

THE CONTROL AND SPREAD OF WILDFIRES III

New Mexico Supercomputing Challenge

Final Report

April 6, 2011

Team: 82

Melrose High School

Team Members:

Richard Rush

Kyle Jacobs

Randall Rush

Brandon Mitchell

Sponsoring Teachers:

Allen Daugherty

Rebecca Raulie

TABLE OF CONTENTS

1. Problem Explanation
2. Solving the Problem
 - 2.1. Variables
 - 2.1.1. Fuel Load
 - 2.1.2. Moisture Content
 - 2.1.3. Wind Direction
 - 2.1.4. Wind Speed
 - 2.1.5. Topography
 - 2.2. Procedures
 - 2.2.1. Burn
 - 2.2.2. Spread
 - 2.2.3. Wind
 - 2.2.4. Topography
 - 2.3. Mathematical Model
 - 2.3.1. Setup
 - 2.3.2. Spread
 - 2.3.3. Burn
 - 2.3.4. Topography
3. Previous Work
4. Goals
5. Progress
 - 5.1. Translating From StarLogo TNG to NetLogo
 - 5.2. Beta Trials
6. Executive Summary
7. Works Cited
8. Acknowledgments
9. Current Code in NetLogo

Problem Explanation

Annually in the US over 4000 people die from fires, and 25,000 are injured. (FEMA.gov) Uncontrolled fires also cause an estimated annual damage of over 8 billion dollars. For this reason, we thought it important to continue our previous year's project on modeling how a fire spreads and how to best control it.

We did this as we felt that our two previous year's work did not satisfactorily complete the goal of providing firefighters with a usable tool that will predict a fire's behavior.

This year we have changed modeling programs from StarLogo TNG to NetLogo to help accomplish our original goal of creating a tool that can be used by the fire-fighters themselves, or by a central command center, on, or off the field to direct personnel to where they are needed most. Changing programs will allow us to further improve the realism and utility of the model by increasing the modeling area and importing satellite images and topographical data to make it as useful and effective as possible.

Solving the Problem

We were able to develop a good model during the last two years. However, we have not accomplished some of our goals, such as importing satellite images, topographical maps, and creating a more detailed modeling area. We had been using StarLogo TNG, as it was easy to learn and to use due to its unique format. However, we were limited in program versatility and in how large of an area we could model. Changing programs to NetLogo would allow us to increase the modeling area, import topographical data, and layer a satellite image over the area, thus making the model more relevant and realistic. NetLogo is an agent based model. Our model works by placing fire agents in the interface with fuel, moisture, wind, and topography. The fire then consumes fuel and creates a new fire moving it in a direction and speed according to the variables affecting it.

Variables

Even though we are changing programs, the basic procedures and variables have remained the same.

Fuel Load—The amount of fuel available per square area of land. This represents different vegetation types and densities. This will be represented in our model by each patch being assigned a specific number. The higher the number, the greater the fuel load. The fuel load in an area can change greatly, on one side of the fence there can be a ploughed field which has an extremely low fuel load, and on the other side of the fence, there can be land enrolled in CRP (Conservation Reserve Program) which can have a very high fuel load. The fuel load in a single pasture can change too, depending on what kind of grass grows from place to place and the quality of the soil across a field. We can assign any amount of fuel to a patch, however as we are trying to create a model to be used by the firemen in the field we will limit the amount of different fuel levels making it more user friendly. We will draw in the fuel as this data cannot accurately be obtained from recorded sources as the fuel load in an area will change from season to season and from year to year.

Moisture Content—The amount of moisture in the area. Often the land will be very dry before a thunder storm, therefore, lightning can easily start a fire. As the storm progresses it may rain, increasing the moisture, retarding the fire's progress, and possibly even putting the fire out. This variable can also represent the growing stage of the vegetation represented and humidity. This is represented in our model by increasing or decreasing the fire's ability to spread. The greater the moisture or humidity, the less chance the fire will spread.

Wind Direction—Wind direction plays a vital role in fire behavior and control. If there is a wild fire, the fire fighters may have had it under control or could have had a firebreak set up ahead of it, but if the fire suddenly changes direction, it may become uncontrolled once again necessitating a strategy change.

Wind Speed—Wind speed is also very important in fires. A fire under a light wind might not ordinarily cross a firebreak or obstacle such as a road, but if the wind were blowing enough, the road would barely slow down a raging fire. We will use these numbers to represent the different wind speeds:

0—0-3 mph

3—15-25 mph

1—3-10 mph

4—over 25 mph

2—10-15 mph

Topography—The heat that radiates from a fire tends to rise, therefore a fire will advance much more quickly up a hill than down a hill. The wind will also be affected by the terrain. A hill can provide a wind break on one side and channel wind down a canyon on the other side of the hill, thus changing wind speeds and directions. We have experimented with draw in hills and elevation; however our best option is to take advantage of NetLogo's ability to import topographical information from outside sources.

Procedures

Our model runs on these basic procedures:

1) Burn

- a) Test to see if patch color has fuel
- b) If so, then subtract one from the fuel amount
- c) If not, then have a 66% chance of dying (This gives a fire a chance at crossing a firebreak with sparks and tumbleweeds).

(The 66% was determined through trial and error relying on local firefighter's experience of fire behavior.)

2) Spread

- a) Test to see if a random number between 0-100 is less than or equal to dryness.
- b) If so, then choose a random number between 0-360 for a direction
 - i) Then create a new fire and send it 1 step in the chosen direction.

3) Wind

- a) Select a random number between 0-45 and add it to the wind direction
- b) Then divide wind speed by 2 and add 0.5 and take that many steps in the chosen direction.

4) Topography

- a) Test if patch height is less than patch height ahead.
 - i) If so, then go forward the difference between patch height and patch height ahead times 4.
 - ii) Create a new fire
- b) Test if patch height is greater than patch height ahead.
 - i) If so, then go back the difference between the patch height ahead and patch height times 1.
 - ii) Have a 50% chance of dying.

Our topography procedure uses numbers that will cause the fire to race up a hill and slow down when it goes down the hill. These particular numbers have not been verified due to the fact that we have not been able to incorporate actual topographical data to compare against an actual fire. This procedure has posed an obstacle in reaching our stated goals.

Mathematical Model

The mathematical formulas that our model follows are stated below.

Setup

F_1 = first fire agent generated

F_1 is placed on an x y grid according to ...

$x \sim U[-200.5, 200.5)$

$$y \sim U[-200.5, 200.5)$$

Where U is uniform distribution.

(The size of the grid is variable and we can change it to model a larger area, or a smaller area in greater detail.)

F_i is placed on a “patch”.

A “patch” is an area that is centered on an x y grid where $x_0, y_0 \in \mathbb{Z}$

A “patch” includes...

$$x \in [x_0 - 0.5, x_0 + 0.5)$$

$$y \in [y_0 - 0.5, y_0 + 0.5)$$

Spread

f_i = fire agent i

t = time

P = probability

The probability that a fire sparks another fire is ...

$$P(f_i \text{ produces } f_j), [t, t+1)$$

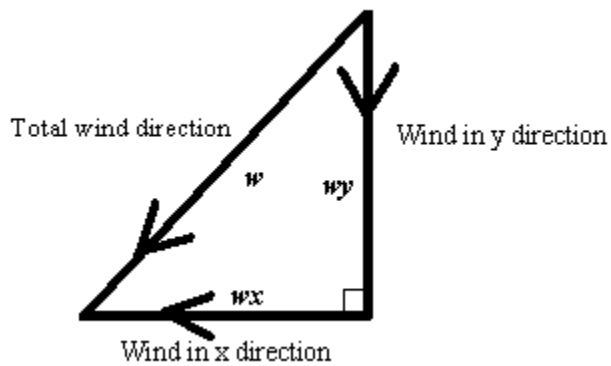
The location of f_j is...

$$x_j = x_i + \cos \Theta_r + \mathbf{w}x$$

$$y_j = y_i + \sin \Theta_r + \mathbf{w}y$$

Where “ Θ_r ” is a random angle uniformly distributed on $[0, 2\pi)$

Where “ \mathbf{w} ” is a wind vector composed of x and y directions



Burn

L_k = fuel level in “patch” K

The fuel level decreases by the number of “fires” on that “patch”

$$L_{kt+1} = \max(L_{kt} - n, 0)$$

Where “n” is the number of fire on “patch K”

If $L_{kt} = 0$, all agents on “patch K” “die”, in the time interval, $(t, t+1]$

Topography

If there is a positive grade

$$F_1 \text{ moves forward (marginal difference)(4)}$$

If there is a negative grade

$$F_1 \text{ moves backwards (marginal difference)(1)}$$

Previous Work

Two years ago, we created an accurate representation of the way that a fire spreads, and responds to variables such as fuel load, moisture, and wind speed and direction. We used this to model a fire crossing a road under different circumstances, such as with high wind speed and mowed bar ditches, or perhaps a low wind speed, but with over grown ditches. We verified our model by showing it to the local volunteer fire fighters and having them tell us if our model accurately represented real fire behavior or not.

Last year, we improved the model so that it also incorporated elevation changes, and how wind reacts to the changes in elevation. Additionally, we determine an equation to find the effective distance of a fire-break based on the angle of approach which is as follows:

— when D is the effective distance, w the width of the fire-break, and θ the angle of the fire's approach. The equation measures the distance of a fire-break or road that the fire has to cross. We then used the model to determine the best fire-fighting techniques for different scenarios, and the best design of a properly placed fire break.

We tested several different firebreak designs with the purpose of protecting a small residential area. From the different designs we tested we determined that to best protect a small residential area is to create a V shaped fire break around the area, with the fire-fighters “staging” the fire just inside of it. Another firebreak we tested had similar results, so we statistically analyzed them. From this we determined that the V shaped break was superior. This particular fire break will prevent the fire from destroying the protected area, however, it creates two smaller fires that can grow and spread individually.

From the incorporation of fire fighting agents in our model we learned the importance of a central command system in the field. Our model represented a group of firemen running to the nearest fire and putting it out. Occasionally this had some interesting results. When the fire would cross the firebreak in one place all of the firemen would run to do the job that only required one fireman. While all the firemen were occupied, the fire would cross the firebreak behind them and get away from them. Sometimes the problem would be exactly opposite.

Occasionally it would have been advantageous to the firefighters to have concentrated on one section of the fire, but instead they were scattered out and the fire got away from them.

Goals

Our goals this year included changing our modeling languages from StarLogo TNG to NetLogo. We wanted to do this due to the fact that we had essentially “maxed out” StarLogo TNG with the vast number of agents involved, the small modeling area, and the complexity of the code. With NetLogo we are not constrained by the blocks of code provided to us. Another reason that we wanted to switch was because with NetLogo, it is possible to import actual topographical data from GIS sources and satellite maps. This is very important in creating an accurate model that is usable by fire fighting forces.

Progress

Currently, we are on the verge of creating a model usable by local fire departments. We have most of our previous program translated into NetLogo. With the exception of topography and firefighting units, we have our program operating exactly the same. We have also layered Google Earth images on top of our model. These images do not affect the model. Rather, they offer a visual aid as to exactly where the fire is burning and provide a guide to draw in certain variables such as elevation and fuel.

Translating from StarLogo TNG to NetLogo

The conversion from StarLogo TNG to NetLogo was an interesting one. There is a line to be drawn between them as far as commands and logic go, but it's a curvy line full of gray areas and holes. This is because they both take and add different things from their parent, StarLogo, almost as if taking different evolutionary paths.

For example, NetLogo treats its patches (each separate unit of ground) almost as agents. Although they can't move, you can ask a patch to do something or assign it new variables. In StarLogo TNG, you can't directly ask a patch to do anything; instead, the agent must manipulate

it or observe it and act according to one of the three built in variables patches have – color, height, and location (x coordinate, y coordinate). These new capabilities are useful to us because we can assign numerical values to a patch and call it 'fuel' whereas in StarLogo TNG we had to color code fuel loads for the agents to recognize how much fuel was on the patch. While patch color still does change in NetLogo, it is strictly for the benefit of the observer and doesn't otherwise affect the model at all. Furthermore, rather than having a "landscaper" agent reach each individual patch and change its values, we can make global adjustments much easier and more efficiently.

In general, most of the logic is the same although the phrasing is a bit different. Most differences are subtle. Some examples from our code include:

"if {[(int patch color / 10) * 10] = green}" in StarLogo TNG

Becomes in NetLogo: "if fuel != 0".

"Set shape (model picture)" in StarLogo TNG

is now "set-default-shape (agent breed name) "(model name)" " in NetLogo.

With a working knowledge of both programs, most procedures can be converted fairly smoothly. However, the small differences between them made it function a bit better in NetLogo. It was a delicate procedure, but it went over smoothly.

Beta Trials

With the assistance of Dr. Shroader, a professor at Eastern New Mexico University, we set up models of Melrose and Floyd we proportioned the map to the code. Then we modeled a fire on a mountain side between Mora and Las Vegas, New Mexico. We used our draw in elevation and fuel methods to set it up. Our model behaved rationally and logically, however we do not have actual verification for it. The following screenshots are of a fire at Floyd with the wind set out of the west at approximately 20 miles per hour with 0 percent humidity.

A Model of a Possible Fire at Floyd

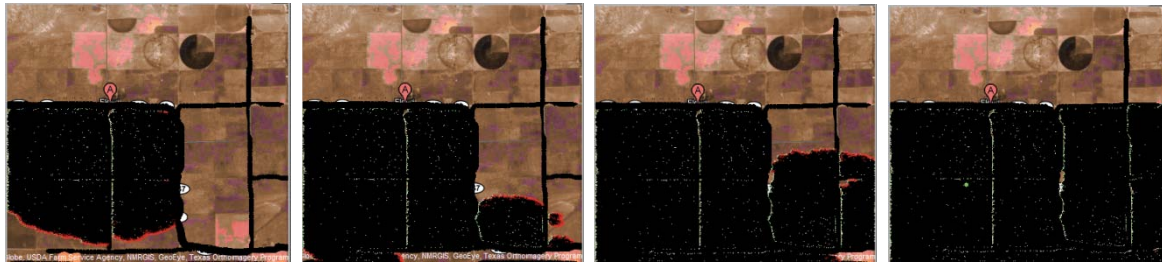


Fuel Layer

Graphic Layer

Running Code

Running Code



Running Code

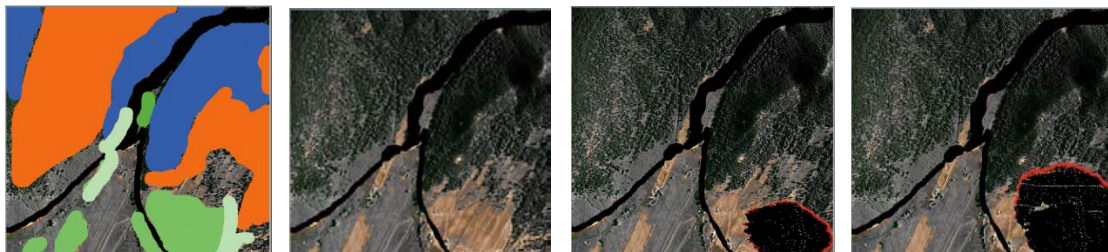
Running Code

Running Code

Running Code

In this particular model a fire was started to the south side of Highway 267 west of Floyd, New Mexico. We can see that the highway provided a good firebreak on the north side of the fire and significantly slowed it on the east side. The fire did cross it though and continued to cross county roads that it met at a 90 degree angle. However, the county road running with the fire stopped the fire's progression north.

A Model of a Possible Fire Near Mora

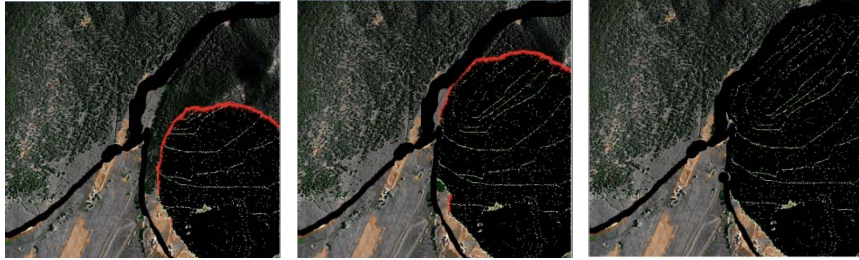


Fuel Layer

Graphic Layer

Running Code

Running Code



Running Code

Running Code

Running Code

In this model the humidity was set at 25 percent with a wind speed of approximately 20 miles per hour. The fire started to the south of a mountain side on a plain with the wind out of the west. After the fire started the wind shifted so that it was out of the south. In this model the fire did not cross the stream bed or the roads as it progressed up the mountain side. We estimated elevation changes and drew them in for testing purposes. When the fire is completely burned, one can see topographical lines. This was where the fire jumped up a “stair step” in elevation and continued to burn across a flat area until it reached another “stair step.” This stair step format of a typical topographical map causes our burns to be erratic and inaccurate. This is why it is important for us to incorporate actual GIS or similar elevation data into our model so that we can set up actual hills instead of steps.

While using draw in elevation methods we have modeled prospective controlled burns. The Forrest Volunteer Fire Department was to help oversee a controlled burn in a canyon of the Cuneva Basin near Ima, New Mexico. However, high winds were predicted the day of the burn. We offered to model the burn beforehand. We found that without more extensive control measures, the fire would quickly get out of control, possibly destroying homes in the nearby area. The burn was canceled before we could relay the results to the rancher planning the event. This provided us with greater insight to the utility of the program in real life.

Executive Summery

This year, we furthered our goals by changing modeling languages and importing satellite images. This has made our model more realistic, versatile, and useful than before, while retaining the ease of use from our previous models. We have used this program to model a proposed burn and we were able to identify certain areas of concern.

We have begun to see more of the potential our model has as a tool to predict controlled burns and future wildfires as well as to be able to develop a plan of action to prevent such a fire or to provide a plan to minimize damage to life and property.

However, we still have work to do. We are still trying to import topographical data into our model in such a way that it can be utilized. This is a significant obstacle that stands in our way of creating a usable product. Once we do get this done though, we will be able to fine tune the topographical aspect of our model. This will provide us with what we ultimately set out to get: an accurate tool that fire fighters, property owners, or even insurance agents can use to predict the fire behavior of a current fire, prepare for a future fire, or test the safety of a proposed controlled burn.

Works Cited

Holecheck, Pieper, Herbel, Range Management, Principles and Practices., 3.

"Fire." wikipedia the free encyclopedia. 2008. 20 Nov 2008

"Fire." FEMA.gov. 2006. 30 Mar 2009

Harris , Tom. "How Fire Works." HowStuffWorks.com. 1998. 20 Nov 2008

David Rush, Forrest Volunteer Fire Department, Training Coordinator

Addison-Wesley, Chemistry of Firefighting, copyright 1990

Bender, Douglas, Kramer, Statistical Methods for Food and Agriculture, copyright 1982

Acknowledgements

We would like to thank Dr. Schroeder of Eastern New Mexico University for taking time to go over our project with us.

We would also like to thank David and Ray Rush of the Forrest Fire Department for helping us verify our model.

Current Code in NetLogo

```
breed [fires fire]
patches-own [fuel elevation]
```

```
to setup
```

```
  set-default-shape fires "fire"
```

```
  clear-all
```

```
  create-fires 1
```

```
  ask fires [
```

```
    setxy startx starty
```

```
    set color red
```

```
  ]
```

```
  ask patches [
```

```
    set fuel 5
```

```
    colorGrass
```

```
    set elevation 0
```

```
  ]
```

```
end
```

```
to colorGrass
```

```
  if fuel = 5
```

```
    [set pcolor green]
```

```
  if fuel = 4
```

```
    [set pcolor green + 1]
```

```
  if fuel = 3
```

```
    [set pcolor green + 2]
```

```
  if fuel = 2
```

```
[set pcolor green + 3]
if fuel = 1
  [set pcolor green + 4]
if fuel = 0
  [set pcolor black]
if fuel = 6
  [set pcolor orange]
if fuel = 7
  [set pcolor blue]
end
```

```
to fireStart
  ask fires [
    burn
    spread
    if windSpeed > .5 [
      wind
    ; wind+slope
    ]
    hill
  ]
end
```

```
to burn
  ask patch-here [
ifelse fuel != 0 [
  set fuel (fuel - 1)
```



```
] [  
if random 3 != 1  
  [ask turtles-here [die]  
  ]  
]  
colorGrass  
]  
end
```

```
to spread  
  if random 100 <= dryness [  
    set heading random 360  
    hatch 1 [forward 1]  
  ]  
end
```

```
to wind  
  set heading random 45 + windDirection  
  forward .1 + (windSpeed / 2)  
end
```

```
to wind+slope  
  
end
```

```
to hill  
  carefully [  
    ]
```

```
if (elevation > [elevation] of patch-ahead 1)
  [ask fires-here [forward (([elevation] of patch-ahead 1 - elevation) * 0.5)]]
if (elevation < [elevation] of patch-ahead 1)
  [ask fires-here [forward (([elevation] of patch-ahead 1 - elevation) * 2)]]
]
[die]
end
```

```
to drawfuel
  if mouse-down? [
    ask patch mouse-xcor mouse-ycor [
      ask patches in-radius tdraw [set fuel (fdraw) colorgrass]
    ]
  ]
end
```

```
to drawelevation
  if mouse-down? [
    ask patch mouse-xcor mouse-ycor [
      ask patches in-radius tdraw [set elevation (edraw) set pcolor white]
    ]
  ]
end
```

```
to drawhill
  if mouse-down? [
```

```

ask patch mouse-xcor mouse-ycor [
  ask patches in-radius int (tdraw * 1.7) [set elevation (edraw / 8) set pcolor orange]
  ask patches in-radius int (tdraw * 1.6) [set elevation (edraw / 8) * 2 set pcolor blue]
  ask patches in-radius int (tdraw * 1.5) [set elevation (edraw / 8) * 3 set pcolor blue + 1]
  ask patches in-radius int (tdraw * 1.4) [set elevation (edraw / 8) * 4 set pcolor red + 1]
  ask patches in-radius int (tdraw * 1.3) [set elevation (edraw / 8) * 5 set pcolor red + 2]
  ask patches in-radius int (tdraw * 1.2) [set elevation (edraw / 8) * 6 set pcolor red + 3]
  ask patches in-radius int (tdraw * 1.1) [set elevation (edraw / 8) * 7 set pcolor red + 4]
  ask patches in-radius int (tdraw * 1.0) [set elevation (edraw / 8) * 8 set pcolor red + 5]

  ask patches in-radius tdraw [set elevation edraw set pcolor green]
]
]
end

```

to drawhill2

```

if mouse-down? [
  ask patch mouse-xcor mouse-ycor [
    ask patches in-radius int (tdraw * 1) [set elevation (edraw / -8) set pcolor orange]
    ask patches in-radius int (tdraw * 1.6) [set elevation (edraw / -8) * 2 set pcolor blue]
    ask patches in-radius int (tdraw * 1.5) [set elevation (edraw / -8) * 3 set pcolor blue + 1]
    ask patches in-radius int (tdraw * 1.4) [set elevation (edraw / -8) * 4 set pcolor red + 1]
    ask patches in-radius int (tdraw * 1.3) [set elevation (edraw / -8) * 5 set pcolor red + 2]
    ask patches in-radius int (tdraw * 1.2) [set elevation (edraw / -8) * 6 set pcolor red + 3]
    ask patches in-radius int (tdraw * 1.1) [set elevation (edraw / -8) * 7 set pcolor red + 4]
    ask patches in-radius int (tdraw * 1.0) [set elevation (edraw / -8) * 8 set pcolor red + 5]
  ]
]

```

```
ask patches in-radius tdraw [set elevation edraw set pcolor green]
]
]
end
```

```
to colors
```

```
if elevation = 0 [set pcolor black]
if elevation = -2 [set pcolor red]
if elevation = -3 [set pcolor blue]
if elevation = 2 [set pcolor red]
if elevation = 3 [set pcolor blue]
end
```