

MMTP Analysis (Code Doc.)

Evan w/ cr: Anthony and Alexis



HARVARD
UNIVERSITY

Current Progress

1. **Updated** Python2 syntax to Python3
2. Wrote an **explanation** of the `batch_local_diff_params.py` script
3. **Proposed** a possible way to do nonuniform efficiencies + some analysis (**Slide 31**)
4. **Found** where the z values are inconsistent and will update soon (`GeoOctuplet.hh`)



Code Explanation (Uniform Efficiencies)

`batch_local_diff_params.py`



General Overview

1. **Anthony** has code that can run the simulation for set uniform efficiencies.
2. He **loops** over this code
3. In the loop, all pcb's are fixed, then **random entries** can be chosen to be killed [optional]

```
def loop():
    fullEff = 0.1;
    for i in range(10):#np.linspace(0,1,11):
        outDir = "/Users/evancraft" + "/fullEff%.2f"%fullEff
        os.mkdir(outDir)
        main(nKillPCBJobs=2,nKillPCBs=0,fullEff=round(fullEff,2),outDir=outDir)
        fullEff = fullEff + .1;

if __name__ == "__main__":
    loop()
```

Increments efficiencies (fullEff) uniformly from .1 to 1

Calls main function to run simulation
for each of these values

loop()



Jobs

Technicalities

1. Each particular simulation is a “**job**”
2. The job is defined by its **efficiency parameters** and the **number of pcb's** to be killed
3. We can then **execute this job** or rather run the simulation

```
# Input: None
# Output: None
def main(nKillPCBJobs = 1, nKillPCBs = 16, fullEff = 1.0, outDir = "/Users/evancraft/Documents/"):
    jobs = make_jobs(nKillPCBJobs = nKillPCBJobs, nKillPCBs = nKillPCBs, fullEff = fullEff, outDir = outDir)

    # Run in parallel
    starttime = time.time()
    pool = multiprocessing.Pool(processes=6) #use all available cores, otherwise specify the number you want as an argument
    for key, vals in jobs.items():
        pool.apply_async(run_job, args=tuple(), kwds=vals)
        time.sleep(3)
    pool.close()
    pool.join()

    print('That took {} seconds'.format(time.time() - starttime))
```

1. main() receives the (efficiency) job information from loop()
2. job is then generated by make_job()

After being generated, the job is run

main()



Generating the Jobs



Generating the Jobs

1. Generates a set of random configurations (**geometry**)
2. For each configuration, **kills** a set number of pcb's
3. Killed pcb's are given the **dead efficiency**
4. Non-killed pcb's are given **fullEff** from the loop

All efficiencies are set to be the fullEff

Information is fed to kill_pcb()

make_jobs()

```
# Input:
# - Number of different configurations of randomly killed PCBs
# - Number of PCBs to kill
# - The dead PCB efficiency
# Output: Dictionary of job args (job_id, job args)
def make_jobs(nKillPCBJobs = 0, # Number of different configurations of randomly killed PCBs
              nKillPCBs = 0, # Number of PCBs to randomly kill
              fullEff = 1.0, # Efficiency of full efficient PCB
              dead_PCB_eff = 0.0, # Efficiency for killed PCBs
              outDir = "/Users/evancraft/Documents/"):
    jobs = {}
    job_full_eff = {'nJobs':1,
                    'outDir': outDir,
                    'nEvents':10**4,
                    'chamber':'large',
                    'bkgRate':0,
                    'efficiencies':formatMMEffString([fullEff]*NPCBS)} #RealEff))

    # jobs.setdefault('job_full_eff',{}).update(job_full_eff)

    for j in range(nKillPCBJobs):
        job = copy.deepcopy(job_full_eff)
        effs = [float(i.replace("'", "")) for i in job['efficiencies'].split(',')]
        job['efficiencies'] = formatMMEffString(kill_pcb(effs, nKillPCBs, dead_PCB_eff))
        jobs.setdefault('job_{}_PCBs_killed_id_{}'.format(nKillPCBs, j), {}).update(job)

    return jobs
```

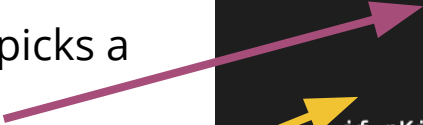
1. Generates a set of random configurations (geometry)
2. For each configuration, kills a set number of pcb's
3. Killed pcb's are given the dead efficiency
4. Non-killed pcb's are given fullEff from the loop

```


# Input:
#     - List of efficiencies
#     - Number of PCBs to kill
#     - The dead PCB efficiency
# Output: list of efficiencies with killed PCBs
def kill_pcb(effs, nKillPCBs = 0, dead_PCB_eff = 0):
    killed = []
    for i in range(nKillPCBs):
        pcb = int(round(random.uniform(0, len(effs)-1)))
        while pcb in killed:
            pcb = int(round(random.uniform(0, len(effs)-1)))
        killed.append(pcb)
        effs[pcb] = dead_PCB_eff
    if nKillPCBs == -1:
        effs = []
        for i in range(8):
            pcb = int(round(random.uniform(0, 15)))
            x = [0]*16
            x[pcb] = 1
            effs += x
        print("Number of PCBs, total efficiency: {}, {}".format(len(effs), np.sum(effs)))
    return effs

```

randomly picks a
pcb to kill



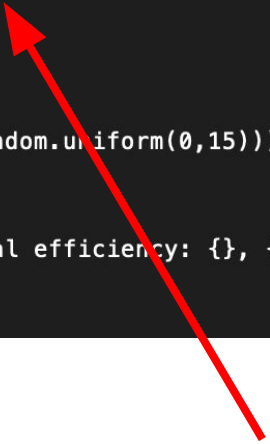
kill that index in
the eff array



kill_pcb()

Returns the updated
efficiency array to
run_jobs()

(while loop) if pcb
is already killed,
pick a different
one






Executing the Job



Executing the job

1. We can then call `batch_local.py`
2. This calls the C++ end of the code to actually run the simulation

```
# Input: job arguments
# Output: None, calls the job operation
def run_job(nJobs = 1,
            outDir = "/Users/evancraft/Documents/", # Not currently used
            nEvents = 100,
            chamber = 'large',
            bkgRate = 1,
            efficiencies = [1.0]*NPCBS):
    os.system( " python python/batch_local.py -j {} -a \"-n {} -ch {} -b {} -tree -uvr -e {}\" -o {}".format(nJobs,
nEvents,
chamber,
bkgRate,
efficiencies,
outDir))
```



Feeds information to the C++ side of the code to be run

run_job()

the 1990s, the number of people in the United States who are 65 years of age or older has increased by 50 percent, and the number of people 75 years of age or older has increased by 100 percent. The number of people 85 years of age or older has increased by 200 percent. The number of people 95 years of age or older has increased by 400 percent. The number of people 100 years of age or older has increased by 1,000 percent. The number of people 105 years of age or older has increased by 2,000 percent. The number of people 110 years of age or older has increased by 4,000 percent. The number of people 115 years of age or older has increased by 8,000 percent. The number of people 120 years of age or older has increased by 16,000 percent. The number of people 125 years of age or older has increased by 32,000 percent. The number of people 130 years of age or older has increased by 64,000 percent. The number of people 135 years of age or older has increased by 128,000 percent. The number of people 140 years of age or older has increased by 256,000 percent. The number of people 145 years of age or older has increased by 512,000 percent. The number of people 150 years of age or older has increased by 1,024,000 percent. The number of people 155 years of age or older has increased by 2,048,000 percent. The number of people 160 years of age or older has increased by 4,096,000 percent. The number of people 165 years of age or older has increased by 8,192,000 percent. The number of people 170 years of age or older has increased by 16,384,000 percent. The number of people 175 years of age or older has increased by 32,768,000 percent. The number of people 180 years of age or older has increased by 65,536,000 percent. The number of people 185 years of age or older has increased by 131,072,000 percent. The number of people 190 years of age or older has increased by 262,144,000 percent. The number of people 195 years of age or older has increased by 524,288,000 percent. The number of people 200 years of age or older has increased by 1,048,576,000 percent. The number of people 205 years of age or older has increased by 2,097,152,000 percent. The number of people 210 years of age or older has increased by 4,194,304,000 percent. The number of people 215 years of age or older has increased by 8,388,608,000 percent. The number of people 220 years of age or older has increased by 16,777,216,000 percent. The number of people 225 years of age or older has increased by 33,554,432,000 percent. The number of people 230 years of age or older has increased by 67,108,864,000 percent. The number of people 235 years of age or older has increased by 134,217,728,000 percent. The number of people 240 years of age or older has increased by 268,435,456,000 percent. The number of people 245 years of age or older has increased by 536,870,912,000 percent. The number of people 250 years of age or older has increased by 1,073,741,824,000 percent. The number of people 255 years of age or older has increased by 2,147,483,648,000 percent. The number of people 260 years of age or older has increased by 4,294,967,296,000 percent. The number of people 265 years of age or older has increased by 8,589,934,592,000 percent. The number of people 270 years of age or older has increased by 17,179,869,184,000 percent. The number of people 275 years of age or older has increased by 34,359,738,368,000 percent. The number of people 280 years of age or older has increased by 68,719,476,736,000 percent. The number of people 285 years of age or older has increased by 137,438,953,472,000 percent. The number of people 290 years of age or older has increased by 274,877,906,944,000 percent. The number of people 295 years of age or older has increased by 549,755,813,888,000 percent. The number of people 300 years of age or older has increased by 1,099,511,627,776,000 percent. The number of people 305 years of age or older has increased by 2,199,023,255,552,000 percent. The number of people 310 years of age or older has increased by 4,398,046,511,104,000 percent. The number of people 315 years of age or older has increased by 8,796,093,022,208,000 percent. The number of people 320 years of age or older has increased by 17,592,186,044,416,000 percent. The number of people 325 years of age or older has increased by 35,184,372,088,832,000 percent. The number of people 330 years of age or older has increased by 70,368,744,177,664,000 percent. The number of people 335 years of age or older has increased by 140,737,488,355,328,000 percent. The number of people 340 years of age or older has increased by 281,474,976,710,656,000 percent. The number of people 345 years of age or older has increased by 562,949,953,421,312,000 percent. The number of people 350 years of age or older has increased by 1,125,899,906,842,624,000 percent. The number of people 355 years of age or older has increased by 2,251,799,813,685,248,000 percent. The number of people 360 years of age or older has increased by 4,503,599,627,370,496,000 percent. The number of people 365 years of age or older has increased by 9,007,199,254,740,992,000 percent. The number of people 370 years of age or older has increased by 18,014,398,509,481,984,000 percent. The number of people 375 years of age or older has increased by 36,028,797,018,963,968,000 percent. The number of people 380 years of age or older has increased by 72,057,594,037,927,936,000 percent. The number of people 385 years of age or older has increased by 144,115,188,075,855,872,000 percent. The number of people 390 years of age or older has increased by 288,230,376,151,711,744,000 percent. The number of people 395 years of age or older has increased by 576,460,752,303,423,488,000 percent. The number of people 400 years of age or older has increased by 1,152,921,504,606,846,976,000 percent. The number of people 405 years of age or older has increased by 2,305,843,009,213,693,952,000 percent. The number of people 410 years of age or older has increased by 4,611,686,018,427,387,904,000 percent. The number of people 415 years of age or older has increased by 9,223,372,036,854,775,808,000 percent. The number of people 420 years of age or older has increased by 18,446,744,073,709,551,616,000 percent. The number of people 425 years of age or older has increased by 36,893,488,147,419,103,232,000 percent. The number of people 430 years of age or older has increased by 73,786,976,294,838,206,464,000 percent. The number of people 435 years of age or older has increased by 147,573,952,589,676,412,928,000 percent. The number of people 440 years of age or older has increased by 295,147,905,179,352,825,856,000 percent. The number of people 445 years of age or older has increased by 590,295,810,358,705,651,712,000 percent. The number of people 450 years of age or older has increased by 1,180,591,620,717,411,303,424,000 percent. The number of people 455 years of age or older has increased by 2,361,183,241,434,822,606,848,000 percent. The number of people 460 years of age or older has increased by 4,722,366,482,869,645,213,696,000 percent. The number of people 465 years of age or older has increased by 9,444,732,965,739,290,427,392,000 percent. The number of people 470 years of age or older has increased by 18,889,465,931,478,580,854,784,000 percent. The number of people 475 years of age or older has increased by 37,778,931,862,957,161,709,568,000 percent. The number of people 480 years of age or older has increased by 75,557,863,725,914,323,419,136,000 percent. The number of people 485 years of age or older has increased by 151,115,727,451,828,646,838,272,000 percent. The number of people 490 years of age or older has increased by 302,231,454,903,657,293,676,544,000 percent. The number of people 495 years of age or older has increased by 604,462,909,807,314,587,353,088,000 percent. The number of people 500 years of age or older has increased by 1,208,925,819,614,629,174,706,176,000 percent. The number of people 505 years of age or older has increased by 2,417,851,639,229,258,349,412,352,000 percent. The number of people 510 years of age or older has increased by 4,835,703,278,458,516,698,824,704,000 percent. The number of people 515 years of age or older has increased by 9,671,406,556,917,033,397,649,408,000 percent. The number of people 520 years of age or older has increased by 19,342,813,113,834,066,795,298,816,000 percent. The number of people 525 years of age or older has increased by 38,685,626,227,668,133,590,597,632,000 percent. The number of people 530 years of age or older has increased by 77,371,252,455,336,267,181,195,264,000 percent. The number of people 535 years of age or older has increased by 154,742,504,910,672,534,362,390,528,000 percent. The number of people 540 years of age or older has increased by 309,485,009,821,345,068,724,781,056,000 percent. The number of people 545 years of age or older has increased by 618,970,019,642,690,137,449,562,112,000 percent. The number of people 550 years of age or older has increased by 1,237,940,039,285,380,274,899,124,224,000 percent. The number of people 555 years of age or older has increased by 2,475,880,078,570,760,549,798,248,448,000 percent. The number of people 560 years of age or older has increased by 4,951,760,157,141,521,099,596,496,896,000 percent. The number of people 565 years of age or older has increased by 9,903,520,314,283,042,199,193,993,792,000 percent. The number of people 570 years of age or older has increased by 19,807,040,628,566,084,398,387,

```
Namespace(NET3=False, a="-n 10000 -ch large -b 0 -tree -uvr -e '0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0'  
", d=False, j='1', o='/Users/evancraft/fullEff0.10')
```


SIMULATION SUMMARY:

0 muons triggered out of 8 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits

0 uniform eff,
no dead

SIMULATION SUMMARY:

74 muons triggered out of 660 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits

.3 uniform eff,
no dead

SIMULATION SUMMARY:

932 muons triggered out of 3135 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits

.5 uniform eff,
no dead

Non-Uniform Efficiencies (Random)

Non-Uniform Efficiencies (Random)

1. We can play around with the killed pcb efficiencies
2. Fix all pcb's to fullEff
3. Randomly choose some of these to be otherEff

```

# Input:
#     - Number of different configurations of randomly killed PCBs
#     - Number of PCBs to kill
#     - The dead PCB efficiency
# Output: Dictionary of job args (job_id, job args)
def make_jobs(nKillPCBJobs = 0, # Number of different configurations of randomly killed PCBs
              nKillPCBs = 0, # Number of PCBs to randomly kill
              fullEff = 1.0, # Efficiency of full efficient PCB
              dead_PCB_eff = 0.0, # Efficiency for killed PCBs
              outDir = "/Users/evancraft/Documents/"):
    jobs = {}
    job_full_eff = {'nJobs':1,
                    'outDir': outDir,
                    'nEvents':10**4,
                    'chamber':'large',
                    'bkgRate':0,
                    'efficiencies':formatMMEffString([fullEff]*NPCBS)} #RealEff))

    # jobs.setdefault('job_full_eff',{}).update(job_full_eff)

    for j in range(nKillPCBJobs):
        job = copy.deepcopy(job_full_eff)
        effs = [float(i.replace("'", "")) for i in job['efficiencies'].split(',')]
        job['efficiencies'] = formatMMEffString(kill_pcb(effs, nKillPCBs, dead_PCB_eff))
        jobs.setdefault('job_{}_PCBs_killed_id_{}'.format(nKillPCBs, j), {}).update(job)

    return jobs

```

Actual Execution

make_jobs()

1. Set fulleff to be 80% (example)
2. For each configuration, kill a set number of pcb's
3. Set killedEff to be 50% (example)
4. Non-killed pcb's are given the fullEff from the loop

```
# Input: None
# Output: None
def main(nKillPCBJobs = 1, nKillPCBs = 16, fullEff = 1.0, outDir = "/Users/evancraft/Documents/"):
    jobs = make_jobs(nKillPCBJobs = nKillPCBJobs, nKillPCBs = nKillPCBs, fullEff = fullEff, outDir = outDir)

    # Run in parallel
    starttime = time.time()
    pool = multiprocessing.Pool(processes=6) #use all available cores, otherwise specify the number you want as an argument
    for key, vals in jobs.items():
        pool.apply_async(run_job, args=tuple(), kwds=vals)
        time.sleep(3)
    pool.close()
    pool.join()

    print('That took {} seconds'.format(time.time() - starttime))
```

We just need to update the killedeff here to be 50%. As of now, it is not set (runs default of 0.0)

main()

After being generated, the job is run and it will emulate being a non-uniform distribution

Real bb5 Efficiencies

Real bb5 Efficiencies (Hack)

1. We could just kill all of the pcb's and set the killed efficiency (individually) to be the real efficiency
2. A bit of a workaround, but straightforward to code

All efficiencies are set to be the fullEff

```
# Input:
# - Number of different configurations of randomly killed PCBs
# - Number of PCBs to kill
# - The dead PCB efficiency
# Output: Dictionary of job args (job_id, job args)
def make_jobs(nKillPCBJobs = 0, # Number of different configurations of randomly killed PCBs
             nKillPCBs = 0, # Number of PCBs to randomly kill
             fullEff = 1.0, # Efficiency of full efficient PCB
             dead_PCB_eff = 0.0, # Efficiency for killed PCBs
             outDir = "/Users/evancraft/Documents/"):
    jobs = {}
    job_full_eff = {'nJobs':1,
                   'outDir': outDir,
                   'nEvents':10**4,
                   'chamber':'large',
                   'bkgRate':0,
                   'efficiencies':formatMMEffString([fullEff]*nPCBS)} #RealEff))

    # jobs.setdefault('job_full_eff',{}).update(job_full_eff)

    for i in range(nKillPCBJobs):
        job = copy.deepcopy(job_full_eff)
        effs = [float(i.replace("'", "")) for i in job['efficiencies'].split(',')]
        job['efficiencies'] = formatMMEffString(kill_pcb(effs, nKillPCBs, dead_PCB_eff))
        jobs.setdefault('job_{}_PCBs_killed_id_{}'.format(nKillPCBs, j), {}).update(job)

    return jobs
```

make_jobs()

1. What we'll do is leave this alone, and just set all the efficiencies as normal
2. Instead, we can kill all the pcb's and when killing them, set them to be the real efficiency

~~randomly picks a
pcb to kill~~

```
# Input:
#     - List of efficiencies
#     - Number of PCBs to kill
#     - The dead PCB efficiency
# Output: list of efficiencies with killed PCBs
def kill_pcb(effs, nKillPCBs = 0, dead_PCB_eff = 0):
    killed = []
    for i in range(nKillPCBs):
        pcb = int(round(random.uniform(0, len(effs)-1)))
        while pcb in killed:
            pcb = int(round(random.uniform(0, len(effs)-1)))
        killed.append(pcb)
        effs[pcb] = dead_PCB_eff
    if nKillPCBs == -1:
        effs = []
        for i in range(8):
            pcb = int(round(random.uniform(0, 15)))
            x = [0]*16
            x[pcb] = 1
            effs += x
        print("Number of PCBs, total efficiency: {}, {}".format(len(effs), np.sum(effs)))
    return effs
```

kill_pcb()

Instead of
randomly picking
a pcb, (for) loop
through all of the
pcb's and hard set
to the realEff

Non-Uniform Efficiencies (Layers)

Non-Uniform Efficiencies (Layers)

1. We can again play around with the killed pcb efficiencies
2. The pcb list is ordered (128 entries), so just choose to kill particular ranges in the list (i.e. 0-7 corresponds to layer 1 and so on)

NPCBS = 128 # 16 per layer

RealEff = [0.5695, 0.9597, 0.9868, 0.985, 0.9825, 0.9835, 0.9849, 0.9844,



I believe that the pcb's are **ordered**, so if I wanted to mess with layer 1, I would be messing with indices 0-7

So when I'm killing the pcb's on say layer 1, I would just loop over 0-7

loop()

~~randomly picks a
pcb to kill~~

```
# Input:
# - List of efficiencies
# - Number of PCBs to kill
# - The dead PCB efficiency
# Output: list of efficiencies with killed PCBs
def kill_pcb(effs, nKillPCBs = 0, dead_PCB_eff = 0):
    killed = []
    for i in range(nKillPCBs):
        pcb = int(round(random.uniform(0, len(effs)-1)))
        while pcb in killed:
            pcb = int(round(random.uniform(0, len(effs)-1)))
        killed.append(pcb)
        effs[pcb] = dead_PCB_eff
    if nKillPCBs == -1:
        effs = []
        for i in range(8):
            pcb = int(round(random.uniform(0, 15)))
            x = [0]*16
            x[pcb] = 1
            effs += x
        print("Number of PCBs, total efficiency: {}, {}".format(len(effs), np.sum(effs)))
    return effs
```

Instead of
randomly picking
a pcb, (for) loop
through only
indices 0-7

kill_pcb()

and the rest h

5 eff

1

4/8 pcb's on layer 3 set to .25 eff

Question from Anthony

For instance, what happens if 1/2 of the PCBs on 2/8 layers have very low efficiency and the rest have around 90%.

SIMULATION SUMMARY:

```
8089 muons triggered out of 9040 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Result: 89%

```
9001 muons triggered out of 9582 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Baseline: 94%

Other Examples

```
(base) Evans-MBP-5:oct_sim-master evancraft$ python3 python/batch_layers.py
Namespace(NET3=False, a="-n 10000 -ch large -b 0 -tree -uvr -e '1.0,1.0,1.0,0.2,
1.0,0.2,1.0,0.2,1.0,1.0,1.0,1.0,1.0,1.0,0.5,0.5,0.5,0.5,0.5,0.5,0.5,1.0,1.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,
1.0,1.0,1.0,1.0'", d=False, j='1', o='/Users/evancraft/fullEff1.00')
Output directory :: /Users/evancraft/fullEff1.00/batch-2020-11-17-19h55m06s
```

In this example,

$\frac{3}{8}$ pcb's on layer 1 set to .2 eff

$\frac{7}{8}$ pcb's on layer 3 set to .5 eff

Other Examples

SIMULATION SUMMARY:

```
8057 muons triggered out of 8942 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Result: 90%

```
9001 muons triggered out of 9582 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Baseline: 94%



Trigger Project (Code)

Evan w/ cr: Anthony and Alexis



Current Progress

1. **Updated** Python2 syntax to Python3
2. Wrote an **explanation** of the `batch_local_diff_params.py` script
3. **Proposed** a possible way to do nonuniform efficiencies + some analysis (**Slide 31**)
4. **Found** where the z values are inconsistent and will update soon (`GeoOctuplet.hh`)



Code Explanation (Uniform Efficiencies)

`batch_local_diff_params.py`



General Overview

1. **Anthony** has code that can run the simulation for set uniform efficiencies.
2. He **loops** over this code
3. In the loop, all pcb's are fixed, then **random entries** can be chosen to be killed [optional]

```
def loop():
    fullEff = 0.1;
    for i in range(10):#np.linspace(0,1,11):
        outDir = "/Users/evancraft" + "/fullEff%.2f"%fullEff
        os.mkdir(outDir)
        main(nKillPCBJobs=2,nKillPCBs=0,fullEff=round(fullEff,2),outDir=outDir)
        fullEff = fullEff + .1;

if __name__ == "__main__":
    loop()
```

Increments efficiencies (fullEff) uniformly from .1 to 1

Calls main function to run simulation
for each of these values

loop()



Jobs

Technicalities

1. Each particular simulation is a “**job**”
2. The job is defined by its **efficiency parameters** and the **number of pcb's** to be killed
3. We can then **execute this job** or rather run the simulation


```
# Input: None
# Output: None
def main(nKillPCBJobs = 1, nKillPCBs = 16, fullEff = 1.0, outDir = "/Users/evancraft/Documents/"):
    jobs = make_jobs(nKillPCBJobs = nKillPCBJobs, nKillPCBs = nKillPCBs, fullEff = fullEff, outDir = outDir)

    # Run in parallel
    starttime = time.time()
    pool = multiprocessing.Pool(processes=6) #use all available cores, otherwise specify the number you want as an argument
    for key, vals in jobs.items():
        pool.apply_async(run_job, args=tuple(), kwds=vals)
        time.sleep(3)
    pool.close()
    pool.join()

    print('That took {} seconds'.format(time.time() - starttime))
```

1. main() receives the (efficiency) job information from loop()
2. job is then generated by make_job()

After being generated, the job is run

main()



Generating the Jobs



Generating the Jobs

1. Generates a set of random configurations (**geometry**)
2. For each configuration, **kills** a set number of pcb's
3. Killed pcb's are given the **dead efficiency**
4. Non-killed pcb's are given **fullEff** from the loop

All efficiencies are set to be the fullEff

Information is fed to kill_pcb()

make_jobs()

```
# Input:
# - Number of different configurations of randomly killed PCBs
# - Number of PCBs to kill
# - The dead PCB efficiency
# Output: Dictionary of job args (job_id, job args)
def make_jobs(nKillPCBJobs = 0, # Number of different configurations of randomly killed PCBs
             nKillPCBs = 0, # Number of PCBs to randomly kill
             fullEff = 1.0, # Efficiency of full efficient PCB
             dead_PCB_eff = 0.0, # Efficiency for killed PCBs
             outDir = "/Users/evancraft/Documents/"):
    jobs = {}
    job_full_eff = {'nJobs':1,
                   'outDir': outDir,
                   'nEvents':10**4,
                   'chamber':'large',
                   'bkgRate':0,
                   'efficiencies':formatMMEffString([fullEff]*NPCBS)} #RealEff))

    # jobs.setdefault('job_full_eff',{}).update(job_full_eff)

    for j in range(nKillPCBJobs):
        job = copy.deepcopy(job_full_eff)
        effs = [float(i.replace("'", "")) for i in job['efficiencies'].split(',')]
        job['efficiencies'] = formatMMEffString(kill_pcb(effs, nKillPCBs, dead_PCB_eff))
        jobs.setdefault('job_{}_PCBs_killed_id_{}'.format(nKillPCBs, j), {}).update(job)

    return jobs
```

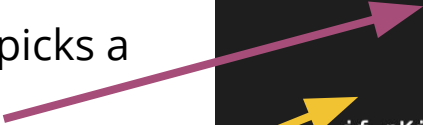
1. Generates a set of random configurations (geometry)
2. For each configuration, kills a set number of pcb's
3. Killed pcb's are given the dead efficiency
4. Non-killed pcb's are given fullEff from the loop

```


# Input:
#     - List of efficiencies
#     - Number of PCBs to kill
#     - The dead PCB efficiency
# Output: list of efficiencies with killed PCBs
def kill_pcb(effs, nKillPCBs = 0, dead_PCB_eff = 0):
    killed = []
    for i in range(nKillPCBs):
        pcb = int(round(random.uniform(0, len(effs)-1)))
        while pcb in killed:
            pcb = int(round(random.uniform(0, len(effs)-1)))
        killed.append(pcb)
        effs[pcb] = dead_PCB_eff
    if nKillPCBs == -1:
        effs = []
        for i in range(8):
            pcb = int(round(random.uniform(0, 15)))
            x = [0]*16
            x[pcb] = 1
            effs += x
        print("Number of PCBs, total efficiency: {}, {}".format(len(effs), np.sum(effs)))
    return effs

```

randomly picks a
pcb to kill



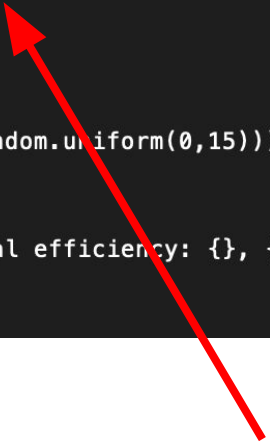
kill that index in
the eff array



Returns the updated
efficiency array to
run_jobs()

kill_pcb()

(while loop) if pcb
is already killed,
pick a different
one






Executing the Job



Executing the job

1. We can then call `batch_local.py`
2. This calls the C++ end of the code to actually run the simulation

```
# Input: job arguments
# Output: None, calls the job operation
def run_job(nJobs = 1,
            outDir = "/Users/evancraft/Documents/", # Not currently used
            nEvents = 100,
            chamber = 'large',
            bkgRate = 1,
            efficiencies = [1.0]*NPCBS):
    os.system( " python python/batch_local.py -j {} -a \"-n {} -ch {} -b {} -tree -uvr -e {}\" -o {}".format(nJobs,
nEvents,
chamber,
bkgRate,
efficiencies,
outDir))
```



Feeds information to the C++ side of the code to be run

run_job()

Examples

```
Namespace(NET3=False, a="-n 10000 -ch large -b 0 -tree -uvr -e '0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,  
0.0,0.0,0.0,0.0,0.0'", d=False, j='1', o='/Users/evancraft/fullEff0.10')
```

In this example, all pcb's are set to .1 and I told it to kill 127 pcb's

SIMULATION SUMMARY:

0 muons triggered out of 8 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits

0 uniform eff,
no dead

SIMULATION SUMMARY:

74 muons triggered out of 660 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits

.3 uniform eff,
no dead

SIMULATION SUMMARY:

932 muons triggered out of 3135 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits

.5 uniform eff,
no dead

Non-Uniform Efficiencies (Random)

Non-Uniform Efficiencies (Random)

1. We can play around with the killed pcb efficiencies
2. Fix all pcb's to fullEff
3. Randomly choose some of these to be otherEff

```

# Input:
#     - Number of different configurations of randomly killed PCBs
#     - Number of PCBs to kill
#     - The dead PCB efficiency
# Output: Dictionary of job args (job_id, job args)
def make_jobs(nKillPCBJobs = 0, # Number of different configurations of randomly killed PCBs
             nKillPCBs = 0, # Number of PCBs to randomly kill
             fullEff = 1.0, # Efficiency of full efficient PCB
             dead_PCB_eff = 0.0, # Efficiency for killed PCBs
             outDir = "/Users/evancraft/Documents/"):
    jobs = {}
    job_full_eff = {'nJobs':1,
                   'outDir': outDir,
                   'nEvents':10**4,
                   'chamber':'large',
                   'bkgRate':0,
                   'efficiencies':formatMMEffString([fullEff]*NPCBS)} #RealEff))

    # jobs.setdefault('job_full_eff',{}).update(job_full_eff)

    for j in range(nKillPCBJobs):
        job = copy.deepcopy(job_full_eff)
        effs = [float(i.replace("'", "")) for i in job['efficiencies'].split(',')]
        job['efficiencies'] = formatMMEffString(kill_pcb(effs, nKillPCBs, dead_PCB_eff))
        jobs.setdefault('job_{}_PCBs_killed_id_{}'.format(nKillPCBs, j), {}).update(job)

    return jobs

```

Actual Execution

make_jobs()

1. Set fullEff to be 80% (example)
2. For each configuration, kill a set number of pcb's
3. Set killedEff to be 50% (example)
4. Non-killed pcb's are given the fullEff from the loop

```
# Input: None
# Output: None
def main(nKillPCBJobs = 1, nKillPCBs = 16, fullEff = 1.0, outDir = "/Users/evancraft/Documents/"):
    jobs = make_jobs(nKillPCBJobs = nKillPCBJobs, nKillPCBs = nKillPCBs, fullEff = fullEff, outDir = outDir)

    # Run in parallel
    starttime = time.time()
    pool = multiprocessing.Pool(processes=6) #use all available cores, otherwise specify the number you want as an argument
    for key, vals in jobs.items():
        pool.apply_async(run_job, args=tuple(), kwds=vals)
        time.sleep(3)
    pool.close()
    pool.join()

    print('That took {} seconds'.format(time.time() - starttime))
```

We just need to update the killedeff here to be 50%. As of now, it is not set (runs default of 0.0)

main()

After being generated, the job is run and it will emulate being a non-uniform distribution

Real bb5 Efficiencies

Real bb5 Efficiencies (Hack)

1. We could just kill all of the pcb's and set the killed efficiency (individually) to be the real efficiency
2. A bit of a workaround, but straightforward to code

All efficiencies are set to be the fullEff

```
# Input:
# - Number of different configurations of randomly killed PCBs
# - Number of PCBs to kill
# - The dead PCB efficiency
# Output: Dictionary of job args (job_id, job args)
def make_jobs(nKillPCBJobs = 0, # Number of different configurations of randomly killed PCBs
             nKillPCBs = 0, # Number of PCBs to randomly kill
             fullEff = 1.0, # Efficiency of full efficient PCB
             dead_PCB_eff = 0.0, # Efficiency for killed PCBs
             outDir = "/Users/evancraft/Documents/"):
    jobs = {}
    job_full_eff = {'nJobs':1,
                   'outDir': outDir,
                   'nEvents':10**4,
                   'chamber':'large',
                   'bkgRate':0,
                   'efficiencies':formatMMEffString([fullEff]*nPCBS)} #RealEff))

    # jobs.setdefault('job_full_eff',{}).update(job_full_eff)

    for i in range(nKillPCBJobs):
        job = copy.deepcopy(job_full_eff)
        effs = [float(i.replace("'", "")) for i in job['efficiencies'].split(',')]
        job['efficiencies'] = formatMMEffString(kill_pcb(effs, nKillPCBs, dead_PCB_eff))
        jobs.setdefault('job_{}_PCBs_killed_id_{}'.format(nKillPCBs, j), {}).update(job)

    return jobs
```

make_jobs()

1. What we'll do is leave this alone, and just set all the efficiencies as normal
2. Instead, we can kill all the pcb's and when killing them, set them to be the real efficiency

~~randomly picks a
pcb to kill~~

```
# Input:
#     - List of efficiencies
#     - Number of PCBs to kill
#     - The dead PCB efficiency
# Output: list of efficiencies with killed PCBs
def kill_pcb(effs, nKillPCBs = 0, dead_PCB_eff = 0):
    killed = []
    for i in range(nKillPCBs):
        pcb = int(round(random.uniform(0, len(effs)-1)))
        while pcb in killed:
            pcb = int(round(random.uniform(0, len(effs)-1)))
        killed.append(pcb)
        effs[pcb] = dead_PCB_eff
    if nKillPCBs == -1:
        effs = []
        for i in range(8):
            pcb = int(round(random.uniform(0, 15)))
            x = [0]*16
            x[pcb] = 1
            effs += x
        print("Number of PCBs, total efficiency: {}, {}".format(len(effs), np.sum(effs)))
    return effs
```

Instead of
randomly picking
a pcb, (for) loop
through all of the
pcb's and hard set
to the realEff

kill_pcb()

Non-Uniform Efficiencies (Layers)

Non-Uniform Efficiencies (Layers)

1. We can again play around with the killed pcb efficiencies
2. The pcb list is ordered (128 entries), so just choose to kill particular ranges in the list (i.e. 0-7 corresponds to layer 1 and so on)

NPCBS = 128 # 16 per layer

RealEff = [0.5695, 0.9597, 0.9868, 0.985, 0.9825, 0.9835, 0.9849, 0.9844,



I believe that the pcb's are **ordered**, so if I wanted to mess with layer 1, I would be messing with indices 0-7

So when I'm killing the pcb's on say layer 1, I would just loop over 0-7

loop()

~~randomly picks a
pcb to kill~~

```
# Input:
# - List of efficiencies
# - Number of PCBs to kill
# - The dead PCB efficiency
# Output: list of efficiencies with killed PCBs
def kill_pcb(effs, nKillPCBs = 0, dead_PCB_eff = 0):
    killed = []
    for i in range(nKillPCBs):
        pcb = int(round(random.uniform(0, len(effs)-1)))
        while pcb in killed:
            pcb = int(round(random.uniform(0, len(effs)-1)))
        killed.append(pcb)
        effs[pcb] = dead_PCB_eff
    if nKillPCBs == -1:
        effs = []
        for i in range(8):
            pcb = int(round(random.uniform(0, 15)))
            x = [0]*16
            x[pcb] = 1
            effs += x
        print("Number of PCBs, total efficiency: {}, {}".format(len(effs), np.sum(effs)))
    return effs
```

Instead of
randomly picking
a pcb, (for) loop
through only
indices 0-7

kill_pcb()

and the rest h

5 eff

4/8 pcb's on layer 3 set to .25 eff

Question from Anthony

For instance, what happens if 1/2 of the PCBs on 2/8 layers have very low efficiency and the rest have around 90%.

SIMULATION SUMMARY:

```
8089 muons triggered out of 9040 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Result: 89%

```
9001 muons triggered out of 9582 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Baseline: 94%

the 1990s, the number of people in the United States who are 65 years of age or older has increased by 50 percent, and the number of people 75 years of age or older has increased by 75 percent. The number of people 85 years of age or older has increased by 150 percent. The number of people 95 years of age or older has increased by 300 percent. The number of people 100 years of age or older has increased by 500 percent. The number of people 105 years of age or older has increased by 1,000 percent. The number of people 110 years of age or older has increased by 2,000 percent. The number of people 115 years of age or older has increased by 4,000 percent. The number of people 120 years of age or older has increased by 8,000 percent. The number of people 125 years of age or older has increased by 16,000 percent. The number of people 130 years of age or older has increased by 32,000 percent. The number of people 135 years of age or older has increased by 64,000 percent. The number of people 140 years of age or older has increased by 128,000 percent. The number of people 145 years of age or older has increased by 256,000 percent. The number of people 150 years of age or older has increased by 512,000 percent. The number of people 155 years of age or older has increased by 1,024,000 percent. The number of people 160 years of age or older has increased by 2,048,000 percent. The number of people 165 years of age or older has increased by 4,096,000 percent. The number of people 170 years of age or older has increased by 8,192,000 percent. The number of people 175 years of age or older has increased by 16,384,000 percent. The number of people 180 years of age or older has increased by 32,768,000 percent. The number of people 185 years of age or older has increased by 65,536,000 percent. The number of people 190 years of age or older has increased by 131,072,000 percent. The number of people 195 years of age or older has increased by 262,144,000 percent. The number of people 200 years of age or older has increased by 524,288,000 percent. The number of people 205 years of age or older has increased by 1,048,576,000 percent. The number of people 210 years of age or older has increased by 2,097,152,000 percent. The number of people 215 years of age or older has increased by 4,194,304,000 percent. The number of people 220 years of age or older has increased by 8,388,608,000 percent. The number of people 225 years of age or older has increased by 16,777,216,000 percent. The number of people 230 years of age or older has increased by 33,554,432,000 percent. The number of people 235 years of age or older has increased by 67,108,864,000 percent. The number of people 240 years of age or older has increased by 134,217,728,000 percent. The number of people 245 years of age or older has increased by 268,435,456,000 percent. The number of people 250 years of age or older has increased by 536,870,912,000 percent. The number of people 255 years of age or older has increased by 1,073,741,824,000 percent. The number of people 260 years of age or older has increased by 2,147,483,648,000 percent. The number of people 265 years of age or older has increased by 4,294,967,296,000 percent. The number of people 270 years of age or older has increased by 8,589,934,592,000 percent. The number of people 275 years of age or older has increased by 17,179,869,184,000 percent. The number of people 280 years of age or older has increased by 34,359,738,368,000 percent. The number of people 285 years of age or older has increased by 68,719,476,736,000 percent. The number of people 290 years of age or older has increased by 137,438,953,472,000 percent. The number of people 295 years of age or older has increased by 274,877,906,944,000 percent. The number of people 300 years of age or older has increased by 549,755,813,888,000 percent. The number of people 305 years of age or older has increased by 1,099,511,627,776,000 percent. The number of people 310 years of age or older has increased by 2,199,023,255,552,000 percent. The number of people 315 years of age or older has increased by 4,398,046,511,104,000 percent. The number of people 320 years of age or older has increased by 8,796,093,022,208,000 percent. The number of people 325 years of age or older has increased by 17,592,186,044,416,000 percent. The number of people 330 years of age or older has increased by 35,184,372,088,832,000 percent. The number of people 335 years of age or older has increased by 70,368,744,177,664,000 percent. The number of people 340 years of age or older has increased by 140,737,488,355,328,000 percent. The number of people 345 years of age or older has increased by 281,474,976,710,656,000 percent. The number of people 350 years of age or older has increased by 562,949,953,421,312,000 percent. The number of people 355 years of age or older has increased by 1,125,899,906,842,624,000 percent. The number of people 360 years of age or older has increased by 2,251,799,813,685,248,000 percent. The number of people 365 years of age or older has increased by 4,503,599,627,370,496,000 percent. The number of people 370 years of age or older has increased by 9,007,199,254,740,992,000 percent. The number of people 375 years of age or older has increased by 18,014,398,509,481,984,000 percent. The number of people 380 years of age or older has increased by 36,028,797,018,963,968,000 percent. The number of people 385 years of age or older has increased by 72,057,594,037,927,936,000 percent. The number of people 390 years of age or older has increased by 144,115,188,075,855,872,000 percent. The number of people 395 years of age or older has increased by 288,230,376,151,711,744,000 percent. The number of people 400 years of age or older has increased by 576,460,752,303,423,488,000 percent. The number of people 405 years of age or older has increased by 1,152,921,504,606,846,976,000 percent. The number of people 410 years of age or older has increased by 2,305,843,009,213,693,952,000 percent. The number of people 415 years of age or older has increased by 4,611,686,018,427,387,904,000 percent. The number of people 420 years of age or older has increased by 9,223,372,036,854,775,808,000 percent. The number of people 425 years of age or older has increased by 18,446,744,073,709,551,616,000 percent. The number of people 430 years of age or older has increased by 36,893,488,147,419,103,232,000 percent. The number of people 435 years of age or older has increased by 73,786,976,294,838,206,464,000 percent. The number of people 440 years of age or older has increased by 147,573,952,589,676,412,928,000 percent. The number of people 445 years of age or older has increased by 295,147,905,179,352,825,856,000 percent. The number of people 450 years of age or older has increased by 590,295,810,358,705,651,712,000 percent. The number of people 455 years of age or older has increased by 1,180,591,620,717,411,303,424,000 percent. The number of people 460 years of age or older has increased by 2,361,183,241,434,822,606,848,000 percent. The number of people 465 years of age or older has increased by 4,722,366,482,869,645,213,696,000 percent. The number of people 470 years of age or older has increased by 9,444,732,965,739,290,427,392,000 percent. The number of people 475 years of age or older has increased by 18,889,465,931,478,580,854,784,000 percent. The number of people 480 years of age or older has increased by 37,778,931,862,957,161,709,568,000 percent. The number of people 485 years of age or older has increased by 75,557,863,725,914,323,419,136,000 percent. The number of people 490 years of age or older has increased by 151,115,727,451,828,646,838,272,000 percent. The number of people 495 years of age or older has increased by 302,231,454,903,657,293,676,544,000 percent. The number of people 500 years of age or older has increased by 604,462,909,807,314,587,353,088,000 percent. The number of people 505 years of age or older has increased by 1,208,925,819,614,629,174,706,176,000 percent. The number of people 510 years of age or older has increased by 2,417,851,639,229,258,349,412,352,000 percent. The number of people 515 years of age or older has increased by 4,835,703,278,458,516,698,824,704,000 percent. The number of people 520 years of age or older has increased by 9,671,406,556,917,033,397,649,408,000 percent. The number of people 525 years of age or older has increased by 19,342,813,113,834,066,795,298,816,000 percent. The number of people 530 years of age or older has increased by 38,685,626,227,668,133,590,597,632,000 percent. The number of people 535 years of age or older has increased by 77,371,252,455,336,267,181,195,264,000 percent. The number of people 540 years of age or older has increased by 154,742,504,910,672,534,362,390,528,000 percent. The number of people 545 years of age or older has increased by 309,485,009,821,345,068,724,781,056,000 percent. The number of people 550 years of age or older has increased by 618,970,019,642,690,137,449,562,112,000 percent. The number of people 555 years of age or older has increased by 1,237,940,039,285,380,274,899,124,224,000 percent. The number of people 560 years of age or older has increased by 2,475,880,078,570,760,549,798,248,448,000 percent. The number of people 565 years of age or older has increased by 4,951,760,157,141,521,099,596,496,896,000 percent. The number of people 570 years of age or older has increased by 9,903,520,314,283,042,199,193,993,792,000 percent. The number of people 575 years of age or older has increased by 19,807,040

```
(base) Evans-MBP-5:oct_sim-master evancraft$ python3 python/batch_layers.py  
Namespace(NET3=False, a="-n 10000 -ch large -b 0 -tree -uvr -e '1.0,1.0,1.0,0.2,  
1.0,0.2,1.0,0.2,1.0,1.0,1.0,1.0,1.0,1.0,0.5,0.5,0.5,0.5,0.5,0.5,0.5,1.0,1.0,1.0,  
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,  
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,  
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,  
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,  
1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,  
1.0,1.0,1.0,1.0'", d=False, j='1', o='/Users/evancraft/fullEff1.00')  
Output directory :: /Users/evancraft/fullEff1.00/batch-2020-11-17-19h55m06s
```

Other Examples

SIMULATION SUMMARY:

```
8057 muons triggered out of 8942 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Result: 90%

```
9001 muons triggered out of 9582 muons that should trigger
0 triggers with spoiled uv hits
n=1: 0 | n=2: 0 | n=3: 0 | n=4: 0
0 bkg uv hits total
0 extra trigger events
0 events where triggers were only made with bkg hits
```

Baseline: 94%