ICE ICE POWER BABY

New Mexico Supercomputing Challenge
Final Report
April 3rd 2018

RRCA-1
Rio Rancho Cyber Academy

Dylan Martinez
Christopher Meyer
Gabe Turner

Harry Henderson

Table of Contents

Executive Summary

This study will compare a **normal refrigerator** to a **forced refrigerator** [**normal refrigerator** with modified time dependent thermostat]. Both refrigerators are identical and have the same initial conditions except where noted and will be in a house that receives all the power it needs during the day from solar panels and from the grid during the night. Note there is a small amount of power loss due to transmission over power lines from the power plant.

A **normal refrigerator** will use a set amount of energy over a given time [ie over a 10 day period an average power per hour can be calculated].

A **forced refrigerator** is set to its lowest temperature possible without freezing/damaging the food [eg thermostat is set to a narrow range just above safest coldest temperature] during the day and then the refrigerator is not powered during the night until it reaches is safest warmest temperature without spoiling the food [eg thermostat is set to a narrow range just below safest warmest temperature]. The forcing of colder temperatures during the day even with allowing warmer temperatures at night will result in an increased overall power use as compared to **normal refrigerator**, however, energy loss over power lines will be reduced.

> The **hypothesis** is that a **forced refrigerator** that maximizes power from the sun while minimizing power from the grid will use less overall power than a **normal refrigerator** after power loss due to transmission over power lines is taken into account.

Introduction

Energy loss is an all encompassing problem that if not dealt with effectively, can bring devastating results. Everyday fossil fuels are used all around the world to power and run daily functions. Functions such as running household appliances consume immense amounts of energy each day. Such appliances include: refrigerators, ovens, and microwaves. This is the main purpose of this project, to focus on the conservation of energy for the biggest power consuming household appliance, the refrigerator.

Description

This project hopes to aid in the resolution of the problem. The structure of this project involves separating the model into daytime & nighttime cycles. During the daytime, the refrigerator will be forced just above the coldest applicable temperature. Throughout the entirety of the daytime cycle and at night time, the refrigerator's thermostat will be set just below the highest applicable temperature. Both temperature ranges are set by the USFDA. Thus, the refrigerator will not need power unless it goes above 40°F, and even so, it will require very minimal power to maintain 40°F. Therefore, the refrigerator will use little to no power obtained from the grid during the nighttime cycle. The goal is to use only power obtained from the solar panels to power the refrigerator. During the entirety of these periods the refrigerator door will remain shut.

Description Cont.

The total power of a refrigerator using a normal thermostat range that gets power from the grid will be approximately 10% more than the rated consumption of the refrigerator due to energy losses over power lines. The number is 10% because this is the approximate amount of energy that is lost when to heat when energy is transferred from residential areas to the power grid. Forcing the refrigerator to be colder (33°F) will use more energy than if the refrigerator was allowed to fluctuate between the normal thermostat range even if the refrigerator uses no power during the night. We hope to show this forced diurnal cycle will use less total power than a refrigerator hooked solely to the grid.

Background

The original end goal of this project was to conserve energy that is once lost by the relocation of energy from one appliance to another. What was expected of the end result for the model was to accurately simulate, and provide conscientious results on forcing a refrigerator to its coldest temperature. The reasoning behind forcing the refrigerator to its coldest temperature was to test a hypothesis that forcing a refrigerator to it's coldest temperature would be a more efficient way of using energy and another method of storing power locally and not on the grid. The initial idea was a result of collaborative brainstorming with our teacher, Harry Henderson, after which we expanded on the idea. The idea was expanded by using a solar panel for power in order to minimize power from the grid. The idea was also expanded to include thinking of the refrigerator as a battery bank.

Research

The initial research that needed to be gathered was based off the temperatures that refrigerated food can be held at. This data was found through the CDC (check to ensure this is accurate).This research needed to be gathered in order to make the initial stages of the model. The next piece of research that was conducted as the average dimensions and energy use of typical refrigerators (single-door, top-freezer). This was found through Energy Star's online resources (see bibliography). This research (although not used yet) will be used at the final stages of the project and will be used to determine if the project is energy-saving or not.

Coding

In the first model (Appendix A), a background was made that included 10 clouds, a ground and a sky but this left a large black space in the middle of the screen. It has also been decided that this area of code is not needed so it will likely be taken out in the future.

In the next model (Appendix B), a procedure called create-suns was created which makes a sun. Also, the sun breed was created so the sun can move throughout the day as it will possibly be added later. Next a procedure named progress was created, which will move the sun and clouds but as stated before, the background is irrelevant and will be taken out later.

In the next model (Appendix D), 3 new breeds were added: an internal-air breed, an external-air breed, and a fridge breed. Even though these breeds have not actually been added into the model, most of them will not be used later in the model. The internal-air is going to be the air inside the fridge and these will have their own temperature variables. The external-air is going to be the air outside the fridge and they will also have a temperature variable, even though the breed external-air was created, it will not be used since the warming be a constant number. The fridge breed was not used later as we changed the fridge from being a breed to being patches. The patch color inequalities had to be changed to make the bottom half of the screen become green and the top half become cyan.

In the next model (Appendix E), a create-fridges procedure was made, which made a fridge, with a center at (0, 13), set its size to 25, and its shape was made into a rectangular container. This procedure was also placed in the setup procedure.

Coding Cont.

In the next model (Appendix F), an internal-airs procedure was added, which makes 20 internal-air breeds. Their color was turned to red, their shape to be a circle, and the xcor to be a random number from 0 to 15 and the ycor to be a random number from 0 to 25. This procedure was also added to the setup procedure.

In the next model (Appendix G) we added a procedure named fill-fridge, this was used to make the internal air agents to spread out inside the fridge. We did this by making a random x coordinate and y coordinate and if this was not on a white patch then it would repeat this until it was. This means that the internal air is in the fridge. The only problem with this model was that we did not make enough internal air agents.

In the next model (Appendix H), we added the variable heat which is the degrees (in Fahrenheit) a certain agent is. Also, we made the agents all get made inside the setup procedure rather than having separate procedures for each breed. Next, we added a go procedure which will be used to move everything around. We also added a procedure called fridgerator which makes the fridge by changing patch color based on inequalities. After, we added a bounce procedure which makes it to where when an internal air agent gets near the edge then it turns around and goes the opposite way. However the bounce procedure did not work since the turtles headings were still wrong.

Coding Cont.

In the next model (Appendix I), we added a move procedure and we updated the bounce procedure. First, if the internal air agent has a temperature between 32 and 41 and the patch it is on is white then it will slightly turn both directions randomly and then go forward. This will cause it to not just go forward and rather slightly move. Also, to resolve the bounce problem above we decided to change the heading rather than making it turn so then it will flip 180 degrees around. The main problem with this model is the area where it must be in a certain temperature range to be able to move since in a real life situation this would not happen.

In the next model (Appendix J), we added a variable called thermostat which is a value that is either 1 or 0(like binary). We also made it to where if the average temperature is higher than 40 degrees then it will turn on the thermostat. If the average temperature is lower than 35 degrees then it will turn the thermostat off. Also we added an area in the code where it would warm and cool the air particles depending on if it gets near the edge or near the blue space in the center. If it gets near the blue space in the center and the thermostat is on then it will cool the air particles. We also added a procedure called cool which makes the blue square. Also we made the adjust procedure which would keep the air's temperature to be between 42 and 32. We took this out since the air can be other values above and below this.

In the next model (Appendix K), we added ticks into the model which is a sense of time for it to go through the go procedure 1 time. Also we made it to where every time the go procedure is ran 1 time and the thermostat is on then it will increase the variable thermostat-on-time by 1.

Coding Cont.

In the next model (Appendix L), we made it to where the heating and cooling values are variables rather than constant values so that way we are able to change them. The heating value was the amount of heat that is given off when the fridge gets to the side or how good the insulation is. The cooling value was how much colder it got when the air got near it or how efficient the cooling mechanism was. Also, we made it to where the internal air won't go through the blue square when the thermostat is off.

In the next model (Appendix M), we changed the heat range to be at 34 instead of 35 however we also had 33 at times for some models. Also we added a variable called flag which is the amount of times the thermostat either turns on or off. The way this works is when by creating another variable called x, this is going to be the thermostat value that was used the previous time the go procedure was used. If x does not equal thermostat, then it increases the flag by 1. Once the flag gets to 2 then the program stops. If the program is still running, then x is set to be thermostat and this cycle is done again until flag does equal 2. We used this to calibrate our model to find a 50% on-off cycle and to find out about how long 1 cycle on and off was. However in the end we did not use the cycle times since we found our values were wrong.

Coding Cont.

In the next model (Appendix N), we added time into it to find out how many times it turned on and off throughout the day and night. We also made it to where we cooled the temperature to be around 33 degrees on average during the day and 41 during the night. The number 33 was chosen so then it is the coldest it can be during the day without making the food freeze. 41 was chosen during the night since we want to have the highest temperature so then we don't have to have the fridge turn on for a long period of time. We made these by adding a new part to the go procedure which is nearly the same as the day part except for the temperature range. Also, we renamed flag to be dflag or the amount of times it changes in the daytime. We did the same thing with the variable y, which has the same function as x, and with nflag. We had y get set to be the thermostat value at the exact time it switches from day to night. However, the thermostat-on-time was getting added 2 times when the thermostat was on. This was because the algorithm that increased the thermostat-on-time was in the program twice. We also found that the dflag and nflag were very similar values which meant the time scale might of been wrong.

In the next model (Appendix O), we wanted to find how much time it took to heat from 32 degrees to 40 degrees to confirm that the time scale was wrong. This value should be a few hours if we had very poor insulation and should last possibly a couple days with very good insulation. However we found that these values were not taking this long and were in fact taking less than an hour (shown in the data). With this data though we were able to find that 400000 ticks per day was a better scale.

Coding Cont.

In the next model (Appendix P), we changed the ticks per day to be 400000. This model was built for a 10 day time frame but so it can switch on and off throughout the day we add a new variable called counter which is the time throughout 1 day. This still had the problem in the thermostat-on-time that will be fixed later.

Data

Data was gathered using NetLogo's Behavior Space. It was determined that in order to verify and validate the model, the parameters that would give a 50% on-off cycle needed to be found. This needed to be found in order to get the model to work like a real refrigerator. The first experiment produced invalid data due to an error in the code (See Appendix Q). The error that occurred was that in the Behavior Space experiment the heating and cooling rates were meant to change but the code made these change at a constant rate. This error led the data to always give a 70% on-off cycle regardless of heating & cooling values.

This was the second experiment performed in Behavior Space (see Appendix R for data). There were more values but these specific 5 were chosen because they yielded accurate and relevant results. These specific values were chosen because they exhibit the 50% on-off cycle. Because of the geometry used in the model the cooling was 10 times the value of the heating. the surface area of the outside of the fridge is about 10 times more than the area of the cooling area inside the fridge. Another experiment needed to be run because although this experiment yielded some data, it didn't yield enough data to verify and calibrate the model.

Data Cont.

In this data table (see Appendix S), there are three columns representing the three components: cooling, warming, and the average temperature of the fridge which all correspond to values on the table. Each row represents the change after each run in the program (1 day). The cooling column represents how much the refrigerator is cooled down when it starts cooling. The warming column represents how much the refrigerator is warmed up when the cooling component isn't on. The final column represents the average temperature for that run.

This experiment (see Appendix T for data) was run the same way as the last one, except the step was included and a day flag as well as a night flag. Although, in this experiment, some changes were made in Behavior Space to ensure that most of the cooling and warming values were constant. Although the error that occurred that required fixing had to do with the step. The step should not have remained constant, which was what needed to be fixed. The day flag represented how many times the refrigerator turned on or off during the day and the night flag represented how many times the refrigerator turned on or off during the night.

The next experiment (see Appendix U for data) that was conducted is flawed because the refrigerator kept turning on and off throughout the night (night-flag). So, what this data shows is the values of the warming and cooling, the step, the thermostat value, day-flag, and the night-flag. The day flag is a marker of how many cycles occur during the day time hours (AM). The Night flag is also also a marker of how many cycles occur during the night time hours (PM).

Data Cont.

The next data set (see Appendix V for data) accessible is flawed due to the need of more data to demonstrate the hypothesis as true and because the thermostat is extremely high. It was gathered using the previous gathered data while adding a few more things to it. The first thing was that the average of an off cycle was multiplied by 24 to get the ticks per day. The next step was to calculate which number will give a good insulation. Good insulation will allow the refrigerator to stay cool longer. After this number was found (400000), the model was rerun using the half of that number (200000) for each day/night. The selected grey rows are the relevant ones due to them having warming values that are 10 times the cooling values. One significant finding that resulted from this specific experiment is that the night flag is about half the day flag. This was expected because at night, the refrigerator is off for most of the night until it reaches the higher threshold of the temperature ranges, but even when it is turned on uses no energy because it uses the energy that was stored during the day. During the day, the flag is higher because it takes energy to force the refrigerator to it's coldest temperature, but after this occurs, the refrigerator turns off, storing energy.

Results

The results that were gathered from all of the above data wasn't quite what was preferred. The most recent data verified that the model could work in a real world situation. Although, due to the limitation of time, not enough data could be gathered to verify that this would be an energy saving concept if exposed to real-world factors. Some of which would include opening the refrigerator door opening throughout the day, different outside temperatures on certain days, percent cloud cover that blocks the solar panels from producing as much energy, etc. Due to the inability to verify if the hypothesis was energy-saving when exposed to environmental conditions such as those listed above, certain restrictions and assumptions had to be made. One of these assumptions was that the refrigerator door was always closed during the model.

Conclusion

The hypothesis was verified to be a energy saving concept through the model. Are hypothesis was verified through multiple Multiple Behavior Space Experiments which allowed us to gather a lot of data. With this data we demonstrated and can support our hypothesis to the fullest with no limits to each experimental run in the future. If this is made into a two-year project, then the data will be expanded to include real-world factors such as those listed in the results. Another way that this idea could be carried out would be to force the refrigerator to it's coldest temperature at sunset, and then storing the energy throughout the night. Which might be a future step of the project.

Appendix A

```
breed [cloud clouds]  ;defines clouds as a breed


to setup  ;makes a setup procedure
  clear-all  ;clears everything from before
  ask patches ;has the patches do something
  [
    if pycor < -25  ;if a patch is below -25 on the y axis
  [
    set pcolor green  ;it turns the patch green
  ]
    if pycor > 50   ;if a patch is above 50 on the y axis
    [
      set pcolor cyan  ;it turns the patch cyan
    ]
  ]
  create-clouds  ;calls on the create-clouds procedure
end    ;ends procedure


to create-clouds    ;starts the create-clouds procedure
create-cloud 10     ; makes 10 clouds
  [
    set color white   ;makes color white
    set size 10       ;makes size 10
    set shape "cloud" ;makes the shape a cloud
    set xcor random 200   ;makes the xcor anything from 0 to 200
    set ycor 95       ;makes the ycor 95
  ]

end           ;ends procedure
```

Appendix B

```
    [
      set pcolor cyan  ;it turns the patch cyan
    ]
  ]
  create-clouds  ;calls on the create-clouds procedure
  create-suns    ;calls on the create-suns procedure
end    ;ends procedure



to create-clouds    ;starts the create-clouds procedure
create-cloud 10     ; makes 10 clouds
  [
    set color white   ;makes color white
    set size 10       ;makes size 10
    set shape "cloud" ;makes the shape a cloud
    set xcor random 200   ;makes the xcor anything from 0 to 200
    set ycor 95       ;makes the ycor 95
  ]

end           ;ends procedure

to create-suns     ;starts the create-suns procedure
  create-sun 1     ; makes 1 sun
  [
    set color yellow  ;makes the sun's color yellow
    set shape "sun"   ;makes the shape a sun
    set size 10       ;makes the size 10
    set xcor -95      ;sets the x coordinant to be -95
    set ycor 50       ;sets the y coordinant to be 50

  ]
end                ;ends procedure




to progress ; Procedure that simulates clouds moving throughout the day.


end        ;ends program
```

Appendix C

```
breed [internal-air internal-airs] ;defines the internal air procedure
breed [external-air external-airs] ;defines the external air procedure
breed [fridge fridges]             ;defines the fridge procedure
breed [cloud clouds]  ;defines clouds as a breed
breed [sun suns]      ;defines suns as a breed

to setup  ;makes a setup procedure
  clear-all  ;clears everything from before
  ask patches ;has the patches do something
  [
    if pycor < 1 ;if a patch is on the bottom half of the screen then...
    [
    set pcolor green   ;...it makes the patch green
    ]
    if pycor > 0 ;if a patch is on the top half of the screen then...
    [
      set pcolor cyan  ;...it makes the patch cyan
    ]
  ]
   create-clouds  ;calls on the create-clouds procedure
  create-suns    ;calls on the create-suns procedure
end    ;ends procedure
```

Appendix D

```
to setup  ;makes a setup procedure
  clear-all  ;clears everything from before
  ask patches ;has the patches do something
  [
    if pycor < 1 ;if a patch is on the bottom half of the screen then...
    [
    set pcolor green   ;...it makes the patch green
    ]
    if pycor > 0 ;if a patch is on the top half of the screen then...
    [
      set pcolor cyan  ;...it makes the patch cyan
    ]
  ]
   create-clouds  ;calls on the create-clouds procedure
  create-suns    ;calls on the create-suns procedure
    create-fridges              ;calls on the create fridges procedure
end    ;ends procedure
```

Appendix E

```
to create-fridges      ;this makes the create-fridges procedure
  create-fridge 1      ;makes 1 fridge agent
  [
    set color white     ;makes the color of the fridge white
    set shape "container"     ;makes the shape a container which is essentially a rectangle
    setxy 0 13            ;sets its center to be at the coordinants (0,13)
    set size 25           ;has the size of the fridge to 25
  ]
end
```

Appendix F

```
   create-clouds  ;calls on the create-clouds procedure
  create-suns    ;calls on the create-suns procedure
    create-fridges             ;calls on the create-fridges procedure
  create-internal-airs   ;calls on the create-internal-airs procedure
end    ;ends procedure
```

```
to create-fridges     ;this makes the create-fridges procedure
  create-fridge 1     ;makes 1 fridge agent
  [
    set color white    ;makes the color of the fridge white
    set shape "container"    ;makes the shape a container which is essentially a rectang
    setxy 0 13          ;sets its center to be at the coordinants (0,13)
    set size 25         ;has the size of the fridge to 25
  ]
end


to create-internal-airs    ;this makes the create-internal-airs procedure
 create-internal-air 20    ;makes 20 internal-airs
  [
   set color red            ;sets there color to red
   set shape "circle"       ;sets the shape to a circle
   set xcor random 15       ;make the xcor be anywhere from 0 to 15 on the x
    set ycor random 25      ;make the ycor be anywhere from 0 to 25 on the y

   ]

end                         ;ends procedure
```

Appendix G

```
to fill-fridge                      ;makes the fill-fridge procedure of the internal air
  setxy random-xcor random-ycor     ;makes the x coordinate and y coordinate a random number
  if pcolor != white                ;if the color is not white...
  [
    fill-fridge                     ;it will call the fill fridge procedure again
  ]
end                                 ;ends the procedure
```

Appendix H

```
turtles-own [heat]


to setup ; Creates the ground, sky, procedure for creating clouds, procedure for creating the sun, and proc
clear-all                    ;clears everything
  ask patches                ;has the patches do something
  [
    if pycor < -95           ;if the y coordinate is smaller than -95 then...
  [
    set pcolor green         ;...it makes the patch green
  ]
    if pycor > -95           ;if the y coordinate is larger than -95 then...
    [
      set pcolor cyan        ;...it will change the patch to be green
     fridgerator             ;and also it will call upon the fridgerator procedure
   ]
  ]
  create-cloud 10 ;makes 10 cloud agents
  [
    set color white  ;makes color white
    set size 10      ;makes size 10
    set shape "cloud";makes shape cloud
    set xcor random 200  ;sets the x coordinate to be between 0 and 200
    set ycor 95      ;sets the y coordinate to be 95
  ]


to fridgerator                    ;makes the fridgerator procedure
  if pxcor > -33 and pxcor < 33 and pycor > -95 and pycor < 33
;if the x coordinate is between -33 and 33 and the y coordinate is between -95 and 33 then...
  [
    set pcolor white   ;...make the patch color white
  ]
end                    ;ends the procedure

|
to bounce ; Procedure that prevents the air particles inside the fridge from going outside of the fridge
ask internal-air    ;has the breed internal air do something
  [
    if patch-ahead 2 != white   ;if the patch 2 spaces ahead is white then
    [
      left 180                  ;turn left 180 degrees
      forward 1                 ;and go forward 1 space
    ]
  ]
end                             ;ends the procedure

to go      ;this makes the go procedure
  bounce   ;calls upon the bounce procedure
end        ;ends the current procedure
```

Appendix I

```
to move                      ;
 if heat >= 32 and heat <= 41 and pcolor = white
 ;if the heat is between 32 & 41 and the color is white
  [
    left random 25  ;go left a random amount of degrees between 0 and 25
    right random 25 ;go right a random amount of degrees between 0 and 25
    forward 1       ;go forward 1 space
  ]
end                 ;ends procedure

to bounce ; Procedure that prevents the air particles inside the fridge from going outsid
ask internal-air                     ;has the internal air do something
  [
    if patch-ahead 1 != white        ;if the patch ahead of the internal air is not white
    [
      set heading heading - 180      ;makes the heading 180 degrees smaller
      forward 1                      ;goes forward 1 space
    ]
  ]
end                                  ;ends the procedure
```

Appendix J

```
  set thermostat 1          ;this sets the thermostat value to be 1 which is on
end                         ;ends the procedure


to go                       ;starts the go procedure
ask internal-air            ;has the internal air do what is below
  [
   ;adjust                  ;calls upon the adjust procedure
   if mean [heat] of internal-air > 40 ;if the average temperature of the internal air is warmer than 40 degrees
    [
      set thermostat 1                 ;turn the thermostat on
    ]
   if mean [heat] of internal-air < 35 ;if the average temperature of the internal air is colder than 35 degrees
    [
      set thermostat 0                  ;turn the thermostat off
    ]
    move                                ;calls on the move procedure
   bounce                               ;calls on the bounce procedure
  ]
end                          ;ends the go procedure


to bounce ; Procedure that prevents the air particles inside the fridge from going outside of the fridge
;ask internal-air                  ;did not need, already in an ask internal-air above
; [
  if [pcolor] of patch-here = cyan    ;if the patch ahead is cyan then
    [
      set heading heading + 180        ;turn backwards
      forward 1                        ;move forward 1
      set heat heat + .001             ;change the heat variable to be .001 more which means it warmed up

      ]
  if [pcolor] of patch-here = blue and thermostat = 1     ;if the patch ahead is blue and the thermostat is on
    [
     set heading heading + 180              ;turn backwards
     forward 1                              ;move forward 1
     set heat heat - .1                     ;change the heat variable to be .1 less which means it cools down
    ]
;  ]
end


to cool                   ;starts the cool procedure
  if pxcor < 10 and pxcor > -10 and pycor > -10 and pycor < 10  ;if a patch is between 10 and -10 on the x and y
  [
    set pcolor blue              ;set the patch to be blue
  ]
end                              ;ends the procedure


;to adjust
;if heat > 42
;  [
;    set heat 42
;  ]
;if heat < 32
;  [
;    set heat 32
;  ]
;end
```

Appendix K

```
  tick                      ;this adds a tick to the total amount of ticks
  if thermostat = 1         ;if the thermostat is on
  [
   set thermostat-on-time thermostat-on-time + 1     ;increases the thermostat-on-time to be 1 larger than before
  ]
end             ;ends the current procedure
```

Appendix L

```
      set heat heat + warming
    ;it makes an internal air particle's temperature to be the warming value plus it's heat
      ]
  if [pcolor] of patch-here = blue       ;if the patch ahead is blue
    [
     set heading heading + 180      ;turn around
     forward 1              ;go forward 1 space
      if thermostat = 1    ;if the thermostat value is 1 or on
      [

     set heat heat - cooling
    ;it makes an internal air particle's temperature to be the cooling value minus it's heat
      ]

    ]
;  ]
end
```

Appendix M

```
  if mean [heat] of internal-air > 40       ;if the average temp is more than 40 degrees
   [
     set thermostat 1                      ;turn the thermostat on
   ]
  if mean [heat] of internal-air < 34       ;if the average temp is less than 34 degrees
   [
     set thermostat 0 ; Thermostat is off
   ]
   move                 ;calls upon move procedure
  bounce                ;calls upon bounce procedure
 ]
 tick                   ;adds 1 tick
 if thermostat = 1      ;if the thermostat is on
 [
  set thermostat-on-time thermostat-on-time + 1     ;add 1 to the thermostat-on-time
 ]
 if x != thermostat      ;if x is not thermostat
     [
       set flag flag + 1  ;set the value of flag to be increased by 1
     ]
 if flag = 2              ;if the flag is at 2
     [
     stop                 ;stop the program
     ]
 set x thermostat         ;sets the x value to be the same as the thermostat value
end
```

Appendix N

```
  ifelse ticks >= 857781                 ;if it is less than or equal to 857781
[
   if mean [heat] of internal-air > 34        ;if the average is warmer than 34 degrees
  [
    set thermostat 1                          ;turn the thermostat on
  ]
 if mean [heat] of internal-air < 32          ;if the average is less than 32 degrees
  [
    set thermostat 0                          ;the thermostat is off
  ]
    if thermostat = 1                  ;if the thermostat equals 1
    [
      set thermostat-on-time thermostat-on-time + 1       ;add 1 to thermostat-on-time
    ]
    if x != thermostat                ;if x is not the thermostat value
    [
      set dflag dflag + 1             ;add 1 to dflag
    ]
    set x thermostat                  ;set x to be thermostat
  ]
  [
    if ticks = 857782                 ;if the ticks are exactly 857782
    [
     set y thermostat                 ;make the value of y be equal to thermostat
    ]
   if mean [heat] of internal-air > 42    ;if the average is warmer than 42 degrees
  [
    set thermostat 1                 ;turn the thermostat on
  ]
    if mean [heat] of internal-air < 40    ;if the average is coolder than 40 degrees
  [
    set thermostat 0                 ;turn the thermostat off
  ]
    if thermostat = 1                ;of the thermostat is on
[
 set thermostat-on-time thermostat-on-time + 1  ;increase the thermostat-on-time by 1
]
if y != thermostat         ;if y is not equal to thermostat
    [
      set nflag nflag + 1   ;increase nflag by 1
    ]
set y thermostat       ;set y to be thermostat
  ]
  move
 bounce
]
tick
if thermostat = 1          ;if thermostat equals 1
[
 set thermostat-on-time thermostat-on-time + 1    ;increase the thermostat-on-time by 1
```

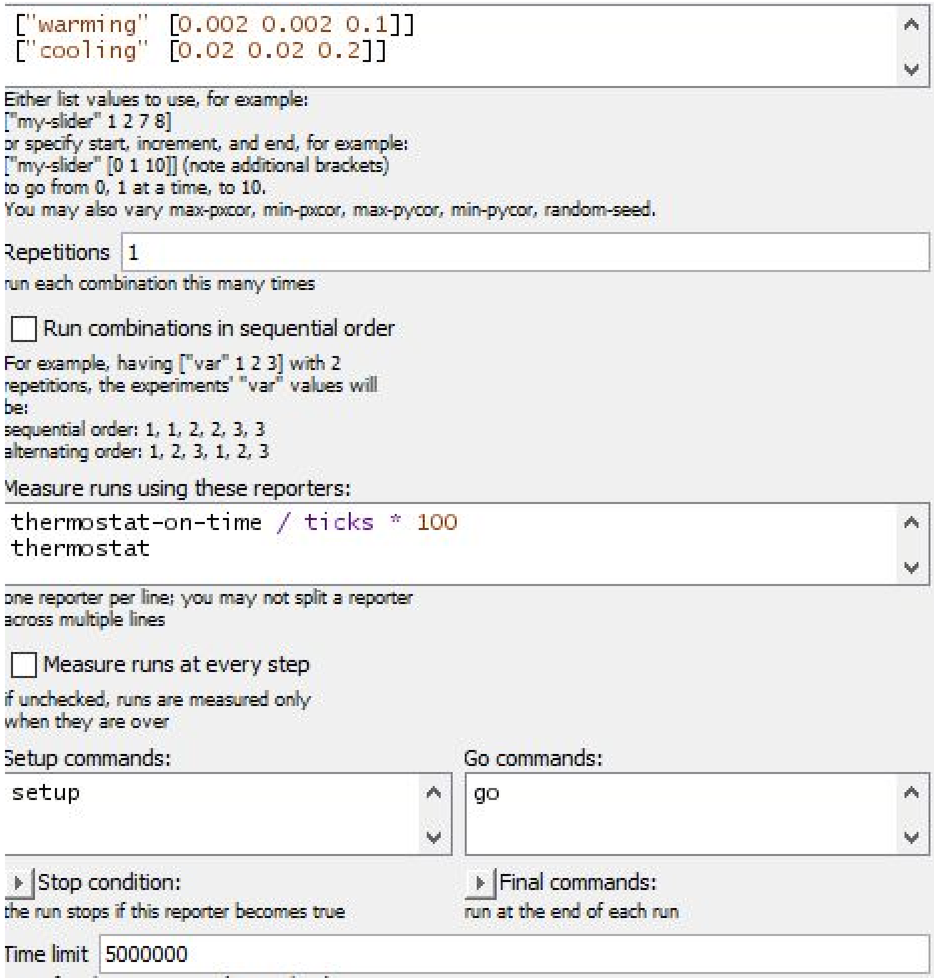
Appendix O

```
create-internal-air 100
[
  set heat 32 ; Minimum temperature without compromising food integrity
  set color red
  set size 2
  set shape "circle"
  fill-fridge
 ]

   if mean [heat] of internal-air > 40  ;if the average temperature is more than 40 degrees
  [
    set thermostat 1   ;the thermostat turns on
  ]
 if mean [heat] of internal-air < 32    ;if the average temperature is less than 32 degrees
  [
    set thermostat 0 ;the thermostat turns off
  ]
```

Appendix P

```
ifelse counter <= 200000  ; daytime
[
  if mean [heat] of internal-air > 34 ;if the average temperature is warmer than 34 degrees
  [
    set thermostat 1    ;turn thermostat on
  ]
  if mean [heat] of internal-air < 32 ;if the average temperature is colder than 32 degrees
  [
    set thermostat 0 ;turn thermostat off
  ]
  if thermostat = 1    ;if the thermostat is on
  [
    set thermostat-on-time thermostat-on-time + 1   ;increase thermostat-on-time by 1
  ]
  if x != thermostat  ;if x is not thermostat
  [
    set dflag dflag + 1    ;increase dflag by 1
  ]
  set x thermostat          ;set x to be thermostat
]  ; end daytime
[  ; start nighttime
  if counter > 200000       ;if counter is equal to 200000
  [
    set y thermostat        ;set y to be thermostat
  ]
  if mean [heat] of internal-air > 42     ;if the average temperature is warmer than 42 degrees
  [
    set thermostat 1    ;turn the thermostat on
  ]
    if mean [heat] of internal-air < 40    ;if the average temperature is colder than 40 degrees
  [
    set thermostat 0    ;turn the thermostat off
  ]
  if thermostat = 1    ;if the thermostat is on
  [
   set thermostat-on-time thermostat-on-time + 1  ;increase the thermostat-on-time by 1
  ]
  if y != thermostat        ;if y is not equal to thermostat
  [
    set nflag nflag + 1   ;increase nflag by 1
  ]
  set y thermostat         ;set y to be the same as thermostat
] ; end nighttime
move
bounce

ick
: thermostat = 1    ;if thermostat is on

set thermostat-on-time thermostat-on-time + 1 ;increase thermostat-on-time by 1

et counter counter + 1   ;increase counter by 1
: counter >= 400000      ;if counter is greater than or equal to 400000

set counter 0           ;reset the counter to be 0
```

Appendix Q


["warming" [0.002 0.002 0.1]]
["cooling" [0.02 0.02 0.2]]
Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.
Repetitions 1
run each combination this many times
Run combinations in sequential order
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3
Measure runs using these reporters:
thermostat-on-time / ticks * 100
thermostat
one reporter per line; you may not split a reporter across multiple lines
Measure runs at every step
if unchecked, runs are measured only when they are over
Setup commands:
setup
Go commands:
go
Stop condition:
the run stops if this reporter becomes true
Final commands:
run at the end of each run
Time limit 5000000

Appendix R

warming	cooling	steps
0.018	0.18	45212
0.014	0.14	57442
0.01	0.1	80699
0.006	0.06	132442
0.002	0.02	399022

Appendix S

Cooling	Warming	Ave. Temperature
0.009	0.11	40.25895
0.01	0.12	40.95495
0.005	0.06	40.99555
0.011	0.13	41.45681
0.012	0.14	42.13566
0.006	0.07	42.15619
0.007	0.08	43.18329
0.008	0.09	43.79331
0.009	0.1	44.18039
0.01	0.11	44.51301
0.011	0.12	44.95034
0.012	0.13	45.29824
0.011	0.11	48.9095
0.012	0.12	48.97629
0.009	0.09	49.01784
0.01	0.1	49.11289
0.008	0.08	49.2946
0.007	0.07	49.40251
0.006	0.06	49.46968

Appendix T

warming	cooling	[step]	thermosta	dayflag	nightflag
0.006	0.08	3431122	35.99423	102	100
0.006	0.1	3431122	28.7726	114	112
0.006	0.08	3431122	36.01927	102	100
0.006	0.08	3431122	36.10554	102	100
0.006	0.08	3431122	35.96794	102	100
0.006	0.08	3431122	35.99989	102	100
0.006	0.08	3431122	35.72412	102	100
0.006	0.08	3431122	36.03938	103	101
0.006	0.08	3431122	35.86116	102	100
0.006	0.1	3431122	28.73946	114	111
0.006	0.08	3431122	35.95407	102	100
0.006	0.08	3431122	36.02265	102	100
0.006	0.1	3431122	28.81297	114	112
0.006	0.1	3431122	28.72302	114	112
0.006	0.12	3431122	23.91224	122	119
0.006	0.1	3431122	28.82378	114	112

Appendix U

2							
3						1715561	
4					number of ticks for 24 hours		
5					3431122	400000	200000
6			from 32 to 40				
7	[run numb	warming	ticks		# min	# hours	
8	12	0.015	33393		14.01463	2.00358	
9	11	0.014	35977		15.09911	2.15862	
10	10	0.013	38688		16.23688	2.32128	
11	9	0.012	41983		17.61975	2.51898	
12	8	0.011	46016		19.31235	2.76096	
13	7	0.01	50671		21.266	3.04026	
14	6	0.009	55912		23.46558	3.35472	
15	5	0.008	63129		26.49447	3.78774	
16	4	0.007	72094		30.25697	4.32564	
17	3	0.006	83879		35.20299	5.03274	
18	2	0.005	101296		42.51269	6.07776	
19	1	0.004	127229		53.39646	7.63374	
20							

Appendix V

warming	cooling	[step]	thermosta	dflag	nflag
0.006	0.08	4000000	3721.758	166	73
0.008	0.1	4000000	3952.027	240	114
0.012	0.08	4000000	7422.263	137	86
0.01	0.1	4000000	4939.648	259	122
0.01	0.08	4000000	6170.646	180	100
0.006	0.12	4000000	2473.414	219	80
0.01	0.12	4000000	4102.864	301	143
0.008	0.12	4000000	3304.504	273	117
0.012	0.12	4000000	4936.25	326	163
0.006	0.1	4000000	2968.781	199	80
0.012	0.1	4000000	5921.176	240	125
0.008	0.08	4000000	4955.118	179	85

[run number]	warming	cooling	[step]	thermostat-on-time / ticks * 100	dflag	nflag	
1	0.006	0.08	4000000	3721.758	166	73	0.305439
5	0.008	0.1	4000000	3952.027	240	114	0.206215
10	0.012	0.08	4000000	7422.263	137	86	0.327354
8	0.01	0.1	4000000	4939.648	259	122	0.191601
7	0.01	0.08	4000000	6170.646	180	100	0.260714
3	0.006	0.12	4000000	2473.414	219	80	0.244147
9	0.01	0.12	4000000	4102.864	301	143	0.164414
6	0.008	0.12	4000000	3304.504	273	117	0.187179
12	0.012	0.12	4000000	4936.25	326	163	0.149284
2	0.006	0.1	4000000	2968.781	199	80	0.261649
11	0.012	0.1	4000000	5921.176	240	125	0.2
4	0.008	0.08	4000000	4955.118	179	85	0.276515

Bibliography

- Critical Temperatures for Food Service. (n.d.). Retrieved 2018, from http://www.idph.state.il.us/about/fdd/fdd_fs_foodservice.htm
- FSIS. (n.d.). Retrieved 2018, from https://www.fsis.usda.gov/wps/portal/fsis/topics/food-safety-education/get-answers/food-safety-fact-sheets/safe-food-handling/refrigeration-and-food-safety/ct_index
- Food Safety information Council, Knowing your Fridge. Retrieved 2018, From http://www.logan.qld.gov.au/__data/assets/pdf_file/0003/4089/knowingyourfridge.pdf
- Heat pump and refrigeration cycle. (2018, March 21). Retrieved 2018, from https://en.wikipedia.org/wiki/Heat_pump_and_refrigeration_cycle
- I. (n.d.). Retrieved 2018, from http://physics.csustan.edu/Ian/HowThingsWork/Topics/Temperature/ThermoLaws/Refrigerators.htm
- M. H., F. (2012, June 22). When it's hot, help your refrigerator keep its cool. Retrieved from https://www.consumerreports.org/cro/news/2012/06/when-it-s-hot-help-your-refrigerator-keep-its-cool/index.htm
- North Devon Council, (n.d.). Temperature control: Food safety tips. Retrieved 2018, from http://www.northdevon.gov.uk/business/food-hygiene-and-safety/food-safety-tips/temperature-control/
- (n.d.). Retrieved 2018, from http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node25.html
- Refrigerator Thermometers: Cold Facts about Food Safety. (2017, November 15). Retrieved from https://www.fda.gov/food/resourcesforyou/consumers/ucm253954.htm

Acknowledgements

All thanks for this project go to Harry Henderson for helping guide us through this project, Harpreet Bhullar, AnnNett Delaney, and Mary Danos who helped us prepare for our UNM presentation, Bob Robey for being there to help us through our initial work, and everyone else who has helped us along the way.