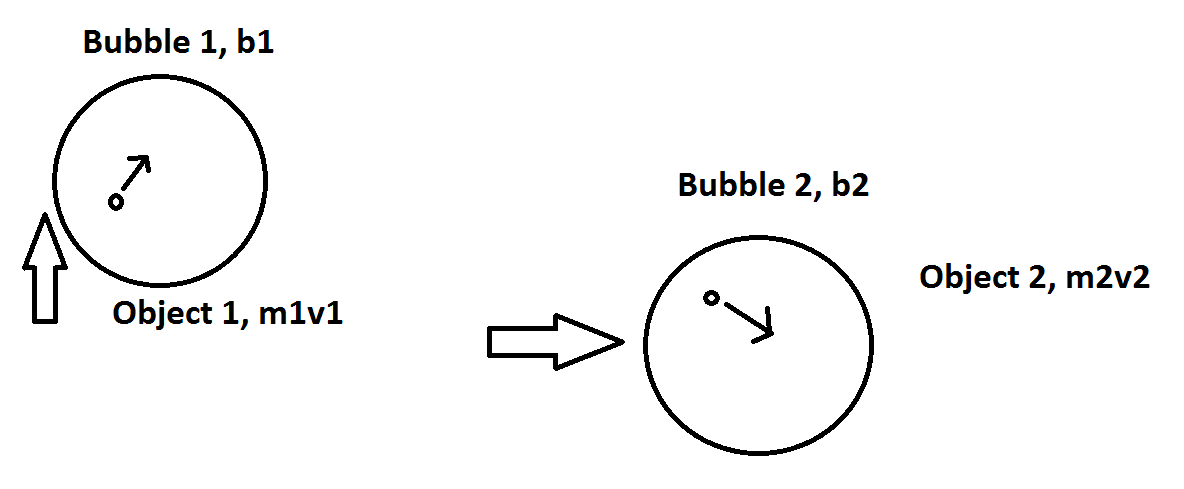
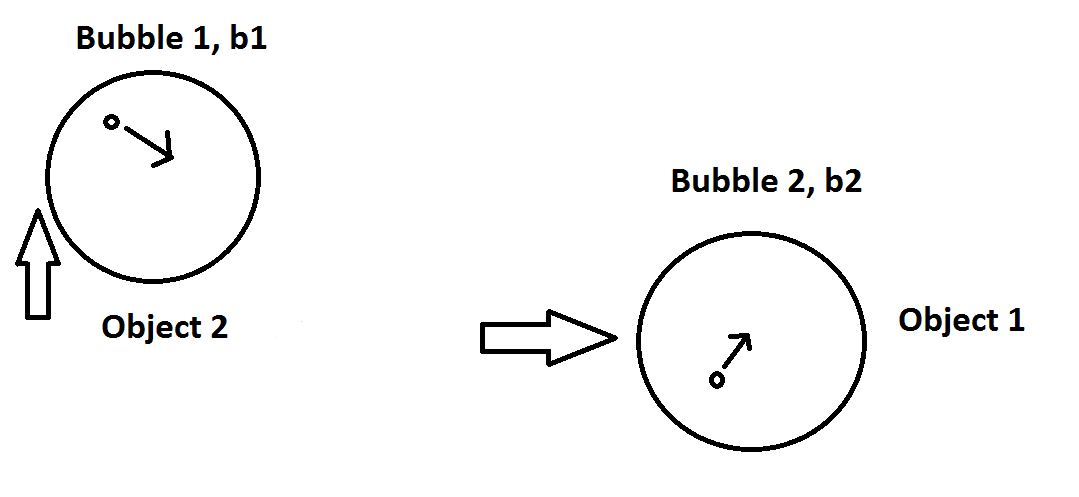
Pre-swap:



Object 1 has a momentum of m1v1 relative to bubble 1, while object 2 has a momentum of m2v2 relative to bubble 2, bubble 1 has a velocity b1, and bubble 2 has a velocity b2.

Assumptions: Momentum is conserved, swap is instantaneous, swap removes no energy from objects being swapped, bubbles themselves are massless, directions of motion are unaltered in each individual object’s reference frame, swap does not affect bubble velocities, and objects are swapped to the same point in the new bubble as they occupied in the old one copy-pasta style.

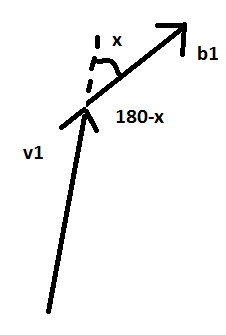
Post-swap:



Clearly, objects must retain their individual masses, but the jump between reference frames will impact their velocities if momentum is conserved. Below all velocity variables are vectors, while mass is a scalar.

Consider object 1. Pre-swap, Relative to the planet it has a momentum . After being swapped, is will have momentum . We want momentum to be conserved, thus we require that (the mass cancels out).

Now, since the information we probably have is the magnitude and direction of all these velocity vectors, I’ll skip past component-wise manipulations and instead handle magnitude combination of vectors with different directions.



Let v1 and b1 have a difference in lateral angle of and a difference in azimuthal angle of . We combine these angles into the overall angle difference between the two vectors: . It was important here that .

Then

Similarly, for the angles between s1 and b2 and

.

So we need to balance

Where s1 is the only unknown.

Isolating s1:

Then it is only a matter of writing a script that accepts the 7 unknowns and spits out the result. Keep in mind that the swap velocity s1 has the same direction as v1.

We use the same logic and equation for any projectile being swapped, redefining variables as necessary.