MATH96012 Project 2

Matthew Cowley 01059624

November 28, 2019

Part 2

As we can see from the graph as M gets large OMP parallelised performs the best with number of threads=4 giving only slightly better performance. This is probably because the computer does not actually have access to 4 true cores. This result is expected because the code has been parallelised across m. It is interesting that the python code outperforms the serial fortran code for large M. This might be due to the row-major of python or something working behind the scenes to improve performance. But for small m we see that fortran serial code is quicker.

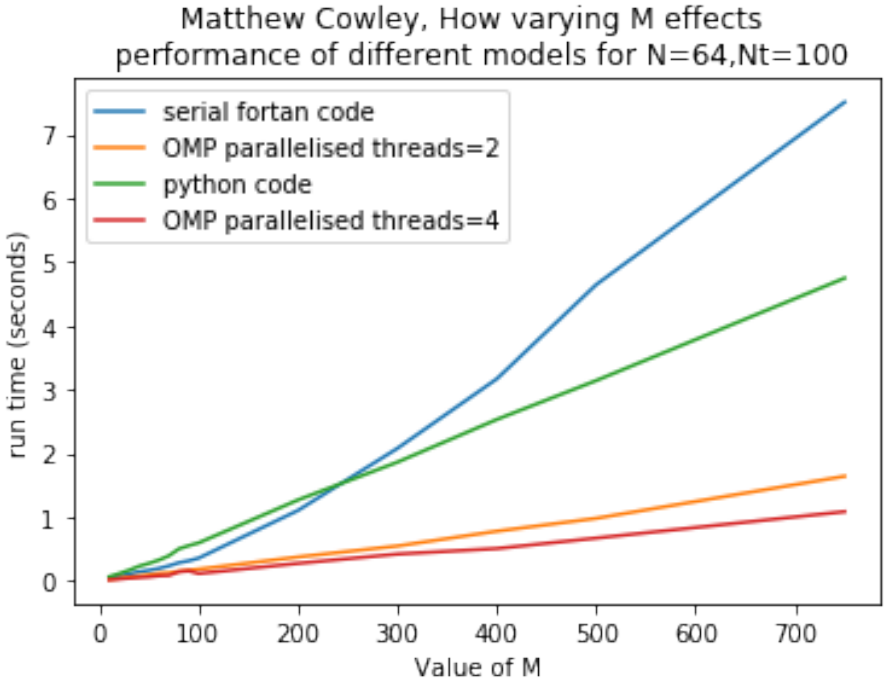

Figure 1: How varying m effects performance

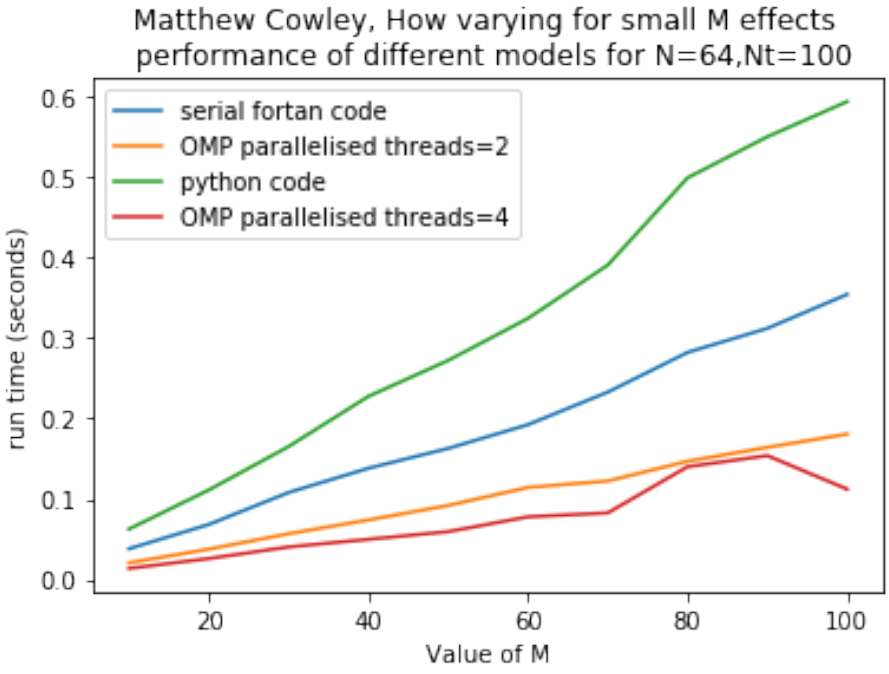

Figure 2: How varying small m effects performance

When we vary N the speed up is not as significant, but to be expected as we didn't parallise across N. Similar results follow when n and m are varyied.

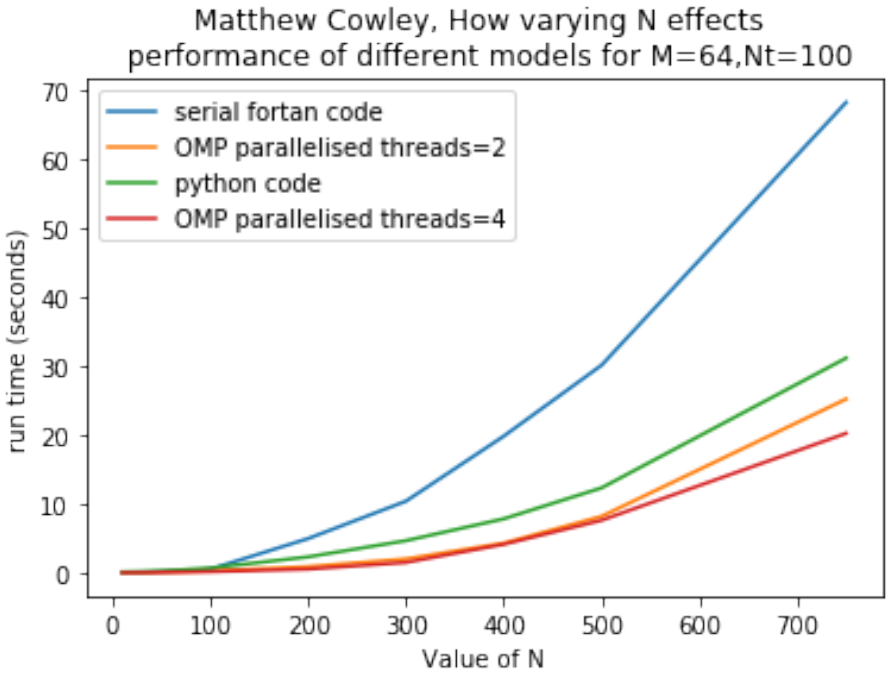

Figure 3:

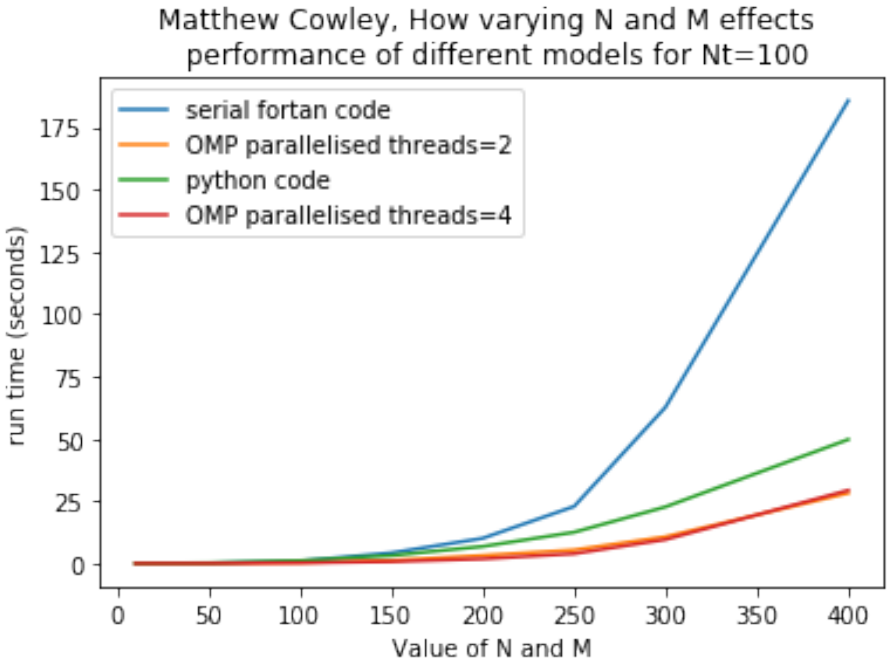

Figure 4:

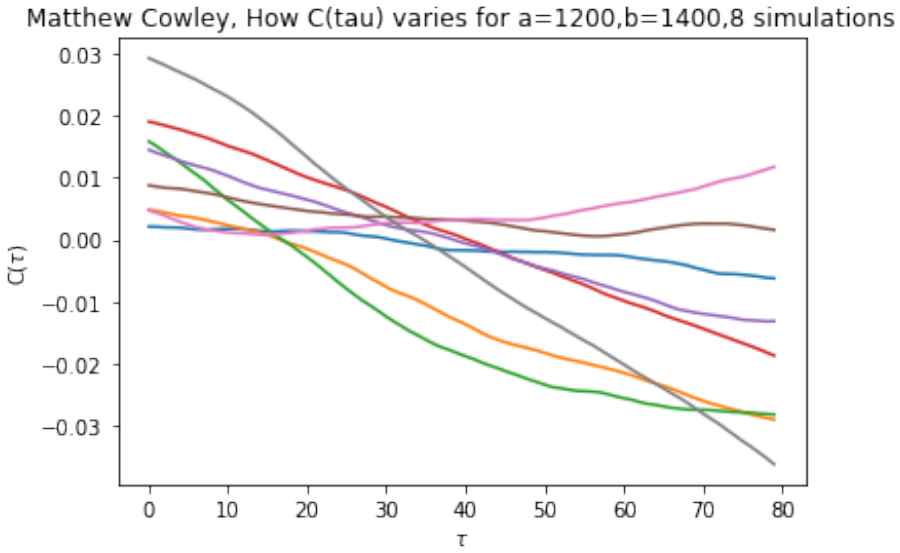

Figure 5:

Part 3

As b-a increases C(tau) appears to increase. But as C(tau) is very small in general. C(tau) also varys alot but when averaged does tend towards a single value.

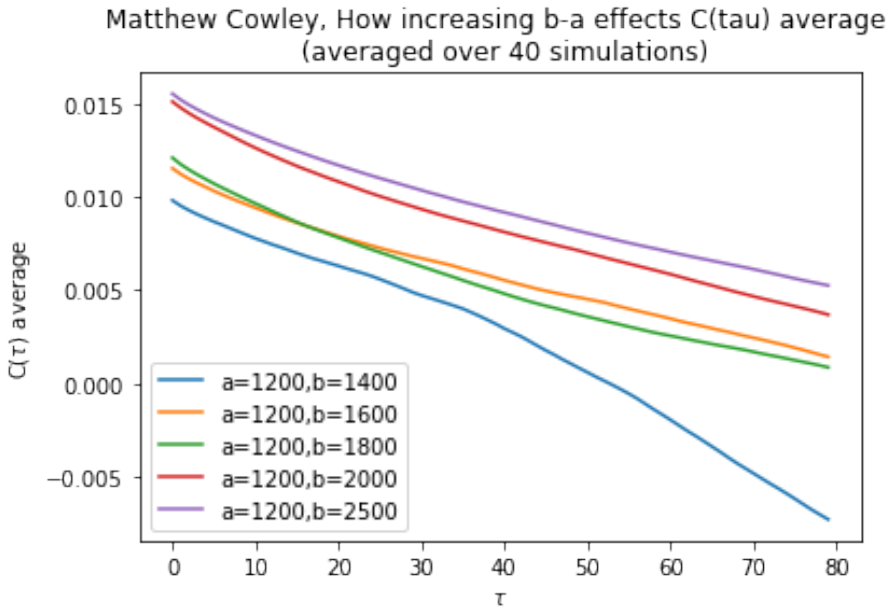
Matthew Cowley, How increasing b-a effects C(tau) average
(averaged over 40 simulations)
C(τ) average
0.015
0.010
0.005
0.000
−0.005
0
10
20
30
40
50
60
70
80
τ
a=1200,b=1400
a=1200,b=1600
a=1200,b=1800
a=1200,b=2000
a=1200,b=2500

Figure 6: