

# Comparision of efficiency of STM methods

*Alan Yang*

*9/26/2019*

## **paragraph to be inserted into main text**

We conducted a sensitivy analysis on computation efficiency of the two methods using the R software. The computation time in seconds and the computation storage in bytes of the two methods were compared. The parameters that were varied were the number of states and the number of cycles in our cohort model. The number of states varied from 2 to 62, incremented by, 5 while the number of cycles varied from 12 to 1320, incremented by 12. The minimum number of states in a cohort model is 2 and a cohort model with more than 60 states is not very common, and should there be more than 60 states alternative approaches such as microsimulation models are often used instead. The number of months in 110 years (modelling an individual's lifetime) is 1320, and an increment of 12 cycles represents annual incrementation.

In order to capture and calculate transition and state rewards using the cohort trace approach, temporary states had to be created, since the transition reward for a state is only obtained when it is first visited. Without loss of generality, we assumed there are no absorbing states and every state can be visited from any other state. Also, the transition probability matrix for both approaches was randomly sampled such that each entry is between 0 and 1 and each row adds up to 1. The rewards vectors and matrices for both approaches were also randomly sampled from appropriate distributions. We set the seed of R's random number generator to assure reproducible results.

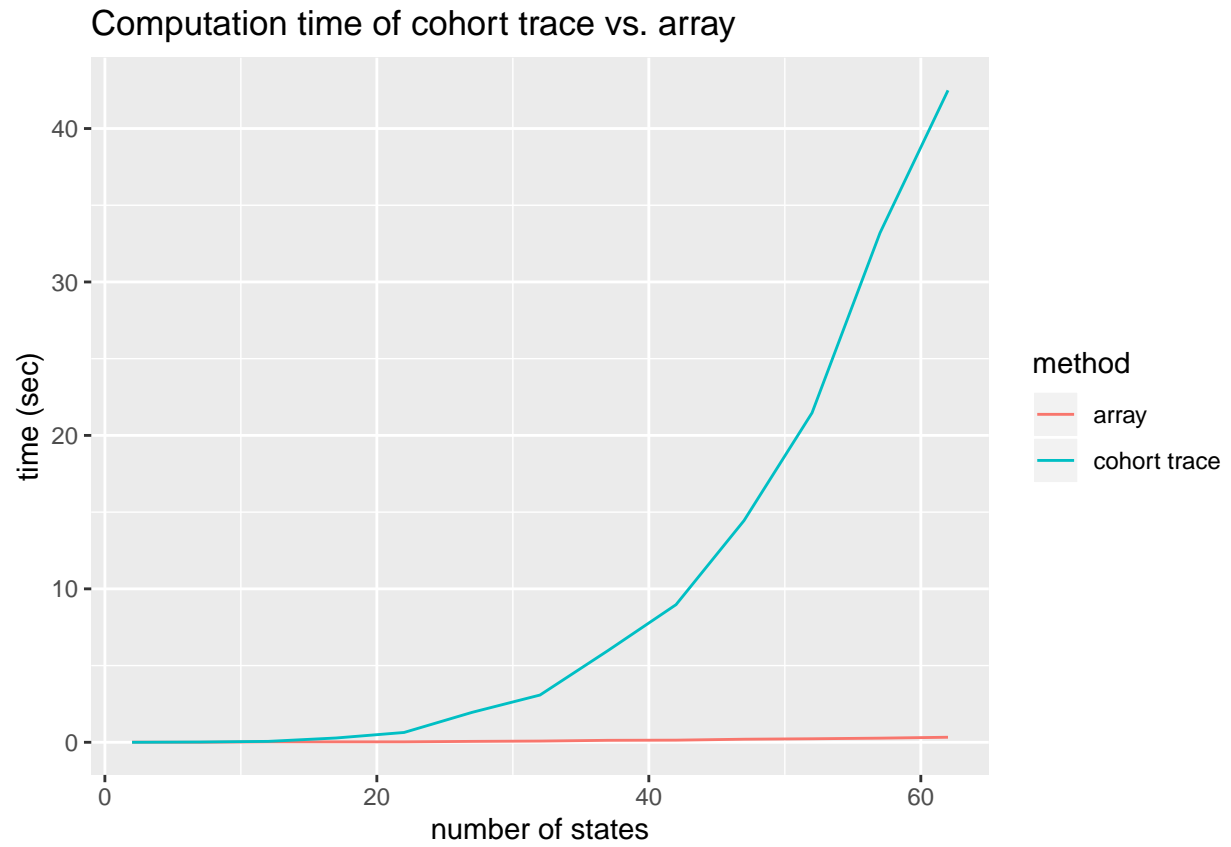
At 1320 cycles, as the number of states increases, the run time of the cohort trace approach increases almost exponentially while that of the array approach stays invariant, and significantly less. Looking at the three-dimensional figure that illustrates how computation time of the two approaches vary as the number of states and the number of cycles increase simultaneoulsy, we see that the run time of the cohort trace approach increases when either the number of states or the number of cycles increases. On the contrary, the run time of the array approach is significantly less and is invariant to increases in either the number of states or the number of cycles.

At 1320 cycles, as the number of states increases, the storage of both approaches increases. When the number of states is less than approximately 52, the cohort trace approach takes up less storage. But when the number of states is above approximately 52, it takes up more storage. Looking at the three-dimensional figure that illustrates how computation storage of the two approaches vary as the number of states and the number of cycles increase simultaneoulsy, we see that the storage increases as when either the number of states or the number of cycles increases. Both approaches take up approximately the same amount of storage before around 50 states, but after 50 states the cohort trace approach takes up more.

From our sensitivity analyis, we conclude that the array approach is computationally superior than the cohort trace approach. It is much faster and does not run any slower when the number of cycles or the number of states increases. And, it takes up less storage as the number of states becomes sufficiently large. The code and the output are included in the supplementary material.

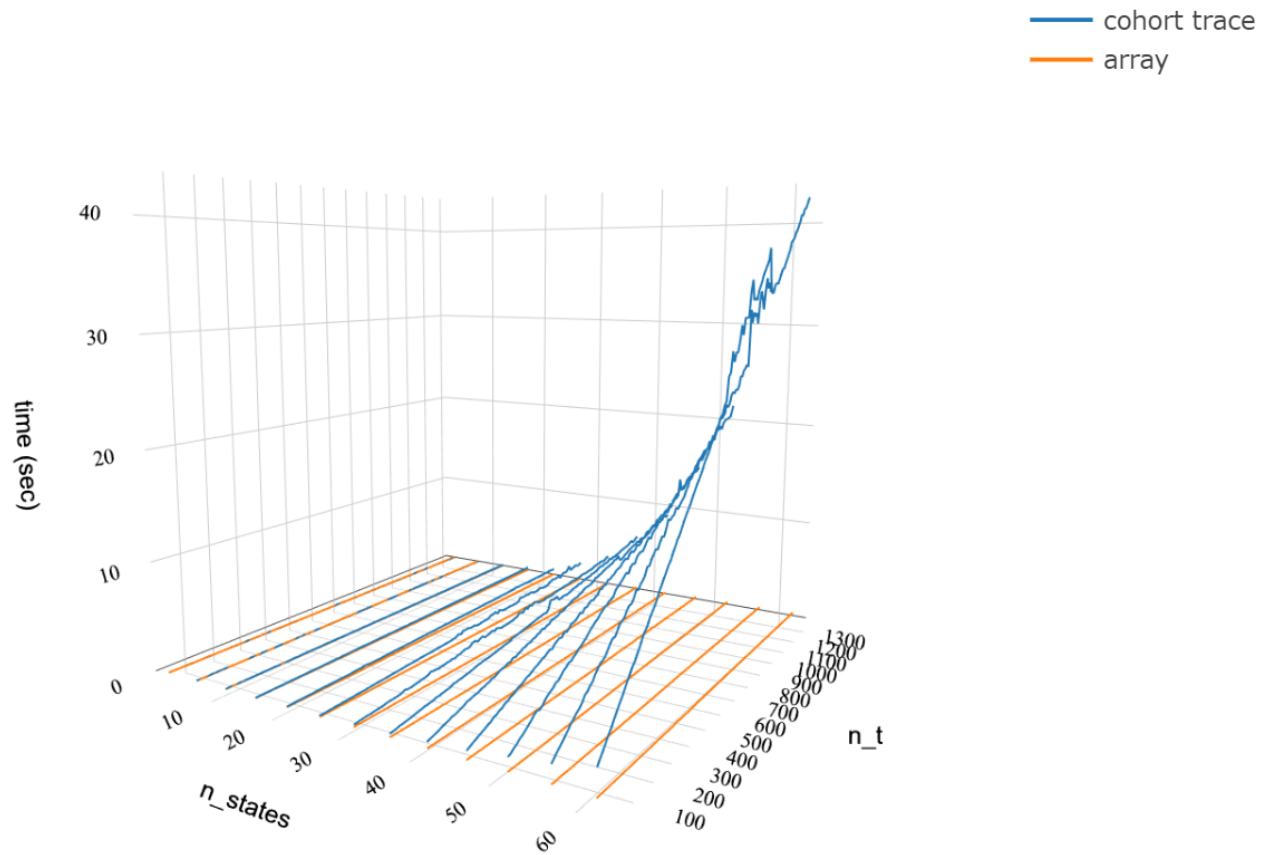
## **Compare time:**

2D line graph - by state, at 1320 cycles



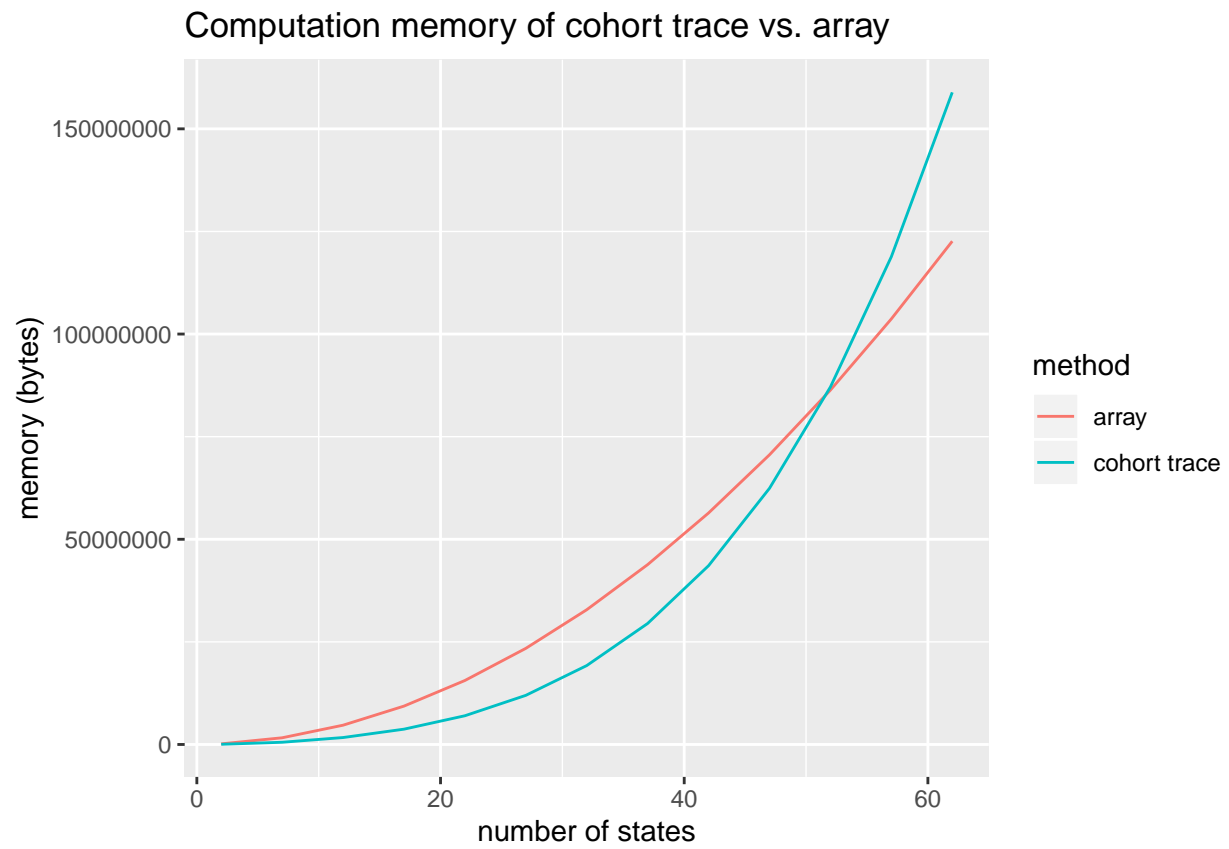
3D Scatter plot:

### 3D graph of $n\_state$ , $n\_t$ and time



#### Compare storage:

2D line graph - by state, at 1320 cycles



3D Scatter plot:

3D graph of  $n_{state}$ ,  $n_t$  and memory

