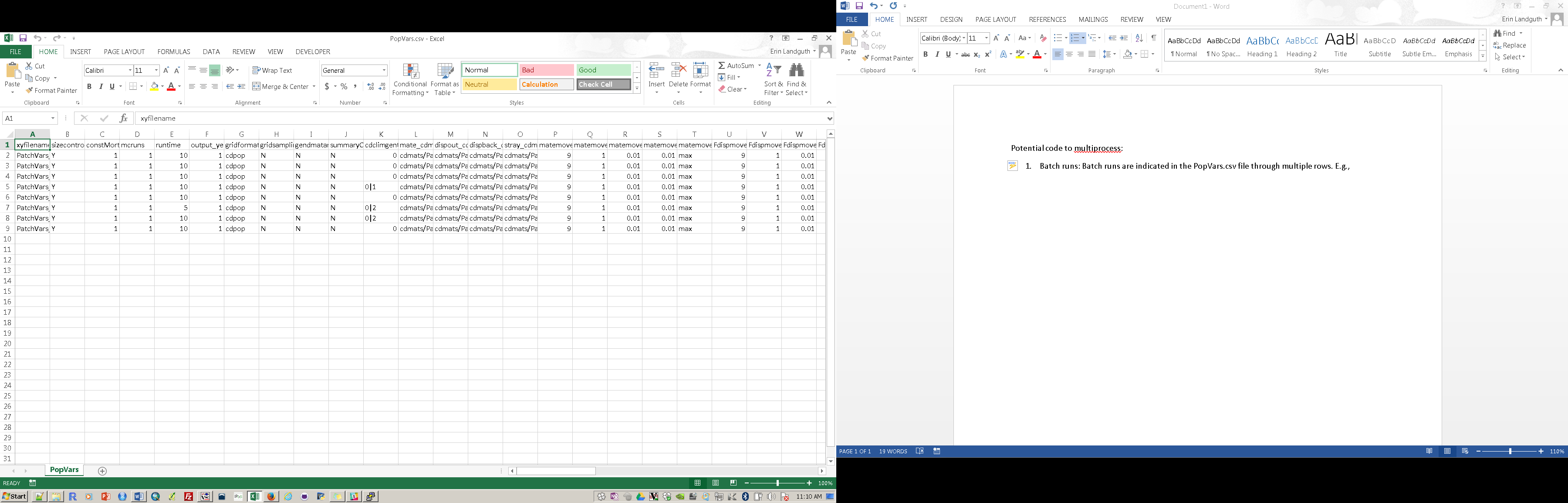
**Potential code to multiprocess:**

1. *Batch runs*: Batch runs are indicated in the PopVars.csv file through multiple rows. E.g., 8 batches below would be indicated in specified output folder as batchrun0mcrun0, batchrun1mcrun0, ….



In CDmetaPOP.py, this is the first for loop ~ ln111:

# ----------------------------------------

# Begin Batch Looping - assign processors

# ----------------------------------------

# This loop is defined by the number of rows in inputvariables.csv

for ibatch in xrange(nSimulations):

1. *Monte Carlo runs*: Or mcruns field in the PopVars.csv file. This is the number of replicates per batch and indicated in the specified output folder as batchrun0mcrun0, batchrun0mcrun1, … In CDmetaPOP.py, this is the second for loop ~ ln311

# ---------------------------------------------

# Begin Monte-Carlo Looping

# ---------------------------------------------

# xrange(mcruns) is typically 10 - 50...and it takes a long time.

for ithmcrun in xrange(mcruns):

1. Ln447: The first module is DoPreProcess(). This has potential for multiprocessing. Each patch location gets initialized independently. However, this is only done once, so I am not sure it is worth the time?
2. Ln 517: GetMetrics() comes next and could also be multiprocessed. Summary statistics are calculated on each patch. This module is also called again later.
3. Ln539: DoUpdate() is called 3 times and could be multiprocessed. Each patch is looped through and individuals within each patch get ‘updated’ through age, size, growth, maturity, other states. DoUpdate() is found in CDmeataPOP\_Modules.py on ln1124. Then on ln1149 is where multiprocessing could start:

# ---------------------------------------------------------------------

# Begin loop through subpopulations updating tasks at appropriate times

for isub in xrange(len(K)):

1. Ln553: The third for loop starts the time steps and this can not be in parallel. However, functions imbedded have possibility.
2. Ln578, Ln654, and Ln656: DoCDClimate(), AddIndividuals(), and DoStochasticUpdate() I do not think can be multiprocessed. These modules update parameters either systematically or stochastically. However, AddIndividuals() could be as this just adds more individuals to each patch independently. This is used rarely though.
3. Ln678 DoMate(): The first big module that loops through individuals to find mate partners. Patches are connected here and this is an individual-based process. I think this would take quite a bit of work to make multiprocessed. Everything would have to be copied to each processor increasing RAM need. Worth looking into.
4. Ln695 DoOffspring(): Females produce offspring. This has potential. Each mated female produces a litter. So definitely, if not through parallel process, some other speed up can be considered here.
5. Ln 712 DoUpdate(): See above
6. Ln726 DoEmigration(): See notes on DoMate(), same idea, but this moves individuals around. For example in CDmeatPOP\_Emigration.py Ln156 – 361 decides where each individual goes. However on Ln363, the ‘packing algorithm’ begins. This is a density dependent process that as independent across each patch. Each isub could go to a PU:

# -------------------

# Packing is selected

# -------------------

if packans == 'packing':

# Loop through each subpopulation

for isub in xrange(len(K)):

There is also a function within this module: CalculateDispersalMetrics() on ln696 which summarizes stats produced from this metric and could be shipped off to PUs, e.g., on ln714:

# Loop through each subpop

for isub in xrange(len(OffDisperseIN)):

1. Ln744: DoMortality() I think could be shipped off to PUs too. This calculates additional mortality at each patch and is independent. E.g., ln57:

# Loop through each patch

for isub in xrange(len(K)):

1. Ln758 DoUpdate() See above, this is the last call.
2. Ln770 DoImmigration(): Similar ideas as DoEmigration(). This moves individuals around again. Packing happens again here (Ln720 in CDmetaPOP\_Immigration.py). In addition, Age 0 offspring are added into each patch – Ln1112 – which is independent.
3. Ln787 DoMortality(): Another motality call, see above.
4. Ln804 GetMetrics(): Another summary stats call, see above.
5. End Time loop and Ln839 DoPostProcess(): For each Monte Carlo, summarize across time and output files.