



Figure 1: The speedup compared with the number of bodies in the simulation.

If we analyse Figure 1, we clearly see that the speedup of the parallelised implementation improves as we increase the number of bodies in the simulation. However, for all of the simulations except one (where number of bodies = 6000) the speedup is less than 1 ($S(p) < 1$), meaning the sequential code is running more quickly than the parallelised program. The reasoning behind this is that in our implementation the time spent performing the operations involved in each iteration is very small, and when we parallelise the loop containing these iterations we introduce a significant overhead due to the spawning and management of each thread. This overhead does not significantly increase when increasing the number of bodies, and thus as we scale the problem it has less of an impact - hence the positive gradient of the graph. Due to this overhead the parallel programming is only likely to increase speedup (and efficiency) when each iteration requires a sufficient amount of processor time or in our case when there are a sufficient number of iterations that need to be performed.

All of the speedups on the graph were acquired with ($p = 12$) processors. The best efficiency we achieved was clearly when number of bodies = 6000 and this was $E(p) \approx \frac{1.02}{12} = 0.085 \equiv 8.5\%$. This is a long shot from the $E(p) \approx 0.3939 \equiv 39.39\%$ (and speedup $S(p) = 26$) that was predicted in step 5, I am sure that if we involved more bodies then the speedup would improve and the efficiency would get closer to the predicted value. However, our method when predicting the speedup/efficiency ignores any overhead and thus our it is expected that the results our real-life simulations yield will fall short of the the theoretical predictions.

Should we increase the scale of the problem even further I believe the performance of the parallel program would continue to improve to an extent, and it would clearly run more quickly than the sequential program (as it already does marginally for 6000 bodies). However, from the decrease in gradient of the curve in Figure 1 I believe that there would come a point when the speed up approach a constant value i.e. $S(p) = k = \text{constant}$. At this point the parallelised program would still outperform the sequential program however I doubt that increasing the problem size would have any significant effect on the speedup. As a result, I don't believe the parallelisation would ever reach the efficiency or speedup that was predicted in the earlier step, but again due to the limitations of the prediction this is to be expected.