

Oil Spills in the Ocean

*New Mexico Supercomputing Challenge
Final Report
November 10, 2009*

*Team 109
Quemado High School*

Team Members:
Justin Miller

Project Mentor:
Mrs. Laura Larisch

Table of Contents:

3- Thesis Statement

4- Introduction

5-Hypothesis

5- Materials

6-Methods

7-Data

23-Results

24- Bibliography

Thesis Statement

I will be making a computer simulation about what happens to oil spills when they fall on one of the many ocean currents. Oil density is the key when making the oil spread throughout the ocean.

Introduction

Most people have heard of an oil spill of some sort, be it land or water. I am going to be working on an informative paper on oil spills in the ocean. The primary goal of this project is to determine how the oceans water temperature and the oil density affect an oil spill. The secondary goal is to design a simulation that demonstrates how the oil will move when each of these variables are added.

The simple question is, “how is an oil spill affected by the ocean and its many variables”. This question requires much dedication to time and learning the information on this subject. To create an efficient model, uncontaminated water was necessary. After that, all of the variables will be added. Then the oil spill will be added. Instead of having many oil densities, I will have one oil density. The density will be the average density of crude oil. The simulation will give an example of how the oil density makes the oil behave in that certain environment.

Since oil spills are not just about the main oil spill, I will also allow the user to change certain variables values. Scientific principles of oil are important to developing a valid summary of oil spill. Oil density and viscosity are very important to the success of the simulation. That is the overview of the oil spill project.

Hypothesis

My hypothesis is that the oil will move with the ocean current. I also think that when the water temperature gets too high, the oil density will decrease; when the water temperature gets too low, the oil density will increase. In addition, I think that when the oil density is high the oil will move slower than when the oil density is lower.

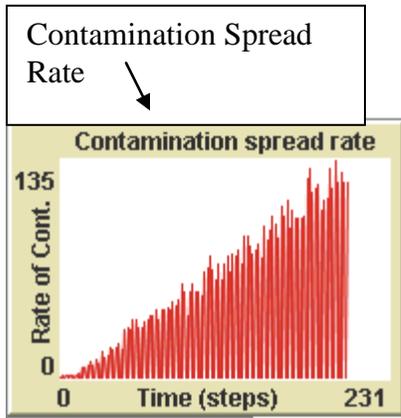
Material list

- Computer
- NetLogo (Computer simulation making software)
- Notebook binder
- Internet resources
- Human resources

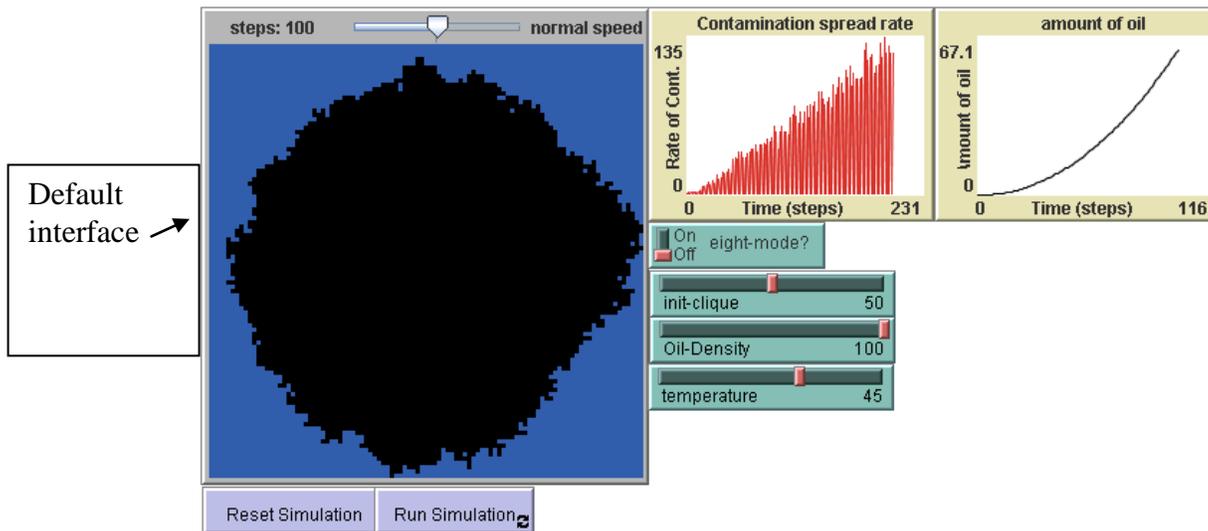
Methods

Step 1 is to research how ocean density and the viscosity of oil affect an oil spill. Step 2 involves programming the NetLogo application to make a simulation of an oil spill. Step 3 is the process of understanding the graphs obtained from the simulation. Step 4 intergrades all of the compiled data from the project.

Data:



Time progressed in steps



Inside the computer simulation, I first have the simulation globals. A simulation global is a variable that has values all through out the simulation. Not just one object. In this simulation, I had two globals. They are color-mode and clique. First, color-mode stands for 0 = water, 1=oil with oil. Second, there is clique. This global gets a value that returns how much water is contaminated with oil. The variable, clique is the basic counter for the graphs. The graphing mechanism watches that specific variable. Those are the global variables inside my simulation.

The format to type the globals in NetLogo for this specific simulation is:

```
globals [
  color-mode ;; 0 = water, 1=oil w/ oil
  clique ;; how much water is contaminated with oil
]
```

For those familiar to NetLogo, you might think that the second part of the simulation code is the breeds. Actually, the next part in my simulation is “patches-own”. The variable “patches-own” is a code that has a value that tells the computer what the patches will be. Without “patches-own”, the simulation would not know what variable to track the amount of oil with, it would not know what variable to remember what patch was first contaminated, and it would not know what variable to use for tracking which patch just was contaminated. The patches-own are amount-oil, first contaminated, and just-spilt? First, amount-oil tracks how much oil has been spilt and is being spread over the ocean surface. First-contaminated tracks witch patch was the first to be contaminated with oil. Finally, just-spilt? Keeps track witch patch or patches was just contaminated with oil. They keep track of what is happening inside the simulation.

The proper format to code these inside NetLogo is:

```
patches-own [
  amount-oil   ;; tracks how much oil is spread
  first-contaminated   ;; clocks which was the first patch to be contaminated
  just-spilt?   ;; tracks which patch just was contaminated
]
```

Next in the simulation codes, I have the setup. Now, the setup is a very important part of the simulation. It resets everything and tells all the patches to get into there starting position. In the setup, I first have “to setup”. This tells the simulation to expect setup codes. Next, I have “ca”. This resets the simulation. It is an abbreviation for clear all. After that, I have “set color-mode 0”. That sets the color mode to 0. If you recall, in the globals I had a global called color-

mode. This sets the value to that variable. After that, I have, “set clique 0”. This sets the variable clique to 0. Next on the list I have, “ask patches”. This asks the patches something. Next there is “[set first-contaminated -1
 set amount-oil 0
 set just-spilt? false
 recolor]”.

This is what the computer asks the patches. This code first sets the first contaminated patch to -1. Then it sets the amount of oil to 0. This is the reset code for the oil. After that the computer sets the question, was the oil just spilt to false. Then the simulation sets a recolor event. This colors everything back to normal; before the oil spill. Then I ask if anything else happens. To do this I type “ifelse seed-one?”. This asks if anything else happens, check to see if there is one seeded. If there is one seeded, the computer seeds one again, then it seeds randomly. Then I have an update action, this updates the simulation. In the setup code, I have “do-plots”. This plots the events happening. In addition to the setup code, I have two final pieces of code. The first is “set temperature 45”. This sets the water temperature to 45. I chose 45 because that is an average ocean temperature. The final piece of code for the setup code is “set Oil-Density 72”. This sets the oil density to 72%. This is an average oil density for crude oil. Then I tell the computer to end the setup code. To do this I typed, “end”. Next, I have to tell the computer what is seed-one, and how to react to it. To do this I first typed “to seed-one”. This told the computer what the next lines of code are going to be doing. Next I have “ask patch 0 0”. This asks the patch at coordinates (0,0) a question. The question the computer asks is expressed in this code, “[spill-oil]” This asks the patch to spill the oil. Finally, I tell the computer to end this block of code, I do this by typing, “end”. After telling the computer what to do when the “seed-one” or “seed-one?”

is asked, I need to tell the computer what to do when “seed-random” is called. To do this I first, like always type, “to seed-random”. Like always, this tells the computer that the next block of code is to seed-random. Next, I have, “ask patches with [amount-oil = 0]”. This asks the patches with the amount of oil equal to 0 a question. The Question it asks is expressed this way, “[if (random-float 100.0) < init-clique [spill-oil]]”.

This asks if the random float is equal to 1000 and greater than the initial clique is. If it is, spill the oil. Then I told the computer to end this block. To do this, I typed, “end”.

After I have told the computer all of that, I need to tell the computer how to react when the simulation is ran. To run the simulation, I have a button named go. Therefore, I need to tell the computer how to “go”. To this I have “to go” as the first line of code for the next block of code. After I tell the computer what the following lines of code are for, I tell the computer what the following lines of code are. First, I have “if all? patches [amount-oil > 0]”. This asks if all the patches had an amount of oil smaller than 0. It will return either true or false. If the question returns true, the following line of code is used, “[stop]”. This stops the simulation. Next, I wanted to ask the patches another question. To do this, once again I typed, “ask patches”. I still have to ask the patches something. I needed to ask them if the amount of oil is smaller than 0. To do this I typed “[if amount-oil > 0]”. This will return either true or false. If it returns true, the next line of code is ran. The next line of code is “[spread-oil]]”. What this does is if the expression returns true, the oil will start to spread over the simulated oil surface. Now that the computer has done alot of calculations, I updated the simulation. To do this I simply typed, “update”. It is that simple. The computer reads each step in ticks. The computer needs to know when to count for each step or tick. At the current position of this block of code, it would be a

good time to tell the computer to count a tick. So tell the computer to count a tick, I type “tick”. When the computer reads this line of code, it will count a tick. Since the plots are being by this time, I tell the computer to update the plots. To do this, I type “do-plots”.

If you can recall, one of my goals was to show how temperature affect how the oil distributes over the ocean surface. Next in the “go” block of code, I tell the computer what to do when the temperature is at a specific value. First, I tell the computer what to do when the ocean temperature is become higher than normal. I typed this into the computer to do this, “if temperature < 50

[set Oil-Density 40]”. This asks if the temperature is greater than 50. It returns either true or false. If it returns true, it sets the oil density to 40. This is a very low oil density for crude oil. Next, I did the exact opposite. The next line of code is “if temperature > 30

[set Oil-Density 100]”. This asks if the temperature is less than 30. It returns either true or false. If it returns true, it sets the oil density to 100. Now that I have told the computer what to do when the ocean temperature changes, I decided to go to the extreme. I decided to allow the user to change the temperature to a freezing point where the water would be completely frozen and the oil would be extremely dense. Crude oil does not have an exact freezing point. The reason for it being that crude oil is made up of hundred or even thousands of different materials. Each having its own unique freezing point. To tell the computer what to do if the user changes the water temperature to an extremely low temperature, I typed the following,

“if temperature < -25

[user-message "The water temperature has become too low to accurately simulate the oil spill anymore. The ocean surface would be completely frozen and the oil would be too dense to move. The simulation is no longer accurate."

set temperature 35]”.

This tells the computer that if the water temperature become below -25 a message is shown. The message will tell the user that the water temperature has become to low to accurately simulate the oil spill anymore. The ocean surface would be completely frozen and the oil would be too dense to move. The simulation is no longer accurate. The computer automatically makes the water temperature rise to 35. There is one more temperature command that needs to be covered. The last line of code before I end this block of code is,

```
“if temperature > 100
  [ set Oil-Density 2 ] “.
```

This asks if the temperature is grater than 100, the oil density will automatically lower to two. That is the end of the run block of code. Therefore, I need to end it. To do this I typed, “end”.

Now I need to tell the computer how to spread the oil. The first line of code is, “to spill-oil”. This like always tells the computer what the next block of code is used for. The next line of code is,

```
“if first-contaminated = -1
  [ set first-contaminated ticks
    set just-spilt? true ]”.
```

This asks if the first contaminated patch is equal to -1. It will return either true or false. If it returns true, the computer sets the first contaminated tick. Next, it asks if the variable “just-spilt” has been set. The next line of code is

```
“set amount-oil amount-oil + 1
  recolor”.
```

This is what the computer does if the variable “just-spilt” has been set. The computer sets the amount of oil to 1 then recolor’s the surface. It will recolor the surface to blue. There is one last line of code that needs to be programmed in before I can end this block. The last line of code is, “if temperature < 50

[set Oil-Density 40]”. This asks if the temperature is smaller than 50 set oil density to 40. Now I can end this block of code. To do this I typed “end”.

The next block of code before the simulation is done. The first line of code is, “to update”. This tells the computer that the next block of code is for updating. The next line of code is, “ask patches with [just-spilt?]”. This asks if the patches that I’ve just spilt. The answer is returned either true or false. If this returns true the next line of code is ran. The next line of code is

```
“[ set just-spilt? false
  spill-oil ]”.
```

This sets the variable “just-spilt” to false and it spills the oil. Now I can end the block of code. To do this I typed, “end”.

The next blocks of code are partners. They run together to do specific functions. This is the blocks of codes: to recolor ;; patch procedure

```
ifelse color-mode = 0
  [ recolor-normal ]
  [ ifelse color-mode = 1
    [ recolor-by-when-spilt ]
    [ recolor-by-when-spilt ] ]
end
```

```
to recolor-normal ;; patch procedure
```

```
  ifelse first-contaminated >= 0
```

```
    [ set pcolor black ]
```

```
    [ set pcolor blue ]
```

```
end
```

```
to recolor-by-when-spilt ;; patch procedure
```

```
  ifelse first-contaminated = -1
```

```
    [ set pcolor black ]
```

```
    [ set pcolor scale-color yellow first-contaminated world-width 0 ]
```

```
end
```

```
to recolor-by-times-heard ;; patch procedure
```

```
  set pcolor scale-color green amount-oil 0 world-width
```

```
end
```

```
;;; mouse handling
```

```
to spread-oil-with-mouse
```

```
  if mouse-down?
```

```
    [ ask patch mouse-xcor mouse-ycor
```

```
      [ spill-oil ]
```

```
        display ]
end

;;; plotting procedures

to setup-plots
  set-current-plot "Successive Differences"
  set-plot-y-range 0 (count patches / 5)
end

to do-plots
  let new-clique count patches with [amount-oil > 0]
  set-current-plot "amount of oil"
  plot (new-clique / count patches) * 100
  set-current-plot "Contamination spread rate"
  ifelse clique = 0
    [ plot 1 ]
    [ plot new-clique / clique ]
  set-current-plot "Contamination spread rate"
  plot new-clique - clique
  set clique new-clique
end
```

These are the coloring functions. What they do is recolor the patches to blue or black. That is the basic function of all those codes. After you have written all that code, this is what the all the code put together should look like: globals [

```
color-mode ;; 0 = water, 1=oil w/ oil
```

```
clique ;; how much water is contaminated with oil
```

```
]
```

```
patches-own [
```

```
amount-oil ;; tracks how much oil is spread
```

```
first-contaminated ;; clocks which was the first patch to be contaminated
```

```
just-spilt? ;; tracks which patch just was contaminated
```

```
]
```

```
;;; setup procedures
```

```
to setup [seed-one?]
```

```
ca
```

```
set color-mode 0
```

```
set clique 0
```

```
ask patches
```

```
[ set first-contaminated -1
```

```
set amount-oil 0
```

```
set just-spilt? false
```

```
recolor ]
```

```
ifelse seed-one?
```

```
  [ seed-one ]
```

```
  [ seed-random ]
```

```
update
```

```
do-plots
```

```
set temperature 45
```

```
set Oil-Density 72
```

```
end
```

```
to seed-one
```

```
  ;; spill the oil in the middle
```

```
  ask patch 0 0
```

```
    [ spill-oil ]
```

```
end
```

```
to seed-random
```

```
  ;; seed with random number of oil sources governed by init-clique slider
```

```
  ask patches with [amount-oil = 0]
```

```
    [ if (random-float 100.0) < init-clique
```

```
      [ spill-oil ] ]
```

```
end
```

to go

if all? patches [amount-oil > 0]

[stop]

ask patches

[if amount-oil > 0

[spread-oil]]

update

tick

do-plots

if temperature < 50

[set Oil-Density 40]

if temperature > 30

[set Oil-Density 100]

if temperature < -25

[user-message "The water temperature has become too low to accurately simulate the oil spill anymore. The ocean surface would be completely frozen and the oil would be too dense to move.

The simulation is no longer accurate."

set temperature 35]

if temperature > 100

```
[ set Oil-Density 2 ]
```

```
end
```

```
to spread-oil ;; patch procedure
```

```
  let neighbor nobody
```

```
  ifelse eight-mode?
```

```
    [ set neighbor one-of neighbors ]
```

```
    [ set neighbor one-of neighbors4 ]
```

```
  ask neighbor [ set just-spilt? true ]
```

```
end
```

```
to spill-oil ;; patch procedure
```

```
  if first-contaminated = -1
```

```
    [ set first-contaminated ticks
```

```
      set just-spilt? true ]
```

```
  set amount-oil amount-oil + 1
```

```
  recolor
```

```
if temperature < 50
```

```
  [ set Oil-Density 40 ]
```

```
end
```

to update

ask patches with [just-spilt?]

[set just-spilt? false

spill-oil]

end

;;; coloring procedures

to recolor ;; patch procedure

ifelse color-mode = 0

[recolor-normal]

[ifelse color-mode = 1

[recolor-by-when-spilt]

[recolor-by-when-spilt]]

end

to recolor-normal ;; patch procedure

ifelse first-contaminated >= 0

[set pcolor black]

[set pcolor blue]

end

to recolor-by-when-spilt ;; patch procedure

```
ifelse first-contaminated = -1
  [ set pcolor black ]
  [ set pcolor scale-color yellow first-contaminated world-width 0 ]
end
```

```
to recolor-by-times-heard ;; patch procedure
  set pcolor scale-color green amount-oil 0 world-width
end
```

```
;;; mouse handling
```

```
to spread-oil-with-mouse
  if mouse-down?
    [ ask patch mouse-xcor mouse-ycor
      [ spill-oil ]
      display ]
end
```

```
;;; plotting procedures
```

```
to setup-plots
  set-current-plot "Successive Differences"
  set-plot-y-range 0 (count patches / 5)
```

end

to do-plots

let new-clique count patches with [amount-oil > 0]

set-current-plot "amount of oil"

plot (new-clique / count patches) * 100

set-current-plot "Contamination spread rate"

ifelse clique = 0

[plot 1]

[plot new-clique / clique]

set-current-plot "Contamination spread rate"

plot new-clique - clique

set clique new-clique

end

Results:

The amount of poisoned water increases as the amount of time progresses. The oil surface area increases rapidly in warmer temperature, the heat makes the oil less viscous and thus letting it flow more fluid like atop the water. After the temperature increases enough the oils, viscosity will be the same as the water and will mix causing massive pollution.

Bibliography:

(2009). Density, Mass, SG of Liquids, Retrieved November 12, 2009.

http://www.simetric.co.uk/si_liquids.htm.

(2009). How Do You Clean Up An Oil Spill, Retrieved November 5, 2009.

<http://www.ceoe.udel.edu/oilspill/cleanup.html>.

(2009). Office of Response and Restoration/Oil Types, Retrieved October 27, 2009.

http://response.restoration.noaa.gov/type_topic_entry.php.

(2009). What happens When An Oil Spill Occurs, Retrieved November 4, 2009.

<http://www.teachersdomain.org/resource/ess05.sci.ess.watcyc.oilspill/>.

(2009). Oil Spills: Impact on the Ocean, Retrieved December 10, 2009.

<http://www.waterencyclopedia.com/Oc-Po/Oil-Spills-Impact-on-the-Ocean.html>.

(2010). Use of Computer Simulation in Oil Spill Response Training, Retrieved January 12, 2010,

<http://sim.sagepub.com/cgi/content/abstract/41/5/175>.