Design Study Report - Team 15

Jaime Martinez (jmartinez66)

Jane McDowell (jmcdowell7)

Katie Addison (kpeatman3)

Kevin Jones (kjones314)

Context

The purpose of this document is to compare and contrast four different approaches that mimic heat diffusion. Each program analyzed will simulate the diffusion of heat on a two-dimensional metal plate. Fick's Law shall be used to model the spreading of temperature in the system; the law states that a concentration of a substance will over time spread based on its diffusion constant. Higher concentration gradients yield faster spreading.

Simulations will approximate the solution to a given input using a difference equation. For two-dimensional diffusion, we do so by cutting a metal into a grid of equal sized cells which are bounded orthogonally. This allows for a simple procedure to simulate the diffusion of temperature, repeatedly calculating each cell's temperature based on the average of its neighbors. Each of the programs being analyzed will cater differently to several non-functional considerations: performance, precision, evolvability, usability, etc. Compiled data will enable reflection on what the trade-offs are amongst the programs.

Research Questions

1. What is the overall affect of varying the grid size with respect to performance and does it have any impact on precision?
2. What acceptable limits can be set without significantly impacting performance or precision?
3. What program variant consumes the most system resources?
4. Which of the set or capped variables pose the greatest impact on system resources?

Subjects

The following subjects will be evaluated on how they affect the non-functional requirements:

1. Higher precision primitives (doubles): Uses doubles to hold temperature values.
2. Lower precision primitives (floats): Uses floats to hold temperature values.
3. Wrapper object containers (Float): Uses Floats to hold temperatures values.
4. Class object for lattice representation (class): Uses a complex class object to represent the different cells in the dissected plate.
5. GUI representation (package): This facilitating program simply aggregates the other four programs and provides a simple user interface for parameter input.
6.

Experimental Conditions

This study will be performed on various systems with highly differing specifications; from different operating systems to laptops, desktops, and even virtual machines. The programming language of choice for the study shall be Java. The programs all require five parameters of input with varying requirements:

- Dimension: Specifies how fine the plate is to be dissected, the number of cells it will contain equals d^2.
- Top temperature: The constant temperature affecting the system from the top edge of the grid.
- Bottom temperature: The constant temperature affecting the system from the bottom edge of the grid.
- Left Temperature: The constant temperature affecting our system from the left edge of the grid.
- Right temperature: The constant temperature affecting our system from the top edge of the grid.

Additionally, the provided dimension must be an integer number greater than zero and less than or equal to 25. The external temperatures can be any real number within the range of 0 to 100 with precision limited by container type utilized. Confounding factors include near infinite execution which shall be limited by setting a maximum number of iterations, as well as determining lack of significant change.

Independent Variables

Name of Variable	Description	Unit
dimension	the grid size of the plate (number of rows by number of columns),	an integer value between 1 and 25
maxIterations	the maximum number of iterations that the diffusion simulation can run	an integer value set to 100
relChgStopCriteria	the minimum value of relative change for the computation to continue	an integer value set to 1.00
topEdgeTemp, bottomEdgeTemp, leftEdgeTemp, rightEdgeTemp	the edge temperatures of the plate or grid which retain a constant value throughout program execution	initially integer values between 0 and 100 supplied by the user and then parsed to the data type called for in the specific program selected (either double or float)
varSelection	the user selection that determines which program variant to use	Primitive ‘double’ array Primitive ‘float’ array Wrapped ‘Float’ array Primitive ‘double’ object

Dependent Variables

Name	Description	Unit
memory	Total memory used by program. Affected by relChgStopCriteria and dimension.	bytes and megabytes
runtime	Total memory available at runtime or the free memory at runtime. Affected by relChgStopCriteria	bytes and megabytes
numIterations	Total number of times the simulation was performed. Affected by relChgStopCriteria and by edge temperatures.	integer
totalTime	Total execution time of the program. Affected by changes in dimension and maxIterations.	milliseconds

Metrics

Table 1

Name	Definition
dimension	Dictates how many distinct cells constitute the plate, via dimension squared. Dimension is an integer value.
maxIterations	Specifies what is the max number of times the diffuse method can loop. This is held within an integer type.
relChgStopCriteria	Prohibits iterations if there is not a sufficient relative change during an iteration. An integer type is used to hold this quantity.
topEdgeTemp	Is the constant temperature in degrees Celsius emanating from the top edge of the plate. This value is represented via either a double, float, or Float depending on the program.
bottomEdgeTemp	Is the constant temperature in degrees Celsius emanating from the bottom edge of the plate. This value is represented via either a double, float, or Float depending on the program.

leftEdgeTemp	Is the constant temperature in degrees Celsius emanating from the left edge of the plate. This value is represented via either a double, float, or Float depending on the program.
rightEdgeTemp	Is the constant temperature in degrees Celsius emanating from the right edge of the plate. This value is represented via either a double, float, or Float depending on the program.
memory	Denotes how much memory the program used. The unit of measure is bytes.
runtime	Denotes how much runtime memory the program has used or has free. The unit of measure is bytes.
numIterations	Is how many times the program loops before reaching a stopping condition. This is tracked via an integer value, incremented once per iteration.
totalTime	How long from start to finish does the program take to execute the diffusion simulation. The unit of measure here is milliseconds.

Variable Summary

Table 2 - Independent and Dependent Variables

Question #	***Independent Variables***	***Dependent Variables***
1	*dimension*	*runtime*
2	*dimension, maxIterations, relChgStopCriteria*	*runtime, memory, numIterations*
3	*varSelection*	*runtime, memory*
4	*dimension, maxIterations, relChgStopCriteria*	*runtime, memory*

Methods

The experiment was conducted by executing the program on the command line via a script, which both supplied the different command line options as well as parsed the output for use in a table. For a given set of input, output data was collected from each program variant and recorded in an Excel spreadsheet. For the first data collection, the experiment was conducted using constant edge values and varying the values of dimension, maxIterations and relChgStopCriteria. For the second data collection, the experiment was conducted varying the value of dimension and relChgStopCriteria, while keeping the edge temperatures and the maximum number of iterations constant. For the third data collection, the experiment was conducted varying the temperature values and the value of dimension while keeping the values of maxIterations and relChgStopCriteria constant. The edge values were selected randomly however they were not generated using a specific tool. A total of 60 separate trials were conducted in this manner on mac OS X 64-bit architecture. A subset of the varied program criteria was conducted on a virtual machine running Ubuntu 3.11.0-17 64-bit architecture as well as Windows 7 64-bit architecture, a total of 10 separate trials per platform. Additionally, a set of 6 separate trials were performed per OS platform varying the heap size. The totalMemory() and freeMemory() methods from Java's Runtime class were used to obtain and calculate the program memory usage, and rounded to a precision of two significant digits. The currentTimeMillis() method from Java's System class was use to obtain and calculate the total program execution time and reported without rounding or altering the result. The table below details the arguments used for the different subjects.

Table 3 - Correlation of arguments to subjects

Subject of Study	Variable	Values used for trials	Tabulated values	Tossed out
Performance to grid size	Dimension	3,10, 25, 50, 100, 500, 1000	10, 25, 100	3, 50, 500, 1000
Performance to iterations	maxIterations	25, 40, 80, 100, 400, 500	25, 100, 500	40, 80, 400
Precision to relative temperature change	relChgStopCriteria	0.0, 0.05, 0.1, 0.25, 0.37, 0.5, 0.75, 0.87, 1.0	All listed	None

Analysis Techniques

The decision to represent the results in the GUI by printing out the temperature values as opposed to animation removed the need for setting the relChgStopCriteria to a lower value to accomplish smooth transition for the sake of the display. As a result, the data was analyzed by correlating computations of the lowest runtime and memory consumption data for each trial and determining the lowest acceptable value for maxIterations without forcing the program to exit. Outlying values were used insomuch as they helped make specific design decisions with respect to limiting the Dimension and setting the maxIterations and relChgStopCriteria, but once those decisions were made the outlying values were tossed out (not included in the results table). Error bounds were reported and used as one of the justifications for capping the value of maxIterations.

Although the data was interpolated for expected trends, no formal regression testing software was utilized.

Results

Table 4 Results of constant edge values and dimension, maxIterations = 25, varying relChgStopCriteria		mac OS X 64-bit Free mem 241 mb / Avail Mem 245 mb			
Parameters	Results	double	float	Float	object
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705584
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	19	19	26	66
relative change stop criteria 0.0	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705608
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	19	19	29	64
relative change stop criteria 0.05	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705608
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	19	19	28	64
relative change stop criteria 0.1	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705608
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	19	20	26	64
relative change stop criteria 0.25	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705608
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	19	19	27	65
relative change stop criteria 0.37	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705608
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	19	20	27	65
relative change stop criteria 0.5	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705608
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	21	20	28	66
relative change stop criteria 0.75	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705608
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	18	19	26	67
relative change stop criteria 0.87	# iterations performed	25	25	25	25
Dimension = 25	used memory (bytes)	1352768	1352768	2705480	2705584
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 25	runtime ms	19	20	27	67
relative change stop criteria 1.0	# iterations performed	25	25	25	25

Table 5 Results of constant edge values and dimension, maxIterations = 100, varying relChgStopCriteria		mac OS X 64-bit Free mem 241 mb / Avail Mem 245 mb			
Parameters	Results	double	float	Float	object
Dimension = 25	used memory (bytes)	1352768	1352768	4058112	2705480
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 100	runtime ms	21	21	34	67
relative change stop criteria 0.0	# iterations performed	100	100	100	100
Dimension = 25	used memory (bytes)	1352768	1352768	4058112	2705480
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 100	runtime ms	22	22	34	65
relative change stop criteria 0.05	# iterations performed	100	100	100	100
Dimension = 25	used memory (bytes)	1352768	1352768	4058112	2705480
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 100	runtime ms	21	21	33	67
relative change stop criteria 0.1	# iterations performed	100	100	100	100
Dimension = 25	used memory (bytes)	1352768	1352768	4058112	2705480
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 100	runtime ms	22	22	34	65
relative change stop criteria 0.25	# iterations performed	100	100	100	100
Dimension = 25	used memory (bytes)	1352768	1352768	4058152	2705456
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 100	runtime ms	21	22	36	64
relative change stop criteria 0.37	# iterations performed	90	90	90	90
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705480
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 100	runtime ms	21	21	32	66
relative change stop criteria 0.5	# iterations performed	65	65	65	65
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705520
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 100	runtime ms	20	21	31	66
relative change stop criteria 0.75	# iterations performed	43	43	43	43
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705448
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 100	runtime ms	21	21	29	64
relative change stop criteria 0.87	# iterations performed	37	37	37	37
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705536
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 100	runtime ms	19	20	30	64
relative change stop criteria 1.0	# iterations performed	32	32	32	32

Table 6 Results of constant edge values and dimension, maxIterations = 500, varying relChgStopCriteria		mac OS X 64-bit Free mem 241 mb / Avail Mem 245 mb			
Parameters	Results	double	float	Float	object
Dimension = 25	used memory (bytes)	1352768	1352768	12174152	2705520
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	11.61	2.58
Maximum # Iterations 500	runtime ms	22	22	41	68
relative change stop criteria 0.0	# iterations performed	500	500	500	500
Dimension = 25	used memory (bytes)	1352768	1352768	8116128	2705472
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	7.74	2.58
Maximum # Iterations 500	runtime ms	22	23	39	69
relative change stop criteria 0.05	# iterations performed	355	355	355	355
Dimension = 25	used memory (bytes)	1352768	1352768	6763456	2705472
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	6.45	2.58
Maximum # Iterations 500	runtime ms	22	24	36	68
relative change stop criteria 0.1	# iterations performed	261	261	261	261
Dimension = 25	used memory (bytes)	1352768	1352768	4058112	2705480
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 500	runtime ms	22	22	34	65
relative change stop criteria 0.25	# iterations performed	138	138	138	138
Dimension = 25	used memory (bytes)	1352768	1352768	4058152	2705456
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 500	runtime ms	21	24	34	67
relative change stop criteria 0.37	# iterations performed	90	90	90	90
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705448
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 500	runtime ms	21	22	33	65
relative change stop criteria 0.5	# iterations performed	65	65	65	65
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705520
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 500	runtime ms	20	21	31	66
relative change stop criteria 0.75	# iterations performed	43	43	43	43
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705448
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 500	runtime ms	20	22	31	66
relative change stop criteria 0.87	# iterations performed	37	37	37	37
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705536
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 500	runtime ms	20	20	29	65
relative change stop criteria 1.0	# iterations performed	32	32	32	32

Table 7 Results of constant edge values and maxIterations, varying dimension, and relChgStopCriteria

Vary Dimension and relChgStopCriteria		mac OS X 64-bit Free mem 241 mb / Avail Mem 245 mb			
Parameters	Results	double	float	Float	object
Dimension = 10	used memory (bytes)	1352768	1352768	1352768	1352768
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	1.29	1.29
Maximum # Iterations 100	runtime ms	6	7	12	27
relative change stop criteria 0.0	# iterations performed	100	100	100	100
Dimension = 10	used memory (bytes)	1352768	1352768	1352768	1352768
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	1.29	1.29
Maximum # Iterations 100	runtime ms	5	5	8	25
relative change stop criteria 1.0	# iterations performed	33	33	33	33
Dimension = 25	used memory (bytes)	1352768	1352768	4058112	2705480
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	3.87	2.58
Maximum # Iterations 100	runtime ms	21	23	35	64
relative change stop criteria 0.0	# iterations performed	100	100	100	100
Dimension = 25	used memory (bytes)	1352768	1352768	2705440	2705536
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 100	runtime ms	21	20	28	63
relative change stop criteria 1.0	# iterations performed	32	32	32	32
Dimension = 100	used memory (bytes)	4058272	4058160	36522272	23227520
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	3.87	3.87	34.83	22.15
Maximum # Iterations 100	runtime ms	184	180	214	487
relative change stop criteria 0.0	# iterations performed	100	100	100	100
Dimension = 100	used memory (bytes)	4058176	4058248	14879552	23227576
Top 100, Bottom 0, Left 75, Right 50	used memory (mb)	3.87	3.87	14.19	22.15
Maximum # Iterations 100	runtime ms	178	175	188	468
relative change stop criteria 1.0	# iterations performed	32	32	32	32

Table 8 Results of constant maxIterations and relChgStopCriteria, varying edge values and dimension

Vary Temperatures and Dimension		mac OS X 64-bit Free mem 241 mb / Avail Mem 245 mb			
Parameters	Results	double	float	Float	object
Dimension = 10	used memory (bytes)	1352768	1352768	1352768	1352768
Top 75, Bottom 0, Left 0, Right 50	used memory (mb)	1.29	1.29	1.29	1.29
Maximum # Iterations 100	runtime ms	4	5	7	17
relative change stop criteria 1.0	# iterations performed	22	22	22	22
Dimension = 25	used memory (bytes)	1352768	1352768	2705544	2705456
Top 75, Bottom 0, Left 0, Right 50	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 100	runtime ms	17	18	26	70
relative change stop criteria 1.0	# iterations performed	23	23	23	23
Dimension = 100	used memory (bytes)	4058224	4058200	12174208	20522096
Top 75, Bottom 0, Left 0, Right 50	used memory (mb)	3.87	3.87	11.61	19.57
Maximum # Iterations 100	runtime ms	170	167	184	435
relative change stop criteria 1.0	# iterations performed	23	23	23	23
Dimension =10	used memory (bytes)	1352768	1352768	1352768	1352768
Top 30 Bottom 10, Left 25, Right 40	used memory (mb)	1.29	1.29	1.29	1.29
Maximum # Iterations 100	runtime ms	4	4	5	14
relative change stop criteria 1.0	# iterations performed	14	14	14	14
Dimension = 25	used memory (bytes)	1352768	1352768	1352768	2705456
Top 30 Bottom 10, Left 25, Right 40	used memory (mb)	1.29	1.29	1.29	2.58
Maximum # Iterations 100	runtime ms	19	17	24	59
relative change stop criteria 1.0	# iterations performed	13	13	13	13
Dimension = 100	used memory (bytes)	4058184	4058216	9468888	19169608
Top 30 Bottom 10, Left 25, Right 40	used memory (mb)	3.87	3.87	9.03	18.28
Maximum # Iterations 100	runtime ms	171	172	184	439
relative change stop criteria 1.0	# iterations performed	13	13	13	13
Dimension =10	used memory (bytes)	1352768	1352768	1352768	1352768
Top 10 Bottom 50, Left 50, Right 80	used memory (mb)	1.29	1.29	1.29	1.29
Maximum # Iterations 100	runtime ms	4	4	8	18
relative change stop criteria 1.0	# iterations performed	28	28	28	28
Dimension = 25	used memory (bytes)	1352768	1352768	2705456	2705448
Top 10 Bottom 50, Left 50, Right 80	used memory (mb)	1.29	1.29	2.58	2.58
Maximum # Iterations 100	runtime ms	19	19	27	64
relative change stop criteria 1.0	# iterations performed	25	25	25	25
Dimension = 100	used memory (bytes)	4058224	4058224	13526920	23227736
Top 10 Bottom 50, Left 50, Right 80	used memory (mb)	3.87	3.87	12.9	22.15
Maximum # Iterations 100	runtime ms	178	173	185	453
relative change stop criteria 1.0	# iterations performed	25	25	25	25

Table 9 Analysis of Tables 4, 5, 6	mac OS X 64-bit	
Vary relChgStopCriteria and maxIterations	Free mem 241 mb / Avail Mem 245 mb	
maxIterations	relChgStopCriteria	# iterations
25	0.00	cap reached
	0.05	cap reached
	0.10	cap reached
	0.25	cap reached
	0.37	cap reached
	0.50	cap reached
	0.75	cap reached
	0.87	cap reached
	1.00	cap reached
100	0.00	cap reached
	0.05	cap reached
	0.10	cap reached
	0.25	cap reached
	0.37	90.00
	0.50	65.00
	0.75	43
	0.87	37
	1.00	32.00
500	0.00	cap reached
	0.05	355.00
	0.10	261.00
	0.25	138.00
	0.37	90.00
	0.50	65.00
	0.75	43
	0.87	37
	1.00	32.00

Table 10 Analysis of Table 7		mac OS X 64-bit			
Vary Dimension and relChgStopCriteria		Free mem 241 mb / Avail Mem 245 mb			
Dimension	relChgStopCriteria	highest runtime (ms)	program variant	highest memory use (mb)	program variant
10	0.00	27.00	object	1.29	equal
	1.00	25.00	object	1.29	equal
25	0.00	64.00	object	3.87	Float
	1.00	63.00	object	2.58	Float/object
100	0.00	487.00	objecct	34.83	Float
	1.00	468.00	object	22.15	object

Table 11 Analysis of Table 8		mac OS X 64-bit			
Vary Temperatures and Dimension		Free mem 241 mb / Avail Mem 245 mb			
Dimension	Top, Bottom, Left, Right	highest runtime	program variant	highest memory use	program variant
10	75, 0, 0 50	17	object	1.29	equal
25	75, 0, 0 50	70	object	2.58	Float, object
100	75, 0, 0 50	435	object	19.57	object
10	30, 10, 25, 40	14	object	1.29	equal
25	30, 10, 25, 40	59	object	2.58	object
100	30, 10, 25, 40	439	object	18.28	object
10	10, 50, 50, 80	18	object	1.29	equal
25	10, 50, 50, 80	64	object	2.58	Float, object
100	10, 50, 50, 80	453	object	22.15	object

Discussion

In the creation of the five programs and many trials performed, the team was able to gain a further appreciation for the intricacies of comparing and contrasting software. At the start we had our assumptions, some which proved to be true, and some not. We thought from a memory and run time standpoint that the ordering of efficiency would be: float, double, Float, and object. The data solidified and debunked some of the projections.

We encountered some phenomena such as the memory consumption of double and float being on par. Even though double offers more precision via more bits, it did not consume more memory. The team believes this can be rationalized due to both primitive objects taking the same amount of memory on a 64 bit machine. Another odd case is with Float taking up more memory than object. This result varies depending on relChgStopCriteria. If it is below 0.5, and sufficient iterations occur, then Float consumes more memory; above 0.5 the memory usage is equal on small dimensions, but on bigger dimensions the memory usage is higher on object. This represents the fact that Float takes more memory than double regardless of the object(s) it is contained in.

To improve this study we feel a 32-bit machine would have helped in further solidifying why float and double consumed equal memory. The anomalous case of Float using more memory than the object approach was noted and led to selecting a relChgStopCriteria value of 1.0 for our programs. In hindsight it is easy to see know why we were partly correct in some of our projections when the experiments clearly highlighted the nuances.

Conclusions

1. The overall affect of varying the grid size is that runtime and memory consumption increases as the dimension variable is increased. It has a no affect on precision, only that it allows a wider representation of the changing temperature values to be displayed to the user.
2. It is acceptable to set limits on the grid size as the first result above proved it has no impact on precision and further, the size limitations of the GUI display contribute to limiting this variable for ease of viewing. Furthermore, increasing the value of maxIterations significantly increases runtime and, in combination with a low value for relChgStopCriteria, presents serious memory constraints. It is evident that for a set relChgStopCriteria value, raising the maxIterations above a certain value in which program execution can still proceed does not have any affect on precision. Finally, given that the GUI is not employing animation for display purposes, the need for a low value of relChgStopCriteria is not necessary and thus should be limited so that in combination with maxIterations it does not contribute to performance degradation.
3. Both the ArrayFloatObjectPlate and LatticePlate implementations consume the most memory resources, being relatively equal to each other but about a 200 % increase over the other two implementations. The LatticePlate had the highest execution times of all programs, in most cases an increase of more than 300 % greater than the other implementations. The combination of both maxIterations and relChgStopCriteria, when not capped or restricted, posed the highest threat to system resources.