Water, Water Everywhere

Super Computing Challenge

Final Report

April 4, 2012

Team Number 49

Freedom High School

Team Members:

Chris Marroquin

Seth Morgan

Jake Wright

Team Teacher:

Richard Foust

Team Mentor:

Joe Vertrees

Executive Summary

Our project was an attempt to model Intel's water consumption, and the effect that this consumption has on the neighboring Village of Corrales. Our team created, using Net Logo, a model to show the effects of Intel's well upon the Corrales aquifer, and to derive possible theoretical solutions to this ongoing dilemma. Our research proved to be beneficial in so far as it taught us what we needed to know to implement our model. In the future we plan on improving the data output and the all-around data entry in the code.

Introduction

Water is a precious resource that carries many uses other than just human consumption, such as: showering, landscaping and even as the cooling process during the creation of computer chips. Well water depletion is an ongoing problem in the residential Corrales area. This is due to the presence of Intel and their industrial sized well. This well pumps 7.78 billion gallons of water in the time span of a year. Intel uses a large sum of water that seems to be consuming more than the surrounding residential area. Our research, and our model, show that this water consumption has been having a deleterious effect upon neighboring, smaller wells.

Description/Methodology

In order to collect background data we needed to know where to look and what to find. To do this we sought help from our mentor Mr. Joe Vertrees, he then informed us of the USGS, the government agency which monitors water levels and use throughout the nation, and he also provided us with necessary “starting” information about the geology of the Rio Grande rift zone, and water movement within it. The methods we used were mainly on-line research and putting said research into play within our model.

Model

Our model is fairly complex. It uses many sliders and variables to actually show the effects of Intel’s over pumping of water on the Corrales area. This model can also be used to show the effect that any major well has on the aquifer, and dependent local human population.

The sliders in our model are used to easily change the data output so the model can adapt to any geographical area. The main slider “Porosity” changes the porosity of the ground, allowing the model to be used to show differing recharge rates for aquifers. The second most important slider “Ground-Water” allows the person controlling the model to change amount of starting ground water in their region. The main well is controlled my two sliders Intel-Well-Efficiency” and “Intel-Well-depth”, one effects the water consumption and the other effects the well depth. This allows the user to control the effects the main well has on the neighboring wells. There are also sliders for the residential wells to change their water consumption and depth. This allows the model to be applied to any area of land or region of the planet. Also to simulate the different weather climates we have a slider “Add-Water” that can increase or decrease the amount of rainfall.

The two plots in our project show the global soil moisture in the form of a graph and a monitor. Both plots display the results of a function which calculates the average water content of the aquifer. Having both the graph and monitor allows the user to have two sources of data and more accurately apply the data to their research. Also at the start of the test, the monitors will show how much water is in the region before any wells are drilled.

The buttons in our model are used to set up the world with all the variables in place from the sliders. First we have the “Setup” button that sets up the world, this includes the slider variables from the wells to the physics of the world. The “residential” button sets up all the local wells with all variables already in place. The third button “Intel” sets up the main well with all its variables already set by the sliders. We put this button third because the residents are usually in the area before the main well is drilled. Lastly the “go” puts this world in motion, its turns the physics and the rain on. This allows the rain to fall and the ground water to move or be used.

Results

Using Netlogo our results show that Intel uses a large amount of water. Most people generally use the ticks as a “counter” we used actual time. With our porosity at exactly .45, our ground water at 800, and the Intel-well-efficiency at 100, running for 10 seconds at normal speed, the model shows that the Intel well does take a large sum of water from the residential area. The soil moisture stays pretty much the same though slowly decreasing as time goes on.

Applications Used

For our model we only used Net Logo, with it we were able to apply real world effects that can easily show what is going on and possible solutions to outcomes presented within our model.

Most Significant Achievement

One of our most significant achievements for our project would have to be how our model's user interface turned out looking. The amount of time we have put into it really shows. We are extremely proud of our model and hopefully we will continue improving it from what we have done.

Acknowledgements/Thanks

Mr. Joseph Vertrees and Mr. Richard Foust have been a huge help with our project. Without these two people our project would not be where it is today. Joe helped us a lot with contributing and helping us debug our model. It seems as though it is not much, but it really is considering our average inexperience with Net Logo programming. Without Joe we most likely would not have as much as we do today, Richard helped just as much as Joe. Mr. Richard Foust is a great leader, leading us to work harder and give us more insight into what our project could use or did not need. He has been a constant source of motivation to this group. We are very thankful for these two people giving their time to help us complete this project. However, it was not just they who deserve thanks, it is everyone who helped us get this far.

Program Code

```
globals [
  sky-top      ;; y coordinate of top row of sky
  earth-top    ;; y coordinate of top row of earth
  soil-moisture  ; overall soil moisture
  water-added  ;total water added
  water-depletion
]


breed [drops drop]     ; packets of Irrigation (water)
breed [groundwaters groundwater]   ; packets of groundwater (groundwater)
breed [H2Os H2O]     ; packets of Water Molecules (humidity)
breed [water waters]


to Setup
  ;; (for this model to work with NetLogo's new plotting features,
```

```
;; __clear-all-and-reset-ticks should be replaced with clear-all at
;; the beginning of your setup procedure and reset-ticks at the end
;; of the procedure.)
;; (for this model to work with NetLogo's new plotting features,
;; __clear-all-and-reset-ticks should be replaced with clear-all at
;; the beginning of your setup procedure and reset-ticks at the end
;; of the procedure.)
;; (for this model to work with NetLogo's new plotting features,
;; __clear-all-and-reset-ticks should be replaced with clear-all at
;; the beginning of your setup procedure and reset-ticks at the end
;; of the procedure.)
__clear-all-and-reset-ticks

create-water Ground-Water
[
  set color blue
  set heading 180
```

```
    set size .5

    set shape "circle"

    waterplacment



  ]



  set-default-shape drops "drop"

  set-default-shape groundwaters "dot"

  set-default-shape H2Os "square"

  setup-world

  set water-added 1

  set soil-moisture 12

  set water-depletion 1

  plot soil-moisture

  plot water-depletion


end
```

```
to setup-world
  set sky-top max-pycor - 1
  set earth-top 1
  ask patches [  ;; set colors for the different sections of the world
    if pycor > sky-top [  ;; space
      set pcolor scale-color white pycor 22 15
    ]
    if pycor <= sky-top and pycor > earth-top [ ;; sky
      set pcolor scale-color blue pycor -20 20
    ]
    if pycor < earth-top
      [ set pcolor brown ] ;; earth
    if pycor = earth-top ;; earth surface
      [ update-Porosity ]  ;surface changes color according to Porosity value
  ]
end
```

```
to residential

  r-well-1

  r-well-2

  r-well-3

  r-well-4

  r-well-5

  r-well-6

end

to r-well-1

ask patches with [pxcor = 15  and pycor <= 3 and pycor >= Residental-well-1]



  [set pcolor white]



    ask patches with [pxcor = 15 and pycor <= 2.5  and pycor >= Residental-well-1 + 2]
```

```
    [set pcolor grey]

end

to r-well-2

ask patches with [pxcor = 10  and pycor <= 3 and pycor >= Residental-well-2]



  [set pcolor 16]



    ask patches with [pxcor = 10 and pycor <= 2.5  and pycor >= Residental-well-2 + 2]



    [set pcolor grey]

end

to r-well-3

ask patches with [pxcor = 5  and pycor <= 3 and pycor >= Residental-well-3]

  [set pcolor red]



    ask patches with [pxcor = 5  and pycor <= 2.5  and pycor >= Residental-well-3 + 2]
```

```
    [set pcolor grey]
end
to r-well-4
ask patches with [pxcor = -5  and pycor <= 3 and pycor >= Residental-well-4]

  [set pcolor 27]

    ask patches with [pxcor = -5 and pycor <= 2.5  and pycor >= Residental-well-4 + 2]

     [set pcolor grey]
end
to r-well-5
ask patches with [pxcor = -10  and pycor <= 3 and pycor >= Residental-well-5]

  [set pcolor 85]

    ask patches with [pxcor = -10 and pycor <= 2.5  and pycor >= Residental-well-5 + 2]
```

```
    [set pcolor grey]

end

to r-well-6

ask patches with [pxcor = -15  and pycor <= 3 and pycor >= Residental-well-6]


 [set pcolor 25]


    ask patches with [pxcor = -15 and pycor <= 2.5  and pycor >= Residental-well-6 + 2]


    [set pcolor grey]


end


to Intel
```

```
  ask patches with [pxcor = 0 and pycor <= 5 and pycor >= Intel-well-depth ]

    [ set pcolor yellow ]

  ask patches with [pxcor = 0 and pycor <= 4 and pycor >= Intel-well-depth + 2]

   [set pcolor grey]

end

to go
  run-Irrigate   ;; step Irrigate  ;Irrigate = amount of water added
  ;; if the Add-Water slider has moved update the color of the "earth surface" patches
  ask patches with [pycor = earth-top]
    [ update-Porosity ]
  run-groundwater  ;; step groundwater
```

```
  run-H2O   ;; moves H2O molecules

  set-current-plot "Global Soil-Moisture"
  plot soil-moisture

  ask turtles
  [ fill
    use
    suck
    runoff
  ]
end

to update-Porosity ;; patch procedure  ;change color of the "earth surface"
  set pcolor scale-color green Porosity 1 -.5
end
```

```
to run-Irrigate
  ask drops [
    if not can-move? 0.3 [ die ]  ;; kill them off at the edge
    fd 0.3                        ;; otherwise keep moving
  ]
  create-Irrigate  ; start new irrigation drops from top
  encounter-earth   ; check for "reflection" (runoff) off earth and absorption
end


to create-Irrigate
  ;drops=irrigation water turtles
  ;; don't necessarily create a drop each tick
  ; as Add-Water gets higher make more
  if 10 * Add-Water > random 50 [
    create-drops 1 [
```

```
      set heading 160
      set color blue - 1
      ;; drops only come from a small area
      ;; near the top of the world
      setxy (random 50) + min-pxcor max-pycor
      set water-added water-added + 1  ;gives total water added for the run of the model
    ]
  ]
end

to encounter-earth
  ask drops with [ycor <= earth-top] [
    ; depending on the Porosity either the earth absorbs the groundwater or "reflects" it (runoff)
```

```
    ifelse 100 * Porosity > random 100   ; this controls the prob that a drop will convert to
groundwater (if 100*Porosity>a random #1-100 drop will REFLECT)
      [ set heading 180 - heading
        die] ;; reflect
      [ rt random 45 - random 45 ;; absorb into the earth
        set color green
        set breed water ]
  ]
end



to run-groundwater    ;soil-moisture is a function of: water added, efficiency, temperature and
Porosity
  set soil-moisture 0.99 * soil-moisture + 0.01 * (12 + 0.1 * count water)  ;groundwaters=soil
moisture turtles
  ask groundwaters
  [
    let dist 0.5 * random-float 1
```

```
    ifelse can-move? dist
      [ fd dist ]
      [ set heading 180 - heading ] ;; if we're hitting the edge of the world, turn around
  ]
end


;the following H2O code controls humidity


to add-H2O  ; randomly adds 5 H2O molecules to atmosphere
  if count H2Os < 100
  [
  let sky-height sky-top - earth-top
  create-H2Os 5 [
    set color green
    ; pick a random position in the right side of the sky area
    setxy random-float max-pxcor
          earth-top + random-float sky-height
```

```
  ] ]

end



to remove-H2O ; randomly remove 5 H2O molecules

  repeat 5 [

    if any? H2Os [

      ask one-of H2Os [ die ]

    ]

  ]

end




to run-H2O

  ask H2Os [

    rt random 51 - 25 ;; turn a bit

    let dist 0.05 + random-float 0.1

    ;; keep the H2O in the sky area
```

```
    if [not shade-of? blue pcolor] of patch-ahead dist
      [ set heading 180 - heading ]
    fd dist ;; move forward a bit
  ]
end


to waterplacment
  set xcor random-xcor
  set ycor random -20
end



to fill
  if breed = water
  [
    if [ pcolor ] of patch-ahead 1 = brown and not any? turtles-on patch-ahead 1 ; rain falls thru
black, but not mud or rain
    [ forward 1 ]
```

```
  ]
end

to suck
  if breed = water
  [
    if [pcolor] of patch-ahead 1 = red
    [forward 1]

  ]
end

to use
   fill

   if breed = water
```

```
 [if [pcolor] of patch-ahead 1 = 25

   [die]

 if [pcolor] of patch-ahead 1 = 85

   [die]

 if [pcolor] of patch-ahead 1 = 27

   [die]

 if [pcolor] of patch-ahead 1 = 16

   [die]

 if [pcolor] of patch-ahead 1 = white

   [die]

if [pcolor] of patch-ahead 1 = red

   [die]



if [pcolor] of patch-ahead 1 = yellow

 [if Intel-Well-Efficiency > random 100

[die]

]]
```

```
end



to runoff
 if breed = water
 [
  set heading 90 + random 181            ; net logo starts random with 0
  if [ pcolor ] of patch-ahead 1 = brown and not any? turtles-on patch-ahead 1
   [
    forward .001
   ]
  if [pcolor] of patch-ahead 1 = grey and not any? turtles-on patch-ahead 2  ;allows water to
move past the well pipes
   [
    forward .001
   ]
```

```
  ]
end

to pump-off  ;Turns off the pump on the deep well by changing the color of the patches

  ask patches with [ pxcor = 0 and pycor <= Intel-well-depth + 2 and pycor >= Intel-well-depth ]
    [ set pcolor black ]
end

to pump-on
  ask patches with [ pxcor = 0 and pycor <= Intel-well-depth + 2 and pycor >= Intel-well-depth]
  [ set pcolor yellow ]
end
```