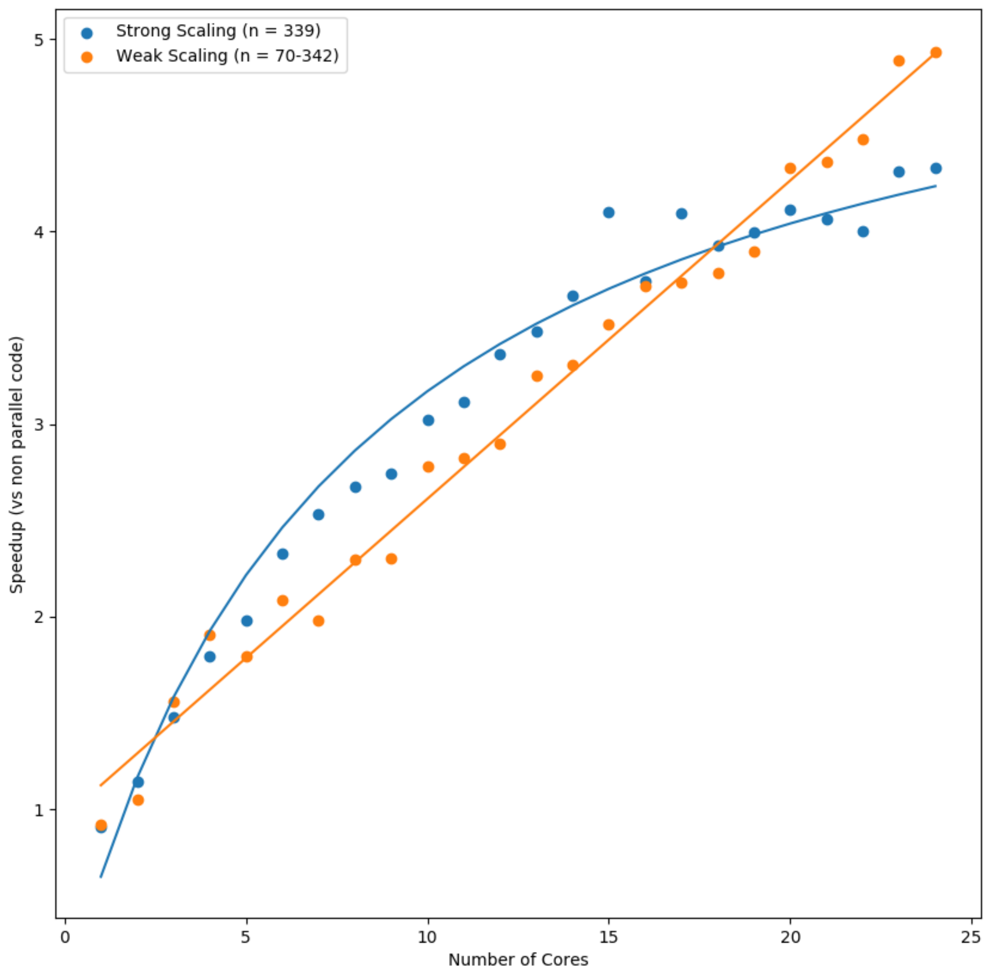
Comparing Strong Scaling and Weak Scaling

I compared two benchmarks for strong and weak scaling, running on Hamilton with between 1 and 24 cores, defining the problem size as n^2, where n is the number of bodies. The blue line shows the speedup for strong scaling with a fixed problem size. The curve follows Amdahls's law's expected shape of $S(p) = \frac{1}{f+(1-f)(1/p)}$ where f is the fraction of the runtime that has to take place serially and p is the number of cores.

The orange line shows the speedup for Gustafson's weak scaling. I scaled n^2 proportionally with the number of cores, meaning each core was given a fixed amount of work. The linear line of best fit follows the expected linear scaling where $S(p) = f + (1-f)p$.



Scipy's curve fit function can be used to estimate f from the strong scaling data = 0.22 for 343 bodies. Using the weak scaling data, we can estimate 1 - f from the gradient of the line of best fit = 0.17, and therefore f = 0.83.

For this simulation, we are usually dealing with a problem of fixed size, and therefore the strong scaling of the code is more relevant to analyse