

CS336 : Parallel & Distributed processing

Project 3 Report

Summary of tasks

For this project the main aim was to understand how circadian clocks work and then actually implement the infrastructure for our simulations. This project focused on incorporating intercellular signaling (also called "coupling") into our model of the circadian clock. We simulated the model with increasing coupling strengths to determine which coupling strengths work best for particular distributions of periods. I lastly measured performance by timing the runs. My coding was updating the methods for `phase_support.c` which is the heart of the simulation and it runs statistical analyses through methods within it.

Tasks

Adding the following methods to `phase_support.c`

My first task was to incorporate support for VIP signaling in the model. I updated the definition of `ParamStruct` to include `VIPStrength` and also allowing `sim_slow` to take a parameter to set as the VIP strength. To implement the VIP coupling and I added a phase Equation to support the `runPhaseSimulation`, as it takes in the current phases and then calculates mean VIP inputs sent from each of the cells. For each phase the VRC value was calculated as well, and all these values combined, they updated the next phases.

Running the phase simulations using `sim_slow`, I got similar return results for the time-series as last week.

```
mkhanyisis-MacBook-Pro:project 3 mggamedz$ ./sim_slow uncoupled_5.phs 5 0.0
setting vip strength 0.000000
ret 0.000000 0.000000
mkhanyisis-MacBook-Pro:project 3 mggamedz$ ./sim_slow coupled_5_1.phs 5 0.1
setting vip strength 0.100000
ret 0.000000 0.000000
mkhanyisis-MacBook-Pro:project 3 mggamedz$ ./sim_slow coupled_5_10.phs 5 1.0
setting vip strength 1.000000
ret 0.000000 0.000000
mkhanyisis-MacBook-Pro:project 3 mggamedz$
```

I then ran `sim_slow` to run the phase simulations and generate the .phs file for the coupled oscillators. My output results when run with `disp_phase` were similar to the expected results from Stephanie.



To generate lots of simulations for one VIP strength starting from random initial conditions, I had to create the generate Gaussian periods function. This method gets tested each time it runs, as it prints the list of Gaussian periods. Running `sim_stats_1s` for the I got this result:

```
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 0 1
Testing generated Gaussian periods:
****
24.304028 24.320503 23.140527 24.208733 24.433165 24.693570 23.820898 2
4.150174 24.023724 24.622083 23.996791 24.236402 23.886011 24.447322 24.
027097 23.568520 23.965225 23.640806 23.147519 24.145263 24.019294 23.71
8755 23.554438 24.492714 23.920324 24.505594 23.462998 24.669890 23.9444
57 24.389617 23.953851 23.251005 24.970381 23.531962 23.854965 23.250015
23.985879 24.807309 24.669407 23.648007 23.648044 24.235051 23.668967
24.015952 24.037904 24.113158 24.019676 23.589367 24.219134 24.763877 2
4.179602 23.388600 22.972789 24.382622 24.096386 24.804973 24.676910 24.
352128 24.034852 24.030909 25.047330 23.982795 24.373338 23.371675 23.63
9000 23.433753 24.005481 24.241816 23.622685 25.224436 23.155289 24.5365
01 23.910726 23.036080 23.629803 24.843475 24.680889 24.793545 24.265516
24.116546 23.820577 24.282716 24.757444 24.920616 23.717100 23.484794
23.341267 23.865838 23.630131 24.181378 24.199489 23.836596 23.966603 2
5.371073 23.938347 24.592037 24.212363 23.663148 23.695549 23.736949 ****
***** DONE *****
begin
end
** begin 2
end2
Event #0 stdev= 0.094794
Event #1 stdev= 0.471302
Event #2 stdev= 0.854351
Event #3 stdev= 1.229961
Event #4 stdev= 1.609562
Event #5 stdev= 1.989159
Event #6 stdev= 2.365993
Event #7 stdev= 2.740422
Event #8 stdev= 3.119799
Event #9 stdev= 3.496702
final stats for trial 0: 3.496702
Ran in 0.041399 seconds
[mkgamedz@nscc project 3]$
```

```
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 0 5
4.332466 3.900654 4.621284 4.884109 4.420533
Ran in 0.191593 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 0.5 5
4.332466 3.900654 4.621284 4.884109 4.420533
Ran in 0.199064 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 0.8 5
4.332466 3.900654 4.621284 4.884109 4.420533
Ran in 0.189811 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 0.1 5
4.332466 3.900654 4.621284 4.884109 4.420533
Ran in 0.188891 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.1 5
0.912903 0.788334 1.014466 2.802297 1.039267
Ran in 0.183691 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1. 5
4.057537 3.561202 4.348075 4.638028 4.145527
Ran in 0.189050 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.05 5
3.338858 2.137350 3.583132 4.213817 3.465449
Ran in 0.187652 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.3 5
0.546255 0.495860 0.635623 0.659564 0.579816
Ran in 0.180537 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.4 5
0.517116 0.469553 0.601766 0.617668 0.557954
Ran in 0.178366 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.6 5
0.493893 0.457062 0.536073 0.600972 0.511108
Ran in 0.176901 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.8 5
0.479082 0.431699 0.512002 0.570466 0.495200
Ran in 0.176885 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.9 5
0.471550 0.424239 0.505322 0.552002 0.500057
Ran in 0.176442 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 1.99 5
0.471550 0.424239 0.505322 0.541486 0.489043
Ran in 0.179995 seconds
[mkgamedz@nscc project 3]$ ./sim_stats_1s 10 100 2 5
0.000000 0.000000 0.000000 0.000000 0.000000
Ran in 0.129949 seconds
[mkgamedz@nscc project 3]$
```


As noticed above, my stats show that at VIP strength less than 1, the standard deviations do not decrease with ease. However at around 1.2 VIP strength the standard deviations decrease significantly, being zero at VIPstrength=2.

Lots of Simulations for multiple VIP strengths

Running sim_stats_ns for multiple VIP strengths, I got this result:

```
VIP strength 0.000000: 4.332466 3.900654 4.621284 4.884109 4.420533
VIP strength 0.100000: 5.104018 4.700558 5.081276 4.467252 5.008208
VIP strength 0.200000: 4.654535 4.554870 4.670632 4.850364 4.713840
VIP strength 0.300000: 4.318232 5.199366 4.611677 4.844898 4.291091
VIP strength 0.400000: 4.926987 4.937625 4.564587 4.814632 4.127325
VIP strength 0.500000: 4.689870 4.724523 4.035441 4.633072 4.532549
VIP strength 0.600000: 4.888723 5.102257 4.890816 4.111214 4.603699
VIP strength 0.700000: 4.322162 4.720510 4.739507 4.469780 4.399096
VIP strength 0.800000: 4.094335 4.675188 4.914634 4.480349 4.584900
VIP strength 0.900000: 5.187675 4.833203 5.184364 4.517783 4.810350
VIP strength 1.000000: 4.884824 4.565618 3.799076 4.251957 4.126727
VIP strength 1.100000: 1.035347 2.917715 1.467843 3.658348 1.855346
VIP strength 1.200000: 0.847191 0.795325 0.929595 0.674217 0.801182
VIP strength 1.300000: 0.788954 0.570330 0.565037 0.617459 0.540306
VIP strength 1.400000: 0.584323 0.656667 0.532623 0.750549 0.555945
VIP strength 1.500000: 0.678095 0.661289 0.564981 0.548685 0.560529
VIP strength 1.600000: 0.564945 0.500242 0.572257 0.571181 0.505861
VIP strength 1.700000: 0.504895 0.457100 0.576209 0.527094 0.483919
VIP strength 1.800000: 0.439467 0.508311 0.465994 0.594142 0.541240
VIP strength 1.900000: 0.527294 0.555109 0.539000 0.557795 0.570602
VIP strength 2.000000: 0.000000 0.000000 0.000000 0.000000 0.000000
Ran in 3.890572 seconds
[mggamedz@nscc project 3]$
```

Running sim_stats_ns for 3 different period distributions(standard deviation 0.1, 0.25 and 0.5), I got :

Standard deviation 0.5

```
[mggamedz@nscc project 3]$ ./sim_stats_ns 10 100 5
VIP strength 0.000000: 4.332466 3.900654 4.621284 4.884109 4.420533
VIP strength 0.100000: 5.104018 4.700558 5.081276 4.467252 5.008208
VIP strength 0.200000: 4.654535 4.554870 4.670632 4.850364 4.713840
VIP strength 0.300000: 4.318232 5.199366 4.611677 4.844898 4.291091
VIP strength 0.400000: 4.926987 4.937625 4.564587 4.814632 4.127325
VIP strength 0.500000: 4.689870 4.724523 4.035441 4.633072 4.532549
VIP strength 0.600000: 4.888723 5.102257 4.890816 4.111214 4.603699
VIP strength 0.700000: 4.322162 4.720510 4.739507 4.469780 4.399096
VIP strength 0.800000: 4.094335 4.675188 4.914634 4.480349 4.584900
VIP strength 0.900000: 5.187675 4.833203 5.184364 4.517783 4.810350
VIP strength 1.000000: 4.884824 4.565618 3.799076 4.251957 4.126727
VIP strength 1.100000: 1.035347 2.917715 1.467843 3.658348 1.855346
VIP strength 1.200000: 0.847191 0.795325 0.929595 0.674217 0.801182
VIP strength 1.300000: 0.788954 0.570330 0.565037 0.617459 0.540306
VIP strength 1.400000: 0.584323 0.656667 0.532623 0.750549 0.555945
VIP strength 1.500000: 0.678095 0.661289 0.564981 0.548685 0.560529
VIP strength 1.600000: 0.564945 0.500242 0.572257 0.571181 0.505861
VIP strength 1.700000: 0.504895 0.457100 0.576209 0.527094 0.483919
VIP strength 1.800000: 0.439467 0.508311 0.465994 0.594142 0.541240
VIP strength 1.900000: 0.527294 0.555109 0.539000 0.557795 0.570602
VIP strength 2.000000: 0.000000 0.000000 0.000000 0.000000 0.000000
Ran in 3.919763 seconds
[mggamedz@nscc project 3]$
```

Standard deviation 0.1

```
[mkgamedz@nsc project 3]$ ./sim_stats_ns 10 100 5
VIP strength 0.000000: 0.869768 0.781644 0.925855 0.977965 0.885105
VIP strength 0.100000: 1.023104 0.940566 1.021525 0.895671 0.998383
VIP strength 0.200000: 0.933983 0.910031 0.933671 0.971460 0.940752
VIP strength 0.300000: 0.864377 1.034791 0.927620 0.967033 0.865269
VIP strength 0.400000: 0.989653 0.988366 0.918538 0.964352 0.827807
VIP strength 0.500000: 0.939899 0.947137 0.803609 0.928137 0.903204
VIP strength 0.600000: 0.981246 1.018267 0.979557 0.827102 0.928021
VIP strength 0.700000: 0.864751 0.951381 0.951023 0.894484 0.881333
VIP strength 0.800000: 0.820649 0.935795 0.977241 0.897590 0.913656
VIP strength 0.900000: 1.037918 0.964503 1.038831 0.902644 0.962054
VIP strength 1.000000: 0.131716 0.124657 0.107853 0.120187 0.112814
VIP strength 1.100000: 0.103841 0.120372 0.102378 0.123171 0.112475
VIP strength 1.200000: 0.115469 0.109194 0.108597 0.098352 0.104464
VIP strength 1.300000: 0.120653 0.102193 0.097320 0.105463 0.092176
VIP strength 1.400000: 0.106573 0.118491 0.098700 0.118680 0.096795
VIP strength 1.500000: 0.118966 0.122018 0.107612 0.104041 0.102944
VIP strength 1.600000: 0.107588 0.099390 0.112530 0.106441 0.099592
VIP strength 1.700000: 0.098039 0.094284 0.107963 0.108894 0.093611
VIP strength 1.800000: 0.089958 0.104448 0.091012 0.115170 0.108895
VIP strength 1.900000: 0.105046 0.106228 0.108149 0.107796 0.110663
VIP strength 2.000000: 0.000000 0.000000 0.000000 0.000000 0.000000
Ran in 3.654859 seconds
[mkgamedz@nsc project 3]$
```

Standard deviation 0.25

```
[mkgamedz@nsc project 3]$ ./sim_stats_ns 10 100 5
VIP strength 0.000000: 2.166322 1.951696 2.316062 2.443589 2.216069
VIP strength 0.100000: 2.554081 2.350907 2.538548 2.230536 2.503849
VIP strength 0.200000: 2.327286 2.277642 2.337515 2.426384 2.354812
VIP strength 0.300000: 2.159077 2.602814 2.305980 2.427797 2.147785
VIP strength 0.400000: 2.464472 2.470144 2.284599 2.410551 2.061245
VIP strength 0.500000: 2.346852 2.363011 2.014958 2.316198 2.267616
VIP strength 0.600000: 2.447589 2.553367 2.443452 2.057422 2.308302
VIP strength 0.700000: 2.161340 2.360143 2.369014 2.237801 2.201977
VIP strength 0.800000: 2.045923 2.343169 2.456943 2.241634 2.292504
VIP strength 0.900000: 2.593171 2.415935 2.600026 2.257205 2.401389
VIP strength 1.000000: 1.538837 0.620848 0.419084 0.544811 0.442603
VIP strength 1.100000: 0.290183 0.321298 0.275565 0.345973 0.301739
VIP strength 1.200000: 0.307973 0.298269 0.306166 0.264253 0.289451
VIP strength 1.300000: 0.316126 0.247827 0.245647 0.272028 0.238861
VIP strength 1.400000: 0.266808 0.290947 0.249191 0.305801 0.256362
VIP strength 1.500000: 0.308680 0.300094 0.259546 0.260341 0.253483
VIP strength 1.600000: 0.268213 0.238470 0.269190 0.267943 0.239636
VIP strength 1.700000: 0.252125 0.221337 0.275994 0.261069 0.230721
VIP strength 1.800000: 0.216361 0.250209 0.226132 0.278342 0.266308
VIP strength 1.900000: 0.259703 0.271314 0.253262 0.267921 0.275453
VIP strength 2.000000: 0.000000 0.000000 0.000000 0.000000 0.000000
Ran in 3.715155 seconds
[mkgamedz@nsc project 3]$
```

A VIP strength of 0.5 allows the most similar set of CT6 crossings in the 10th cycle, as there the deviation is much smaller.

Extensions

None

Collaborators

I got help from Brandon and Zhuofan on this project