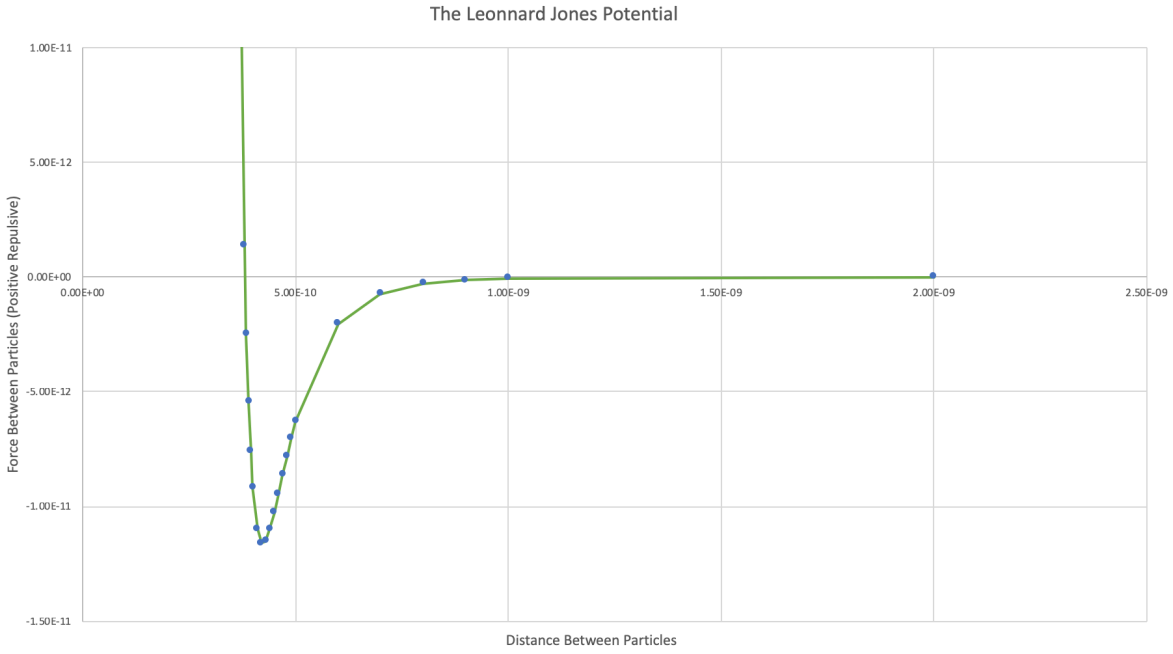
Solution Step 4 **Matt Ingram**

Initial Adaptive Time Stepping Scheme

I based my initial time stepping scheme around the hints given in the assignment and compared the computational efficiency of both by measuring the number of calls to the *updateBody* function. The adaptive time stepping code itself does introduce some extra computations, but these are negligible relative to main body of the function, so I ignored them. Using the convergence plot in Step 3, I decided that a minimum time step of 10^{-4} was sufficiently small to achieve a reasonable amount of accuracy. Had we used this value as a constant time step size, when running the simulation with the input parameters specified in Step 3, the program would require $\frac{10^4}{10^{-4}} = 10^8$ calls to the *updateBody* function. When running simulation with the adaptive time stepping scheme, and varying starting time steps, the results were as follows:

Starting Time Step Size (s)	Call Count to updateBody( )	Final Distance (m)
10	4212584	0.0000000062945712
1	4201286	0.0000000063836525
10^{-1}	4242294	0.0000000064114279
10^{-2}	4304456	0.0000000064377245
10^{-3}	4089109	0.0000000063362192
10^{-4}	4230939	0.0000000064071732

As we can see, even for this relatively simple scheme, the call count was reduced by $\frac{10^8 - 4200000}{10^8} \times 100 = 95\%$ consistently. It's also interesting to note that when this adaptive scheme was used, a much larger starting time step size of 10s can be used, and the algorithm will still converge. This led me to investigate the 'magic' parameter 1×10^{-9}, as I believed it was this factor that was which may have a significant impact upon the convergence of the algorithm. I plotted the formula for the Lennard Jones Potential (LJP) in Excel, as shown in the figure below.



The Lennard Jones Potential (LJP) is a simple numerical model that approximates the interaction between a neutral pair of atoms or molecules at extremely small distances, due to the effects of Pauli repulsion and the (attractive) van der Waals forces. It appears that at around 1×10^{-9}m of separation, the atoms first start to feel the van der Waals forces. As these forces quickly increase, we encounter a 'stiff' region. Our Explicit Euler method, combined with this adaptive time stepping scheme, would become highly inefficient if the time step size was allowed to decrease indefinitely, and so I believe this is why this 'magic' parameter was implemented.

Second Adaptive Time Stepping Scheme

I decided to try to implement a 'better' adaptive time stepping scheme, to try to further reduce the computations required, whilst still maintaining (and possibly increasing) a high level of accuracy. The scheme I tried to implement follows from theory in the lectures. We know that the discretization error in each time step is given by:

$$e(t+h) = y(t+h) - y_h(t+h) = y(t) + h \cdot y'(t) + \frac{h^2}{2} y''(t) + \cdots - \; y(t) - h \cdot y'(t)$$
$$\approx \frac{h^2}{2} y''(t)$$

and thus a good time stepping scheme should always keep this error bounded, such that:

$$\frac{h^2}{2} y''(t) < \epsilon \quad \Rightarrow \quad y''(t) < \sqrt{\frac{2\epsilon}{h^2}}$$

for some tolerance ϵ. Taking our $y(t) = x$ (the position of our particle) then we can quite easily find an approximation of the second derivative $y''(t) = \ddot{x}$, as $\ddot{x} = a = \frac{F}{m}$ by Newton's 2nd Law. Therefore, if we monitor the changes in our force, and decrease the time step sufficiently when the force changes drastically, we will be able to maintain a high level of accuracy. I attempted to implement this adaptive scheme, but even with significant experimentation I could not find a suitable value for ϵ that increased performance over the initial adaptive time stepping scheme.

Multiple Body Adaptive Time Stepping

When multiple bodies are introduced into the simulations, a new issue arises. With only a single, global, time step, we may run into a situation where the majority of the particles are isolated, with only relatively few interacting in a meaningful manner. In this instance, a large time step would be sufficient for the majority of the particles, yet a small time-step would still be applied globally, due to the minimum distance between any two particles being smaller than the given threshold ($\sim 5 \times 10^{-9}$). One possible solution to this problem may be to implement a scheme where each particle is allocated its own adaptive time step. Upon every positional update of any particle, the pairwise distances between the given particle and every other particle would be updated. At the end of the *updateBody* function, the minimum distance of every particle to any other particle would be calculated. Upon the next iteration, the same adaptive time stepping scheme is used as above, but on a per-particle basis: if the minimum distance between a given particle and all other particles is below the given threshold ($\sim 5 \times 10^{-9}$), then the minimum time step is used. Otherwise, the other checks are performed, and the time step for that particle is increased or decreased accordingly. Furthermore, if we were to implement this adaptive time-stepping scheme, we could add more sophisticated functionality. For example, we could implement a scheme where we increased the time step size significantly for particles once they had interacted and repelled from another, as the forces dramatically decrease again, and the problem becomes less stiff. Unfortunately I ran out of time to implement this scheme, but it would definitely be interesting to implement and analyse.