# Rocky Planet Formation as the Universe Ages

**New Mexico**
**Supercomputing Challenge**
**Final Report**
**April 1, 2009**

*Team 18*
*Aspen Elementary*


***Team Members:***
**Devon Conradson**

***Teacher:***
**Zeynep Unal**

***Project Mentor:***
**Steven Conradson**

**Table of Contents**

## Executive Summary

I used the StarLogo software development package to write a program that models the behavior of an outgoing shock wave of heavy elements from a supernova as it goes through space, being affected by the gravitational pull from the other particles, the supernova core, and an arbitrary number of stars in proximity. I used this program to answer the question of whether Rocky Planets would form more easily when the universe was young and the stars were close to each other and would be more difficult as the universe aged and the stars became further apart. The program showed that my hypothesis was mostly correct, heavy elements dispersed through space when there were only a few stars but aggregated into clusters when there were many. What I did not expect, however, was that the actual arrangement of the stars was critical. Clusters of stars, especially near the supernova core, caused high local concentrations of iron particles to occur even when there were only three neighbor stars. This effect of star clusters on occasional runs peaked at intermediate numbers of stars and then decreased at higher ones. So if extremely high concentrations of heavy elements instead of the average concentration is important then intermediate numbers of neighbor stars may be more effective in creating giant molecular clouds and subsequent solar nebulae from which solar systems with rocky planets will form.

## Introduction

Rocky planets like earth are most interesting to us because only they can support life as we know it. Earth-like planets could not have formed directly after the Big Bang because it only created hydrogen, helium, and a tiny amount of lithium and none of the heavier elements that rocky planets and life are composed of, like oxygen, silicon, iron, and even carbon. Up through around oxygen these elements were only made in stars. Elements heavier than oxygen were only made in supernovas that blasted them into space when they exploded. This includes iron, nickel, aluminum, and silicon that are the most common elements in the Earth apart from oxygen. If after a supernova these elements had just continued to disperse through space on their original trajectories radially outwards from the supernova core they would never have condensed into the giant molecular clouds and solar nebulae from which solar systems like ours formed.

Therefore, one or more forces must have altered the trajectories of the supernova ejecta so that instead of simply continuing to separate from each other as their distance from the core increased and they occupied a larger volume of space these heavy elements collected together to form regions where they were highly concentrated. If this had not happened then there never would have been enough of them in a solar system for rocky planets to have formed.

The most likely force that could have condensed heavy elements is gravity, the force responsible for all of the interactions of matter in the universe. Since the dark matter that is supposed to constitute 80–90% of the universe, although it is not understood, is probably evenly distributed in galaxies so that its gravitational force is also even then it is probably stars with their highly localized gravity that pull the heavy elements traveling outwards from supernova together into clumps. Then, when shock waves or other cosmological phenomena that appear to have initiated solar system formation occurred, they swept these bunches of heavy elements into the giant molecular clouds of hydrogen that eventually formed solar nebulae and then solar systems.

But if stars are the origin of the force that concentrates the heavy elements that are drifting through space so they eventually form rocky planets then there is a problem having to do with the age of the universe. Rocky planets could not form after the big bang until after the elements of which they are made had been created in supernovae, so it would have taken some time after the Big Bang before the first rocky planet was created. But if it is true that the heavy elements aggregate because of the gravitational fields of the stars they pass by, then if stars are too far apart these elements will not condense and rocky planets also cannot form. Increasing separation of stars, however, is exactly what will happen as the universe ages. Therefore, at some time in the history of the universe, rocky planets may cease to form and life as we know it will not begin on newly created planets as well.

To test this I first want to see if the heavy elements ejected by a supernova condense if their paths are bent by nearby stars. Since the formation of rocky planets out of solar nebula requires high concentrations of heavy elements in the giant molecular clouds from which they form (perhaps triggered by shock waves from supernovae) then this aggregation is a required step in the process. I then want to test if having fewer stars farther apart so that the gravitational effects are weaker weakens or even stops this aggregation. If it does, then as the universe ages and stars increase their distance from each other rocky planets will form more rarely or not at all.

**Description**

I want to first test if the heavy elements ejected by a supernova condense if their paths are bent by nearby stars. Since the formation of rocky planets out of solar nebula requires high concentrations of heavy elements in the Giant Molecular Clouds from which they form (perhaps triggered by shock waves from supernovae) then this aggregation is a required step in the process. I want to then test if having fewer stars farther apart makes this aggregation less effective. If it does then as the universe ages and stars increase their distance from each other then rocky planets may form only rarely or not at all.

I can do this with a program written in StarLogo TNG that would calculate the trajectories of objects moving through space subjected to the gravitational forces from each other and stars. To do this I need to know the mathematical equations that describe the movements of objects through space. This field is called Celestial Mechanics. Celestial mechanics is the application of Newton's Laws of Motion to the movement of mass through space. (I'm ignoring Einstein's relativistic corrections.) If I know the mathematical equations that describe Newton's Laws then I can develop the algorithms (the step by step procedure) I need to write a program for doing celestial mechanics calculations that describe the motion of masses. The motion and the masses that I am interested in for my project are the heavy elements created and then ejected from a supernova as they move through space and pass by other stars.

The place to begin is Newton's first two Laws of Motion. The first Law is that an object at rest will remain at rest, or an object in motion will remain moving at a constant velocity (velocity is specified by a magnitude – the speed – and the direction) unless acted on by an outside force. The second Law is that applying a force to an object will cause it to accelerate, meaning that either or both the magnitude and the direction of its velocity will change. The algebraic equation that describes this is:

$$\mathbf{F} = m\mathbf{a};$$

where **F** is the force, m is the mass of an object, and **v** is the acceleration. I didn't need Newton's third Law that for every action there is an equal and opposite reaction because I fixed all the stars in place. They therefore only acted on the particles that modeled the heavy elements ejected by the supernova but were not themselves affected by the particles or each other. **F** and **v** are vectors, mathematical objects that have both a value and a direction. Vectors are designated by writing them in bold type. Speed, which is only the magnitude of the velocity, is what is called a scalar number. An important property of vectors that was critical in writing my program is that they can be placed in a Cartesian plane and separated into their x and y components by trigonometric functions that are included in StarLogo, the sine, cosine, tangent, arctangent (Figure 1). The x and y components can be added to each other separately, which provides a way of adding the vectors for force, acceleration, and velocity in the program.

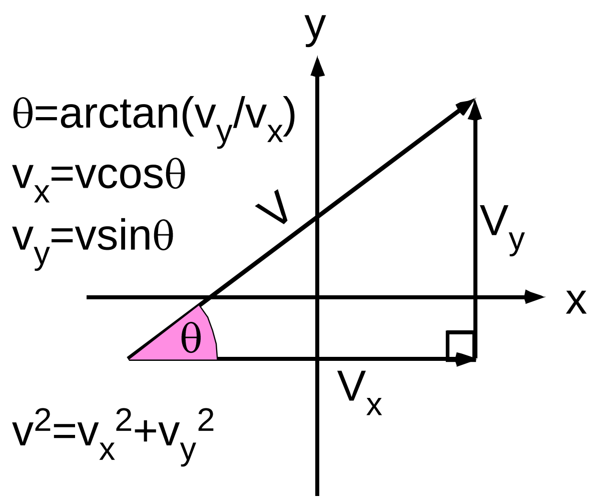



**Figure 1. Vectors.** In a Cartesian plane a vector **v** of length v can be separated into its x component, $v_x$ and its y component, $v_y$, that together make a right triangle with an angle θ. These parameters of the triangle are related to each other by the trigonometric functions sine, cosine, tangent/arctangent, and Pythagorean's

The way that I do the calculation is to do everything in moves that correspond to an interval of time. Starting with all of the particles that have been ejected from the supernova in their original positions, their new positions are:

$$\text{position}_{new} = \text{position}_{original} + (\mathbf{velocity} \bullet \text{time}).$$

In Cartesian coordinates this equation becomes:

$$(x_{new}, y_{new}) = (x_{original} + \mathbf{velocity_x} \bullet \text{time}, y_{original} + \mathbf{velocity_y} \bullet \text{time}).$$

In words, these equations say that the velocity (speed and direction) of an object remains constant unless an external force acts on it, so that its position is its original position displaced by its velocity multiplied by the amount of time it has been moving. I describe the position by using Cartesian coordinates. I separate the total velocity into its x and y components.

Next I have to calculate how the velocities change in the gravitational fields from all of the stars. This is where Newton's second Law is applied.

$$\mathbf{F} = m\mathbf{a} : \mathbf{a} = \mathbf{F}/m$$

If an external force does act on an object then it produces an acceleration (a change in speed *and/or* direction) that is proportional to the force and inversely proportional to its mass.

The change in velocity, the new component of the velocity that happens in one turn, is just like the position, the acceleration times the amount of time that it acts:

$$\mathbf{v}_{\mathbf{new\ component}} = \mathbf{a} \bullet \text{time}: \mathbf{v}_{\mathbf{new\ x\ component}} = \mathbf{a}_{\mathbf{x}} \bullet \text{time}, \mathbf{v}_{\mathbf{new\ y\ component}} = \mathbf{a}_{\mathbf{y}} \bullet \text{time}$$

In words, the acceleration multiplied by the time the object is accelerating with that value produces a new velocity component. Just like with the velocity, I can separate the acceleration into x and y components.

$$\mathbf{v}_{\mathbf{new}} = \mathbf{v}_{\mathbf{original}} + \mathbf{v}_{\mathbf{nc}}: \mathbf{v}_{\mathbf{newx}} = \mathbf{v}_{\mathbf{originalx}} + \mathbf{v}_{\mathbf{ncx}}, \mathbf{v}_{\mathbf{newy}} = \mathbf{v}_{\mathbf{originaly}} + \mathbf{v}_{\mathbf{ncy}}$$

In words, the new velocity is the sum of the original velocity and the new component produced by the force.

Also, note that StarLogo TNG automatically steps forward in time. If I make each turn equal to one unit of time then I can leave the time out of my calculations for both the velocity and the acceleration.

All that remains is the equation that describes the force of gravity. This is well known:

$$F = G \bullet m_1 \bullet m_2 / \text{distance}^2$$

In words, the gravitational force between two objects equals the gravitational constant (I varied this in my program to make it easier to run based on making it slightly below the force necessary so that the particles ejected from the supernova would fall back) multiplied by the masses of the two objects that are attracting each other an divided by the square of the distance. Using these equations we can understand the way in which gravity makes a thrown object (Figure 2): 1) hit the ground; 2) go into orbit; or escape from a planet by accelerating it towards the center of the planet without necessarily changing its speed (that happens from air resistance or when the object is moving outwards from the center and not tangentially).

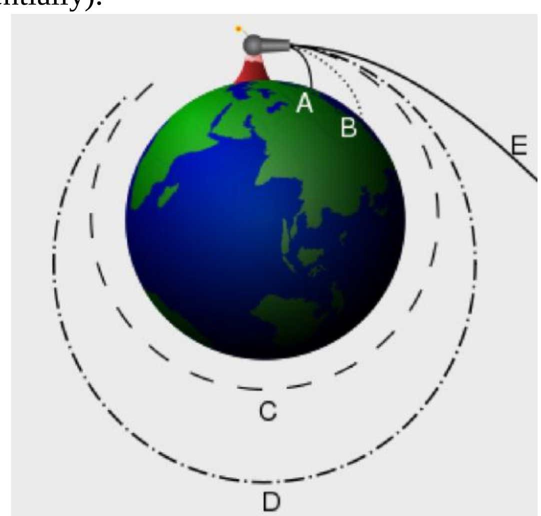



**Figure 2. Gravity and velocity.** Gravitational force will make a cannonball fired at a tangent to the earth's surface accelerate towards the center of the earth. Below orbital velocity the cannonball will hit the earth (A and B). At orbital velocity (C and D) the cannonball is moving as fast as it falls so that it orbits. At escape velocity (E) the cannonball leaves the gravitational well of the earth.

In order to implement these equations I wrote the program “Supernova” that went through the following steps in a two dimensional space that went from -50 through 50 in integer increments:

1. Setup the region of space with the supernova at a particular position (the origin) and a certain number of stars randomly distributed around it (it would be better to position the stars myself but I didn’t time to implement that in the code).
2. Set off the explosion that fires the heavy elements (as particles) into space radially from the supernova, all with the same speed.
3. Calculate their new positions.
4. Remove any particle that is within one unit of the boundaries.
5. Sum the x and y accelerations that result from the gravitational force each particle and star exerts on the other particles.
6. Add this acceleration to the original velocity to get the new velocity.
7. Count the total number of ejecta particles and the number of ejecta particles that are within one unit of each other and write the results into a table.
8. Repeat steps 3-5 up to 600 seconds in 0.2 second increments (one clock tick). Note that I actually split 3 and 4/5 into separate turns because of a quirk in StarLogo that caused some particles to move before the entire turn was complete.

The actual program is shown in the presentation (and possibly as an appendix if it doesn’t make the file too large to download since it is all images of the screens from StarLogo).

Prior to performing the actual runs I tried different combinations of initial velocity, masses for the stars and supernova core and iron particles, and the gravitational constant. I eventually settled on a combination that allowed all of the particles to escape from the space when there were zero stars but in which the gravitational pull from the core was close to being in balance with the original velocities of the particles and in which the masses of the neighboring stars were just below 1/3 that of the supernova core. The ejecta particles were 1/250 the masses of the neighboring stars.

I performed 10–12 runs for three, seven, and eleven neighboring stars. I recorded the image of the two-dimensional space at the very beginning to obtain the positions of the stars and at the end to obtain the positions of the remaining ejecta particles. I plotted the number of ejecta (called “nirons”) particles and the number of particles within one unit of each other, called “close,” as a function of time over the run. Close always begins quite high because all of the particles are contained in a circle of radius one at time=0. There was some kind of an error in the “close” calculation that caused large oscillations in its value for the first up to a dozen or so points, after which it gave correct values

To better evaluate and interpret the data I then the subtracted the results for “nirons” and “close” from those obtained with zero neighboring stars, in which case the ejecta formed a continuously expanding circle, the components of which all eventually left that region of space. They left at different times because they formed a circle whereas the space was a square.

**Results**

*Zero stars baseline.* Doing the calculation with zero stars gives a baseline for comparison with results from calculations with stars. The supernova explosion sends all of the ejecta particles that begin in a circle with radius one radially outwards. With time their velocity decreases because of the gravitational pull of the supernova core. Eventually the first particles come to the boundary of the system and leave it. Because they form a circle within the square perimeter of the calculation space they depart over a period of time. Similarly, because of the circular symmetry of the particles, the “close” parameter decreases in steps that correspond to arriving at a radius for which a set of the particles on the perimeter cross the boundary between being counted in “close” and being beyond its limit as the perimeter increases in size.

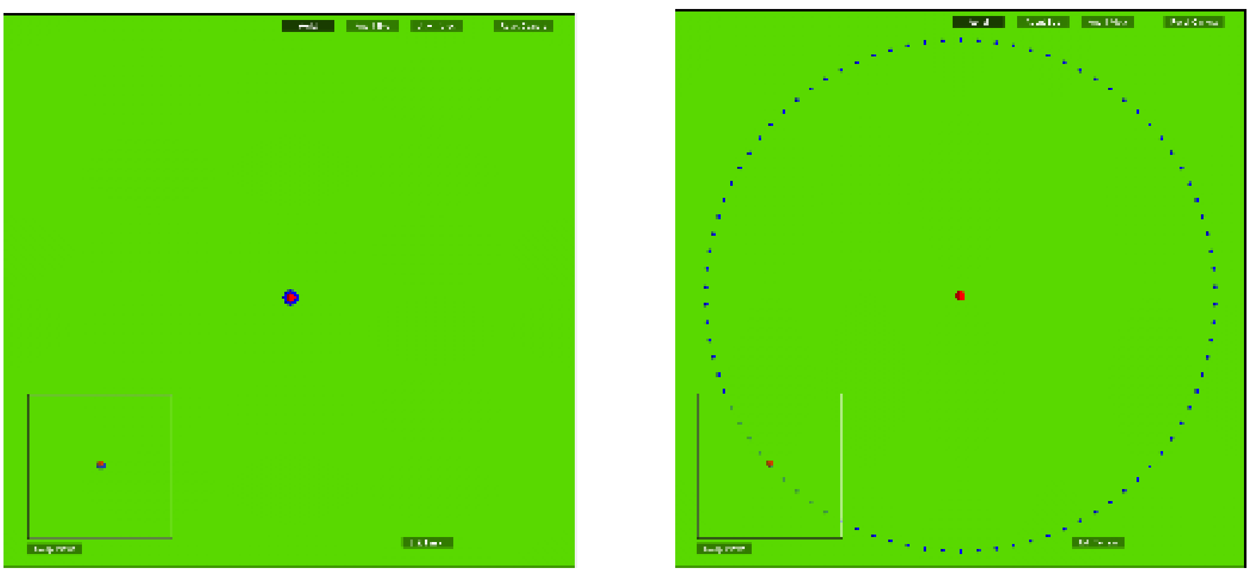


**Figure 3. Zero stars, images.** On the right is the configuration with zero stars at the time = 0. The supernova core is the red circle in the center, the blue particles form a circle of radius one around it. At some time after the explosion the ninety particles that have moved radially outwords from the center so that they still occupy the perimeter of a circle have separated so they can be distinguished from each other.

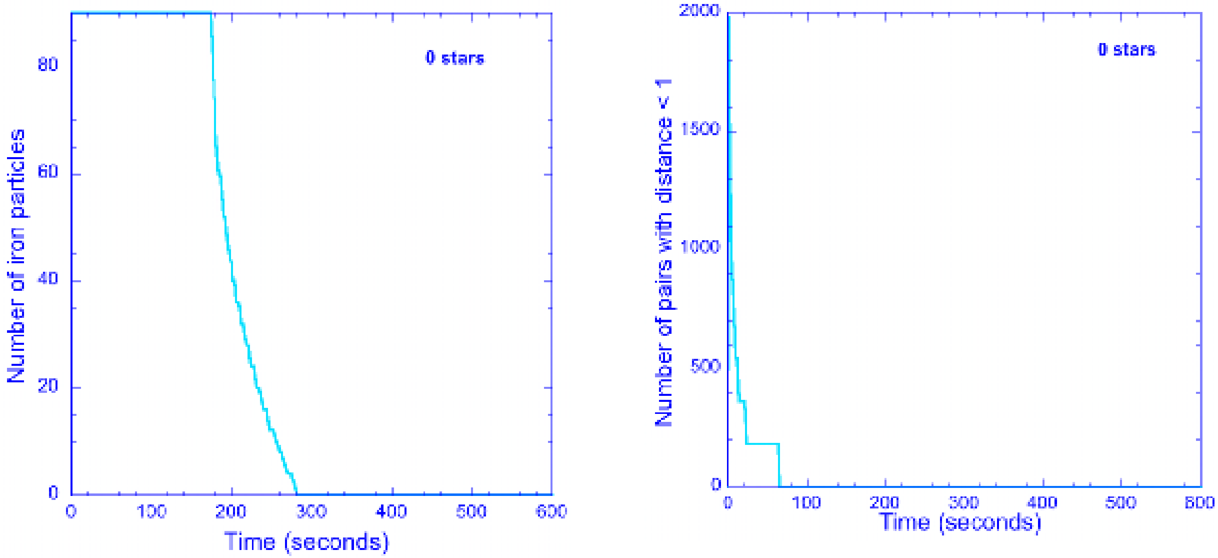



**Figure 4. Zero stars, results.** On the right is the number of particles. On the left is the value of the "close" parameter. The results are discussed in the text.

*One star*. The figures with one star show how its gravitational field perturbs the trajectories of the nearby particles as they pass by. The symmetry of the circle is broken as the particles approaching the star are accelerated towards it and move beyond the perimeter of the circle. As they pass by then swing around the star and are given a large component of velocity towards it perpendicular to their original direction. This can be seen in the close pairs when the perimeter of the circle is at the star. This transverse motion and clustering continue even afterwards, as can be seen in the second picture at the later time.

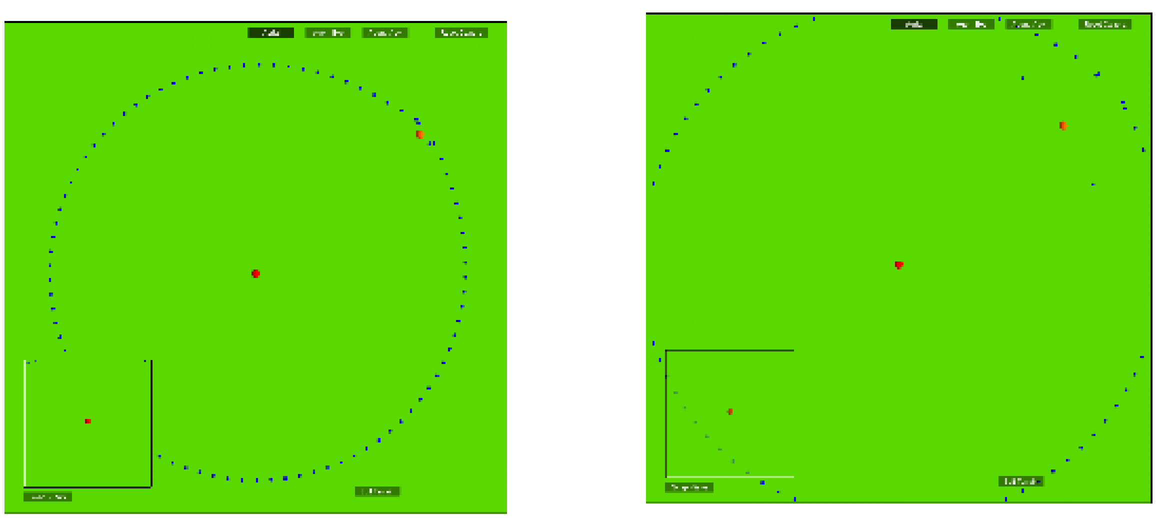


**Figure 5. Effects of one star.** The gravitational field of a single star first accelerates the nearby particles towards it and then gives them a large velocity component towards the star transverse to their original trajectory that causes them to cluster.

*Seven stars.* Eleven run with seven stars gave quite interesting results (Figure 6.) The number of iron particles minus the number with zero stars as a function of time all had the same shape and peak position because this originated in the zero stars calculation. But the spread in the numbers was almost a factor of two, with one calculation being especially low. Despite this the slopes beyond the peak that show the rate of loss of the iron particles are all similar. The "close" parameter displays even more diversity. Eight of the runs remained very close to zero until close to 300 seconds until they rose above the zero star value and then decreased slowly. But three of the runs, and especially two of these, gave "close" values that are so high they are not even comparable. The results appear bimodal instead of following normal statistics where they would be clustered around the average. This kinds of distribution points to completely different kinds of behavior for the majority of the results with low values and the ones with the high values. Interestingly, the runs with the high numbers for "close" do not have high numbers for the particle count. This means that since the same number of particles remain in the space, the ones that are there are closely clustered together.

Three stars gave similar results (not shown), with only one run out of the total that was significantly higher than the others but still much lower than the two with seven stars.

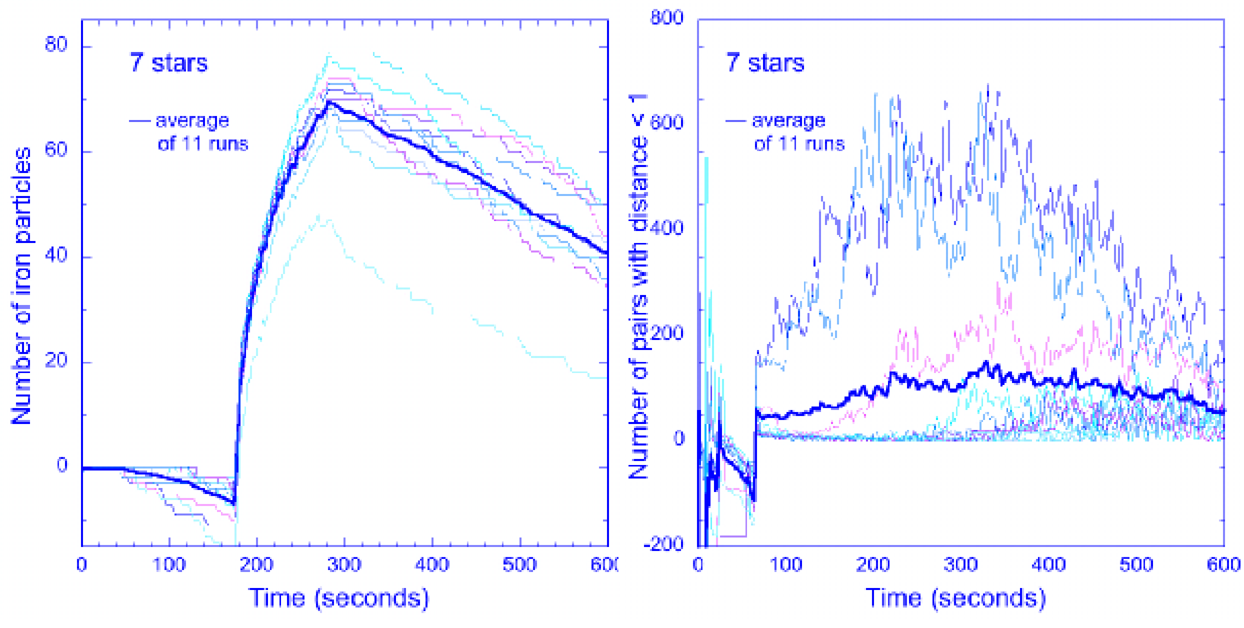



**Figure 6. Seven stars, results.** Left; number of particles minus the number of particles for zero stars at the same time. Right; "close" parameter minus the value for zero stars at the same time.

*Three stars.* The reason for the bimodal behavior can be seen in Figure 7, which shows the results for three stars. In most cases the number of particles was at or close to zero by the end of the 600 seconds and the number of close pairs was also at zero. These values are lower than for seven stars. There were, however, two runs where the number of particles stayed high and the number of close pairs also not only became very high – around 1/3 of the two high values for the seven stars – but was actually increasing with time. Inspection of the pattern of stars and particles showed why. If the stars were clustered then two effects were observed. The first is that the stars in proximity to each other acted as a single more massive star for iron particles away from them that decelerated the ones moving in that direction and could actually cause their directions to reverse and move towards the star group. The second effect is that the stars in a cluster often directed the iron particles towards each other. Especially if the motion was towards the more massive core, the core would trap the particles for a long period giving a very high density around it. As more particles were trapped in the gravitational well of the core with time the number of pairs increases. This same effect is what caused the extreme behaviors for the seven stars. There are differences between the two sets of runs, however. For three stars the slope of the number of particles is smaller than for seven, although it increases towards the end. The value of “close” is smaller, but increases at the end. And the differences from the average are larger for three stars. So the effects of the clustering are at least somewhat dependent on the total number of stars in space.

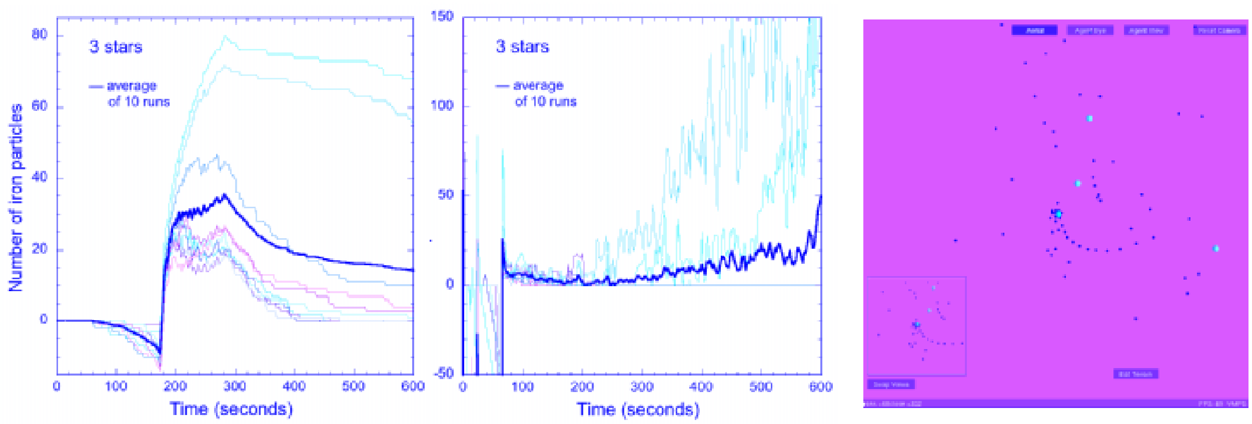



**Figure 7. Three stars, results and image.** Left; the number of particles minus those for zero stars. Center; the value of “close” minus the results for zero stars. Right; the final distribution of particles for the run that gave the largest number for “close.” This configuration of stars caused a large number of iron particles to fall back towards the supernova core where they became trapped in its gravitational well. The high concentration gives the large value for “close.” As more particles fall deeper into the well and the concentration increases with time the number continues to rise, even at the end of the calculation.

*Eleven stars.* On going from seven to eleven stars the trends between three and seven reverse. There is a wider distribution of particle numbers remaining with time. In addition, although the number of close pairs is relatively large, there are no instances where it is overwhelmingly larger than the average. There may, however, be two types of behavior that give larger and smaller values for “close.” I think this is because with a large number of stars a third effect becomes important. The accelerations of the particles towards groups of stars or after curving closely around a star are so large that the particles are no longer trapped in local gravitational wells but are continuously shunted into different areas of space. Since there is no region such as the area around the supernova core that may be a deep gravitational well that collects and traps large numbers of particles in a small area to give a high concentration, the concentration remains relatively low and the value of “close” remains low as well.

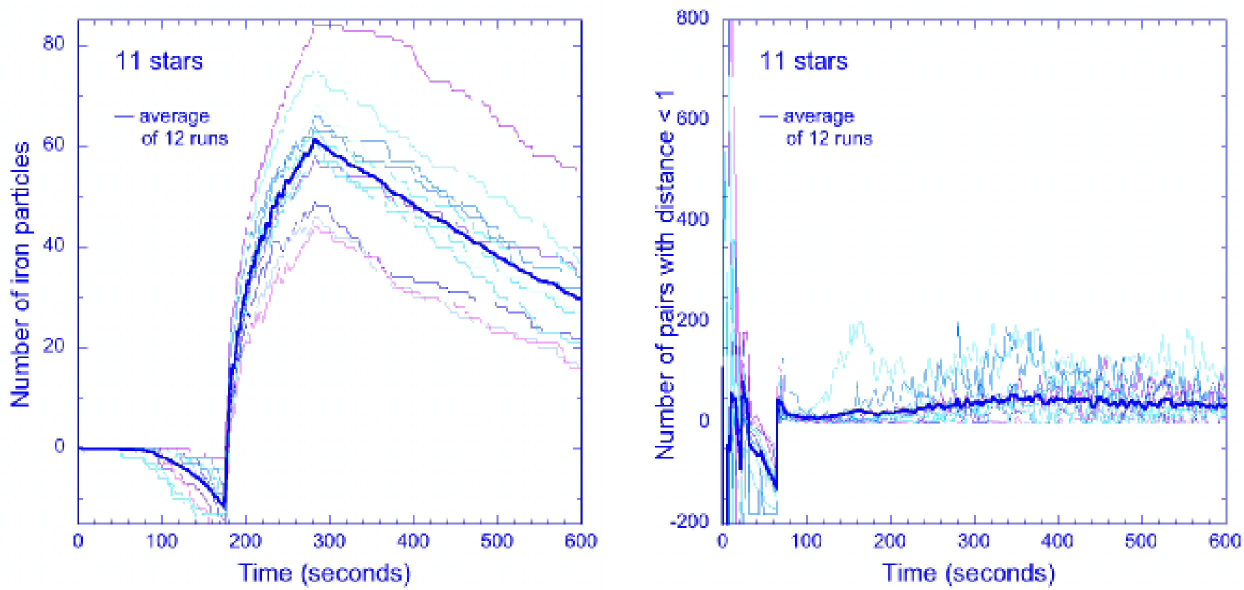



**Figure 8. Eleven stars, results.** Left; the number of particles minus those for zero stars. Right; the value of “close” minus the results for zero stars. Although the distribution of particle numbers is larger than for seven stars, there are no significant outliers for “close” although there do appear to be two types of behavior that give larger and smaller numbers.

*Comparisons of Average Values for Three, Seven, and Eleven Stars.* Comparing only the average results for the three different numbers of stars more clearly shows the trends suggested by the individual results. Seven stars retains the greatest number of iron particles for the longest time. (The peak is because the zero star results have been subtracted and there is a discontinuity at that time.) The number of close pairs is greatest for seven stars. (The bug in the program is evident at shorter times.) Oddly, three stars actually gives a greater number of close pairs at 600 seconds. This must mean that the clustering effect decreases after a certain threshold in the number of stars. But the averaging process does not show the fact that the different distributions of stars mostly give a particular result, with a few distributions giving an enormous number of pairs with time because they divert large numbers of iron particles into orbit around the supernova core.

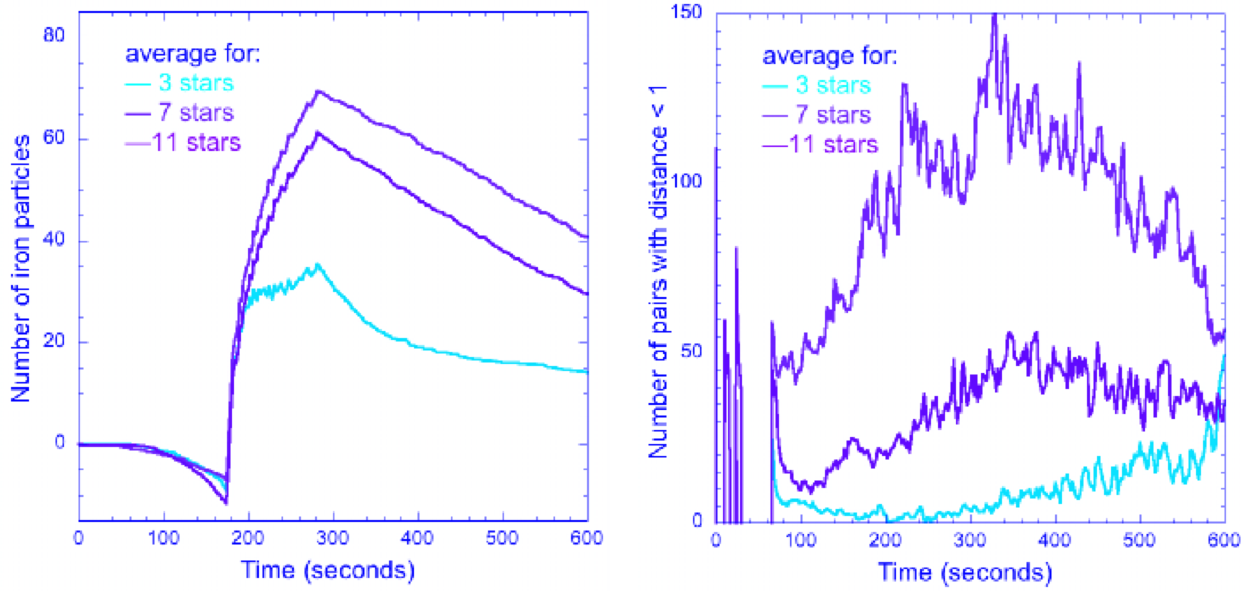



**Figure 9. Comparison of average results for different numbers of stars.** Left; average value of the number of particles minus the zero star results for the different numbers of stars. Right; average value of "close" minus the zero star results for the different numbers of stars. For three and seven stars the averages may not be useful because they may exhibit bimodal behavior instead of statistical.

## Conclusion

Overall, these calculations do demonstrate that heavy elements created in supernovas and then ejected into space by the explosion are concentrated by neighboring stars while in their absence they simply disperse into space without ever aggregating in a way that solar nebular would have high concentrations and form solar systems with rocky planets.

However, the actual arrangement of the stars is critical in this process, especially when the total number of stars is small or intermediate. This unexpected result is because, after being trapped by the gravitational pull of a cluster of stars that are close to each other, the particles eventually cluster around the supernova core or in other gravitational wells. This causes the heavy elements to clump together in a high concentration in case a shock wave from another supernova sweeps through or some similar event occurs; rocky planets are not limited to forming in proximity to the supernova core in which they were created. In the case of high star densities most arrangements of stars give larger numbers of close pairs on average, but there are no longer any cases with very high concentrations of stars in gravity wells.

Therefore, while it does appear likely that as the universe ages and the distance between stars increases the formation of rocky planets will slow down, it may continue at a low level for a very long period of time because there will still be configurations of even a small number of stars that continue to cause them to aggregate into regions with high concentrations.

## Recommendations

It would be useful to continue to refine the masses of all of the objects, the velocities of the particles, and the gravitational constant to attempt to give results that can be compared against our universe. It would be extremely difficult to extend these calculations to three dimensions with StarLogo. StarLogo is not the optimum programming language or environment for these kinds of calculations anyway because its emphasis on agents means it is not easy to determine long range interactions between objects, such as occurs in celestial mechanics.

I have also begun calculating the gravitational potential as a function of position for the various arrangements of stars and have completed additional sets of runs for seven and eleven stars. I am hoping that this will enable better understanding of the clustering phenomenon that I observed and especially what configurations of stars cause it.

## Acknowledgements

I would like to thank my teacher, Ms. Unal, my mentor, my father, and the StarLogo advisors at MIT for all of their assistance and encouragement.

**References**

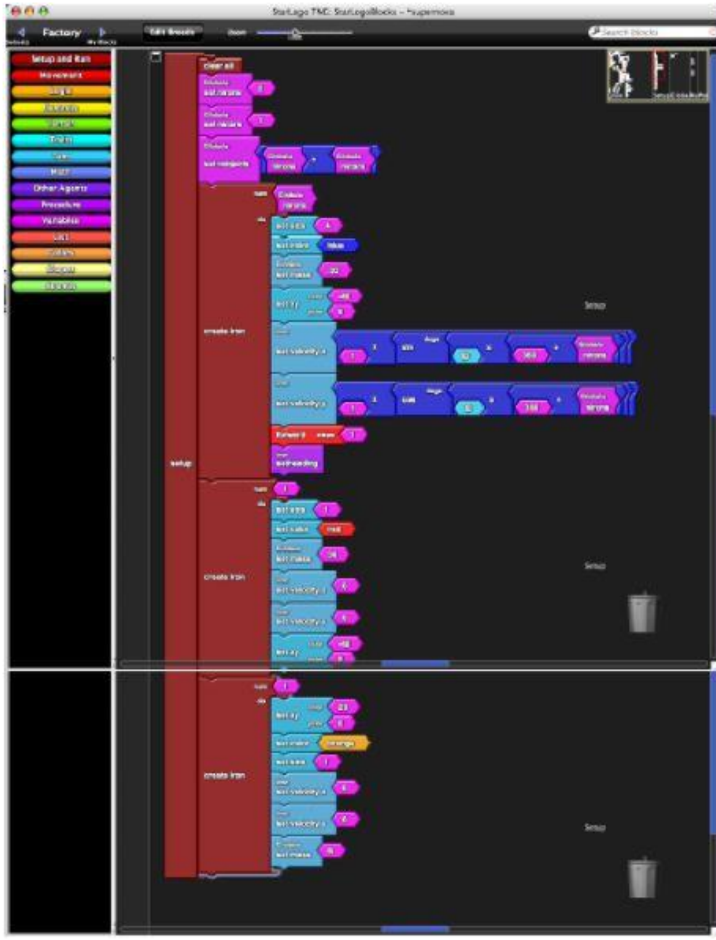

1. L. Bernstein, *lasers.llnl.gov/programs/images/nucleosynthesis_in_a_star.jpg,* and *Science & Technology Review*, Lawrence Livermore National Laboratory, 2007.
2. *commons.wikimedia.org/wiki/File:Big_Bang_nucleosynthesis.svg*, 2007.
3. S. Fudge, *www.k12.nf.ca/gc/Science/Physics3204/Projects2003/SlotA/ProjectA2/index1.html*.
4. L.N. Mitchell, *www.gaston.k12.nc.us/resources/teachers/webquests/Art/webquest/resources.htm*, North Carolina State University.
5. K. Nollett, *images.iop.org/objects/physicsweb/world/20/8/3/PWfea4_08-07.jpg,* Argonne National Laboratory, 2007.
6. H. D. Young, R. A. Freedman, T. R. Sandin, and A. L. Ford. *Sears and Zamansky's University Physics.* Addison Wesley Longman, Boston, 2004.
    A. Zablludoff, *atropos.as.arizona.edu/aiz/teaching/nats102/images/bigbang.jpg,* Arizona State University, 2008.

## Appendix 1: Supernova Program

# Global Variables


These variables were defined in the Global Area. Indexa and indexb are counters needed to do things in the program. Close is not implemented yet but will be used to keep numerical score of how many particles are close to each other.

Note that the stars are called “iron” particles. Since everything interacts identically I only need one type of agent.



# SETUP Module

These commands set up the number of iron particles and stars. I can easily change these.

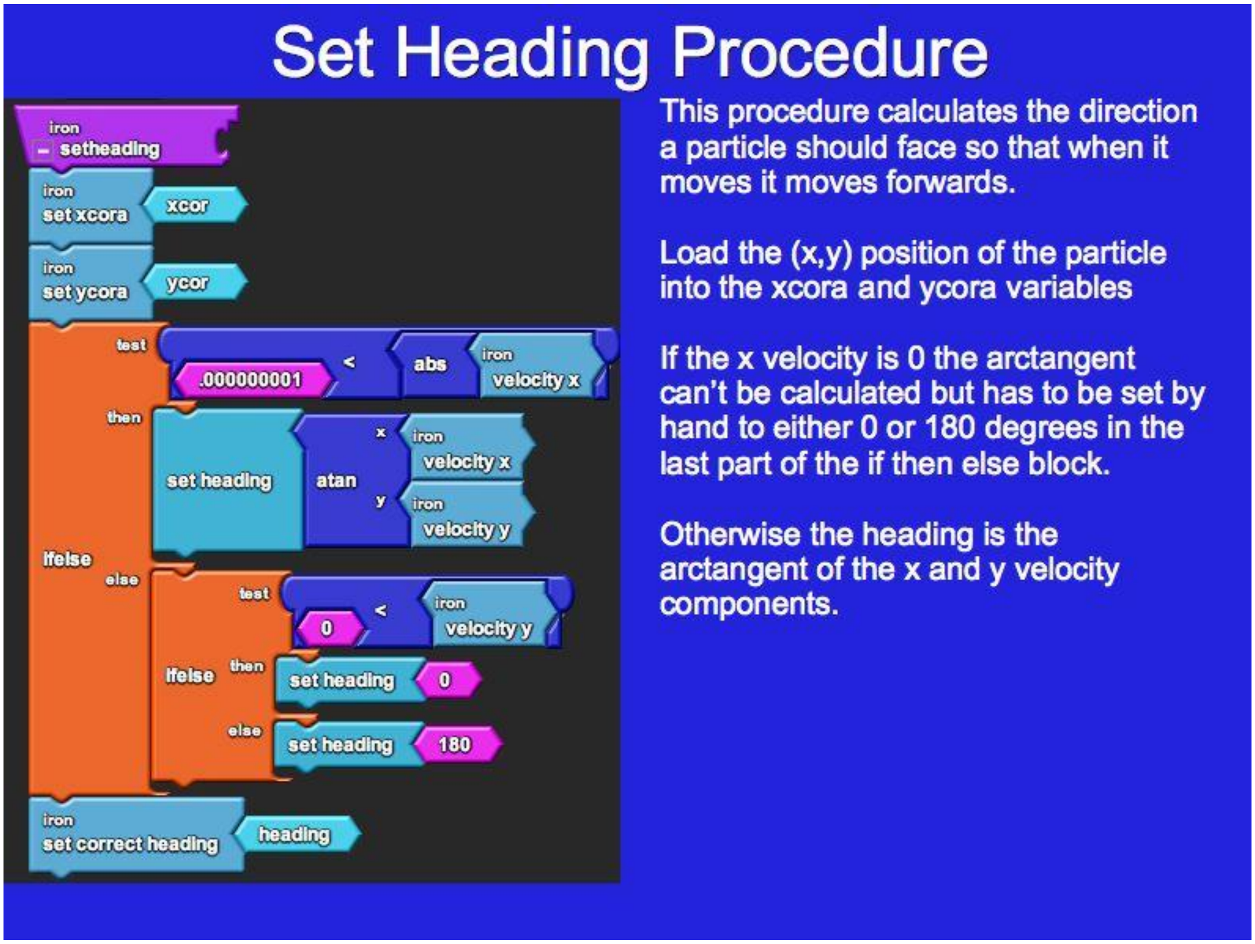
This section puts a ring of iron particles around the supernova and gives them the same speed but with directions radially outward from the core. It moves them away to avoid dividing by zero and gets ready to do the calculation by setting the variable that alternates the program between calculating accelerations and moving the particles to “true.”

This section creates the supernova core at a particular position and with a particular mass.

This section creates the remaiing stars at random positions and with the same mass.

# Iron Block Variables


These are the remaining variables I need to do the calculations. I put them in the iron block.


Set Heading Procedure
iron
setheading
iron
set xcora
xcor
iron
set ycora
ycor
ifelse
test
.000000001
<
abs
iron
velocity x
then
set heading
atan
x
iron
velocity x
y
iron
velocity y
else
ifelse
test
0
<
iron
velocity y
then
set heading
0
else
set heading
180
iron
set correct heading
heading
This procedure calculates the direction a particle should face so that when it moves it moves forwards.
Load the (x,y) position of the particle into the xcora and ycora variables
If the x velocity is 0 the arctangent can't be calculated but has to be set by hand to either 0 or 180 degrees in the last part of the if then else block.
Otherwise the heading is the arctangent of the x and y velocity components.

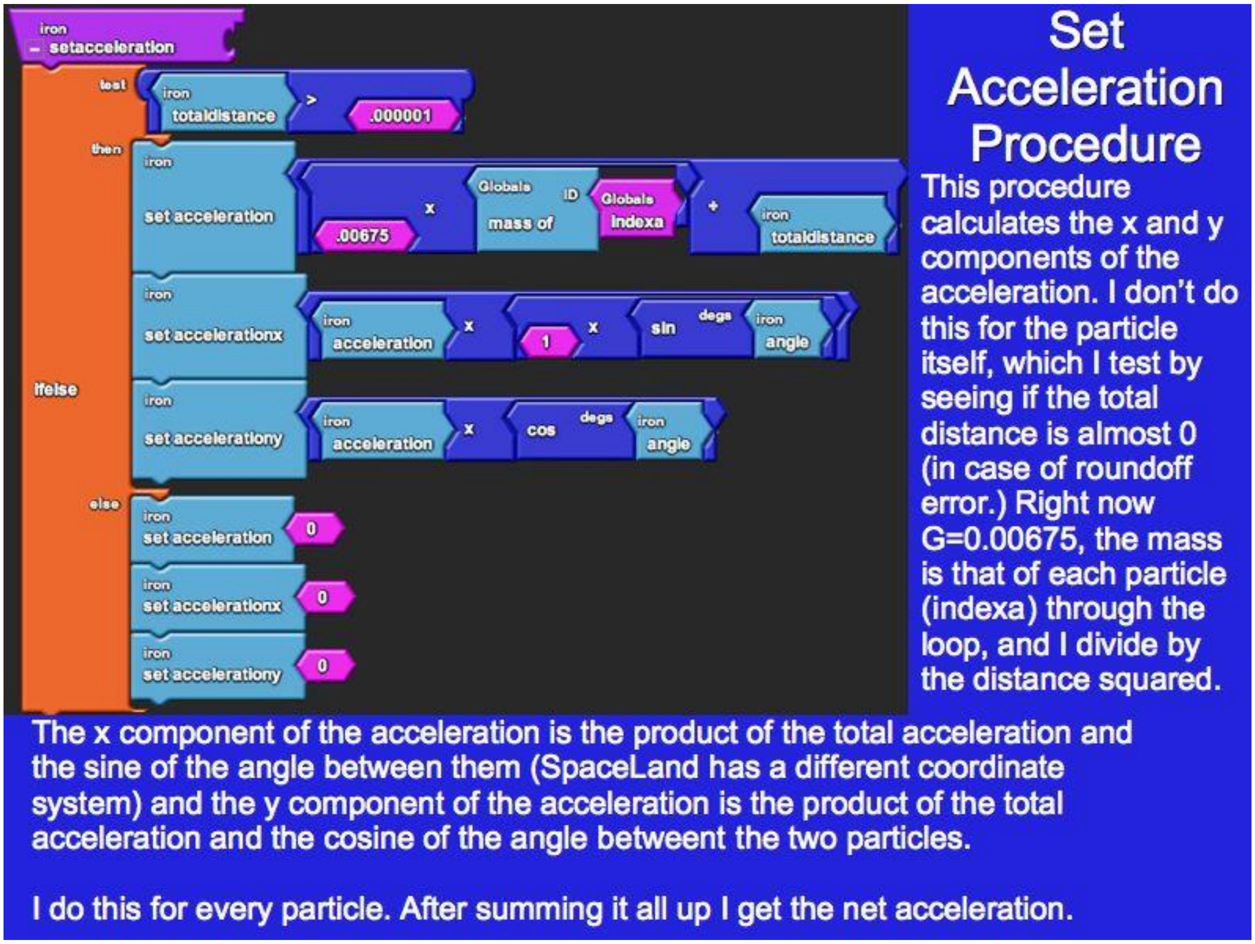
# Move Iron Procedure


Move an iron particle.
Its speed is the square root of its x and y velocities squared (Pythagorean's theorem).
Its direction is already set, so just move it forward.
Set the variables xcora and ycora (its Cartesian x,y coordinates) by calling up the positions.

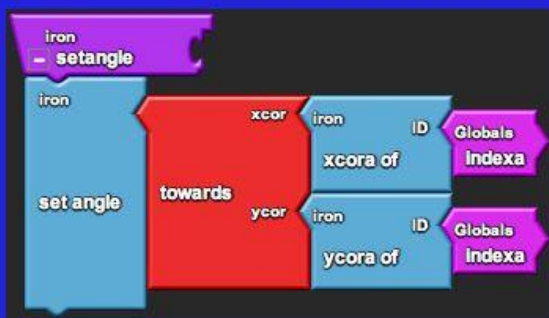
# Set Total Distance Procedure


This procedure finds the distance between two particles.

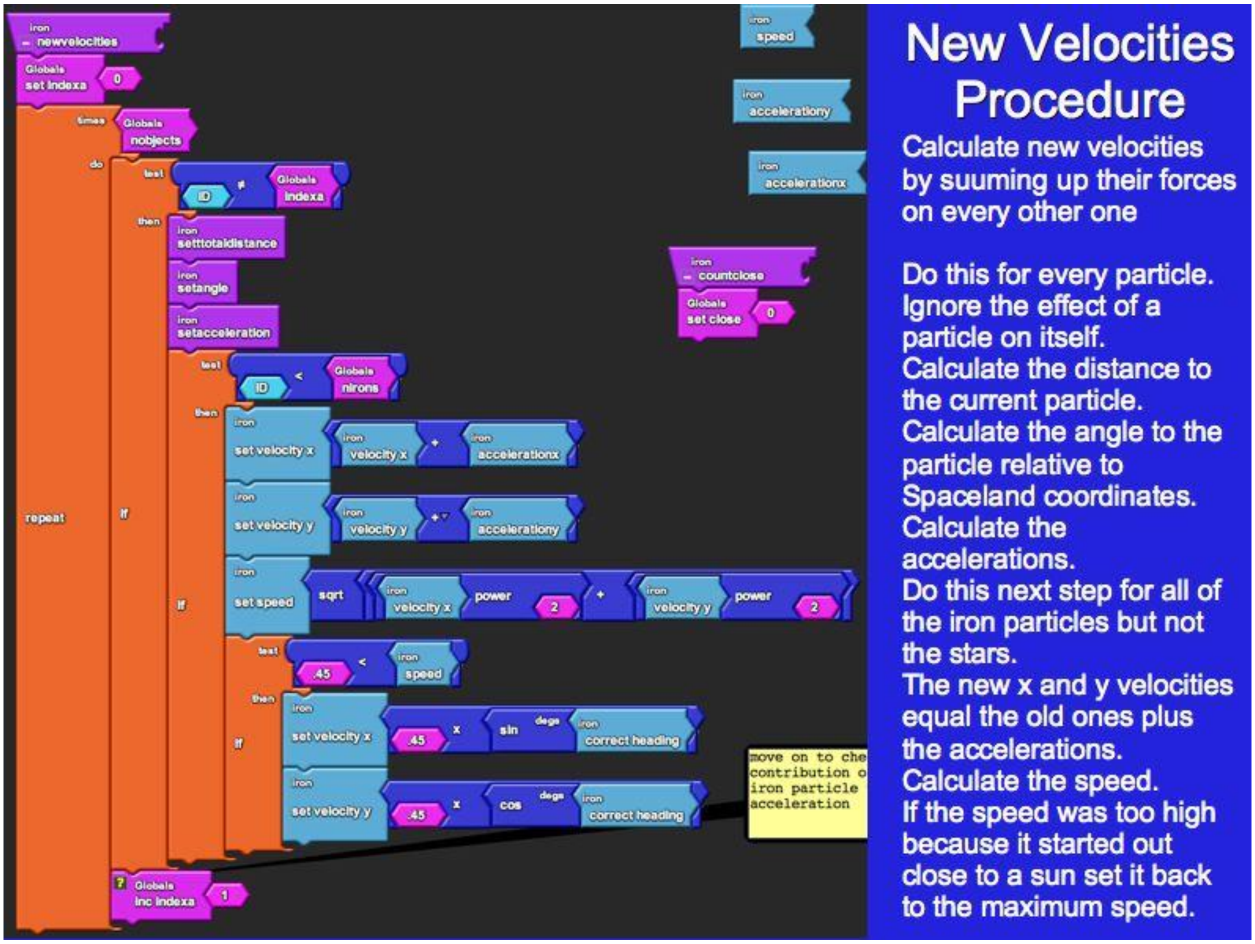
The x component (xdistance) is the difference between the current particle and the particle specified by indexa. The y component is found identically.
The toal distance is the sum of the squares of the x and y distances (Pythagorean's theorem). I leave it squared because that's what goes into the gravitational force so I would just be squaring it again anyway.


iron
setacceleration
test
iron
totaldistance
>
.000001
then
iron
set acceleration
.00675
x
Globals
mass of
ID
Globals
indexa
÷
iron
totaldistance
iron
set accelerationx
iron
acceleration
x
1
x
sin
degs
iron
angle
ifelse
iron
set accelerationy
iron
acceleration
x
cos
degs
iron
angle
else
iron
set acceleration
0
iron
set accelerationx
0
iron
set accelerationy
0
Set Acceleration Procedure
This procedure calculates the x and y components of the acceleration. I don't do this for the particle itself, which I test by seeing if the total distance is almost 0 (in case of roundoff error.) Right now G=0.00675, the mass is that of each particle (indexa) through the loop, and I divide by the distance squared.
The x component of the acceleration is the product of the total acceleration and the sine of the angle between them (SpaceLand has a different coordinate system) and the y component of the acceleration is the product of the total acceleration and the cosine of the angle betweent the two particles.
I do this for every particle. After summing it all up I get the net acceleration.

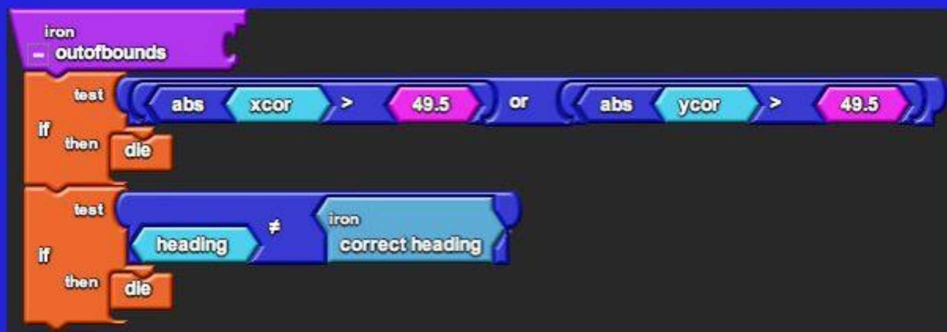
# Set Angle Procedure




This procedure determines the angle between two objects (iron particles) by using the StarLogo "towards" command.

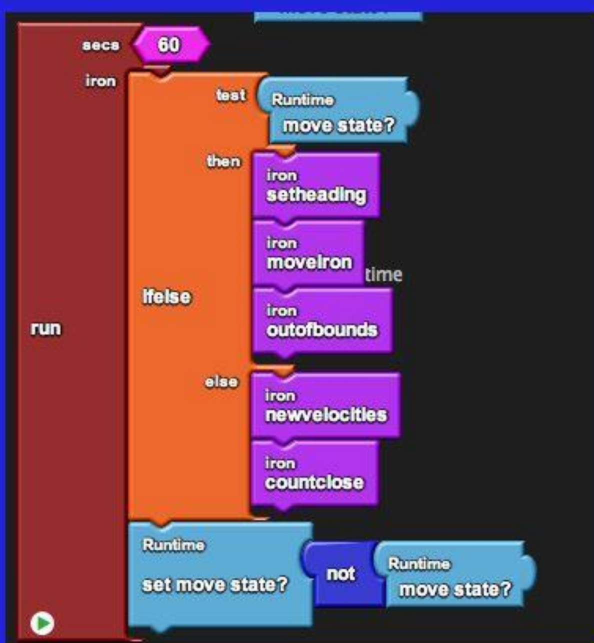

New Velocities Procedure
Calculate new velocities by suuming up their forces on every other one
Do this for every particle.
Ignore the effect of a particle on itself.
Calculate the distance to the current particle.
Calculate the angle to the particle relative to Spaceland coordinates.
Calculate the accelerations.
Do this next step for all of the iron particles but not the stars.
The new x and y velocities equal the old ones plus the accelerations.
Calculate the speed.
If the speed was too high because it started out close to a sun set it back to the maximum speed.
newvelocities
set indexa
repeat
set velocity x
set velocity y
set speed
sqrt
power
correct heading
countclose
set close
inc indexa

# Out Of Bounds Procedure




If a particle leaves SpaceLand (if its x or y coordinate gets very close to 50 or -50) then it dies.

# Run Module




60 seconds at a time

If the move state is true then move the particles
If the move state is false than calculate their new velocities for the next move

To move them calculate their directions, make the move, and if they are not out of bounds then kill them

To calculate their new velocities do it.

Count close isn't implemented yet, it's to keep score.

Toggle the move state.

Keep looping for 60 seconds.