CS 475 Parallel Computing
Instructor: Matthew Meyn
Student: Khuong Luu (luukh)

Project 4

1. What your own-choice quantity was and how it fit into the simulation.

1.1. My own choice of quantity: Number of Wolfs

Each month, the number of wolfs:

- More wolfs are born only when there are more than 1 wolf in the pack.
- If the number of wolfs is greater than 1, there is 50% chance a new wolf is born that month. If this happens, 10 wolfs will be born
- The percentage of chance that the wolf pack attack the deers is proportional to the number of wolfs and the number of deers
- In the event of wolf-attack, the bigger the value of grain height, the less deers are killed because as higher grain, deers can get away from wolf due to lack of sight. Also, the more there are wolfs in the attack, the more grain will be storm down during the chase. If there are no wolf-attack in a month, a quarter of population will die that month
- The wolfs come at the end of the month, when the grain has grown and the deers have been well fed (when both values are updated)
- 2 wolf eat 1 deer. If there are more deers than the required numbers, it's fine, but if there are not enough deers for all the wolf, the wolf population will decrease by (required deers - actual deers)/2

The code that compute and update the number of wolfs is:

```
int ComputeWolf() {
   int tmpWolf = 0;
   tmpWolf += WolfBorn();
   tmpWolf += WolfStarved(); // add negative values
   return tmpWolf;
}
```

```
void UpdateWolf(int tmpWolfs) {
   NowNumWolf += tmpWolfs;
   if (NowNumWolf < 0)
      NowNumWolf = 0;
}
```

The threads for the Wolf has the same control flow as the Grain and the Deer, which is:

```
void Wolf() {
   int tmpWolfs = ComputeWolf();
   #pragma omp barrier // computing barrier 1
   UpdateWolf(tmpWolfs);
   #pragma omp barrier // updating barrier 1
}
```

The overall program sections are organized as following:

```
InitData();
   UpdateFactors(); // init the factor for the 1st time

   for (int iStep = 1; iStep <= NUM_STEPS; ++iStep) {
      out << iStep;
      #pragma omp parallel sections default(none) shared(NowGrainHeight, NowNumDeer,
NowNumWolf, LatestGrainStomped, LatestNumDeersEaten, out)
      {
         #pragma omp section
         {
            Grain();
         }
         #pragma omp section
         {
            Deer();
         }
         #pragma omp section
         {
            Wolf();
         }
         #pragma omp section
         {
            Watcher(out);
         }
      } // omp parallel sections
      out << endl;
   } // for NUM_STEPS
```

Due to the big number of lines of code, please refer to my source code attached in this submission for more specific implementation of the above quantity function.

Because of this quantity, I have also chance they way I compute and update the existing agents as following:

1.2. Height of Grain (NowGrainHeight)

In general, each month, the height of grain:

- grows proportionally to how nice the weather is
- is eaten by deers
- is reduced due to the wolfs chase the deers whenever there's a wolf attack

Each month,

- grain grows an amount of:

tempFactor * precipFactor * 8

where:
tempFactor = exp((-1) * pow((float)((NowTemperature - MIDTEMP)/10), 2.0f))
precipFactor = exp((-1) * pow((float)((NowPrecip - MIDPRECIP)/10), 2.0f))

- grain is eaten by deers in an amount of:

NowNumDeer * ONE_DEER_EATS_PER_MONTH
where ONE_DEER_EATS_PER_MONTH = 0.1 (inch)

- grain is stomped by the chase between wolf and deer in the 50% chance of wolf attack

If a wolf attack happen, the height (in inches) that are stomped down is:
LatestNumDeersEaten * 0.05f;
Where:
LatestNumDeersEaten = **NowNumWolf/2 – NowGrainHeight*2.0;**
or **0** depends on the latest state of the game

1.3. Number of Deers

In general, each month, the number of deers:

- increase when there are enough grain to feed
- decrease due to being eaten by wolfs if there's a wolf attack that month

Each month,

- the number of deer is **computed** by:

```
int ComputeDeers() {
   float requiredGrain = NowNumDeer * ONE_DEER_EATS_PER_MONTH;
   float diffGrain = NowGrainHeight - requiredGrain;
   float diffDeers = diffGrain / ONE_DEER_EATS_PER_MONTH;
   return diffDeers;
}
```

- and is **updated** by:

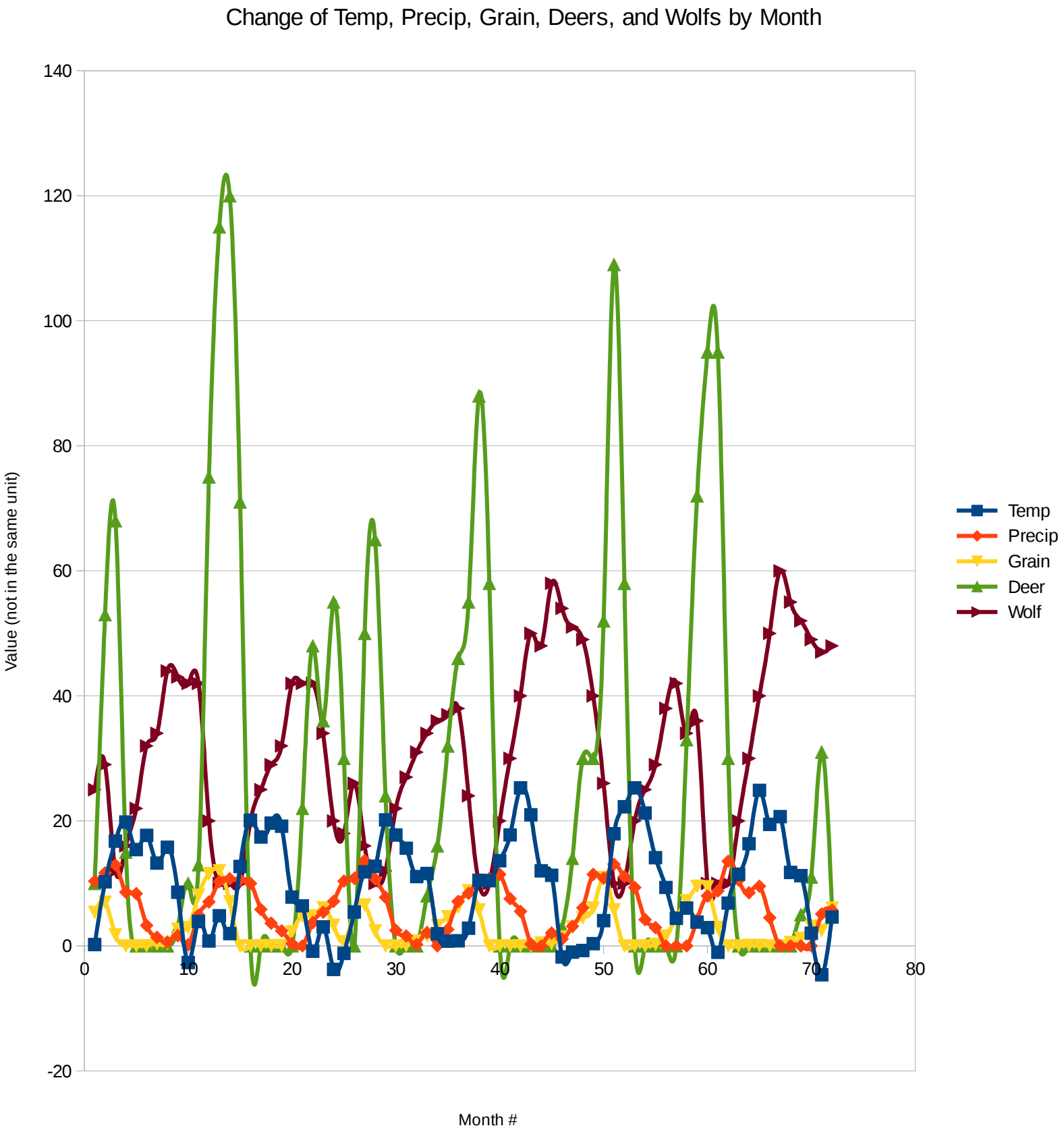
```
void UpdateDeers(int tmpDeers) {
   NowNumDeer += tmpDeers;
   NowNumDeer -= LatestNumDeersEaten;
   if (NowNumDeer < 0)
      NowNumDeer = 0;
}
```

2. A table showing values for temperature, precipitation, number of deers, height of the grain, and your own-choice quantity as a function of month number.

Notes: temperature in C degree and height in inches

Month#	Month	Year	Temp	Precip	Grain	Deer	Wolf
1	1	2018	0.210372	10.3517	5.36986	10	25
2	2	2018	10.2517	11.67	6.9787	53	29
3	3	2018	16.7631	12.9477	1.73242	68	12
4	4	2018	19.8024	8.59645	0	15	16
5	5	2018	15.4236	8.35612	0	0	22
6	6	2018	17.6817	3.27131	0.0174113	0	32
7	7	2018	13.2512	1.37337	0.325416	0	34
8	8	2018	15.7688	0.615148	0.377425	0	44
9	9	2018	8.58316	1.64849	2.66384	3	43
10	10	2018	-2.65064	0.189002	2.96186	10	42
11	11	2018	3.89577	4.94514	8.09785	13	42
12	12	2018	0.796293	7.044	11.5607	75	20
13	1	2019	4.82372	10.2849	12.0171	115	10
14	2	2019	1.97689	10.7414	7.04875	120	10
15	3	2019	12.7128	10.5917	0	71	10
16	4	2019	20.1131	9.999	0	0	20
17	5	2019	17.4118	5.8344	0.028948	0	25
18	6	2019	19.6432	3.63648	0.0319456	0	29
19	7	2019	19.1482	2.39975	0.0360197	0	32
20	8	2019	7.81325	0.365571	2.22518	0	42
21	9	2019	6.4141	0	4.82059	22	42
22	10	2019	-0.845506	3.73124	4.80162	48	42
23	11	2019	3.02598	5.40905	6.07238	36	34
24	12	2019	-3.76088	7.12238	3.30367	55	20
25	1	2020	-1.19136	10.376	0.658259	30	18
26	2	2020	5.42314	10.8129	5.36272	0	26
27	3	2020	11.8416	13.6652	6.55071	50	16
28	4	2020	12.7372	10.7265	2.40801	65	10
29	5	2020	20.169	7.7812	0	24	12
30	6	2020	17.7417	2.48319	0	0	22
31	7	2020	15.6237	1.58386	0.0686959	0	27
32	8	2020	11.086	0.219679	0.804881	0	31
33	9	2020	11.581	2.1379	1.6328	8	34
34	10	2020	1.93833	0.0100948	3.23881	16	36
35	11	2020	0.763164	2.63061	4.63483	32	37
36	12	2020	0.809258	7.11827	6.23305	46	38
37	1	2021	2.84254	8.40823	8.81063	55	24
38	2	2021	10.4951	10.558	5.74611	88	10
39	3	2021	10.426	10.7813	0	58	10
40	4	2021	13.6395	11.4526	0	0	20
41	5	2021	17.7574	7.54683	0.0241583	0	30
42	6	2021	25.2808	5.52255	0.0241634	0	40
43	7	2021	20.9561	0.290568	0	0	50
44	8	2021	12.0156	0	0.459418	0	48
45	9	2021	11.2947	2.05671	0.390015	0	58
46	10	2021	-1.78398	1.09651	1.42028	3	54
47	11	2021	-0.992413	3.13129	3.03568	14	51
48	12	2021	-0.739646	6.08643	4.5092	30	49
49	1	2022	0.346065	11.4401	6.05629	30	40
50	2	2022	4.05345	10.8387	10.961	52	26
51	3	2022	17.9425	13.0678	5.78085	109	10
52	4	2022	22.2474	11.0802	0	58	10
53	5	2022	25.2737	9.34711	0	0	20
54	6	2022	21.2438	4.21559	0.000611837	0	25
55	7	2022	14.1242	2.9291	0.233698	0	29
56	8	2022	9.35376	0	1.58161	0	38
57	9	2022	4.41059	0	4.52454	0	42
58	10	2022	6.06837	0	7.22655	33	34
59	11	2022	3.79872	4.17113	9.54567	72	36
60	12	2022	2.91872	7.9895	9.47059	95	10
61	1	2023	-1.03173	8.785	2.95396	95	10
62	2	2023	6.85015	13.5505	0	30	10
63	3	2023	11.4685	10.5388	0	0	20
64	4	2023	16.3445	8.51301	0.0795865	0	30
65	5	2023	24.8942	9.54056	0.079597	0	40
66	6	2023	19.4391	4.50633	0	0	50
67	7	2023	20.6677	0.00293696	0	0	60
68	8	2023	11.7505	0	0.522024	0	55
69	9	2023	11.2262	0	1.18519	5	52
70	10	2023	2.03632	0	3.1241	11	49
71	11	2023	-4.62825	5.09653	2.46104	31	47
72	12	2023	4.62385	5.73156	6.0216	6	48

3. A graph showing temperature, precipitation, number of deers, height of the grain, and your own-choice quantity as a function of month number.
Notes: temperature in C degree and height in inches


Change of Temp, Precip, Grain, Deers, and Wolfs by Month
Value (not in the same unit)
Month #
140
120
100
80
60
40
20
0
-20
0
10
20
30
40
50
60
70
80
Temp
Precip
Grain
Deer
Wolf

4. A commentary about the patterns in the graph and why they turned out that way.

Temperature and Precip is periodically by sin and cosin function. This is our original intention since the beginning.

When temperature and precip go near the ideal temperature and precip, which are around 10, respectively, the grain grow very high (in inches) and as those weather condition goes away from the those ideal values, the grain height generally tends to be reduced.

The grain height generally follow this pattern but not really stable because of the wolf attacks. The growth of grain helps the number of dear also grows strong because we design that deer growth is proptional to the height of the grain. As the number of deer goes up, the chance of being attacked by wolf also goes up. Once the number of deers is twice as much of the wolfs, the wolf attack. There are occasional attack due to the random chance and also from the fact that the number of the wolf born is also from 50% chance. In every wolf attack, not only the attack decreases the height of the grain but the height of the grain also affect back the number of deers get eaten (since we design that the higher the grain height, the less deer is eaten as the sight of wolf is hidden by the grain height) which in turn affect the grain height eaten by the surving deers afterward. These are the source of the unstabability of the grain height.

We can also notice that the number of deers not only increases very sharp and also decrease very sharp. This is because I have adjusted the value of ONE_DEER_EATS_PER_MONTH parameter from 0.4f to 0.1f. This cause the deer to be very easy to grow as they require fourth times less food to breed. But just as it increase quickly and sharply, it will reach the threshold at which the deers are too visible to the wolf pack (the grain height cannot hide them anymore) and that make the wolf attack which causes the dramatic decrease in the number of deers.

Like the deers, the wolf population pattern does generally behave in a opposite pattern of the deers. However, the wolf population pattern a little unpredictable and in some occation behave out-of-pattern because of many random factors I put in into the ComputeWolfs() function. What makes the general pattern of the wolf population is the deer population. The only major chance the wolf population increases is when the deer population suppasses the "twice as much" threshold and are fed to the wolf by "1 deer is eaten by 2 wolfs" ratio. And as the deers supplies decreases sharply, the number of wolfs die because of starving then increase, which decreases the population. Occasionally and randomly, the wolfs make unusally attack and gain population, which make the pattern of wolf population a bit unpredictable.