14 May 2018
Project #4
OSU CS475 Sp2018
Joshua L. Rogers
rogerjos@oregonstate.edu

Project 4: Functional Decomposition

Because the project was cast as a rough simulation of an ecosystem, I chose to make my own-choice quantity/phenomenon an attempt at simulating the effects of increased greenhouse gas emissions. I made certain assumptions in order to do so meaningfully, namely that the totality of the environment simulated was so small as to be potentially affected by greenhouse gas emission and absorption by the graindeer and grain respectively. Alternately, that the graindeer and grain had prodigious effects on the amount of greenhouse gases in the atmosphere. Come to think of it, I also assumed an atmosphere, but let's not get too bogged down in minutiae.

The simulation was implemented as follows: I complied with the base implementation as described in the project notes, finding that deviation from the given simulation constants and formulae was unnecessary. The code, when run with three threads will produce the basic simulation as described in the project notes. Other global values such as the starting month, starting year, ending year, and initial grain and graindeer quantities may be modified with compilation flags, although by default the code implements the project description precisely.

Next, I implemented my own function Greenhouse() to be included in the compiled program if the thread count is set to four threads by passing the compiler a "-DTHREADCOUNT=4" argument, first adding a global float NowCO2 to track the concentration of greenhouse gas (simplified to only CO2) over or under the baseline. That is, if there is 50% more greenhouse gas in the atmosphere than was initially present NowCO2 would equal 50.0. This value is clamped against -99 because we presume that even perfectly efficient grain cannot scrub *all* of the CO2 from the atmosphere. Also, in a staggering coincidence, this avoids a divide-by-zero issue in the SetWeather() function.

Greenhouse() works in much the same way as GrainDeer() and Grain(), first calculating a local value (NewCO2) based on global state, then updating the global value (NowCO2) when all other global variables are updated. NewCO2, is equal to the number of GrainDeer less the inches of Grain divided by 2.5, a value chosen because we presume that grain is less efficient per inch than a single graindeer is inefficient and also because it makes an interesting graph in which the grain can manage to mitigate the graindeer's environmental effects temporarily.

The NowCO2 global state variable is used in SetWeather(), which calculates the weather for each month. First the GasFactor is calculated, which is NowCO2 expressed as a decimal percentage (e.g., 25% = 0.25) plus 1. Then the GasFactor is incorporated into the temperature calculation by multiplying it by the initial temperature calculation, driving the average temperature up for high levels of CO2 and down for low levels of CO2, simulating the effect of greenhouse gas on temperature averages. Then the GasFactor is incorporated into the random noise by multiplying it by each of the interval bounds, increasing the amplitude of variance at high levels of CO2 and decreasing it at low levels, simulating the volatility that high levels of atmospheric CO2 can cause in weather patterns over shorter timespans. Similarly, the GasFactor is applied to the initial precipitation calculation as a divisor, thus decreasing the amount of precipitation at high levels of CO2 in simulation of the lower levels of precipitation encountered with greenhouse gas influenced climate change. The precipitation randomization is modified just as the temperature randomization is modified in order to simulate the increased volatility that accompanies an increased concentration of CO2.

Admittedly, this approach is less than nuanced. However, I believe that it achieves at least a level of accuracy and realism that befits a project in which extinction-proof graindeer subsist on one-dimensional grain.

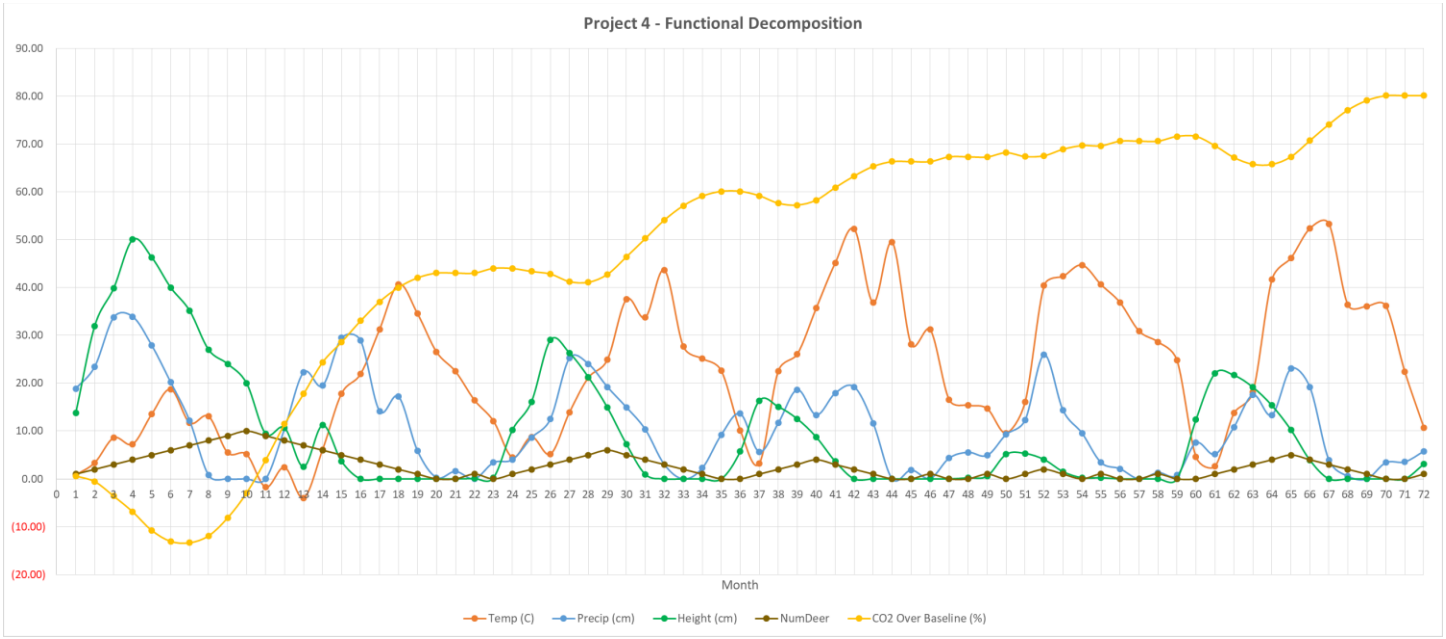
See below an arbitrary result from running the simulation showing values for temperature, precipitation, height of the grain, number of graindeer, and output of Greenhouse() as described above as a function of month number in tabular form. Because seventy-two months were simulated it was necessary to split the table into thirds for the sake of legibility:

Temp (C)	0.84	3.39	8.60	7.29	13.54	18.73	11.73	13.06	5.49	5.12	-1.75	2.42	-3.97	6.11	17.84	21.94	31.24	40.71	34.54	26.54	22.46	16.46	12.09	4.50
Precip (cm)	18.87	23.48	33.78	33.92	27.93	20.20	12.15	0.76	0.00	0.00	0.00	10.54	22.30	19.48	29.47	28.94	14.19	17.21	5.86	0.29	1.57	0.00	3.41	4.02
Height (cm)	13.76	31.98	39.87	50.02	46.32	40.00	35.16	26.98	24.04	19.97	9.45	10.66	2.52	11.23	3.66	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.24	10.25
NumDeer	1	2	3	4	5	6	7	8	9	10	9	8	7	6	5	4	3	2	1	0	0	1	0	1
CO2+ (%)	0.60	-0.57	-3.60	-6.88	-10.76	-13.06	-12.35	-11.89	-8.14	-2.93	3.93	11.44	17.76	24.36	28.60	33.02	37.02	40.02	42.02	43.02	43.02	43.02	44.01	43.97
Month/Year	0/2017	1/2017	2/2017	3/2017	4/2017	5/2017	6/2017	7/2017	8/2017	9/2017	10/2017	11/2017	0/2018	1/2018	2/2018	3/2018	4/2018	5/2018	6/2018	7/2018	8/2018	9/2018	10/2018	11/2018
Month #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Temp (C)	8.78	5.19	13.90	21.19	24.91	37.53	33.74	43.66	27.70	25.15	22.58	10.12	3.23	22.55	26.04	35.70	45.14	52.22	36.81	49.45	28.09	31.25	16.55	15.40
Precip (cm)	8.62	12.53	25.25	24.05	19.22	14.98	10.33	3.00	0.00	2.27	9.23	13.62	5.65	11.67	18.64	13.37	17.93	19.24	11.63	0.00	1.85	0.10	4.35	5.56
Height (cm)	16.12	29.03	26.34	21.26	14.91	7.29	0.94	0.00	0.00	0.00	0.00	5.77	16.36	15.09	12.55	8.74	3.66	0.00	0.00	0.00	0.00	0.00	0.00	0.23
NumDeer	2	3	4	5	6	5	4	3	2	1	0	0	1	2	3	4	3	2	1	0	0	1	0	0
CO2+ (%)	43.36	42.82	41.25	41.10	42.75	46.40	50.25	54.10	57.10	59.10	60.10	60.10	59.20	57.62	57.24	58.27	60.89	63.32	65.32	66.32	66.32	66.32	67.32	67.32
Month/Year	0/2019	1/2019	2/2019	3/2019	4/2019	5/2019	6/2019	7/2019	8/2019	9/2019	10/2019	11/2019	0/2020	1/2020	2/2020	3/2020	4/2020	5/2020	6/2020	7/2020	8/2020	9/2020	10/2020	11/2020
Month #	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

Temp (C)	14.71	9.56	16.08	40.38	42.32	44.71	40.60	36.87	30.91	28.61	24.78	4.60	2.62	13.76	18.19	41.66	46.14	52.36	53.33	36.37	36.11	36.14	22.40	10.74
Precip (cm)	4.90	9.30	12.30	25.93	14.32	9.55	3.41	2.12	0.00	1.32	0.78	7.53	5.14	10.85	17.57	13.34	23.06	19.16	3.94	0.57	0.00	3.40	3.56	5.77
Height (cm)	0.58	5.13	5.32	4.05	1.51	0.24	0.24	0.00	0.00	0.00	0.00	12.38	22.04	21.65	19.15	15.34	10.26	3.91	0.00	0.00	0.00	0.00	0.00	3.10
NumDeer	1	0	1	2	1	0	1	0	0	1	0	0	1	2	3	4	5	4	3	2	1	0	0	1
CO2+ (%)	67.28	68.19	67.38	67.54	68.90	69.67	69.63	70.59	70.59	70.59	71.59	71.59	69.64	67.17	65.76	65.74	67.33	70.71	74.10	77.10	79.10	80.10	80.10	80.10
Month/Year	0/2021	1/2021	2/2021	3/2021	4/2021	5/2021	6/2021	7/2021	8/2021	9/2021	10/2021	11/2021	0/2022	1/2022	2/2022	3/2022	4/2022	5/2022	6/2022	7/2022	8/2022	9/2022	10/2022	11/2022
Month #	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72

See below the above results in graph form, also showing temperature, precipitation, height of the grain, number of graindeer, and output of Greenhouse() as a function of month number.



When the program is run using only three threads, thereby simulating the scenario as described in the project notes, we see the expected sinusoid variance in precipitation and cosinusoid variance in temperature, both moderated somewhat by the randomization function to lend their curves a jagged appearance. Grain height increases are closely aligned with increases in precipitation, but spike and decrease rapidly as the temperature rises above 10C. Deer population, oddly only able increment or decrement its population each month regardless of total population, follows the grain supply, slowly rising as long as more grain is available and falling as grain height drops lower than deer population, thus describing a shallow wave that follows that of the temperature quite closely.

Adding the Greenhouse() function complicates matters by informing both the values of temperature and precipitation as a function of graindeer population. Because the graindeer population is dependent upon the grain height, which is in turn dependent upon the degree to which the temperature and precipitation level conform to their "ideal" values, a feedback loop is formed whereby an increase in grain leads to an increase in graindeer, which, having a greater effect on the CO2 level than the grain, pushes up the CO2 level, resulting in higher temperatures, lower precipitation, and on average a less hospitable environment for grain growth, which causes lower graindeer population. This is portrayed in the data above as follows:

The initial conditions are conducive to grain growth, so the grain height immediately increases much more quickly than the graindeer population can expand, which results in a net decrease in CO2 levels below the baseline value. The result is a colder environment with less volatile weather, which causes the grain growth to be reduced beginning at the 4th month. The graindeer population, however, is still increasing, and even though they outstrip the grain supply and begin to decline at month 10, the CO2 level increases to above baseline for good at month 11 and continues to increase sharply essentially as a function of the deer population due to low grain height resultant from unfavorable conditions until about month 20. As a consequence, the temperature begins to destabilize, peaking at 40C in month 18, which represents a significant departure from the unmodified simulation in which temperatures tend to range from about -5C to 25C. Precipitation is suppressed, and as a result the grain height craters from month 16 to month 23, only recovering when the temperature curve overcomes the CO2 effect and drops to just over 12C in week 23 along with a similar curve-induced increase in the precipitation levels. The grain grows quickly, peaking at just under 30cm in month 26. However, this grain growth brings with it an accompanying increase in the graindeer population. Note the drop in CO2 levels from month 24 to month 28 as the fast-growing grain absorbs the gas. However, inevitably the graindeer population increases, bringing with it another upward slope in the CO2 levels beginning at month 29. Additionally, the temperature once again follows its modified curve upward while the precipitation follows its modified curve downward, again driving grain height to zero from month 32 to 35 while the temperature peaks at over 43C in week 32. Again, the graindeer population drops and another brief period of lower temperatures and increased rain occurs at week 36, but the temperature wave grows increasingly wild, gradually increasing in both randomness and maximum temperature while the precipitation wave also increases in random amplitude and decreases in average precipitation over time as the CO2 value is driven ever upward. This results in a less grain-friendly environment except in the event that the low end of the temperature curve is not randomly driven out of the ideal growth zone and the precipitation during that time is sufficient for grain growth, in which case we see results as shown from week 60 through 62 on the graph. However, once again the opportunistic and quite ineradicable graindeer, whose population has been suppressed since month 44 due to an absence of grain, come surging back and trigger another reversal of the grain's small decrease in surplus CO2. As a result the temperature soars to a new maximum just over 53C, excess CO2 rises to 80% over baseline, and the graindeer population and grain height once again drop to around zero.

Thus we see that the Greenhouse() effects a feedback loop between the various elements of the simulation in which, at least for the constants chosen, the graindeer tend to create an environment not conducive to their or the grain's well-being. Because neither similar temperature and precipitation fluctuations nor the consequent suppression of grain height and graindeer population is present in the simulation when run without the NowCO2 state variable accumulating extra greenhouse gas it is clear that this quantity is actually affecting the simulation.