ECOSYSTEM MODELS

# Fire Model

Terri Donovan

recorded: January, 2010

Today we're going to build a spatial model on a spreadsheet. A spreadsheet is a good way to think about spatial models because by definition every single cell on the spreadsheet has an address. Identified by it column and it's row number. Because of that we know not only what the address is of each cell, but we also know exactly what it's position is relative to each and every other cell on the spreadsheet.

Today we'll do a simple fire model. And we're going to start off my creating a grid of cells, which is going to represent our landscape. And we'll stretch this grid from column B through column P. We'll start at row 6 and drop it down to row 21. And we'll go ahead and give those a border.

Each of these pixels or cells is going to have a fuel level. And we'll represent that with a random number - which is R A N D, open parentheses, closed parentheses. Drag that formula across the spreadsheet and down. And now we have a landscape that's full of pixels. Each of these pixels has a fuel level that's indicated by the value of the random number. A cell close to 0 would have a very low fuel level. And a cell close to 1 would have a very high fuel level.

Let's go ahead and enter the words Burn Threshold in cell A1. In cell C1 we'll let that be an input. Here is where we will enter what our burn threshold is. This means, for example, that any of these cells in this grid that have a fuel level greater than .6 are capable of burning. There is enough fuel there to burn.

To make this easy to see visually, we'll go ahead and select the grid and head up to Conditional Formatting. Then choose Highlight Cells Rules. Then highlight those cells who's value is greater than the value in cell C1. And we'll shade those yellow with dark yellow text. Then we'll press OK.

Now while our grid is still selected, let's go ahead and choose a second Rule. And we'll choose those cells which are less than the value in cell C1. And we'll fill those green. Those are cells that do not have enough fuel to burn. And so they will be safe from burning.

And then finally, even though we don’t have this quite yet on the spreadsheet. Let’s go ahead and add one more rule. And we’ll let cells which are equal to a capital letter B, be filled with light red fill with dark red text. That B will represent a cell that is burned. If I enter a B someplace in this spreadsheet, that shows us, that particular cell has burned.

This whole grid represents our landscape at Time t. So we’ll go ahead and indicate that at Time t. Now just to keep our self organized here, let’s go ahead and enter our key. Yellow are cells that are ‘at risk’ of burning. Green cells are cells that are ‘safe’ from burning. Red cells are cells that have ‘burned’.

Let’s make this look more grid like, but selecting columns B through P, and then shrinking them up so they look like a square. Let’s actually make it big enough to display two decimal points.

Our goal here is this model is, if this is what the landscape looks like at time-step t, then what we want to know is what will this exact same landscape look like in the next time-step. And our time-step can be seconds, hours, minutes, days. It all depends on what you as the modeler choose as the spatial extent of this particular grid, and also the phenomenon of interest.

To make this process a bit quick, let’s just go ahead and select all of our cells. And we’ll head up to the Home tab and copy those. And then select cell R5. And we’ll just paste those straight in. And again we’ll shrink these columns up so that they are more grid-like in appearance. In this way we can see both of our grids at the same time.

This grid over here we’re going to let represent our grid at the next time-step. So we’ll enter Time t + 1. Here is what the grid looks like at time-step t. And here is what it will look like in the next time-step.

The goal here is to enter an equation that will calculate those for us. We’ll go ahead and clear those numbers out. Just so that you can see my full screen. I’ll go ahead and move these over. Our goal here is to think about, an equation that will be entered in the upper left corner. And the equation we have in this cell R6, is going tell us what the fate of the cell B6 upper left corner, was as we move from time-step t to time-step t + 1. In order for this to burn in the next time-step what has to happen?

Well two things. First the fuel load must be greater than the burn threshold. And then second, as Tony had indicated, that one of the neighboring cells needs to be burned as well. Our equation is going to have to handle the fact that we need to access what is the fuel load and compare to the burn threshold. It's also going to have to look and see what's happened in the neighboring cells, which is what makes it a spatial model to begin with. And then we're going to have the spreadsheet return a capital letter B, if it's burned. And we'll have it return the exact same fuel level if it's not burned.

Takes some time and see if you can figure out how to write up formula for cell R6 and then come back and we'll work on it together.

PAUSE the video: Work on a formula for cell R6.

There is a few ways to handle this equation. But we're going to go ahead and start off by entering an IF function. We'll find IF. Remember that the IF function has three arguments. The first is the logical test. This logical test must return true or false. As we start in this cell, we want to know if cell B6 actually burns. So our logical test is going to tell us the conditions which will result in a burn. And if it's burned, we'll return a capital B. And if it's false, we'll just return whatever is in cell B6. Now we have to think about what the logical test is.

Two things must happen. First, the fuel level must be greater than the burn threshold. Second, at least one of the neighboring cells also has burned. We're going to embed an AND function in this argument here. And we'll do that by heading over to the upper left corner, finding the AND function. And our first logistic test is - is cell B6 greater than the burn threshold in cell C1. Let's go ahead and anchor that C1. That needs to be true in order for it to burn. And we can see, right now, cell B6 does not have the threshold we need. But if it did the other condition is that, one of the surrounding cells must also be burned.

We'll go ahead and enter an OR equation in this argument here. Let's head on up to our spreadsheet. Find the OR function. What we need to do is just walk our way around the cell of interest, and see if any of those neighbors had been burned in the previous time-step. You can see up on the formula bar that our OR function has begun. First, let's start up in the upper left hand corner. What we want to know, is cell A5 equal to a capital B? Nope, it's not. Let's move down to the next neighbor. Is cell B5 equal to a capital B? And you can see that, as we go through this OR equation the spreadsheet is entering them up on the formula bar. And it's separating each one of these arguments in the OR function with a comma.  Just to keep things a little bit neater here.  Let's go ahead and enter these straight up in the formula bar.

Now we are on logical test 3.  We need to evaluate, is C5 equal to a B?
Now we can head down.  How about C6?
Now we are on logical test 5.  C7?
Now we are on logical test 6.  B7?
Now we are on logical test 7.  And you can tell that because it's bolded. The place you are in the function, which argument you are on, is actually highlighted.

We got A5, B5, C5, C6, C7, B7.  Now we head over to A7.  And we close it up with A6.  And that basically takes care of our OR function.  This is our equation.  I'm going to go ahead and press OK.  And then we'll head right back up and click on the formula bar, and you can see which pieces of information belong to the equation itself.

So let's just walk through this one more time.  If all of this is your logical test. OK?  So two things must be true for the AND function to ring true.  First, a cell must have a burn level greater than the value in cell C1.  Second, at least one of the neighbors, could be any of them, will also be burned.  If both of those conditions are true, we're going to tell it that the cell has burned.  If they are not true, then we'll just return the value that is in cell B6, which is the fuel level.  When we're finished we press Enter.  And there is our equation.

Now we can drag this all the way across our grid and down the grid.  Now what we can see is what would have happened in the very next time-step.  Lightning strikes, we see that these cells here, have enough fuel and they are neighboring.  And in the next time-step we'll see that those are burned as well.

What's going to happen in the next time-step?  Well what you would expect is if we moved out to Time t + 2, that in the next time-step that this cell would burn and this cell would burn.   None of these other cells have the fuel load for burning.  This one does, and this one does, and they are adjacent to a burn cell.  Then in the next time-step we would expect these to burn.  And we would expect these two to burn, and so on.

So what we're going to do now is write a macro.  And the macro is going to take the time conditions from the starting time, project out one time-step into the future.  Then we're going to take the state of the system in this time step and place it directly on top.  And let this be our time-step of interest. The equations then will keep updating as we go through time. Let's see how that works.

First we'll go up to the View tab, and then choose Macros, Record Macro. And you can name this 'Burn'. And enter a shortcut key and a description. These end up being comments in your Visual Basic code. Now the spreadsheet is faithfully recording the keystrokes that we make. What we want to do is take the time-step for T + 1, copy those values. We'll head over and select cell B6, and then we're going to Paste Values into time-step t. You can see what has happened now. This is the current state, and the spreadsheet is automatically calculating what is going to happened in the next time-step. That is basically all there is to this macro. So we head back up to the View tab, we'll choose Macros, and Stop Recording.

We can run this macro by inserting a picture, or we can choose to insert a button. That is what I'm going to go ahead and use, since it's on my mini toolbar. I'm going to assign that button the macro Burn. And I'll go ahead and label this 'Burn'. I press this macro, again we'll start to see how our spreadsheet fire is spreading across the landscape.

What you can do is just focus on one of these cells. Because what you are seeing, every time you click the macro button, is you're just seeing how it moves from time-step to time-step. And looks like we have stopped right there. Now why have we stopped? We'll it's because all of the neighboring cells now do not contain enough of a burn level to catch fire. And so our fire basically goes out.

Some output statistics that we could compute, would be things like, What proportion of the landscape has actually burned? And Tony showed this graphically during the lecture.

Let's go ahead and create another macro that will reset this landscape automatically for us. We'll go up to the View tab, press Record Macros. And we'll write a new macro. And this macro we'll call Reset Landscape. We can press OK. Now to reset this landscape, all we want to do is start by populating it with random numbers once again. And drag that all the way down. And now we can just press Stop Recording. Add that reset button to our spreadsheet. And label it something that makes some sense.

To begin this model we simply enter a capital B to establish our fire. Then we let this run. Once it's finished, we can record some statistics about this fire behavior. And when we would like to redo it, we can just push the reset button and we're able to start again.

This lets you explore how changing the burn threshold will affect the dynamics of the fire.

Well now that you have your model running, let's go ahead and save the spreadsheet. Save this spreadsheet, and you can call it whatever you would like. You want to make sure that you choose the type as an Excel xlsm, which is a macro enabled workbook. You can save it wherever you would like to.

We would also like to look at the Visual Basic code that was recorded as we clicked through our macro. One thing that we can do is add a little tiny bit of code. So that as we start our model, and we started with a capital B, that the spreadsheet will automatically run through - let's just say 30 time-steps - without you having to press this button.

We're going to go into the Visual Basic code itself and add those few lines of code. So we'll head over to Macros, we're going to View Macros. And we'll find the macro called Burn and we'll edit that macro. And here is the code itself. It's a sub routine called Burn. The typing in green, which begins with a single quote, is a comment that you the modeler will certainly want to get into the habit of entering. These are notes to yourself and the computer does not read those as part of the routine.

We started off by selecting cells R6 through AF21. That represented the landscape and time-step Time t + 1. We copied those values, then we headed over to cell B6 and selected it. Then pasted in the grid, into that time-step t grid, and the sub routine ended.

To let this run for say, 30 or so time-steps, we can do that easily by invoking a counter. And we'll just type in the code 'for counter equals 1 to 30'. When the code hits this, it will invoke a counter, set it to 1, it will run through this code. And when it's done it will find the word 'Next'. And that changes the counter from the number 1 to the number 2. And it will loop through this code again until it hits 30. At which point it will pop out of the sub routine, and you'll be done.

Let's see how this works. We'll reset it. We'll start our burn. If we click Burn, we should run through 30 time-steps in short sequence. And in this particular case, I ran out of places to burn because nothing else had a high level of fuel, the fire pooped out.

Let's reset it and try with a burn threshold [0.4] of a little bit lower value. This time we'll start our burn and we'll let it run for 30 time-steps. And that is basically it.

If you don’t want to have the counter in your code, simply go back up to your  Macros, find the code, and remove the counter.  So you can click this one step at a time to see the results. Actually, that might be preferable, because then you can study and see by comparing time-step Time t to Time t + 1 exactly why the patterns change.  Because you can see those equations directly in the spreadsheet.

That’s about it for this one.

< 00:22:09  END >