1 Thu

A Bash

a) A little bit of Bash each day keeps the doctor away. In the main loop, around line 140, we have

```
read -p "Starting generation ${gen} at location ${state}.\
          Press any key to continue... " -n1 -s
```

Option `-p <prompt>` outputs the `prompt` string before reading user input.
Option `-n <number>` returns after reading the specified `number` of chars.
Option `-s` does not echo the user's input.

2 Fri

On a flight home. Didn't do much except started going over Part B1 again

3 Sat

A Meeting with Julie

a) The meeting this Wednesday was cancelled. In the meantime, Julie suggests reading the Appendix and her candidacy paper. I have largely finished reading the relevant part of the Appendix, so the rest of this week will most likely be spent reading the candidacy paper and checking out the `Loop Parts` in more detail.

b) Part B1 second pass: will also check out the job submission script this time around (final line of `Part_B_GPU_job_1.sh`, but still not going to look into the xmacros).

B Regex

a) Started reading about regular expression, although perhaps not directly relevant to anything.

4 Sun

A Meeting with Julie

a) Currently a bit confused by the job submission script `GPU_XF_job.sh`

b) I think this is because I am not familiar enough with the file structure yet.

c) Also, I am not entirely sure what the `individual number` is for in this script.

5 Mon

A Meeting with Julie

a) Still on the job submission script of `Part B1`. I am confused about when exactly are the `indiv_parent_dir` created (the ones in $`XFproj/Simulations`)

B SLURM

Filename replacement are as follows:

a) `%x` - Job name, which can be given as `--job-name=<name>`

b) `%a` - Job array ID (index) number

c) `%A` - Job Array's master job allocation number

d) More options here: https://slurm.schedmd.com/sbatch.html

Note that the following are not the same:

a) $`SLURM_ARRAY_JOB_ID`: this is the `%A` from above

b) $`SLURM_ARRAY_TASK_ID`: this is the index (`%a` from above)

6 Tue

A Meeting with Julie

a) Started reading Julie's candidacy paper.

7 Wed

A Meeting with Julie

a) Read everything aside from Chapter 5.

b) **TODO:** Go over Chapter 3 again before reading chapter 5.

c) Ask Julie about Figure 6,7 8 and 9 (a discussion on all of chapter 3.1.2 if possible)

d) **Zenith Angle:** The angle at which the muon traveled into the IceCube detector, with respect to the vertical. Source: https://user-web.icecube.wisc.edu/~krosenau/index.html

B Bash

a) A backtick is not a quotation mark. Instance: `Part_B_GPU_job1.sh` line 37

```
for i in ‘seq 1 $NPOP‘
```

Everything we type inside the backticks is executed before the main command (such as `chown`), and the output of the backticked-command is then read by the main command

8 Thu

A Meeting with Julie

a) Regarding Part A of the June 5th entry, from Alex:

> those are created by XF. It's like a backend thing, we never make them ourselves but by virtue of simulating antennas they get made by XF as the location to store the antenna simulation data

In other words, we do not ourselves `mkdir` the $`indiv_dir_parent`.

b) Meeting with Julie today moved to Monday next week.

c) ~~It seems unlikely to be a bug given how long this script has been used, but I am getting an error from the for loop through all frequencies in freqlist. Will ask Alex about it.~~ I'm an indiot. I need to initialize $`GeoFactor` first.

d) Finished `Loop_Parts/Part_B1` second pass, moving on to `Part_B2`. Adding line breaks in the process; **Be careful with the trailing whitespace!** *Mayhap it's better to just leave them as single long lines?*

e) **TODO:** sometime in the future I'll need to figure out what `simulation_PEC.xmacro` and `output.xmacro` do to really understand what `Part_B1` and `Part_B2` are about.

9 Fri

A Meeting with Julie

a) ~~Probably trivial, but it seems like the uan files are already being moved to the correct directory at the end of Part_B2, so I am not sure why we do it again at the beginning of Part_C.~~

b) One of the `mv` command might be extra; will check with Alex.

c) `Part_C` second pass: digging into the python codes; should be able to finish by tomorrow.

B Bash

a) The dollar sign works inside the double quotes so there is no need for string concatenation in the example below:

```
$a=2;$b=2
echo "a * b = $(($a*$b))"
```

The above outputs `a * b = 4`. Note that the $((...)) part is for arithmetics.

b) At some point I should try to figure out exactly what double quotes are for in shell scripts; they seem to be more than just strings?

10 Sat

A Meeting with Amy

a) Get in touch with Nicholas to see what he's up to.

b) Go to Monday's collaboration meeting this week if possible to see if there's a project for me.

B Python

(a) XFintoARA.py

a) instead of using the `%`'s for string interpolation, we could use `f—strings`.

b) inside the curly braces we can call variables, for instance {`g.WorkingDir`} in line 66:

```
uanName = f'{g.WorkingDir}/.../{g.gen}_{indiv}_{freqNum}.uan'
```

c) The `g` above is from line 102: `g = parser.parse_args()`

d) `Line 73`: `mat = [["0" for x in range(n)] for y in range(m)]` is simply python's way to do `mat=zeros(m,n)` in Matlab. (`List Comprehension`)

12 Mon

A Python

(a) XFintoARA.py

The following pertains `line 81 \& 82`

a) `line 81` first turns the third entry in the list `lineList` into a float.

b) `line 82` contains the following: `"%.2f" % 10` which turns `10` into `10.00`. This is NOT modulo operation; this is probably more like string interpolation.

c) Mostly finished reading this python script. Moving on to `Part C` tomorrow.

(b) General

a) To access the `help` message of `argparse`'s `add_argument` function, one can run the command `python3 <filename>.py -h` at the terminal

b) For more info on the `argparse` module one can look through the documentation of `argparse` on https://docs.python.org/3/library/argparse.html and the tutorial at https://towardsdatascience.com/a-simple-guide-to-command-line-arguments-with-argparse-6824c30ab1c3

c) ~~Reviewed file opening (with open, etc)~~

d) mode `w` and `w+`: `w` is write whereas `w+` is read and write

e) apparently with python3, when using `os.chmod` one needs to add `0o` (zero-oh) in front of `777` to grant all read-write-execute, etc. For instance:
`os.chmod("<filename>", 0o770)`
This is because

> In unix conventions, written numbers are assumed to be decimal unless they are prefixed with a `0x` (or `0X`) in which case they are hexadecimal

(https://stackoverflow.com/questions/32729309/what-is-the-purpose-of-the-octal-digit-0-permission) and according to the error message we "use an 0o prefix for octal integers".