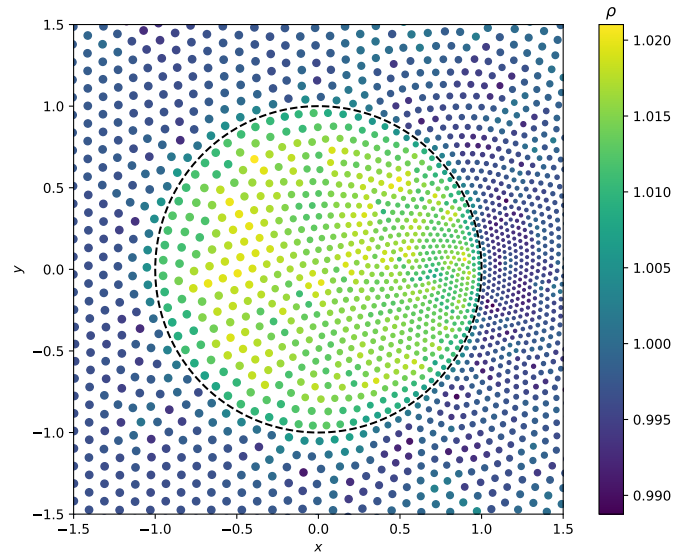


Fig. 1. Packed unit circle with variable resolution, refinement around (0.0, 1.0).

Figure 2 shows the result of packing a starfish with constant resolution, $\Delta s = 0.1$.

Figure 3 shows the result of packing a unit chord NACA 0012 airfoil with variable resolution. The spacing at the leading and trailing edge are set as $\Delta s = 0.00625$.

In order to compare the performance of our proposed algorithm we compare the time taken with that of [1]. The results

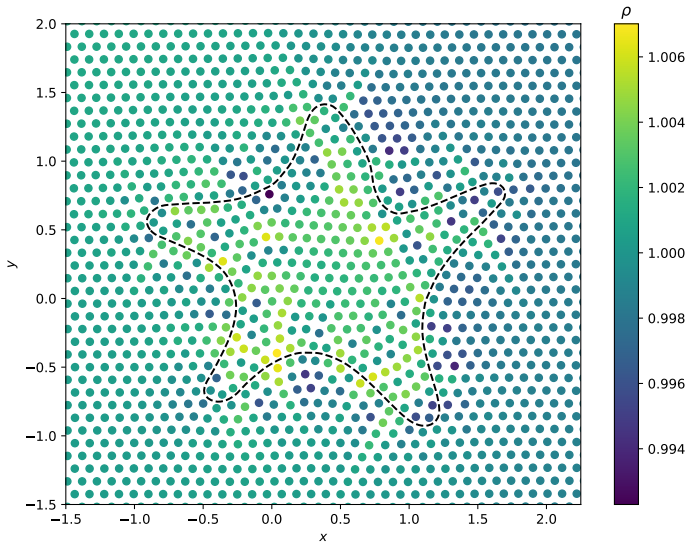


Fig. 2. Packed starfish with constant resolution.

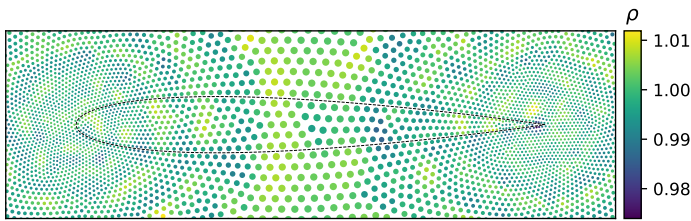


Fig. 3. Packed NACA 0012 airfoil with variable resolution. Refinement around leading and trailing edge.

are shown in Table I. As can be seen the new method can be up to an order of magnitude faster.

TABLE I
TIME TAKEN FOR PARTICLE INITIALIZATION.

Case	Serial			OpenMP		
	[1]	Present	Speedup	[1]	Present	Speedup
Circle	22.26	4.82	4.61x	6.84	1.07	6.39x
Starfish	77.67	6.39	12.15x	21.52	1.49	14.42x

¹ Constant Resolution. Run time in seconds. Min of 5 runs on an Intel core i7-8700 machine.

V. CONCLUSION

This paper introduced a pure SPH particle initialization algorithm using a combination of existing SPH techniques as building blocks. The method is simple, fast, and provides a way to adaptively change the resolution locally for a given configuration. The results indicate that the generated particle distributions are of comparable quality to state-of-the-art methods. The method is expected to be useful in a wide range of industrial and engineering simulations.

While the formulation of the method is suitable for implementation in 3D, we still need to test the method for its performance in 3D. We are also working on some algorithmic improvements and optimizations as well as developing an automatic criterion for terminating the iterations.

ACKNOWLEDGMENT

The authors acknowledge the use of the computing resources of the ACE facility, Department of Aerospace Engineering, IIT Bombay.

REFERENCES

- [1] P. Negi and P. Ramachandran, "Algorithms for uniform particle initialization in domains with complex boundaries," *Computer Physics Communications*, vol. 265, p. 108008, Aug. 2021, doi: 10.1016/j.cpc.2021.108008.
- [2] C. Zhao, Y. Yu, O. J. Haidn, and X. Hu, "Physics-driven complex relaxation for multi-body systems of SPH method," Sep. 26, 2024, *arXiv:2409.17795*. doi: 10.48550/arXiv.2409.17795.
- [3] L. Fu and Z. Ji, "An optimal particle setup method with Centroidal Voronoi Particle dynamics," *Computer Physics Communications*, vol. 234, pp. 72–92, Jan. 2019, doi: 10.1016/j.cpc.2018.08.002.
- [4] L. Vela Vela, R. Sanchez, and J. Geiger, "ALARIC: An algorithm for constructing arbitrarily complex initial density distributions with low particle noise for SPH/SPMHD applications," *Computer Physics Communications*, vol. 224, pp. 186–197, Mar. 2018, doi: 10.1016/j.cpc.2017.10.017.
- [5] K. Van Der Sande and B. Fornberg, "Fast Variable Density 3-D Node Generation", *SIAM J. Sci. Comput.*, vol. 43, no. 1, pp. A242–A257, Jan. 2021, doi: 10.1137/20M1337016.
- [6] S. J. Lind, R. Xu, P. K. Stansby, and B. D. Rogers, "Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves", *Journal of Computational Physics*, vol. 231, no. 4, pp. 1499–1523, Feb. 2012, doi: 10.1016/j.jcp.2011.10.027.
- [7] Y. Zhu, C. Zhang, Y. Yu, and X. Hu, "A CAD-compatible body-fitted particle generator for arbitrarily complex geometry and its application to wave-structure interaction", *J Hydrodyn*, vol. 33, no. 2, pp. 195–206, Apr. 2021, doi: 10.1007/s42241-021-0031-y.
- [8] P.-N. Sun, D. Le Touzé, G. Oger, and A.-M. Zhang, "An accurate SPH Volume Adaptive Scheme for modeling strongly-compressible multiphase flows. Part 1: Numerical scheme and validations with basic 1D and 2D benchmarks", *Journal of Computational Physics*, vol. 426, p. 109937, Feb. 2021, doi: 10.1016/j.jcp.2020.109937.
- [9] A. Haftu, A. Muta, and P. Ramachandran, "Parallel adaptive weakly-compressible SPH for complex moving geometries", *Computer Physics Communications*, vol. 277, p. 108377, Aug. 2022, doi: 10.1016/j.cpc.2022.108377.
- [10] X. Yang and S.-C. Kong, "Adaptive resolution for multiphase smoothed particle hydrodynamics", *Computer Physics Communications*, vol. 239, pp. 112–125, Jun. 2019, doi: 10.1016/j.cpc.2019.01.002.
- [11] S. S. Prasanna Kumar and B. S. V. Patnaik, "A multimass correction for multicomponent fluid flow simulation using smoothed particle hydrodynamics", *International Journal for Numerical Methods in Engineering*, vol. 113, no. 13, pp. 1929–1949, Mar. 2018, doi: 10.1002/nme.5727.
- [12] A. Haftu, A. Muta, and P. Ramachandran, "Parallel adaptive weakly-compressible SPH for complex moving geometries," *Computer Physics Communications*, vol. 277, p. 108377, Aug. 2022, doi: 10.1016/j.cpc.2022.108377.