FARSITE - *High Performance Simulation*
Computational Mathematics and Data Analytics 2021-2022

Lorena Pérez Gálvez 1535868

Artur Llabrés Brustenga 1528792

Xabier Oyanguren Asua 1456628

Basic Case

2. Execucions modificant els paràmetres del fitxer de configuració. Com que no us proporcionem un resultat real per aquest exemple, no heu de treure conclusions sobre com els diferents valors dels paràmetres milloren la simulació. Però si que heu d'observar els efectes per comprendre quin impacte tenen i treure conclusions sobre el temps de d'execució i forma de la propagació. A continuació us indiquem el casos mínims que heu de provar, combinant les diferents opcions entre elles:

All the experiments performed for this section can be found in Table **??** with their corresponding computation times. The resulting simulated perimeter time evolution can be seen in Figures 1 and 2.

Table 1: Results of simulations. Times are computed as the average over 6 executions. Times are in format (min:s) and Stdv. stands for sample standard deviation.

ID	timestep (min)	visible Step (min)	perimeter Resol (m)	distance Resol (m)	enable Crownfire	enableSpot FireGrowth	Time	Time Stdv.
1	5	5	5	5	True	False	0:8.08	0.0:0.7
2	45	45	5	5	True	False	0:5.61	0.0:0.5
3	5	45	5	5	True	False	0:7.18	0.0:0.9
4	45	45	150	150	True	False	0:0.05	0.0:0.01
5	5	5	150	150	True	False	0:0.19	0.0:0.02
6	45	45	150	5	True	False	0:0.19	0.0:0.02
7	45	45	5	150	True	False	0:5.46	0.0:0.4
8	45	45	5	5	False	True	0:4.89	0.0:0.6
9	45	45	5	5	False	False	0:4.79	0.0:0.4
10	45	45	5	5	True	True	0:4.85	0.0:0.6

a. Timestep

As the results of Table 1 suggest (experiment in row one and two only change the *timestep* and the *visibleStep* in consonance), increasing the *timestep* leaving the rest fixed reduces the average computational time (over 6 independent runs), which is consistent with the fact that a bigger *timestep*, given fixed time boundaries means fewer iterations of the algorithm. The *timestep* was increased by a factor nine and the execution time does not decrease even half, meaning it is not really the complicating factor, yet its effect is clear.

With respect to the graphical results in Figures 1 a and b, we see the perimeter was outputted in less iterations, meaning the perimeters in time will be fewer. Since the simulation conditions were simple, we do not really see any quality difference though.

d. visibleStep In row three relative to row one, we have only changed the visible step, multiplying it by nine but leaving the timeStep fixed. This means that data will only be outputted every nine time iterations, which graphically can readily be noted in Figure 1 c relative to a. In this case however, the rest of time iterations in between are still computed even if not outputted. Therefore, there is indeed a computational time reduction, but not as significant as the one we obtained by reducing the timeStep itself (the reduction is only presumably due to the output ammount difference).

b. distaceResolution c. perimeterResolution

By leaving a high timeStep, so we are able to graphically note the differences, we can see the effect of increasing both the perimeterResolution and the distanceResolution from 5 to 150 in row 4 (to be compared with row 2). The average execution time is reduced almost two orders of magnitude, which means these two parameters are very critical in the algorithm. Even if we use a small timestep (row 6, compared with row 1), the execution time reduction is almost two orders of magnitude.

Now if we analyze what happens when we only increase the perimeterResolution (row 7 relative to the 5 and 2), we see that the execution time is also reduced, but not as much as increasing both the perimeter and the distance resolutions, yet roughly the same. On the contrary if we only increase the distanceResolution (row 8 relative to 6, 5 and 2), there is a execution time reduction, but it is roughly noticeable. This means that giving a fixed perimeter resolution and timestep, increasing the distanceResolution does not make the computation much easier. On the contrary increasing the perimeter resolution does affect deeply the computational effort.

Indeed, if we look at the graphical results of increasing only the perimeterResolution leaving the rest fixed (Figure 2 a relative to Figure 1 b), we see the quality of the simulation is decreased even visually (the perimeters outputted are polygonal angular shapes, instead of elliptical curves). Instead, by only changing the distanceResolution (Figure 2 b vs 1 b), the simulation quality is visibly damaged in this case. Much the same way, by increasing both together (Figures 1 d,e vs 1 a,b), the perimeter gets angular polygonal shapes, even for low timeSteps, indicating the effect is mostly due to the perimeterResolution.

This is due to the fact that the perimeterResolution as explained in the lecture is inversely proportional to the number of considered wave propagating nodes in each wavefornt. Thus the increased computation time. As the perimeterResolution increases, the distance btween adjacent propagating nodes in the wavefront is allowed to be bigger before additional nodes are generated in between. This clearly is the cause of the angular shapes of the perimeter. The vertices of the polygon are the nodes that propagate the wavefront. Since there is no additional node appearing between them, the perimeter in between is a straight line. The same happens in fact for the rest of simulation, but since the nodes are so closed to each other, the aliasing is not noticeable.

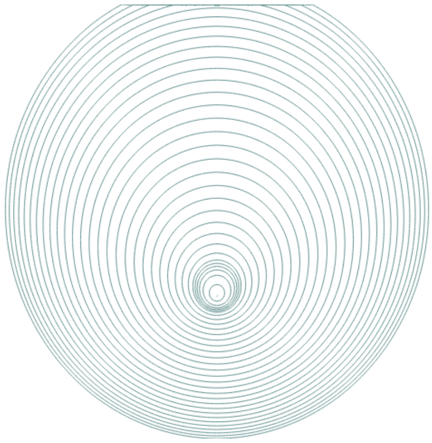
When it comes to the distanceResolution, in reality, we should graphically be able to notice some differences, yet in this simple setting we hardly do. In theory an increase of the distanceResolution allows the wavefront to advance further away form the previous time perimeter without need of computing extra intermediate steps (capped by the distanceResolution, when the step is to be greater). Not only this makes the simulation's quality improve, since very big steps of the perimeter propagation will be finner, but it makes the compiutational time bigger, which is mainly what we observe. Yet, the increment in time (or decrement due to to its decreasing) is not that significant at least in this case, since apparently the distanceResolution is not exceeded in any significant cases.

e. enableCrownfire f. enableSpotFireGrowth

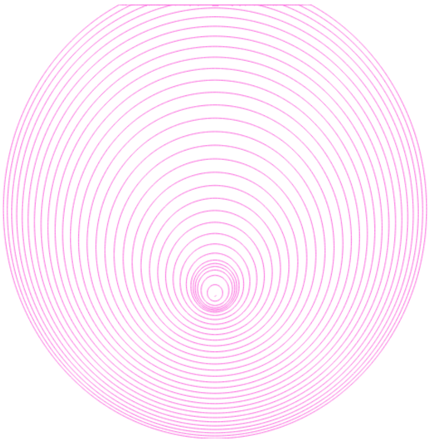
Enabling or not the crown fire alone (leaving spot fire to False), as is done in simulations 2 and 9, the computational time is reduced significantly by disabling it (a whole fifth of the time within one standard deviation). This makes sense since enabling the simulation to take care for crown fires includes an additional item in the simulation to take care of in the calculations and likely the output and input data attention.

Enabling or not the spot fire growth (simulations 8 and 9), only very slightly increases the time in the case of enabling it (about two percent within a single standard deviation). Such an increase, even if slight would make sense since activating spot fire growth sporadically makes the algorithm to make a random choice run, which would propagate the fire to an isolated island in a successful case. Graphically we see no difference (Figures 2 c and d), which suggests no significant extra fire island is generated, in consonance with the almost imperceptible computational time difference.

(a) timestep=5 visibleStep=5 perimeterResolution=5 distanceResolution=5 enableCrownfire=True enableSpotFireGrowth=False

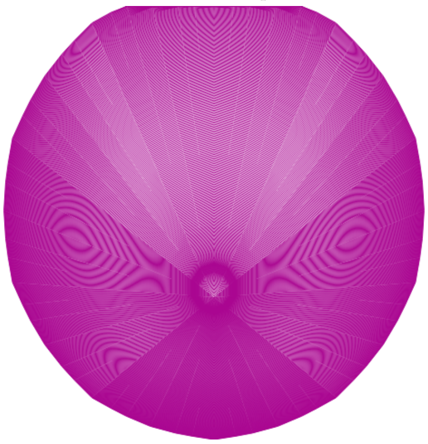


(b) timestep=45 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=True enableSpotFireGrowth=False



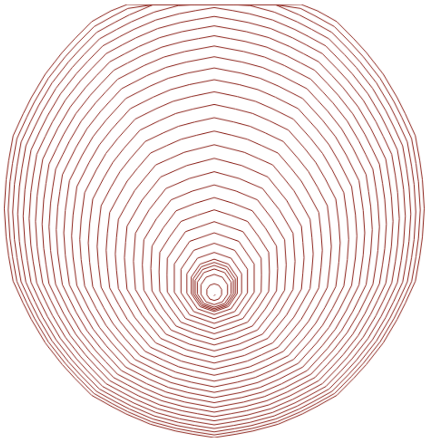
(c) timestep=5 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=True enableSpotFireGrowth=False

(d) timestep=45 visibleStep=45 perimeterResolution=150 distanceResolution=150 enableCrownfire=True enableSpotFireGrowth=False

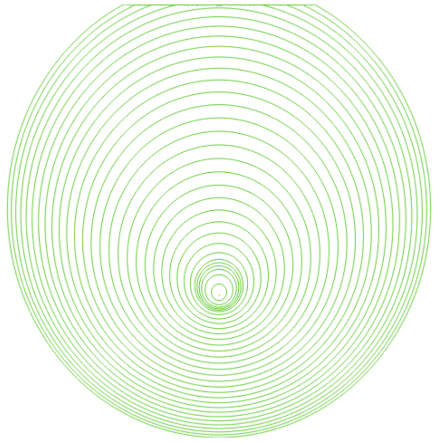


(e) timestep=5 visibleStep=5 perimeterResolution=150 distanceResolution=150 enableCrownfire=True enableSpotFireGrowth=False

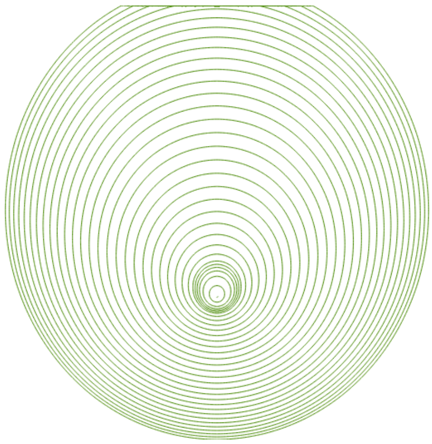
Figure 1: Perimeter of the fire at each outputed time step. Parameters for each simulation are detailed in the subfigure captions.



(a) timestep=45 visibleStep=45 perimeterResolution=150 distanceResolution=5 enableCrownfire=True enableSpotFireGrowth=False

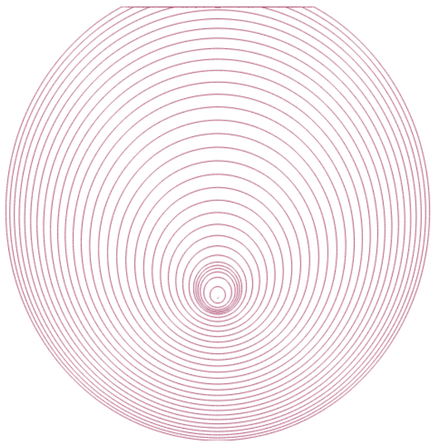


(b) timestep=45 visibleStep=45 perimeterResolution=5 distanceResolution=150 enableCrownfire=True enableSpotFireGrowth=False



(c) timestep=45 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=True

(d) timestep=45 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=False



(e) timestep=45 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=True enableSpotFireGrowth=True

Figure 2: Perimeter of the fire at each outputed time step. Parameters for each simulation are detailed in the subfigure captions.

La Jonquera Case

2. Realitzar execucions modificant el paràmetre de distanceResolution amb 5m i 100m, tenint fixe el perimeterResolution a 5m. Quina diferència esperàveu i quina observeu? Es pot identificar amb aquestes execucions quina és la tècnica de FARSITE per aumentar artificialment la resolució del perímetre. (Useu el QGIS).

As it is explained in the first exercise and the study case, we expect that an increment in the distance resolution will cause each time step to be able to extend the perimeter to a further away orthogonal distance without additional sub-steps in between. Essentially, we expect that when we relax the distanceResolution (increase it), the simulation gets worse in quality, since less sub-timeStep steps will be required in each time iteration (between perimeter reconstructions).

After we executed both cases, as can be seen in Figures 3 b,c that we find precisely what we expected. Indeed, it turns out that by zooming enough, we can observe a certain polygonal angularity in the 100 distanceResolution case, which is due to the fact that in each of the sub-tiemStep steps taken in distanceResolution 5 but not in distanceResolution 100, the small perimeterResolution will generate new points[1], which somewhat artificially makes the resolution of the effective perimeter itself in time increase, without changing the perimeterReoslution parameter. This dynamical timeStep change is the technique suggested by the question.

3. Realitzar execucions modificant el paràmetre de perimeterResolution amb 5m i 100m, tenint fixe el distanceResolution a 5m. Quina diferència esperàveu i quina observeu? (useu el QGIS).

In this other case, (see Figures 3 b,d) we clearly see that increasing the perimeterResolution, as happens in the first and study cases, produces perimeters with each time a more angular/polygonal shape, due to the fact that a larger linear distance between adjacent perimeter active points will be required for the generation of additional wave-front propagator points. So the observed results correspond precisely to what we expected, again.

4. Usant diferents distanceResolution i perimeterResolution (5m i 100m) provar diferents valors de timestep (5m, 10m, 15m, 30m, 50m, 100m). Explicar les diferències entre els valors tenint en compte com funciona FARSITE i què volen dir els paràmetres.

For a small spatial resolution in magnitude (5m in distanceResolution and perimeterResolution), increasing the timeStep makes no significant difference in the resulting perimeters as can be seen in Figures 3 b, e (appart from the fact that we will have an output less frequently). This is because in reality, no matter how big we put the timeStep (within some reasonable bounds), the distanceResolution will control how many sub-timeSteps are dynamically taken between one time iteration and the next one, and the perimeter rediscretization will be performed in each sub-timeStep and not only when the whole step is taken (as explained in the footnote). Thus by having the distanceResolution fixed at a relatively small value of 5m, in reality it dominates the temporal behaviour of the evolution.

Increasing the distance resolutions together with the timeStep on the other hand (Figures 3 f and 4 b), we find the same conclusions as in the previous two quesitons. Just that perhaps now the quality

[1] As can be read in pages 20, 21 of the Farsite documentation [1]): "New vertices must be inserted at the mid-span of a perimeter segment if the distance between vertices is greater than the perimeter resolution. This process has been referred to as rediscretizing. [...] At the end of each sub-timestep each fire polygon is rediscretized to check the perimeter resolution".

of the simulations could get worse, even if the effective number of steps taken (counting the subSteps) should be equivalent. In fact, they visually seem to be equivalent as expected.

5. Realitzar dues execucions modificant el valor de enableCrownFire amb timestep de 10m, distanceResolution de 5m i perimeterResolution de 5m.

We expect that the simulation will get more realistic with the activation of enableCrownfire, since the vegetation of the region will be more accurately taken into account. Yet, since there is no really clear correspondence between the simulations and the real case as we discuss further on, we cannot make this point clear. In fact we can only see some minor differences in Figures 3 b and 3 c (not activating or doing so), when it comes to the sharp inwards propagation edges. Enabling the crown fire seems to avoid them and the fire locally propagates in a planer or more extensive way. Yet the net effect is hardly percievable.

6. Realitzeu 10 execucions amb els mateixos paràmetres de FARSITE amb enableSpotFireGrowth, heu obtingut el mateix resultat a totes les execucions. Expliqueu el motiu.

We have obtained roughly the same results in all execution, as can be seen for some examples in Figures 4 s,e,f. However, the results should be at least slightly different due to the way the spot fires are calculated, which are done in three main steps:

- Lofting: Which consists on determining the plume characteristics and the lofting embers. This is done through a set of equations (34-38 in the Farsite documentation Ref [1]) that are deterministic, there would be the possibility to sample some of the parameters of these equations from a distribution, but according to the documentation parameters like particle diameter are set to a constant value. Therefore no randomness that could change the output is introduced here.
- Flight: Which consists on determining the trajectory of the embers. Similarly to Lofting this is computed using deterministic equations (39-43 in the Farsite documentation [1]).
- Ignition: Which consists on determining if the landed ember will ignite a new fire. Ignition will only be possible if the ember lands on an area with combustible substrate that has not been burned yet. Even in the case of landing in a combustible area ignition is not guaranteed, this is determined by a parameter called ignition frequency. Here is where the randomness that should change the results on each simulation is introduced.

The thing might be that the initialization seed for the random generator is set to be the same when run in different terminal sessions.

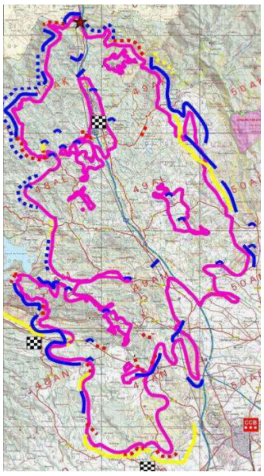
7. Comparar les simulacions (provant diversos paràmetres) amb la realitat (el perímetre reals es troba a la carpeta Real/). No fa falta fer tots els paràmetres contra tots, només provar unes 6 combinacions

We simulate the experiments listed in Table 2, which yield the perimeter time evolutions in Figures 3 and 4, where we can also find the real fire propagation as recorded by the competent authorities. We find that after a wide search of the parameters, no simulation really yielded a perimeter evolution that ends up matching much with reality.

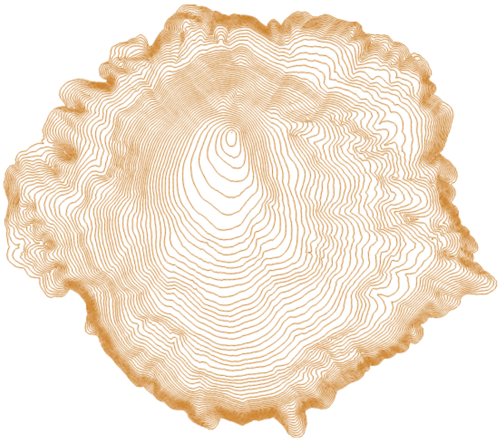
Yet, we find that the most accurate prediction is precisely given by the combinations of parameters that have the perimeter and distance resolutions and the timeStep to the minimum, and the crownFire and SpotFireGrowth parameters activated Figure 3 b.

Table 2: Results of simulations. Times are computed as the average over 1 executions.

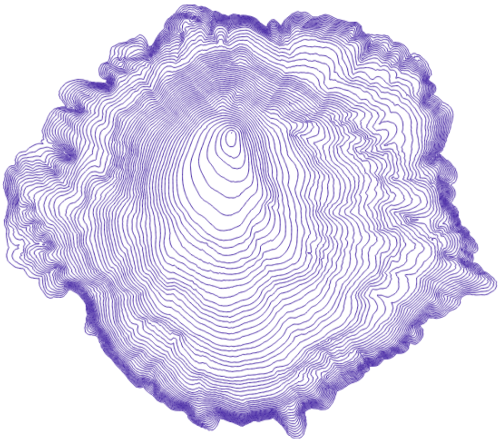
timestep (min)	**visible Step (min)**	**perimeter Resol (m)**	**distance Resol (m)**	**enable Crownfire**	**enableSpot FireGrowth**
5	45	5	5	False	False
5	45	5	100	False	False
5	45	100	5	False	False
100	45	5	5	False	False
100	45	5	100	False	False
100	45	100	5	False	False
10	45	5	5	True	False
10	45	5	5	False	True
5	45	5	5	True	True



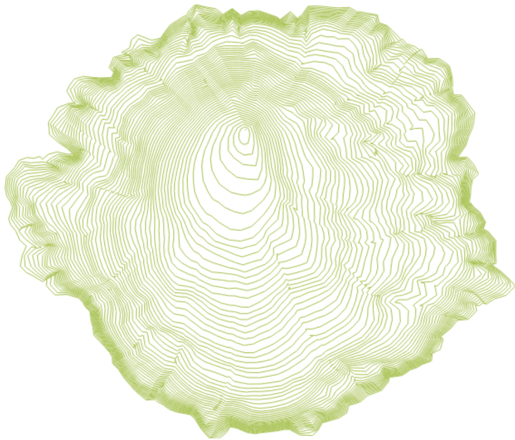
(a) Real final perimeter of the fire (in purple)n as recorded by the authorities.



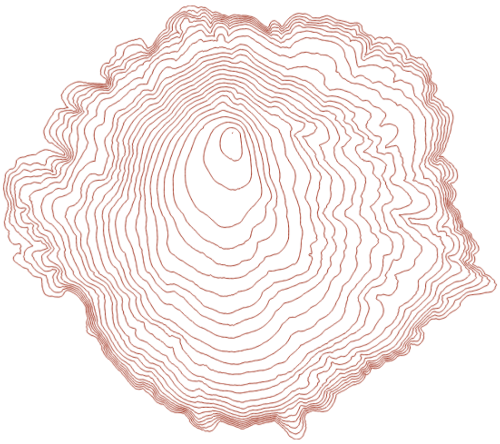
(b) timestep=5 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=False



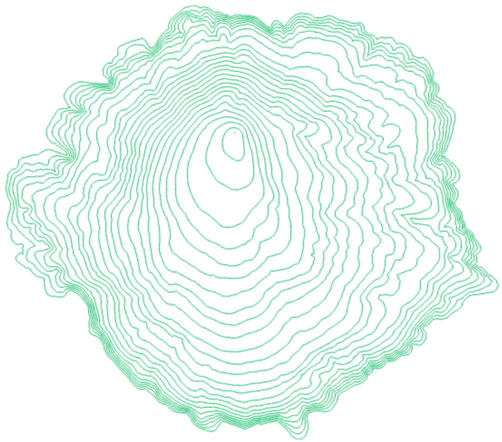
(c) timestep=5 visibleStep=45 perimeterResolution=5 distanceResolution=100 enableCrownfire=False enableSpotFireGrowth=False



(d) timestep=5 visibleStep=45 perimeterResolution=100 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=False

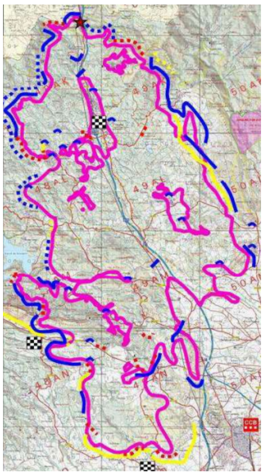


(e) timestep=100 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=False

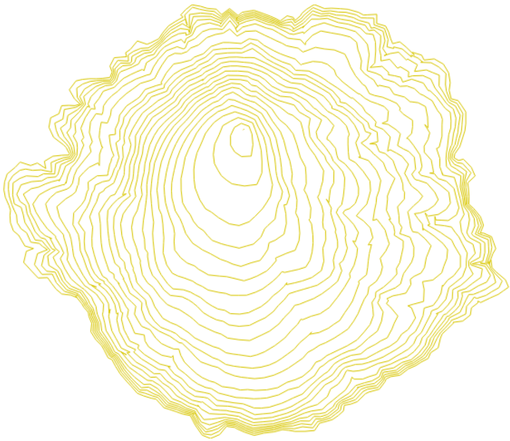


(f) timestep=100 visibleStep=45 perimeterResolution=5 distanceResolution=100 enableCrownfire=False enableSpotFireGrowth=False

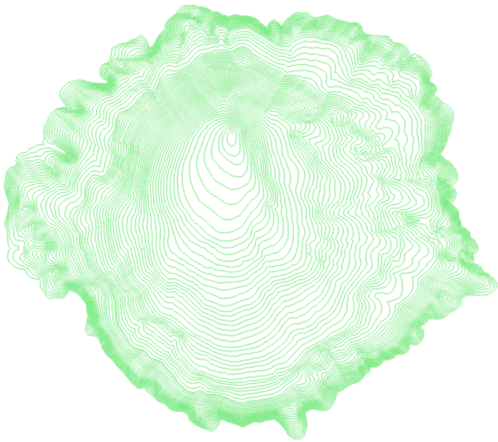
Figure 3: Perimeter of the fire at each outputted time step. The outermost perimeter is strictly the oldest simulation time towards the future. Parameters for each simulation are detailed in the subfigure captions.



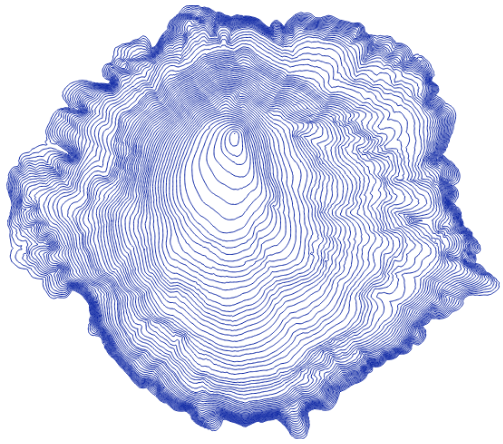
(a) Real final perimeter of the fire (in purple)n as recorded by the authorities.



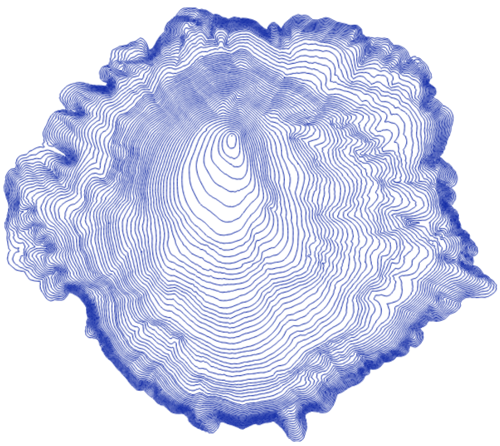
(b) timestep=100 visibleStep=45 perimeterResolution=100 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=False



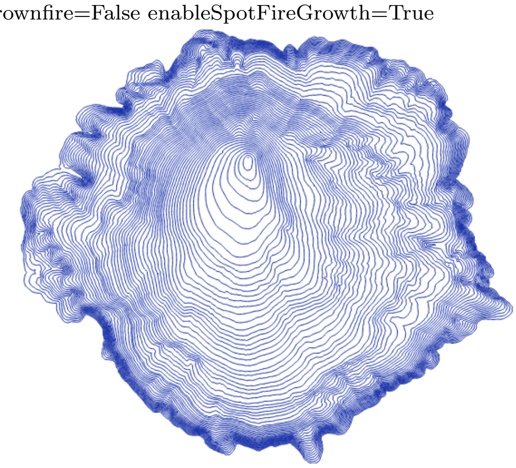
(c) timestep=10 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=True enableSpotFireGrowth=False



(d) Repetition #1: timestep=10 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=True



(e) Repetition #2: timestep=10 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=True2

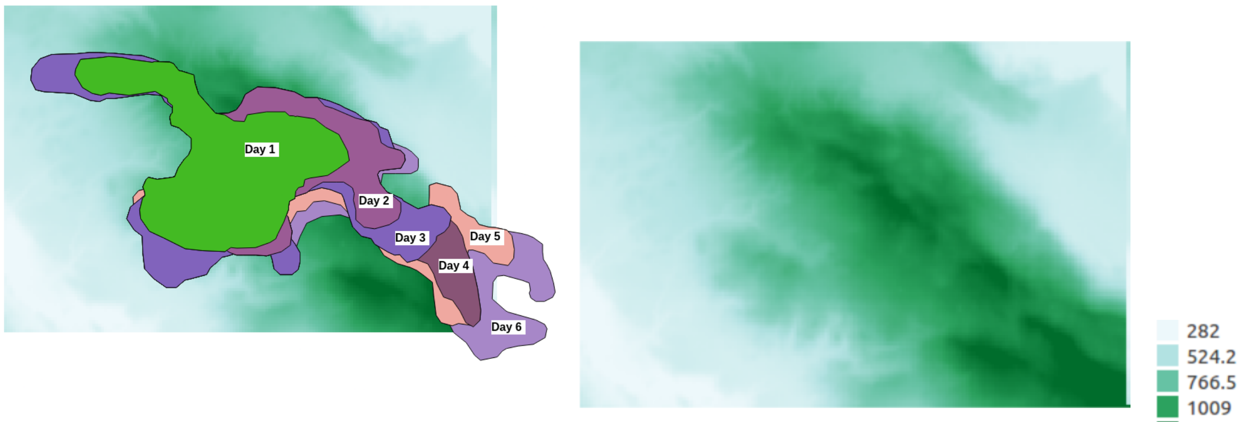


(f) Repetition #3: timestep=10 visibleStep=45 perimeterResolution=5 distanceResolution=5 enableCrownfire=False enableSpotFireGrowth=True3

Figure 4: Perimeter of the fire at each outputted time step. The outermost perimeter is strictly the oldest simulation time towards the future. Parameters for each simulation are detailed in the subfigure captions.

Study Case

We can see in Figure ?? the propagation of the study case fire that happened in reality as recorded by the competent authority when it happened. We see that in four days it arrived to the edge of the orographic map we were provided. This means we will perform the simulations in what follows for four days at most, since longer times would lack comparability with the real case.



(a) Real Propagation **(b)** Underlying orography

Figure 5: (a) Real propagation of the forest fire in 6 days. The outgoing shapes of different filling colors represent the fire perimeter in each consecutive day (the upper most green one being the oldest in time, then the maroon one is the second oldest day and so on). We checked that the overlapped regions were strictly in common in all the layers, meaning we need not plot them all separately. (b) Underlying elevation map for our study case with its corresponding colormap legend in the right. The map is of 168x111 pixels of size 100x100 m.

All the simulation experiments we performed can be seen detailed in Table 3 with their computational times averaged over 3 independent runs and their corresponding standard deviations. The resulting perimeters in the first four days are all plotted in Figures 6 and 7.

Table 3: Results of simulations. Times are computed as the average over 3 executions. Times are in format (min:s) and Stdv. stands for sample standard deviation.

ID	timestep (min)	visible Step (min)	perimeter Resol (m)	distance Resol (m)	enable Crownfire	enableSpot FireGrowth	Time	Time Stdv.
1	30	1440	10	10	True	True	30:17	0:1
2	30	1440	10	10	False	False	29:56	0:1
3	300	1440	10	10	True	True	37:44	0:3
4	30	1440	100	10	True	True	2:12	0:0.4
5	30	1440	10	100	True	True	29:25	0:1
6	30	1440	100	100	True	True	0:4.0	0:0.04
7	30	1440	100	100	False	False	0:3.8	0:0.07
8	30	1440	200	100	True	True	0:1.6	0:0.04
9	30	1440	100	200	True	True	0:4.1	0:0.03
10	30	1440	200	200	True	True	0:1.3	0:0.02
11	30	1440	500	100	True	True	0:1.4	0:0.2
12	30	1440	100	500	True	True	0:4.2	0:0.1
13	30	1440	500	500	True	True	0:0.97	0:0.01
14	30	1440	1000	1000	True	True	0:0.86	0:0.06

To begin with the study, we performed the most detailed simulation the RAM of the virtual machine allowed us (index 1 in the Table). It took a very long average time (30 minutes averaged over 3 runs) which we would like to presumably reduce preserving in the best way possible the quality of its resulting perimeter evolution. It turns out that even if it is the best case simulation (with the finest perimeter and distance resolutions, time-step and with all the extra model complexities activated), it does not give us a resulting prediction that matches much, even visually, with the ground-truth, as can be seen by comparing Figure 6 a and b. This is possibly in part due to the fact that we are not using any detailed simulation for the wind-field which is a very relevant factor for the fire spread. Yet we will consider this first exhaustive simulation as our reference.

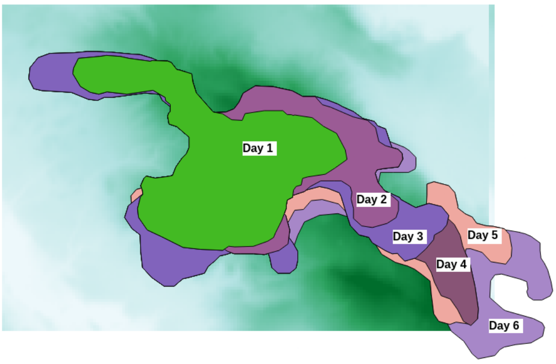
We then try to see what happens when we disable the model complexities of the crown fire and the spot fire growths (simulation 2 in the Table), we see by comparing the results (Figures 6 a and c) that the resulting perimeter evolution does not change significantly at all. Yet, since the computational time is only slightly reduced by about a 1.16%, and as they could be helpful for a more realistic result, we decide to leave them activated for most of the remaining tests.

We then try to change the timeStep increasing it by a factor of ten (experiment 3), in order to check if increasing it is worth the loss of quality-reduction in computational time trade-off. We do see a significant change by looking at the Figures 6 a and d, where we can see that the evolved perimeter is smoothed in the case of a bigger timeStep (d). This lack of detailed grain in the perimeter, even if in this case the overall perimeter still looks similar, will be unacceptable for us. In fact, rather oddly, we find the average computation time to be increased by about 25%! We say this is strange since we expected a higher time-step to reduce the number of required iterations. However this is likely due to the fact that a bigger time-step will imply that in each step the algorithm will try to evolve the perimeter of a wave-front element longer than the allowed distance resolution, which will result in many intermediate steps to be made where one would not expect them. Both because of this last fact (which when increasing the distance resolution might be alleviated), but specially because the perimeter itself suffers an unwanted simplification, we decide to leave the timeStep at 30. In fact, we already saw in Exercise 1 that in the best case, the reduction of this parameter only decreased the computational time by a small fraction, so the loss of quality therein will be worthless.

Next we try increasing the perimeter and distance resolutions by factors of ten independently or together (experiments 4, 5 and 6 respectively). We note there is a very big drop in computational time when we increase the perimeter resolution alone or together with the distance resolution (92.35 and 99.77% respectively), though not that big if only increased the distance resolution, of about 2.86 % (which are due to the reasons exposed in more detail in exercise 1). Since the resulting simulations do not seem to be affected much (see Figures 6 a, e, f and Figure 7 b), where we see that changing the distance resolution does not appreciably affect the result and the perimeter resolution change has only slightly modified some previously sharp edges, we decide to adopt both changes and to test yet further simplifications of the parameters. The slight loss of quality has no comparison with the decrease in computational time.

We try again what happens if we disable the extra model complexity parameters of crown fire and spot fire growth (experiment 7). We now note an even smaller absolute computational time drop relative to our new optimal experiment 6, of about 0.2 seconds. The perimeter evolution on the other hand does not seem to change, thus, we decided to leave them still activated.

We try again to increase the perimeter and distance resolutions, now by a factor of 2, since these are clearly the critical parameters of the model (experiments 8, 9 and 10, only the perimeter, only the distance and both at the same time respectively). We note again a drop in computation time, of about 67.5% in the best case. The drop this time however is not that significant for our time scales, since the simulation takes 1.5 seconds instead of 4 seconds (where the previous absolute change was of about 30 minutes). In fact if we simulate the same conditions but now with a 10 factor increase of the perimeter and distance resolutions (experiment 14) instead of just a factor 2, the time reduction is still of about the same order of magnitude of 75%, with just a drop from the 4 seconds to about 0.9



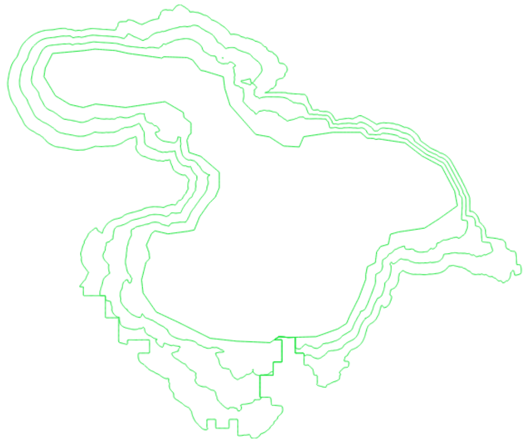
(a) Real Propagation

(b) timestep=30 visibleStep=1440 perimeterResolution=10 distanceResolution=10 enableCrownfire=True enableSpotFireGrowth=True

(c) timestep=30 visibleStep=1440 perimeterResolution=10 distanceResolution=10 enableCrownfire=False enableSpotFireGrowth=False

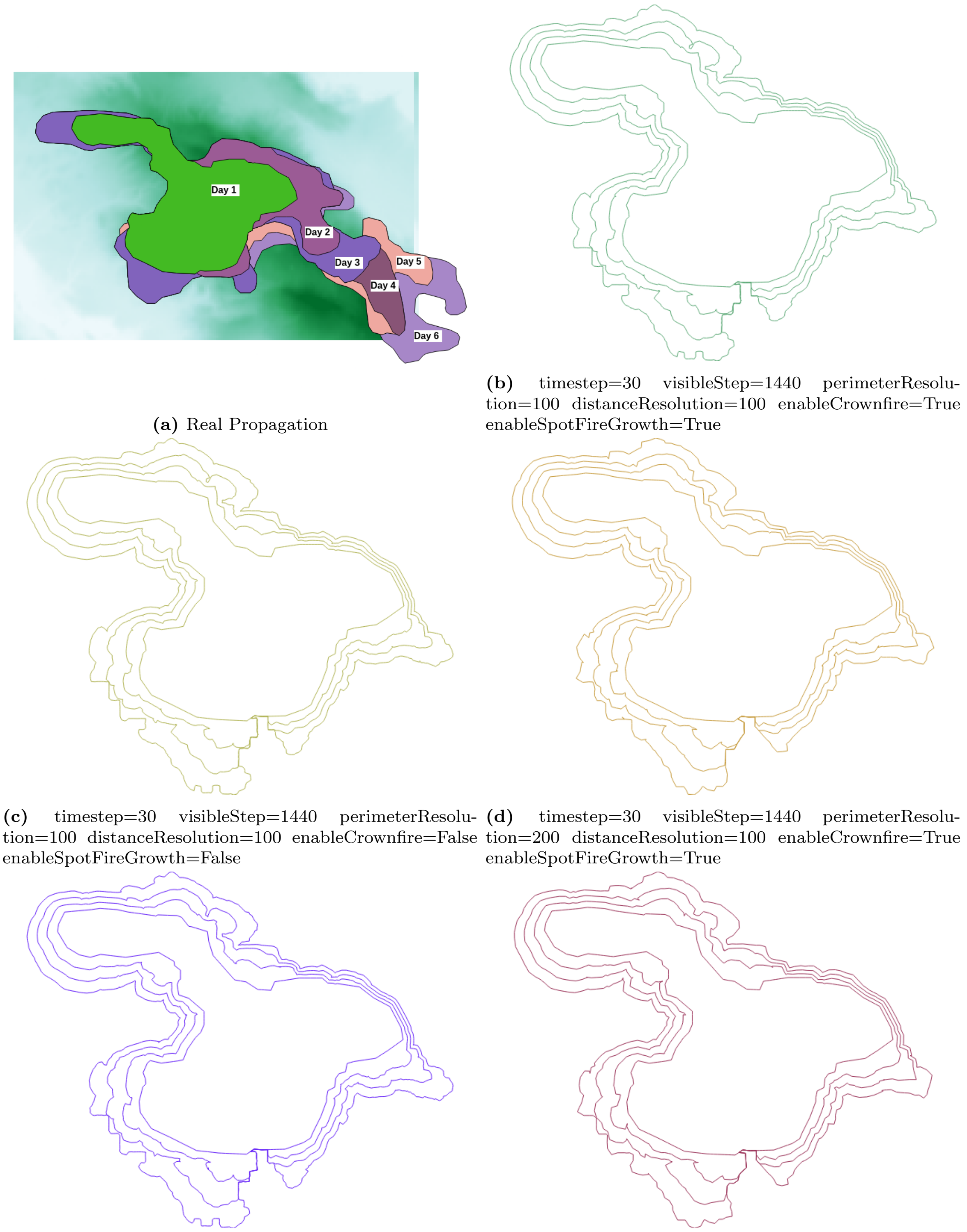
(d) timestep=300 visibleStep=1440 perimeterResolution=10 distanceResolution=10 enableCrownfire=True enableSpotFireGrowth=True

(e) timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=10 enableCrownfire=True enableSpotFireGrowth=True



(f) timestep=30 visibleStep=1440 perimeterResolution=10 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True

Figure 6: Perimeter of the fire at each outputted time step. Parameters for each simulation are detailed in the subfigure captions. Experiments 1-5 of Table 3.



(a) Real Propagation

(b) timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True

(c) timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=100 enableCrownfire=False enableSpotFireGrowth=False

(d) timestep=30 visibleStep=1440 perimeterResolution=200 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True

(e) timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=200 enableCrownfire=True enableSpotFireGrowth=True

(f) timestep=30 visibleStep=1440 perimeterResolution=200 distanceResolution=200 enableCrownfire=True enableSpotFireGrowth=True

Figure 7: Perimeter of the fire at each outputted time step. Parameters for each simulation are detailed in the subfigure captions. Experiments 5-10 of Table 3.

(a) timestep=30 visibleStep=1440 perimeterResolution=**100** distanceResolution=**100** enableCrownfire=True enableSpotFireGrowth=True

(b) timestep=30 visibleStep=1440 perimeterResolution=**200** distanceResolution=**200** enableCrownfire=True enableSpotFireGrowth=True

(c) timestep=30 visibleStep=1440 perimeterResolution=**500** distanceResolution=**500** enableCrownfire=True enableSpotFireGrowth=True

(d) timestep=30 visibleStep=1440 perimeterResolution=**1000** distanceResolution=**1000** enableCrownfire=True enableSpotFireGrowth=True

Figure 8: Perimeter of the fire at each outputted time step. Parameters for each simulation are detailed in the subfigure captions.Experiments 6, 10, 13 and 14 of Table 3.

seconds. We also try an increase of factor 5 with the same result (experiments 11, 12, 13). Also, by only increasing the distance resolution, the reduction in time is negligible again (it even gets bigger within a single standard deviation!) both for the factor 2 and 5 increases. Thus, we consider only increasing the distance resolution not worth it.

Now, if we analyze the results of these simulated perimeters visually (Figures 8 a to f), we see that the perimeter clearly gets a worse quality, as happened with the timestep, just that now it gets more and more polygonal, diffusing the details of the wavefront. With a resolution (both distance and perimeter) of 1000 m, the result is clearly unacceptable, with 500 m the perimeter is as unacceptable as the one we considered for the increasing timeStep. For the 200 m parameter, the result looks quite acceptable, yet we find that the propagation of the elongation in the bottom right side gets impeded, which is a clearly unwanted effect, since in the real case it is this part of the wavefront that gets maximally expanded. Therefore, taking into account the reduction in computation time is minimal in our orders of magnitude, we select 100 m resolutions as the optimal combination (experiment 6).

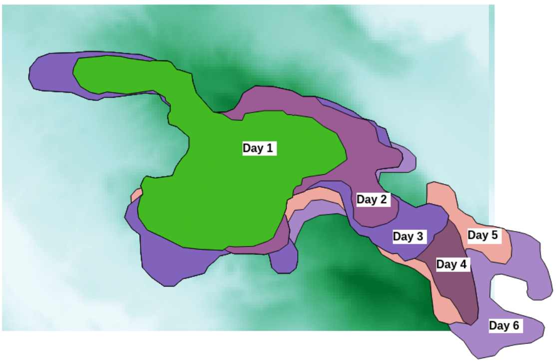
It is interesting to note that the file has a resolution of 168x111 with each pixel of size 100x100 m, meaning that really the difference of the distance and perimeter parameters will be most notable when made bigger than 100 m, which is precisely what we have found.

All in all we decided the optimal parameter combination to be those of experiment 6.

In fact, we finally run the simulation with these "optimized" parameters for four days each by changing the initial fire conditions to each of the real conditions in each day, and see if at any point the simulation starts to match more the reality. We can see these resulting perimeters in Figure 9. The simulation obviously gets more accurate as the initial condition is chosen later on, yet, there is no really good correspondence with the real case in any of the increasing times. For instance, the main difference from the simulation beginning in day 2 to day 3 (inner perimeters in) Figures 9 c and d is that there was a sharp increase of the perimeter towards the top left direction and towards the bottom left, yet, the simulation of day 2 did not catch this. A yet clearer example is the transition from day 3 to day 4 (inner perimeters in) Figure 9 d and e, was a sharp elongation of the perimeter in the bottom right direction. Yet, the simulation was not able to correctly find this. In fact, not even the ac-cutest of the conditions (those of simulation 1 in Table 3) would have found correctly these time-evolutions, as we find in Figure 10, where we performed a four day simulation with initial conditions day 2, 3 and 4, and we find no good match in any of the cases we mentioned above. This calls for an improvement of other parameters we have not been touching, such as the wind conditions.

References

[1] Mark A. Finney. *FARSITE: Fire Area Simulator—Model Development and Evaluation*

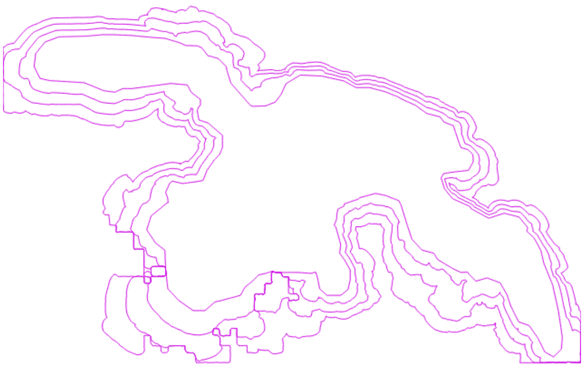


(a) Real Propagation

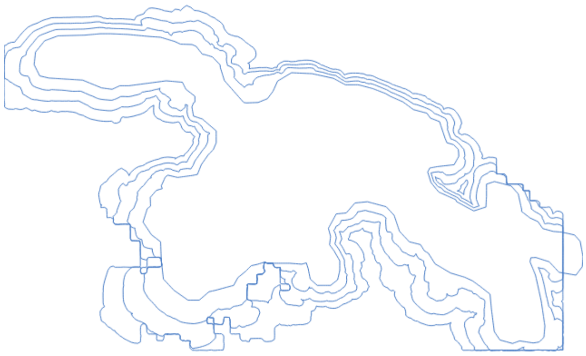
(b) intial time = DAY 1 timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True

(c) intial time = DAY 2 timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True

(d) intial time = DAY 3 timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True

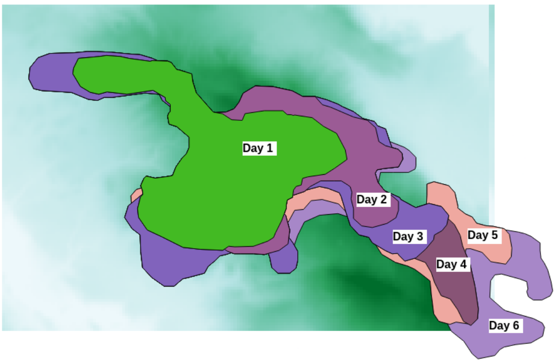


(e) intial time = DAY 4 timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True



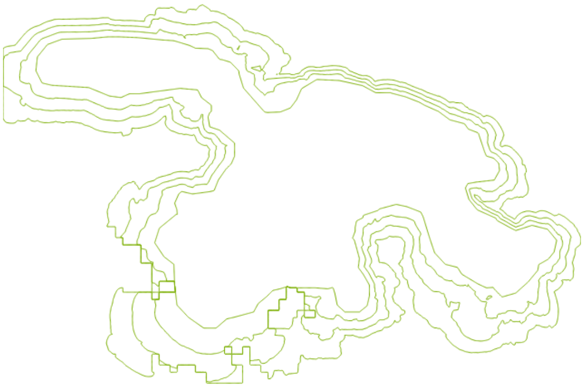
(f) intial time = DAY 5 timestep=30 visibleStep=1440 perimeterResolution=100 distanceResolution=100 enableCrownfire=True enableSpotFireGrowth=True

Figure 9: Perimeter of the fire at each outputted time step with conditions of experiment 6 in Table 3.. Parameters for each simulation are detailed in the subfigure captions. Each simulation has as initial condition a different real perimeter of a different day and each is run for four additional days.



(a) Real Propagation

(b) DAY = 2 timestep=30 visibleStep=1440 perimeterResolution=10 distanceResolution=10 enableCrownfire=True enableSpotFireGrowth=True



(c) DAY = 3 timestep=30 visibleStep=1440 perimeterResolution=10 distanceResolution=10 enableCrownfire=True enableSpotFireGrowth=True

(d) DAY = 4 timestep=30 visibleStep=1440 perimeterResolution=10 distanceResolution=10 enableCrownfire=True enableSpotFireGrowth=True

Figure 10: Perimeter of the fire at each outputted time step with conditions of experiment 1 in Table 3.. Parameters for each simulation are detailed in the subfigure captions. Each simulation has as initial condition a different real perimeter of a different day and each is run for four additional days.