The most important results of the MATLAB profiler are those that have the largest total time and appear at the top of the profile report; these functions have the most intensive iteration and integration of the code. Slow performance is more than likely caused by inefficiencies within the most time/calculation intensive functions and so if you want to improve the code’s performance these are the functions to optimize. A way to make code more efficient would be to optimize the storage of data and calling of data used in your functions. In terms of memory, one way to significantly reduce the amount of memory required is to avoid the creation of unnecessary temporary copies of data. A way to fix this includes pre allocating large arrays, the built in function ‘repmat’ can be especially helpful in avoiding temporary memory storage. When considering memory usage you can use a built-in MATLAB function ‘memUsed’ to see the current usage in the code and adjust accordingly. Another option is to reduce the amount of overhead when storing data by using simple numeric arrays, since they have the least overhead. Any function that is repeatedly called or iterated with poorly allocated or stored data can result in compounding inefficiencies in time and computing power. The use of nested functions can help pass fewer arguments and prevent unnecessary memory by having functions share workspaces.