Growing trees model

Table of content:

1. Mathematical notation

To be specific with transition between two states of something (using probability) we use matrixes and vectors. Vectors contains probability state and the matrixes are state-transition matrixes. How it works?

$\vec{p}(n+1) = M \cdot \vec{p}(n)$

Where: $\vec{p}(n)$ - vector of probability within n-step, M - state-transition matrix.

1.1. Conditions for vector

The vectors are normalized. That means:

$|\vec{p}| = 1$

But the norm is calculate in different way than in Euclidean space:

$$\vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_k \end{pmatrix}$$

$$|\vec{p}| = \sum_{i=1}^{k} |p_k| = 1$$

Where $p_1, p_2, \ldots, p_k$ are probabilities of being in state *1*, *2*, … or *k*.

And because the numbers $p_1, p_2, \ldots, p_k$ are positive valued between 0 and 1 (including them), it is quite simpler condition:

$$|\vec{p}| = \sum_{i=1}^{k} p_k = 1$$

Examples

a) We are working in a company and the boss told us to do something. We have two states: 1- undone, 2- done. But after some time we didn't finished the work, so our probability state is:

   $$\vec{p}(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

   And when the boss will come, he sees undone work.
   But after one hour we will finish the work. Then our state will be:

   $$\vec{p}(t+1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

b) We are learning for an exam. This is hard and our probability to pass is about 75%=0.75. We also have two states during the (future) exam: 1- failed, 2- passed. So our probability vector of passing exam is:

   $$\vec{p}(exam) = \begin{pmatrix} 0.25 \\ 0.75 \end{pmatrix}$$

1.2. Our case using the vector

Those vectors can describe one particle (for example in quantum mechanics) but also can represent whole physical system (like the gas in a bottle). In this work we will use them to describe growing trees. Our system contains 4 state of living tree: sampling, pole, mature, old. Also it contains 1 state of dead tree. So the vector is 5-dimensional:

$$\vec{p}(t) = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{pmatrix}$$

Where p_1 - sampling, p_2 - pole, p_3 -mature, p_4 -old, p_5 -dead tree.

1.3. State-transition matrix

The matrix is used for transition between one and another step of mathematical calculations (also between two step of any simulation). It is describing all the possible transitions as a numbers (representing the possibility of transition between states).
It is important to create proper matrix. First condition is that every column of state-transition matrix has to add into 1:

$$M = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1k} \\ m_{21} & m_{22} & & m_{2k} \\ \vdots & & \ddots & \vdots \\ m_{k1} & m_{k2} & \cdots & m_{kk} \end{pmatrix}$$
$$\quad\ \underbrace{\sum}_{1} \quad \underbrace{\sum}_{1} \quad \cdots \quad \underbrace{\sum}_{1}$$

Examples

a) The coin has two sides: front and back side. When we spin the coin from front side it can change or stay on its side with the same probability of ½. The same situation is with the back side. The transition matrix between first and second state is:

$$M = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

And the vector is describing:

$$p = \begin{pmatrix} front \\ back \end{pmatrix}$$

b) We can see an apple hanging on the tree. Every day it has ½ chance to fall. The fallen apple can't hang again, so the matrix has form:

$$M = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 1 \end{pmatrix}$$

And the vector is describing:

$$p = \begin{pmatrix} hanging \\ fallen \end{pmatrix}$$

1.4. Our case using the matrix

As it was said before, we have 5 different states of the tree. Also, what we should notice, we can't go from the state 1 into 3, apart from the state 2 (there are more conditions similar to this one).
So the state-transition matrix has form:

$$M = \begin{pmatrix} p_{11} & 0 & 0 & 0 & 0 \\ p_{21} & p_{22} & 0 & 0 & 0 \\ 0 & p_{32} & p_{33} & 0 & 0 \\ 0 & 0 & p_{43} & p_{44} & 0 \\ p_{51} & p_{52} & p_{53} & p_{54} & 1 \end{pmatrix}$$

We should remember the condition for this matrix (sum of each column equals 1). Also we can be interested in changing states: growing or dying trees. That change the matrix a bit:

$$M = \begin{pmatrix} 1-p_{grow,1}-p_{death,1} & 0 & 0 & 0 & 0 \\ p_{grow,1} & 1-p_{grow,2}-p_{death,2} & 0 & 0 & 0 \\ 0 & p_{grow,2} & 1-p_{grow,3}-p_{death,3} & 0 & 0 \\ 0 & 0 & p_{grow,3} & 1-p_{death,4} & 0 \\ p_{death,1} & p_{death,2} & p_{death,3} & p_{death,4} & 1 \end{pmatrix}$$

Now we can see that we need only 7 number to have complete behavior of one type of tree on one type of ground. Also we can handle with even more simple model- 2 numbers:

$$M = \begin{pmatrix} 1-p_{grow}-p_{death} & 0 & 0 & 0 & 0 \\ p_{grow} & 1-p_{grow}-p_{death} & 0 & 0 & 0 \\ 0 & p_{grow} & 1-p_{grow}-p_{death} & 0 & 0 \\ 0 & 0 & p_{grow} & 1-p_{death} & 0 \\ p_{death,} & p_{death,} & p_{death,} & p_{death,} & 1 \end{pmatrix}$$

But it depends only on what we want to simulate. More complex model has 7 numbers, and it will not be so hard to implement. At last changing the model (between simple and complex) won't be hard ☺.

2. Advanced calculation of math model

If we want to describe the state after one step we use the equation, as above:

$$\vec{p}(n+1) = M \cdot \vec{p}(n)$$

Using the same equation, but with added numbers:

$$\vec{p}(n+2) = M \cdot \vec{p}(n+1)$$

We can receive:

$$\vec{p}(n+2) = M \cdot \vec{p}(n+1) = M \cdot M \cdot \vec{p}(n) = M^2 \cdot \vec{p}(n)$$

$$\vec{p}(n+2) = M^2 \cdot \vec{p}(n)$$

The general formula is:

$$\vec{p}(k) = M^k \cdot \vec{p}(0)$$

If we want to describe k-step, just multiply the matrix k-times and then multiply the vector by the result matrix. Easy? What about the number k, which is not a integer, but real number?
To get the result, we can go through 2 major ways:

- mathematically precise
- simple and approximate

The result is the same, so the approximation is finding the exact function.

I will show the simple way. To do that, I will take well known example (from physic or chemistry lessons): half-decay of isotopes.

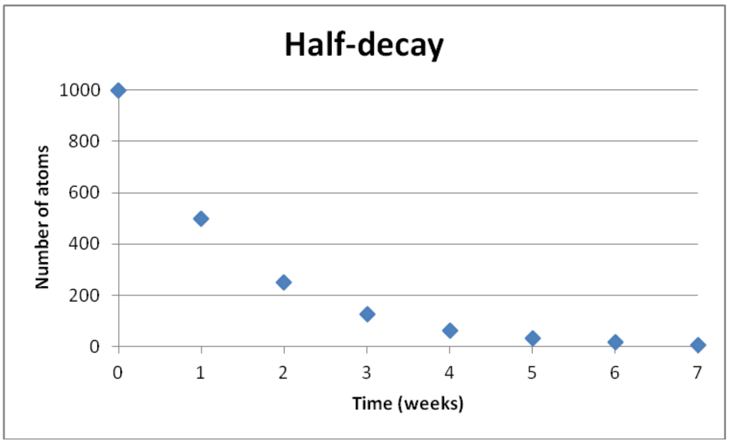
2.1. The law of half-decay

After time of Δt we will have ½ of atoms of one isotope. The time can be from 10^{-9} s to thousands of years. So we can take almost any time and find an isotope which is very similar to our example. Let take the time of 7 days. Iodine-131 (*iodium*) has the time of 8 days. Almost the same ☺.

2.2. Process of decay

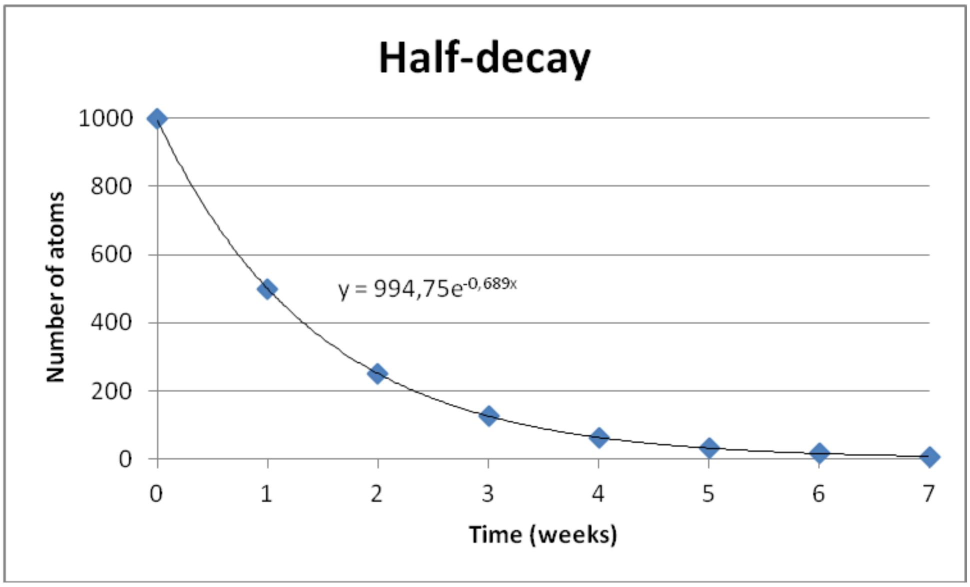
We can take for example 1000 atoms of our strange-iodine. After 7 days we will measure only 500 of atoms (the rest will transform into another type of atoms, which we can't see in our experiment). After another days we will see only 500·½=250 atoms. Then we will see 125, 63, 32, 16, 8 and so on. We can picture the result as a graph:

Week	Atoms
0	1000
1	500
2	250
3	125
4	63
5	32
6	16
7	8



We can see that the number of atoms decreases with time. At first the amount of atoms decreases very fast, then it slows down.

What we can do to describe the numbers better? We can fit a function to the graph. Which? Let take exponent: $f(x) = A \cdot \exp(\alpha \cdot x)$



We can see that in this experiment the fit is not exact (x=0, then f(x)=994.75≠1000). But really it fits. More atoms at first would change the parameters (because there will not be any approximations).

2.3. Equations for first two points

We can try to solve the equations for first two points. Two unknowns and two equations should give exact solution:

$$f(0) = 1000$$

$$f(0) = A \cdot \exp(\alpha \cdot 0)$$

$$f(1) = 500$$

$$f(1) = A \cdot \exp(\alpha \cdot 1)$$

We know that, $\exp(\alpha \cdot 0) = \exp(0) = 1$, so:

$$1000 = A \cdot 1$$

$$500 = A \cdot \exp(\alpha)$$

So the solution for A (first equation):

$$A = 1000$$

Second equation is not so simple:

$$500 = 1000 \cdot \exp(\alpha)$$

$$\frac{500}{1000} = \exp(\alpha)$$

$$\exp(\alpha) = \frac{1}{2}$$

$$\alpha = \ln\left(\frac{1}{2}\right) \approx -0.693147$$

Where ln(x) means the natural logarithm (logarithm with the base of e=2.718…)[1].

Let calculate the third point from our experiment using the numbers:

$$f(2) = 1000 \cdot \exp(-0.693... \cdot 2) = 1000 \cdot \exp(-1.386...) = 1000 \cdot 0.25 = 250$$

Great! That was our result.

2.4. Interpolation

So if we want to calculate the number of atoms in our experiment at any time, we should just put into the equation correct number of weeks and recalculate. It can be even half of week. Half of second too. Any period of time we want.
The only problem is to recalculate time: from seconds into weeks.
1 week = 7·24·60·60=604800 seconds
That means:
1 second = 1/604800 weeks ≈ 0.00000165344 weeks
Very short time.

2.5. Extrapolation

The numbers are plain. They means nothing to us. We should interpret them at first.
If we add some atoms at the beginning of the experiment, the equation will change. First equation ($f(0) = ...$) will be different. And the result will change only the *A* number. This number means how many particles we have at the beginning of the experiment. The second value: α is constant for half-decay law.
Our case of trees is a bit different. Trees can behave exactly the same like atoms in decay, but the description can be different for us. We aren't thinking about number of trees, but about the probability of "living tree" (instead of "dead tree"). That changes the number *A* into 1. Second thing is that we want to say that 90% of trees will last for 50 minutes (=3000 seconds). What makes the equation for α:

---

[1] http://en.wikipedia.org/wiki/Natural_logarithm

$$0.9 = \exp(\alpha \cdot 3000)$$
$$\ln(0.9) = 3000\alpha$$
$$\alpha = \frac{\ln(0.9)}{3000} \approx -0.0000351202$$

Any conditions can be met by the equations here.

3. First look into my model of growing trees.

At first I assumed that we can calculate every "n" seconds (or milliseconds) calculate behaviour of the tree. Second assumption is that have precise numbers describing the possibility of changing the state of the tree. How can we get the numbers? That will be next part of this work.
Also my model contains three variables describing any land:

- Temperature
- Humidity
- Fertility

We can set the numbers as degrees (Celsius, Kelvin, Fahrenheit, …), percent (of water) and percent (of nutrients) (that is the natural representation) or another representation: only real values between 0 and 1. 1 is very hot, 0- very cold. 1- full of water (see, lake), 0- no water at all (lava). The same with fertility. The second model is better for implementation and simulation, but it isn't so good for human thinking ☺. Some people will not understand the numbers.
I will use only the second model, but I will translate natural units into the model.

4. Math model of probability (our case)

To describe what happened with a tree we can use probability.

4.1. Example of calculations (growing):

We have 100 samplings of larch. We want them to grow fast in **best conditions**. Let it be in 50 seconds, 90 of them (90%) will be grown up into poles (or higher), what means that only 10 of them will stay as samplings. Also our simulation will have steps every 2 seconds.
So the number of growing up every step of simulation will be (as in point 2.5):

$$f(n) = A \cdot \exp(\alpha \cdot n)$$

Here:

$$n = \frac{t}{\Delta t} = \frac{50}{2} = 25$$
$$A = 100$$
$$f(25) = 100 - 90 = 10$$

Then:

$$10 = 100 \cdot \exp(25\alpha)$$
$$\exp(25\alpha) = \frac{10}{100}$$
$$25\alpha = \ln(0.1)$$
$$\alpha = \frac{\ln(0.1)}{25} \approx -0.0921$$

So after one step we have that amount of samplings:

$$f(1) = 100 \cdot \exp(-0.0921 \cdot 1) = 100 \cdot 0.912011 \approx 91.2$$

From the other side we can say those numbers as possibilities and number of trees:
N_0 - number of samplings at the beginning, $N(n)$ - number of samplings in n-step, p_{stay} - possibility of staying (not growing up), n - number of steps.

$N(n) = N_0 \cdot p_{stay}{}^n$

$N(1) = N_0 \cdot p_{stay}$

Comparing those equations:

$N(1) = N_0 \cdot p_{stay}$

$f(1) = 100 \cdot 0.912011$

We can say, that in our case:

$p_{stay} = 0.912011$

$N_0 = 100$

There is a condition for p_{stay} (but we assume that the tree can't die now):

$p_{stay} = 1 - p_{grow}$

$p_{grow} = 1 - p_{stay}$

$p_{grow} = 1 - 0.912011 = 0.0879892$

4.2. Conclusion

That calculations were only for one type of trees and one soil. We have assumptions: time, number (or percent) of grown trees, soil, and the result is a possibility of growing up a tree in a constant period of time.

This is very important: the p_{grow} value is constant for one type of trees and one type of soil.

And something to code of any simulation:

For every step we can generate a random real number (0 … 1) and then compare it with the p_{grow} value. Here is a sample of C function (probably C++ is very similar):

```
int growQ(double possibility){
      double randomNumber=drand48()[2];
      if(randomNumber>possibility) return 0; //False- stay at
                                              the current size
      return 1; //True- the tree should grow
      }
```

Second point here is that the same algorithm is for death of trees, but another numbers are in input.

4.3. Math model for death of trees

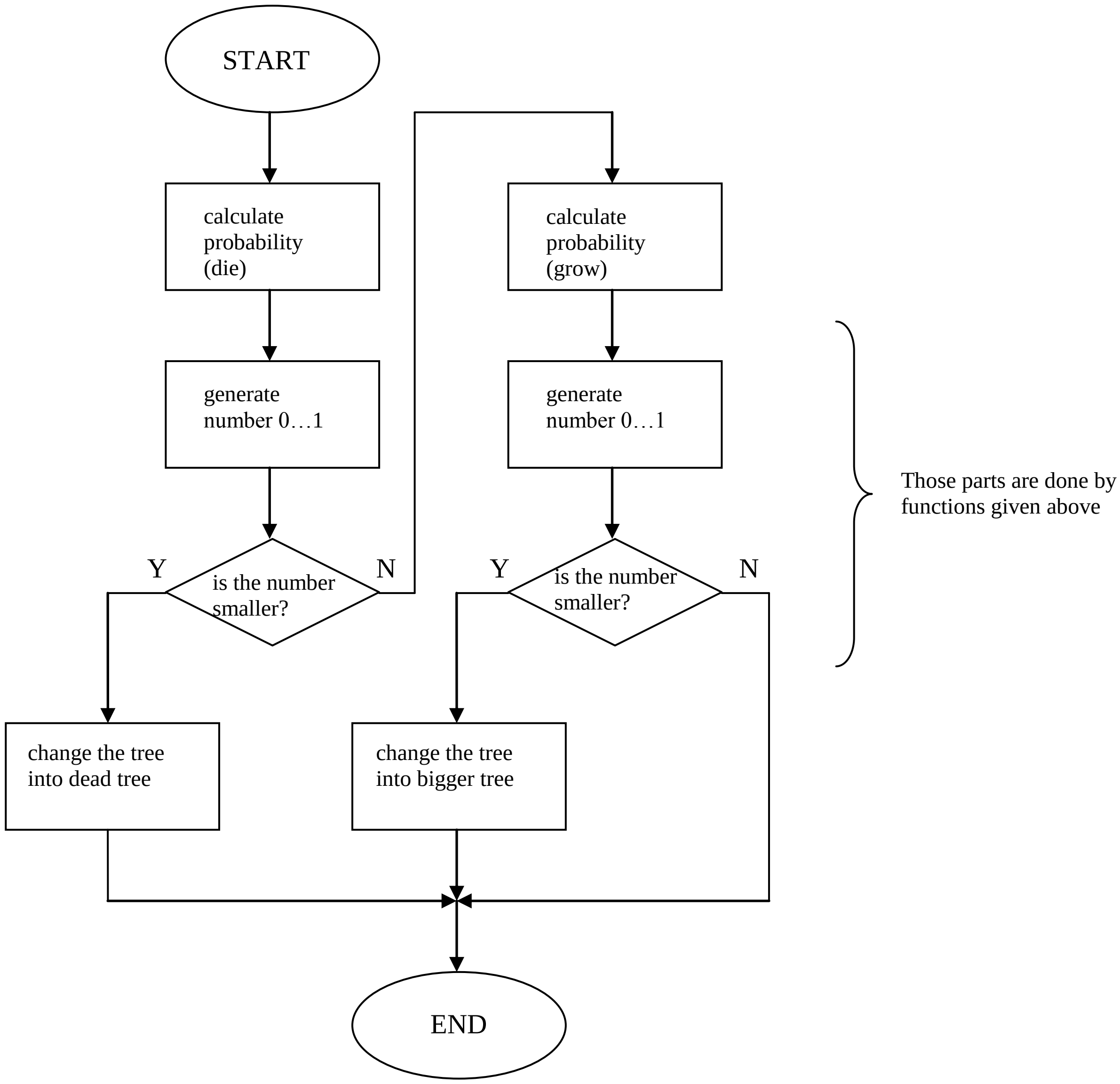
Here I should say that the model for death of trees should be exactly the same as model for growing trees. So we should calculate the probability of death (p_{death}) in the same way (on exact type of land). And then we should make similar function for death of trees:

```
int deathQ(double possibility){
      double randomNumber=drand48();
      if(randomNumber>possibility) return 0; //False- stay
                                                       alive
      return 1; //True- the tree should die
      }
```

[2] Manual for the drand48 function is here: http://pubs.opengroup.org/onlinepubs/7908799/xsh/drand48.html. We can use any other pseudo-random generator for numbers in range (0 … 1).

4.4. Compiling together functions

There are two ways to compile those functions together. First one contains generating random number two times. This diagram shows the algorithm of the process:



This way of compiling functions together gives something different to the math model described at the beginning. Here we have two state-transform matrixes: one containing only death of trees and second- only growing up. It works slightly different to first math model.

Second way is a bit more complicated, but it contains only one number generation. Also it is exact to the math model of state-transform matrixes and state vectors.
At first we calculate probability for death and grow of trees. Then we have to sum them. Why? The sum of possibility of both processes can't be greater than 1. If it is we have to normalize it to one:

$p_{grow} + p_{death} > 1$

$$p'_{grow} = \frac{p_{grow}}{p_{grow} + p_{death}}$$

$$p'_{death} = \frac{p_{death}}{p_{grow} + p_{death}}$$

Then we should generate a random number. Last thing is to conclude, what should be done:

- $0 \ldots\ p_{death}$: the tree is dying
- $p_{death}\ \cdots\ p_{death} + p_{grow}$: the tree is growing
- $p_{death} + p_{grow} \ldots$ 1: the tree stays at the state

The function containing all the things algorithm can be written as:

```
void treeStep(double possibilityDeath, double possibilityGrow){
       double pDeath=possibilityDeath;
       double pGrow=possibilityGrow;
       if(pDeath+pGrow>1){
              //Normalization of possibilities
              pDeath= possibilityDeath/(possibilityDeath+ possibilityGrow);
              pGrow= possibilityGrow /(possibilityDeath+ possibilityGrow);
              }
       double radomNumber=drand48();
       //Here is the conclusion
       if(randomNumber<pDeath){
              //THE TREE IS DYING → tree.die()?
              }
       else{
              if(randomNumber<pDeath+pGrow){
                     //THE TREE IS GROWING → tree.grow()?
                     }
              else{
                     //DO NOTHING → ?
                     }
              }
       }//END of function
```

Which process should be chosen to write in? Really don't know. Maybe there is something there in the code of Widelands? Or one of those is too complicated in any way?

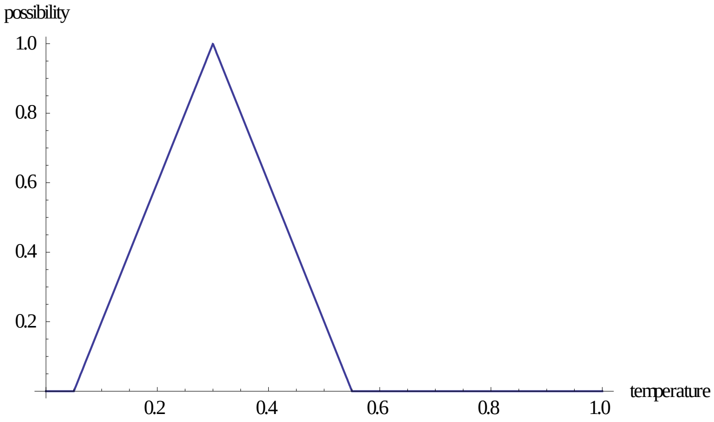
5. Four models for possibility values

I've described how to calculate possibility for one type of soil and one type of tree. But we have more trees and more soils, which can be used. How to calculate all those possibilities? I've made up 4 basic models.

Let take the *larch* again as a tree. Its best conditions for living is rather cold temperature (0.3), and not so wet ground (0.35), and middle fertility of ground (0.5). Every differences change the possibility of growing up or not dying into smaller numbers:

5.1. Symmetric triangle model (simplest model)

We assume that the change of temperature on both sides (into colder or hotter temperatures) causes the same changes of possibilities:



We describe all the possibilities for each feature of soil and at last we multiply the results. That gives us the main result. But we should also multiply the result by the probability of growth or live[3]. Then we have exact numbers for possibilities for each land.

This model needs two parameters for each feature of soil: highest point and width of the triangle. Also it needs a possibility for best parameters.

Maximum complex way contains: different best parameters for living and growing the tree and different widths for each feature of soil. Then different possibilities of growing and living. Also it can be different for each size of the tree (sampling, pole, etc.). That makes ((1+1)·3+1)·2·4=56 parameters for each tree. Lots of numbers.

Let's simplify:

Best parameters will be still different, but the width will be the same in each parameter:

(3+1+1)·2·4=40

Ok, let say, that every size of tree has the same parameters:

(3+1+1)·2=10

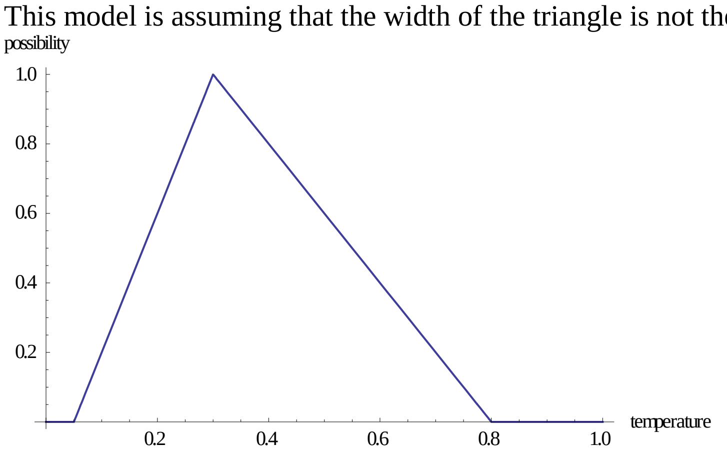
We can say that best parameters for growing are the same as best parameters for living:

3+1+2=6

This is the simplest of the simplest models here. Best conditions are for every size of trees and what we want to calculate (living or growing up). Also the width is the same. Changes only the possibility of death or survival.

5.2. Non-symmetrical triangle model

This model is assuming that the width of the triangle is not the same on both of sides:



This model brings even more parameters which describe the tree:

((1+2)·3+1)·2·4=80 parameters in worst case

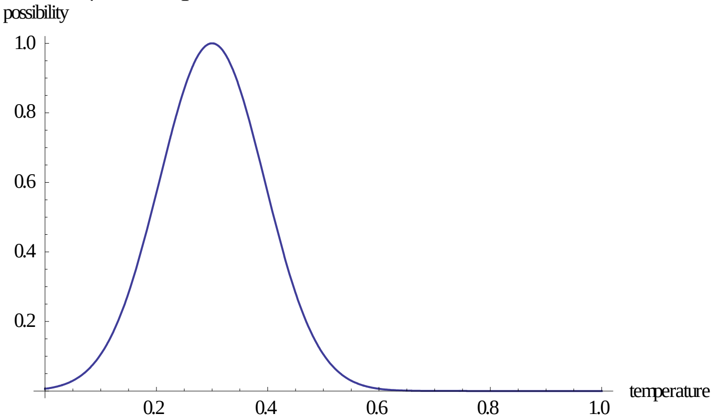
[3] Remember that we defined the death possibility by „1- live possibility".

5.3. Symmetrical gauss model

This is based on gauss function:

$$g(x) := \exp\left(-\frac{(x-\mu)^2}{\sigma^2}\right)$$

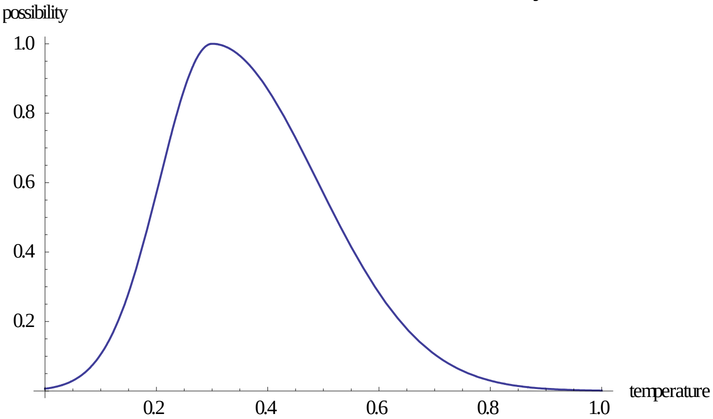
Where μ is the position of maximum on x-axis and σ is a width at half maximum of the function.



The number of parameters needed for this model is between 6 and 56 (like in the first described model).

5.4. Non-symmetrical gauss model

This contains, like in the second model, not symmetric function, based on gauss:



In my opinion this function describes the behaviour in the most complex and natural way. But to describes one tree we need up to 80 parameters. This is way to much.

5.5. Conclusion- best options

To describe one tree we need at least 6 parameters. Two of described models are based on triangle, another two- on gauss function. What is surprising, gauss- based functions are easier to implement in the code. Simple symmetric gauss (3rd model) needs no intervals in implementation (first triangle model can be realised by intervals or some comparing and *abs* function- then intervals are not needed). I would start implementation with gauss based functions.

What is worth noting, gauss function is always greater than 0. There will be always a possibility to stay alive or grow up for a tree in this model.

6. Features of soil

Every soil has its own features: temperature, humidity and fertility. When we have all the land as a one soil, it is very easy to calculate every possibility values. But when we have to mix soils, that can be hard. Best option is to notice, that the map is divided into triangles and each triangle has its vertices on the nodes. There the trees are growing. Also each triangle can be another type of soil.

So each tree has 6 parts of land in its neighbourhood.

The simplest way to mix the features of land is to make a simple arithmetic mean for each of them:

$$fertility_{mean} = \frac{1}{6}(fertility_1 + fertility_2 + fertility_3 + fertility_4 + fertility_5 + fertility_6)$$

…

This numbers should be considered in calculations.

7. Conclusion- first model

My model includes following steps:

- Take a tree (every constant Δt time)
- Find the nearest parts of land
- Calculate mean features of soil
- Calculate possibilities of surviving (→dying) and growing up
- Make a step for found possibilities.

The found values are constant during the game (and with simplest models also with growing tree), so we can keep them in a tree object or somewhere. Then most of steps will be skipped and the calculations will be faster.

How many calculations we will make for whole map?
The biggest map contains is 512x512, what means 262144 places for trees. Then we want to make the steps every 2 seconds. So we have 131072 calculations every 1 second. The calculations should be faster than 0.00763 ms, what is very difficult. Longer time mean that the Widelands will not work properly.

8. Introduction for second model

Second model is a bit more complicated. We can do calculations only once. We can ask another question: "How long the tree will live on the land?" instead of "Will the tree survive?".
The rules for calculating the basics possibilities are the same, but the last equations will change. The possibility theory answers on both questions and gives simple answers how to calculate both things.

I'm sorry, but I can't describe this idea right now. If anyone is interested on it, please contact with me.

9. Appendix- manual

I know that this was hard to create good maps for game in Widelands. My first maps weren't so good because I didn't know how the trees behave on each ground. After some testing I've given numbers in another work (Foresters and woodcutters). This should be somewhere given in a table: features of soils and best conditions for each trees. We can make a simple table like this (this is an example!):

	Soil	Feature		
		Temperature	Humidity	Fertility
Greenland	Meadow 1	0.4	0.7	0.9
	Meadow 2	0.45	0.65	0.8
	Meadow 3	0.5	0.6	0.9
	Meadow 4	0.55	0.55	0.8
	Steppe			
…	…	…	…	…

	Best conditions (living)			Best conditions (growing up)		
Tree	Temperature	Humidity	Fertility	Temperature	Humidity	Fertility
Larch	0.4	0.35	0.5	0.45	0.5	0.6
Spruce	0.25	0.35	0.5	0.4	0.4	0.6
Alder	0.6	0.6	0.4	0.55	0.5	0.5
Aspen	0.6	0.4	0.6	0.45	0.45	0.7
…	…	…	…	…	…	…

Any questions? Just ask.
Any mistakes? Please point them out!
kalkulator_empire@tlen.pl
https://wl.widelands.org/profile/einstein13/
http://student.agh.edu.pl/~rak/widelands/