

邱名彰 100060007

HW01

Theoretically, algorithm 3 shall be the fastest one among the three algorithms and algorithm 1 be the slowest since algorithm 3 replace one vector division a real number division. In the meantime, algorithm 2 shall be faster than algorithm 1 for algorithm 2 does one less vector assignment and subtraction.

We can learn from the plot that the computational time of algorithm 1 and 2 is quite close when matrix dimensions are less than or equal to 800, and the time difference for 1600 dimensional matrix is quite large, which is exactly the same as what we have predicted. The situation for algorithm 2 and 3 is a bit eerie since the experimental data show that algorithm 2 is the fastest when matrix dimension is 1600, though I have done experiment many times and choose the mean as the final data. But we do can see that the time needed for computation of algorithm 3 is less than algorithm 2 except for the case of 1600 dimensional matrix(i.e. m9).

We can also view from the data that the time for computations seems to grow cubically from 100 to 1600 dimensional matrices(i.e. m5 to m9). As the matrix size doubles, the time grows roughly 8 or 9 times, that is, the time complexity would probably be $O(n^3)$ or so.

	Time						
	m3	m4	m5	m6	m7	m8	m9
Algo1	0.003	0.004	0.043	0.282	2.282	21.193	219.665
Algo2	0.003	0	0.046	0.281	2.261	21.086	183.708
Algo3	0	0	0.038	0.246	2.027	19.431	206.096

