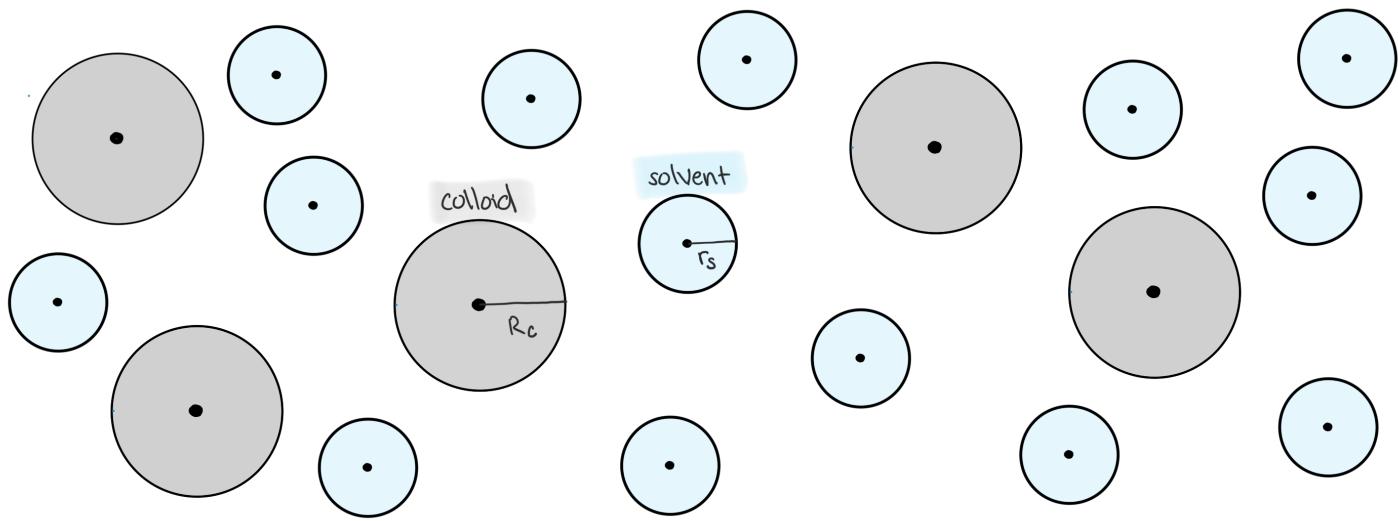


# DPD Colloid Simulations

## TWO TYPES OF PARTICLES

We use DPD to simulate colloid particles in a solvent (such as water) by defining at least two types of DPD particles: one representing individual colloid particles (roughly in the nano-to-micrometer range) and one representing regions of solvent. For a basic colloid simulation both types of particles are spheres and the number of particles is chosen to match the desired colloid volume fraction ( $\phi$ ).



This quickly becomes more complicated than simulating just one type of particle (i.e. water alone) because we now have 2 particle sizes ( $r_s$  and  $R_c$ ), 3 types of interactions

- solvent - solvent particle interactions ( $r_s = r_s$ )
- colloid - solvent particle interactions ( $r_s \neq R_c$ )
- colloid - colloid particle interactions ( $R_c = R_c$ )

and we want to add additional DPD forces to correctly model colloid behavior.

## SYSTEM UNITS

As discussed in the accompanying document on the basics of DPD, the simulation runs in DPD units, which are meaningless in the real world. We can convert DPD units to real-world units for specific systems (as we showed for a DPD simulation of water), but we usually want our colloid simulations to be generalizable to more than one real-world colloid system. Therefore we nondimensionalize our system as much as possible and scale units against a generalized experimental parameter, such as scaling distance in terms of particle radius, or scaling time in terms of particle diffusion time. We should always report data normalized by these reference values.

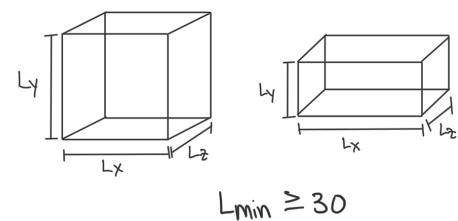
For a colloid simulation we need to set:

- simulation box size ( $L_x, L_y, L_z$ )
- colloid particle volume fraction ( $\phi$ )
- the number density ( $\rho$ )
- solvent particle radius ( $r_s$ ), volume ( $V_s$ ), and mass ( $m_s$ )
- colloid particle radius ( $R_c$ ), volume ( $V_c$ ), and mass ( $m_c$ )
- the cut-off radius ( $r_{cut}$ )
- the system temperature ( $k_B T$ )
- viscous dissipation parameter ( $\gamma_{ij}$ ) and background viscosity ( $\eta_0$ )
- parameters for all the interparticle forces
  - the repulsion parameter ( $a_{ij}$ ) for conservative force
  - parameters for other forces ( $D_0, \kappa, r_0, f_{\text{contact}}, \text{ alternate cut-off radii, etc.}$ )
- simulation timestep ( $\delta t$ ) AKA integration timestep
- recording timestep (AKA trigger or period)
- length of the simulation (the total number of timesteps)

## SIMULATION SIZE

The simulation box can be fixed (with different numbers of colloid particles added to achieve different volume fractions); OR, the total number of colloid particles can be fixed (with the volume of the simulation box changing to achieve different volume fractions). In either case:

- the total number of particles of EACH TYPE must be large enough for the results to be statistically significant
- the simulation box (volume  $V = L_x L_y L_z$ ) does not need to be a cube, but the smallest side length should be  $\geq 30$



## VOLUME FRACTION

Colloid volume fraction is usually low-to-moderate, 5% - 40% (unless you are specifically simulating dense suspensions). A good standard, go-to value is  $\phi = 0.2$  (i.e. 20%).

## NUMBER DENSITY

We usually use the same number density we used to simulate water:  $\rho = 3$

## SOLVENTS

Like when simulating a solvent (i.e. water) alone, we define a solvent particle as 1 DPD bead with a diameter of 1 DPD unit ( $r_s = 0.5$ ), spherical volume ( $V_s = \frac{4}{3}\pi r_s^3$ ), and mass of 1 DPD unit ( $m_s = 1$ ). The number of solvents is calculated by multiplying their total volume by the number density.

Although the solvent particles are given a volume during the simulation creation step, during the force calculations we treat them as point particles (i.e. momentum carriers only). You will notice during the force calculations that we locally redefine  $r_s = 0$  for those steps.

## COLLOIDS

The size of a colloid typically starts at twice the size of a solvent particle : radius  $R_c = 1$  and volume  $V = \frac{4}{3}\pi R_c^3$  ( $\sim 8V_s$  for  $R_c = 1$ ,  $r_s = 0.5$ ). The colloid particle mass is calculated from the particle volume and number density :  $m_c = V_c \rho$

## CUT-OFF RADIUS

The standard value for cut-off radius is  $r_{cut} = 1$ . This was chosen by running a solvent particle simulation and looking at the Radial Distribution Function (RDF)  $g(r)$  to find the distance from the center of a solvent particle where there is the highest probability of finding another solvent particle.

## TEMPERATURE ( $k_B T$ )

There are standard values commonly used for  $k_B T$  in the literature, depending on the system being studied. The main considerations are Brownian (thermal) motion and its effects on the simulation timestep.

A very low temperature ( $k_B T = 0.000001$ ) provides so little thermal motion that there is almost no particle interaction (and the simulation can go very long timesteps with almost nothing happening); while a very high temperature ( $k_B T = 10,000$ ) makes the Brownian forces so large (causing so much wild motion of the particles as Brownian motion dominates their behavior) that you will not be able to see the interactions between particles unless the timestep is extremely small (and simulation run time is very long).

You can check the literature for the most commonly used temperature values for different systems:

- for water simulations we used  $k_B T = 1$  (you can go up to  $k_B T = 10$  if you decrease the timestep).
- for colloid simulations people usually use  $0.05 \leq k_B T \leq 0.1$  (and we usually use  $k_B T = 0.1$ )

## VISCOSITY

The two viscosity parameters  $\eta_0$  (background viscosity) and  $\gamma_{ij}$  (the viscous dissipation scaling parameter) are linked.

- For an equilibrium simulation (no applied flow)  $\eta_0 = 0.3$  and  $\gamma_{ij} = 4.5$  are sufficient for physical behavior and keep simulation run time low
- For a simulation under flow usually  $\gamma_{ij} \geq 45$  is needed for physical behavior, corresponding to  $\eta_0 = 1.1$

It is IMPORTANT to use the same viscosity parameters for ALL simulations of a given system. You CAN NOT form an equilibrium gel at  $\eta_0 = 0.3$   $\gamma_{ij} = 4.5$  and then shear it at  $\eta_0 = 1.1$   $\gamma_{ij} = 45$ .

The viscous dissipation parameter ( $\gamma_{ij}$ ) has a limited range, around  $4.5 \leq \gamma_{ij} \leq 200$ . At very low values of  $\gamma_{ij}$  there is so little dissipation (e.g. friction) that the temperature can not be kept stable and blows up to large values. Similarly, because  $\gamma_{ij}$  and  $\sigma_i$  (the random force scaling parameter) are linked, at very high  $\gamma_{ij}$  the random force becomes so large that temperature again blows up to large values.

The relationship between  $\gamma_{ij}$  and  $\eta_0$  is almost logarithmic, but it is difficult to exactly calculate. Instead, if you want to find the viscosity ( $\eta_0$ ) for a new system at a given temperature you can either:

- For a system with a Newtonian solvent:  
run a shear simulation of the solvent only (~10,000 particles, 10 million timesteps, and a shear rate  $\sim 0.5$ ), and since the solvent is Newtonian you can easily calculate the viscosity from the shear stress
- Without shear:  
Calculate the stress autocorrelation function at quiescent condition and average over many initial states to reduce noise. The integral of the stress autocorrelation function will give you viscosity.

FORCE PARAMETERS are discussed in the section on interactions.

## TIMESTEP

The integration timestep  $\delta t$  is chosen based on the type of particle interactions you are simulating. The goal is to find a happy medium where the integration timestep is small enough that you see all the important interparticle interactions, but also large enough that your simulation doesn't take forever to run. This process is always a little bit fussy because real time is continuous, and discrete timesteps can cause nonphysical behavior. We typically start at  $\delta t = 0.001$  (AKA  $\delta t = 10^{-3}$ ) DPD times and adjust as needed.

## TRIGGER / PERIOD AND DATA OUTPUT

In addition to choosing the integration timestep you should choose the interval at which the simulation data is saved. To reduce simulation run time (and the size of output files) we often do not save every timestep as a frame in the simulation output file. The goal is to optimize most relevant data vs. fastest run time and smallest data files. Certain analysers will also require different precision (for example, to calculate the mean squared displacement (MSD) you will ideally have saved data at every timestep, or at least used a trigger smaller than the particle diffusion time)

The "best" trigger interval is often dynamic (so you get the most data when the most important interactions are happening and save space when they are not) but this requires the ability to adjust playback frame rate if you want to visualize results correctly.

## SIMULATION RUN TIME

The number of timesteps needed to run a simulation will vary with the behavior being simulated. You will need to run test simulations and analyze the results to determine if you ran it long enough.

For equilibrium simulations, the general rule is to run the simulation for 500 "diffusion times" to reach a (quasi) steady state, although the details of the simulation (volume fraction, strength of interparticle interactions, etc.) will cause it to vary slightly (e.g.  $\sim 400 - 1000$ ), so you still need to double check!

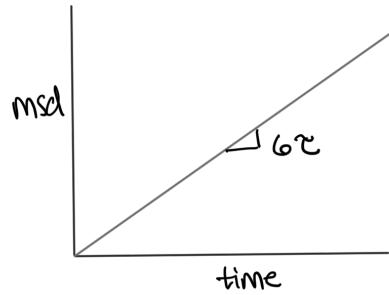
When we say "diffusion time" we are specifically referring to the diffusion time of a bare (single) particle in our chosen solvent - a generalized value that will make our simulations comparable to each other and to experiments.

We do not report simulation results in timesteps or DPD times, but in diffusion times:  $t^* = \text{DPD time (t)} / \text{diffusion time } (\tau)$

The diffusion time of a colloidal particle depends on the simulation temperature and the viscous dissipation parameter. We can estimate the diffusion time with the Stokes-Einstein equation

$$D = \tau = \frac{k_B T}{6\pi \eta_0 R_c}$$

For a new simulation system you should always compare this value with a baseline simulation of the colloid particle in solvent at 5% volume fraction ( $\phi=0.05$ ) simulated with neutral hard-particle colloid-colloid interactions (contact force only, no attraction or repulsion) and normal solvent-solvent and solvent-colloid interactions. Calculate the mean squared displacement (msd) and plot it on linear axes versus time. The slope will give you  $6D$  (AKA  $6\tau$ ).



### A NOTE ON SIMULATION COST

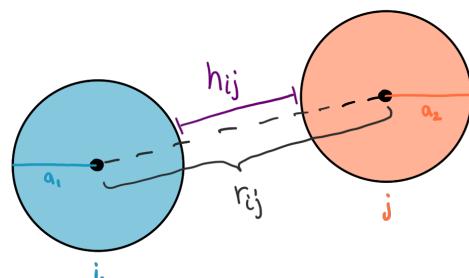
Viscosity, temperature, timestep, and the type of behavior you are simulating all affect total simulation run time, but the biggest factor is the number of particles. Less than 0.5% of the simulation cost is the colloid particles. With particle radii scaled 2:1, roughly EIGHT solvent particles take up the same amount of volume as ONE colloid particle. This means that at moderate volume fractions with 1000 colloid particles you can easily have ~1 million solvent particles! And large simulations at low volume fractions are more computationally expensive and take longer to run.

## DISTANCE MODIFICATION

Before we calculate our interparticle interactions we need to account for the fact that we now have particles of different sizes. This requires an update to DPD's standard center-center distance calculation for particle interaction.

The easiest way to account for particle radius is to convert from center-center distance to surface-surface distance

$$h_{ij} = r_{ij} - (a_i + a_j)$$



this now accounts for any particle radius  $r_p = a_p$  (i.e. colloid-solvent, colloid-colloid, or two colloids of different radii)

We will use  $h_{ij}$  for all our particle interaction calculations, but there will still be times that we input the center-center distance from our simulation script to correctly match the type of distance our built-in DPD functions expect. These are explained in detail later.

## CHOICE OF INTERACTIONS

The main changes to our DPD simulation come from the choice of colloid particle interactions. To do this we add at least 3 new forces:

While DPD is designed for soft-particles, which work great for our coarse-grained regions of solvent, we want our colloids to be hard-particles, which requires the addition of a contact force ( $F_{ij}^{\text{con}}$ ).

We also want to add an alternative particle interaction potential ( $F_{ij}^I$ ), other than conservative force, to give us more options for types of attraction, repulsion (or both) between colloid particles.

And finally, we want to add additional forces ( $F_{ij}^H$ ) to fully capture the hydrodynamics of the colloid behavior.

All this takes of from a "standard" set of DPD forces:

$$\mathbf{F}_i = m_i \frac{d\mathbf{v}_i}{dt} = \sum \mathbf{F}_{ij}^C + \mathbf{F}_{ij}^\Delta + \mathbf{F}_{ij}^R$$

to collid DPD:

$$\mathbf{F}_i = m_i \frac{d\mathbf{v}_i}{dt} = \sum \mathbf{F}_{ij}^C + \mathbf{F}_{ij}^\Delta + \mathbf{F}_{ij}^R + \mathbf{F}_{ij}^I + \mathbf{F}_{ij}^H + \mathbf{F}_{ij}^{\text{CON}}$$

If the simulation package you are using does not have a built in option for adding new forces to the DPD calculation, then you will need to make modifications to its source code, which is what we did with HOOMD-blue.

## CONTACT FORCE

Because DPD moves particles in discrete timesteps by the center of their volume there will often be non-physical particle overlaps that need to be resolved at the start of each timestep.

For a soft particle (like a standard DPD particle), if the timestep is kept small enough then these overlaps never exceed the realistic, physical deformation of the particles, which is why there is no contact force in standard DPD.

To make our colloids behave like hard particles (similar to a Lennard-Jones potential in an MD simulation) we add a contact force to resolve overlaps (and prevent deformation of these particles in our simulations).

particles approach each other	overlap occurs with discrete timestep	apply contact force to resolve overlap	continue simulation
$t = 1$		$t_2$	

In a Lennard-Jones hard potential there is a finite force between two particles that goes to infinity when the particles overlap. In our simulations we can't use infinite numbers (it will crash the code) so the best we can do is a semi-hard potential by adding contact force.

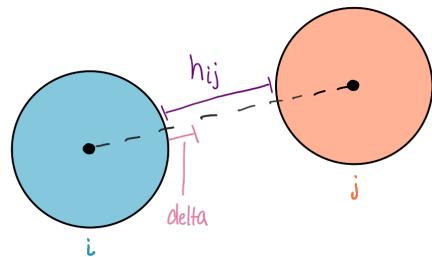
THIS IS A NON-PHYSICAL FORCE. It has no real world counterpart and only exists because of the way our simulations work.

The goal is to make the contact force as large as possible (to quickly resolve overlaps) without crashing the simulation.

We use two calculations for contact force to ensure a smooth transition between overlap and non-contact situations. (as published in Jamali & Boromand 2015). The transition between the two types of contact occurs at a small distance,  $\delta$ , that is comparable to the surface roughness of a colloidal particle.

The softness/hardness of a colloid particle is tuned by a scaling parameter called the contact modulus  $f_{\text{contact}}$  (with real units  $f_{\text{contact}}(k_B T / \delta^3)$ )

$$F_{ij}^{\text{CON}} = \begin{cases} f_{\text{contact}}(1 + h_{ij}) & , h_{ij} \leq 0 \\ f_{\text{contact}}(1 - \frac{h_{ij}}{\delta})^3 & , 0 \leq h_{ij} \leq \delta \\ 0 & , h_{ij} > \delta \end{cases}$$



Our simulations are mesoscale (e.g. primarily in the micrometer range, not the nanoscale), so we typically set  $\delta = 0.001$  (i.e. for a colloid particle  $\sim 1 \mu\text{m}$ ,  $\delta \sim 1 \text{ nm}$ , around the lower limit of our simulations' length scale, where the physics starts to change anyway).

Through trial and error the best contact modulus for simulating hard colloid particles was found to be  $f_{\text{contact}} = 10,000 \text{ } k_B T / r_{\text{cut}}$  although you may choose to adjust it for different simulations.

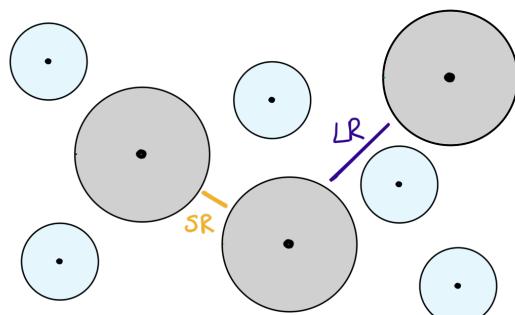
## PARTICLE INTERACTIONS

You can use any type of particle interaction potential to calculate the forces between interacting colloids. Common options are Morse Potential or Asakura-Oosawa Potential.

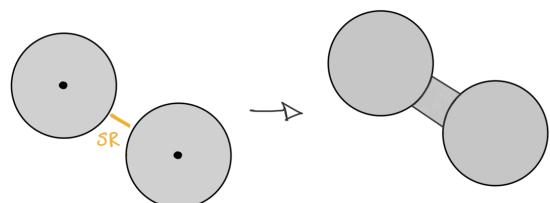
We typically simulate neutrally interacting colloid particles that become attractive due to the addition of a polymer to the solvent and the resulting local changes in polymer concentration (polymer depletion). For these systems we prefer the Morse Potential, which is described in an accompanying document.

## HYDRODYNAMICS

One of the hallmarks of DPD simulations is that the combination of dissipative and random forces conserves momentum in the system both locally and globally. This also ensures that the colloid-solvent interactions effectively preserve the correct multi-body and long-range hydrodynamic (LR) effects on the colloid particles; however, for complete hydrodynamics we need to add an approximation of the short-range lubrication (SR) effects in cases where a solvent particle does not fit between two colloid particles.



If we consider two interacting particles as one dumbbell, (for more details, see the paper: Ball and Melrose (1997) "A simulation technique for many spheres in quasi-static motion under frame-invariant pair drag and Brownian Forces" Physica A: Stat Mech App )



then the angular rotation of the dumbbell can have four internal modes: squeezing, (tangential) shear, pumping, and twisting.

For close spheres at low Reynolds number (such as in our case) you can approximate these forces by using the coefficients from lubrication theory:  $a_{sq}$ ,  $a_{sh}$ ,  $a_{pu}$ ,  $a_{tw}$

Since we do not track rotation of our particles (since they are spherical and isotropic), the main two modes to consider are squeezing mode lubrication and tangential shear lubrication. (and we will see that pumping and twisting will also be negligible anyway).

As derived in Appendix A of Ball and Melrose, squeezing mode lubrication occurs in the normal direction and diverges with inverse surface-surface interparticle distance when two particles contact each other ( $h_{ij} = 0$ )

$$a_{sq} \propto \frac{1}{h_{ij}}$$

Tangential shear lubrication exists at the contact point between two particles, but it diverges with the logarithm of the surface-surface interparticle separation distance.

$$a_{sh} \propto \log\left(\frac{1}{h_{ij}}\right)$$

making it's contribution significantly smaller than the squeezing mode lubrication. This is also true for pumping and twisting modes:

$$a_{sh} \propto a_{pu} \propto a_{tw} \propto \log\left(\frac{1}{h_{ij}}\right)$$

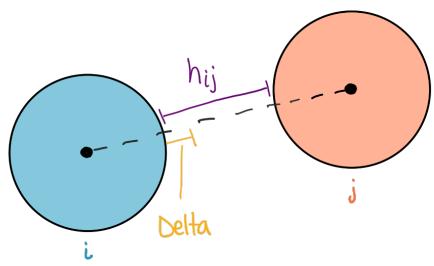
Past research has verified that ignoring the contributions from tangential shear to the short range (close gap) hydrodynamics of particle suspensions does not significantly effect the behavior of the system, and as mentioned above we are also not interested in the other modes in this simulation, so we only need to add squeezing mode lubrication.

$$\mathbf{F}_{ij}^H = \mathbf{F}_{ij}^{sq}$$

The squeezing lubrication is based on a pair drag term ( $\mu_{ij}^H$ ) equal to Ball and Melroses squeezing coefficient:

$$a_{sq} = \mu_{ij}^H = \frac{3\pi\eta_0 R_c^2}{2h_{ij}}$$

Similar to what we did with contact force, we introduce a gap distance Delta to ensure a smooth transition from interparticle contact to short-range squeezing lubrication, and then the long-range (colloid-solvent) hydrodynamics. We typically also set this Delta = 0.001 for the same reasons as when setting the delta in contact force: our simulations are mesoscale (e.g. primarily in the micrometer range), and for a colloid particle  $\sim 1\mu\text{m}$ , this puts Delta  $\sim 1\text{nm}$ , near the lower limit of our simulations' length scale, where the physics starts to change anyway.



Generalizing  $a_{sq}$  for any two particles (radius  $a_i$  and  $a_j$ ) and using the Delta, we calculate hydrodynamic force as:

$$\mathbf{F}_{ij}^H = -a_{sq} (\mathbf{v}_{ij} \cdot \mathbf{e}_{ij}) \mathbf{e}_{ij}$$

where  $a_{sq}$  is defined as

$$a_{sq} = M_{ij}^H = \begin{cases} \frac{3\pi\eta_0}{2\Delta} \left( \frac{a_i + a_j}{2} \right)^2 & , 0 < h_{ij} \leq \Delta \\ \frac{3\pi\eta_0}{2h_{ij}} \left( \frac{a_i + a_j}{2} \right)^2 & , h_{ij} > 0 \end{cases}$$

Finally, Ball and Melrose use the squeezing mode lubrication coefficient as a gamma-like parameter to calculate the random force controlling parameter ( $\gamma_{ij}$ ) during the simulation (as shown in their Appendix B):

$$\sigma_{ij} = \sqrt{2 a_{sq} k_B T}$$

So, we will also update our random force calculation when the additional hydrodynamic force ( $\mathbf{F}_{ij}^H$ ) is being used:

$$\sigma_{ij} = \sqrt{2 \gamma_{ij} k_B T} \rightarrow \sigma_{ij} = \sqrt{2 (a_{sq} + \gamma_{ij}) k_B T}$$

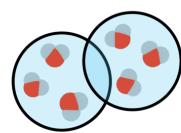
## CALCULATING INTERACTIONS

### SOLVENT-SOLVENT

Solvent-solvent interactions are calculated with the same forces as standard DPD particle interactions. There are no additional particle interaction effects or hydrodynamic effects that need to be considered, and the solvent particles are soft (with repulsion from conservative force) and do not need an added contact force.

$$\mathbf{F}_{ss_i} = m_i \frac{d\mathbf{v}_i}{dt} = \sum \mathbf{F}_{ij}^c + \mathbf{F}_{ij}^d + \mathbf{F}_{ij}^r + \mathbf{F}_{ij}^{Z^0} + \mathbf{F}_{ij}^{K^0} + \mathbf{F}_{ij}^{CON}$$

The soft repulsive conservative force translates to soft, slightly deformable solvent particles. Remember that our solvent particles represent a region of space containing 3  $\text{H}_2\text{O}$  molecules, so the deformation of a solvent particle can be thought of as the overlap of empty space between molecules.



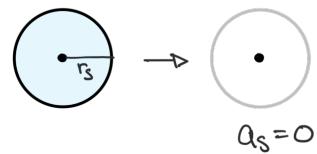
In the simulation script we scale the conservative force parameter ( $a_{ij}$ ) by temperature and cut-off radius:

$$a_{ij} = 25 k_B T / r_{cut}$$

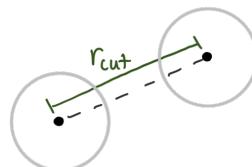
and use the chosen value for the viscous dissipation parameter ( $\gamma_{ij}$ )

Some potential well parameters ( $\kappa$  and  $r_0$ ) can be left set to their standard values, but to ensure other interactions are not calculated, the remaining parameters (potential well parameter  $D_0$ , background viscosity  $\eta_0$ , the contact modulus  $f_{contact}$ ) should all be set to zero.

Although we defined solvent particles with a radius ( $r_s$ ) when we calculate the interparticle forces we will treat them as point particles (momentum carriers only), so we need to set the internally called particle radii to zero ( $a_1 = a_2 = 0$ ).



This effectively makes the surface-surface distance equal the center-center distance ( $h_{ij} = r_{ij} - (0+0)$ ), and the cut-off radius used in the force calculation step and the construction of the neighboring list are the same, the base value  $r_{cut} = 1$



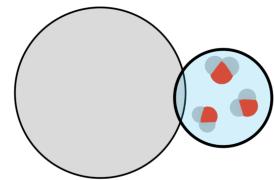
## COLLOID - SOLVENT

Colloid-solvent interactions represent two types of  $ij$ -pairs: colloid ( $i$ ) - solvent ( $j$ ) and solvent ( $i$ ) - colloid ( $j$ ). Both of these will function the same way.

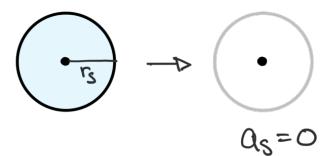
Colloid-solvent interactions are also calculated with the same three forces as standard DPD particle interactions (there are no additional particle interaction effects or hydrodynamic effects that need to be considered between these particle types, and because the solvent particles are soft, no contact force is needed).

$$\mathbf{F}_{cs_i} = m_i \frac{d\mathbf{v}_i}{dt} = \sum \mathbf{F}_{ij}^c + \mathbf{F}_{ij}^d + \mathbf{F}_{ij}^r + \mathbf{F}_{ij}^{z^0} + \mathbf{F}_{ij}^{h^0} + \mathbf{F}_{ij}^{con}$$

Although we will define our colloid particles as hard spheres that do not deform, (with an added contact force) the soft repulsion of the conservative force still correctly simulates colloid-solvent interactions due to the soft deformability of the solvent particles.



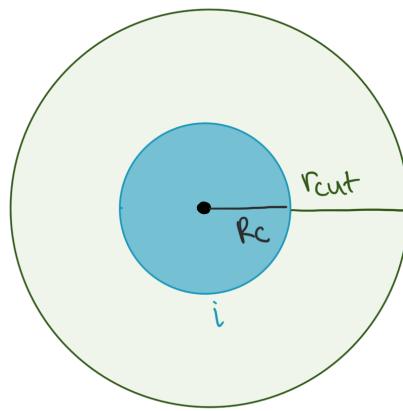
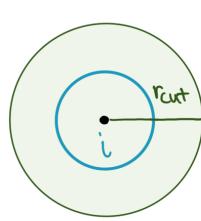
In the simulation script we will again treat the solvent particles as point particles for the force calculation steps, but this time we also need to make two additional distance modifications to the cut-off radius:



- we need to recalculate the cut-off radius to correctly balance the colloid-solvent interactions during the force calculation step, where we will input  $r_{cut}$  as a surface-surface distance
- we need to explicitly convert the value of  $r_{cut}$  used when calculating the neighboring list as a center-center distance

We did not need to do this for solvent-solvent interactions because the solvent particles are point particles of the same size (and both values for the cut-off radius were the same, the original  $r_{cut}=1$ )

For colloid-solvent particles, we need to change the cut-off radius to keep colloid-solvent interactions correctly balanced with the solvent-solvent and colloid-colloid interactions. Because we converted from center-center distance to surface-surface distance calculations, using the same cut-off distance for a (smaller) solvent particle and a (larger) colloid particle covers a dramatically different volume when calculating particle interactions.



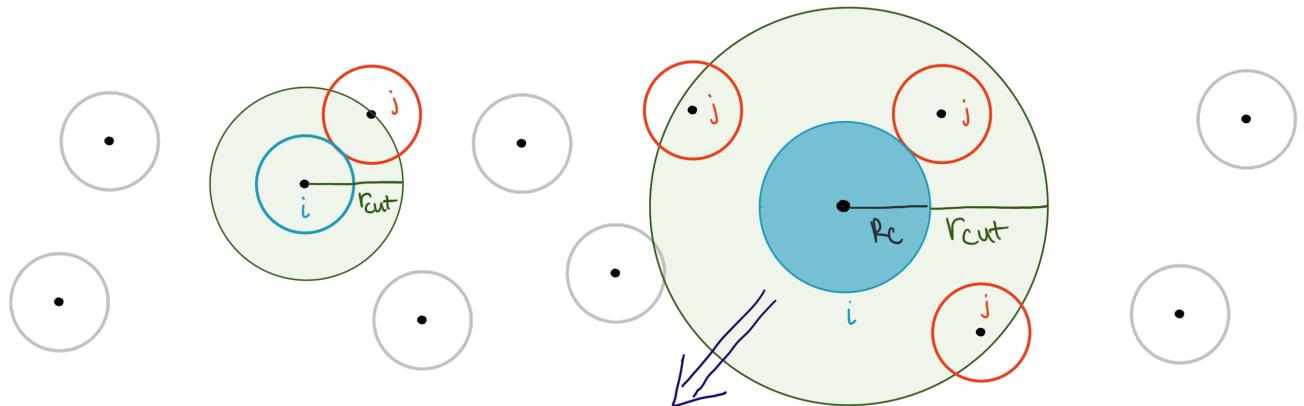
$$V_{ss} = \frac{4}{3}\pi [r_s + r_{cut}]^3 - \frac{4}{3}\pi r_s^3$$

$$V_{cs} = \frac{4}{3}\pi [r_c + r_{cut}]^3 - \frac{4}{3}\pi r_c^3$$

Ex]  $r_s = 0, r_{cut} = 1 \Rightarrow V = 4.19$

$r_c = 1, r_{cut} = 1 \Rightarrow V = 29.3$

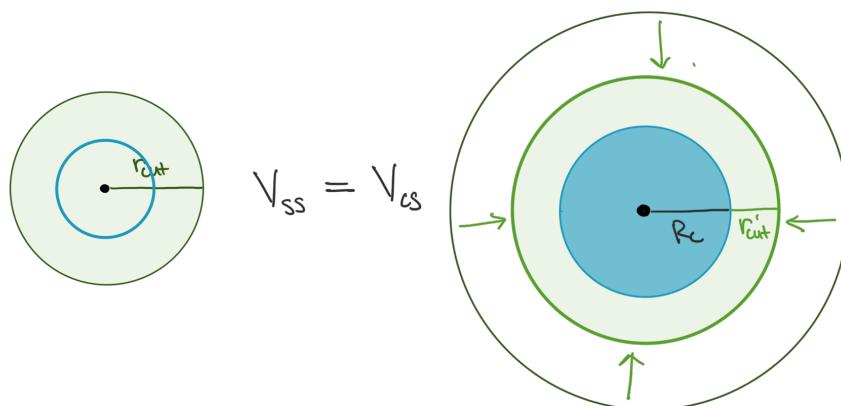
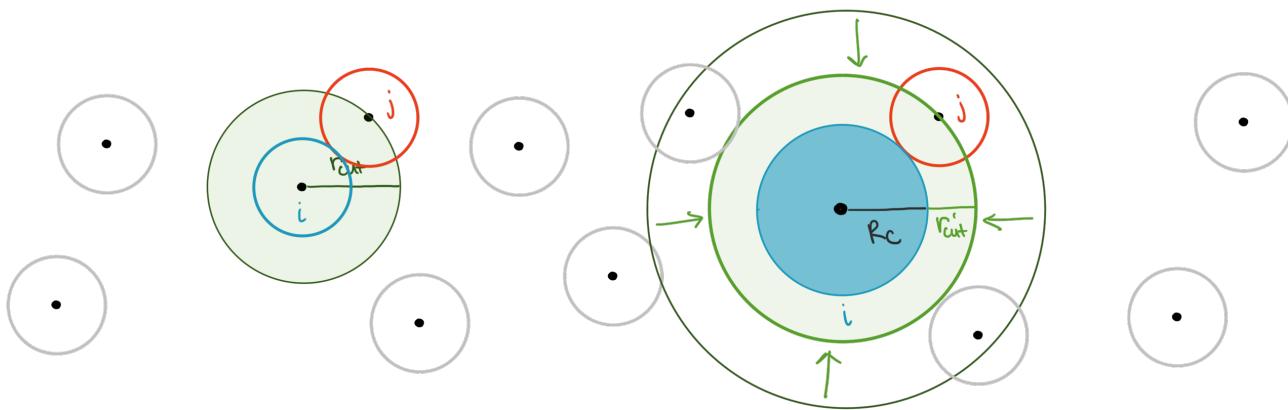
Because the colloid interaction volume is so much larger than the solvent interaction volume, a single colloid particle ends up forming many more interacting  $i_j$ -pairs with neighboring solvent particles.



If the colloid-solvent pairs are clustered together on one side of the colloid particle, this will create an artificially large conservative (repulsive) force on the colloid particle, pushing it away from the solvents. Over the course of the this effect adds up to causing the entire system to artificially phase separate.

This type of phase separation is an artifact of our coarse graining of the solvent. In a real system the colloid is surrounded by tiny water molecules, not solvent particles, and the water molecules do not cluster randomly cluster together on one side of the colloid at a large enough percentage to induce this type of depletion effect.

To fix this we need to decrease the colloid - solvent interaction volume so that it is equal to the solvent - solvent interaction volume during the force calculation step (eliminating the depletion effect).



$$\frac{4}{3}\pi \left[ (r_s + r_{\text{cut}})^3 - r_s^3 \right] = \frac{4}{3}\pi \left[ (R_c + r'_{\text{cut}})^3 - R_c^3 \right]$$

for our solvent particles  $r_j = 0$ :

$$r_{\text{cut}}^3 = (R_c + r'_{\text{cut}})^3 - R_c^3$$

$$(R_c + r'_{\text{cut}}) = \sqrt[3]{r_{\text{cut}}^3 + R_c^3}$$

Because DPD is used to working with center-center distances, we will use input the center-center cut-off distance:  $(R_c + r'_{\text{cut}})$  for the built in neighboring list calculation

and input the surface-surface cut-off distance:  $(R_c + r'_{\text{cut}}) - (0 + R_c) = r'_{\text{cut}}$  for our modified interparticle force calculations.

for our original  $r_{\text{cut}} = 1$  and colloid particles  $R_c = 1$

$$(R_c + r_{\text{cut}}) = \sqrt[3]{1^3 + 1^3} \approx 1.25$$

In our old code you will see this center-center distance approximated as

$$(R_c + r_{\text{cut}}) = r_{\text{cut}} + 0.25 R_c$$

and the surface-surface distance approximated as

$$(R_c + r_{\text{cut}}) - (0 + R_c) = r_{\text{cut}} + 0.25 R_c - 1 = 1 - 0.75 R_c$$

however, this is not quite correct for other colloidal radii.

YOU SHOULD INSTEAD ALWAYS CALCULATE

center center :  $(R_c + r'_{\text{cut}}) = (r_{\text{cut}}^3 + R_c^3)^{1/3}$

surface-surface :  $(R_c + r'_{\text{cut}}) - (0 + R_c) = (r_{\text{cut}}^3 + R_c^3)^{1/3} - R_c$

Now that we have completed the colloid-solvent cut-off radius distance modification we can calculate the forces!

We scale the conservative force parameter ( $a_{ij}$ ) by temperature and the new surface-surface cut-off radius:

$$a_{ij} = 25 k_B T / ((R_c + r'_{\text{cut}}) - (0 + R_c))$$

and use the chosen value for the viscous dissipation parameter ( $\gamma_{ij}$ )

Some potential well parameters ( $\kappa$  and  $r_0$ ) can be left set to their standard values, but to ensure other interactions are not calculated, the remaining parameters (potential well parameter  $D_0$ , background viscosity  $\eta_0$ , the contact modulus  $f_{\text{contact}}$ ) should all be set to zero.

To make the solvent particle a point particle, we set one of the internally called particle radii to zero ( $a_1 = 0$ ) and the other radius to  $R_c$  ( $a_2 = R_c$ ).

We also need to make sure we input the new surface-surface cut-off radius for the force calculation step :  $r_{\text{cut}} = (R_c + r'_{\text{cut}}) - (0 + R_c)$

Finally, we set the cut-off radius for the neighboring list calculation to the new center-center cut-off radius  
 $r_{\text{cut}} = (R_c + r_{\text{cut}})$

### COLLOID - COLLOID

Colloid-colloid interactions are where all the added forces come in. We replace the DPD conservative force with the force from our chosen interaction potential, add a contact force only in the case when two colloid particles overlap (to create hard-particle interactions between our colloids), and add the short-range approximation of squeezing mode lubrication to complete the hydrodynamic forces. Dissipative and random force remain the same.

$$\mathbf{F}_{\text{cc}i} = m_i \frac{d\mathbf{v}_i}{dt} = \sum \mathbf{F}_{ij}^{\text{C}} + \mathbf{F}_{ij}^{\text{D}} + \mathbf{F}_{ij}^{\text{R}} + \mathbf{F}_{ij}^{\text{I}} + \mathbf{F}_{ij}^{\text{H}} + \mathbf{F}_{ij}^{\text{CON}}$$

In the simulation script we set the conservative force parameter to zero ( $a_{ij} = 0$ ) and the viscous dissipation parameter to the chosen value ( $\gamma_{ij}$ ).

The potential well parameters ( $D_0, \kappa, r_0$ ) and the contact modulus ( $f_{\text{contact}}$ ) are all set to their standard values.

Viscosity is set to the chosen background viscosity ( $\eta_0$ ), and the internally called particle radii are set to the colloid particle radius ( $a_1 = a_2 = R_c$ ).

The cut-off radius used to calculate the forces is the standard cut-off radius value ( $r_{\text{cut}}$ ), which here corresponds to a surface-surface distance

but, the cut-off radius used by the neighboring list needs to be adjusted to a center-centers distance that accounts for the size of the particles  
 $\rightarrow r_{\text{cut}} + a_1 + a_2$

