Investigating the Performance and Behaviour of a Forest Fire Simulation

George Brignell-Cash

December 2022

1 Introduction

The forest fire model is a simple case of stochastic spreading whereby its evolution is determined by the properties and behaviour of its nearest neighbours. A forest fire model simulates a forest via a grid of trees and spaces. Fire is then implemented and is allowed to spread through the forest under certain rules. The simulation varies on many input factors of this model and hence this paper details the investigations into the behaviour, convergences and performance of this forest fire model as well as looking into an extension of the model where the direction of wind modifies the capabilities of the fire.

2 Model Convergence

2.1 Introduction

The average behaviour of the model for different starting grids was investigated by varying the grid size N, the probability that a tile is a tree p and the amount of repeats total-seeds, to find the number of steps the model takes until the fire burns out as well as the probability that the fire reaches the bottom of the grid.

To explore how the model converges, the step-count and the probability that the fire spread to the bottom of the grid will be used. The step-count is the amount of steps it takes for the fire to "burn out", i.e, the amount of steps till there is no fire on the grid. The probability the fire reaches the bottom will be a value between 0 and 1 and is calculated through a mean of the Boolean values true or false for many different starting grids.

To begin investigating the model, the results of a 100 by 100 grid were calculated using 1000 repeats.

2.1.1 Grid N=100

The mean step-count was plotted again the tree probability. Which shows an exponential-like curve with a maximum step-count at tree probability 0.6. Since the probability that a tile is a tree can be thought of as the density of the grid, the tree probability will be referred to as the tree density.

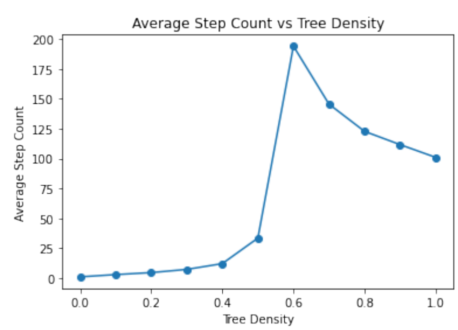


Figure 1: Average Step Count vs Tree Density

The low step-count values between 0.0 and 0.5 is due to the sparseness of the tree tiles restricting the fire's spreading ability and thereby ending the simulation short, leading to lower step-counts. Between these values, the step-count does increase slightly, therefore there is a relationship between the density of the tree tiles and the longevity of the fire.

There is a large difference in step-count values between the 0.5 and 0.6 densities. To get the exact nature what is happening in the model at this point, more data should be collected at a higher resolution. However the densities from 0.6 have a very clear exponential decay relation, where the step-count tends to the size of the grid N. This happens as when the density is higher, the efficiency of the fire spreading through the grid is greater and therefore requires less steps.

The probability that the fire spreads to the bottom of the grid will be referred to as the doomsday probability, as a value of 1 means that the fire is certain to spread across the whole forest.

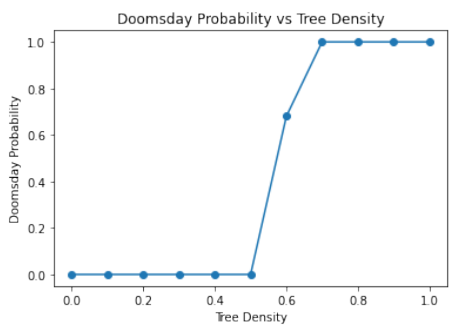


Figure 2: Doomsday Probability vs Tree Density

The doomsday probability was plotted against the tree density. The relationship represented a step function with an exact doomsday probability value of 0 for densities below 0.5 and an exact certain doomsday value of 1 for those above 0.7. Once again the tree density value of 0.6 sticks out as significant as it is the only value in this range that does not sit on one of the steps, and higher resolution data around this value needs to be added.

Overlapping the two plots does not provide much insight apart from emphasising the significance of the 0.6 density point.

2.1.2 Increasing the resolution of data between 0.5-0.7

More precise data was then collected between the 0.5 and 0.7 tree density values.

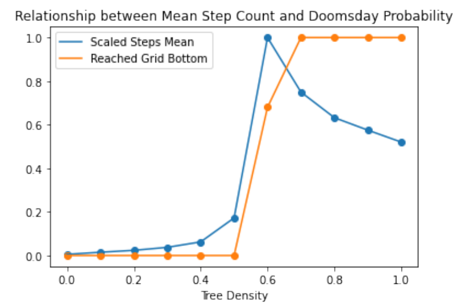


Figure 3: Comparison of Doomsday Probability and Average Step Count

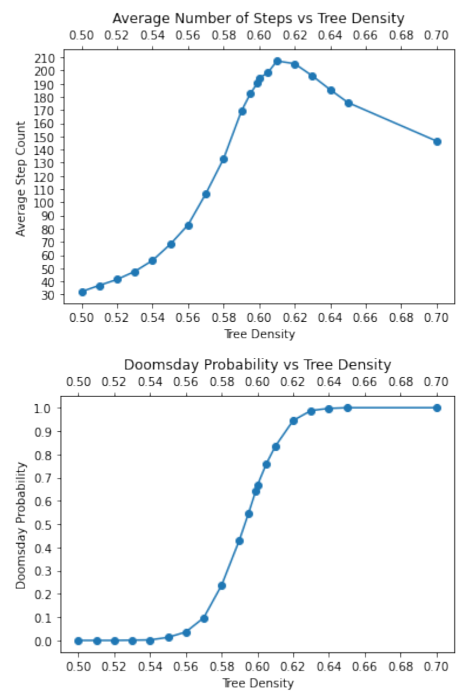


Figure 4: Re-scaled Graphs for the Doomsday Probability and Average Step Count vs Tree Density

The new data produces much smoother curves on both graphs. The 0.6 density value was originally thought of as significant for both doomsday probability vs tree density and step-count vs tree density, however the significance of this value has completely diminished from the doomsday probability plot, as

0.6 is not individualistic from the relationship. The important point to note is that the doomsday probability is 1 at 0.65. On the other hand, the step-count significance of 0.6 has only shifted. It is now seen that the maximum value of the relationship is at 0.61. At this point the combination of the doomsday probability and the spreading efficiency is at its weakest.

2.1.3 Changing repeat count (seeds) to see convergence

Up to this point, each simulation has been run 1000 times and averaged over these runs. In order to reduce the amount of calculations and time spent running these simulations, it may be beneficial to find an optimal number of repeats to use.

Starting with a grid size of N = 100, the step-count and the doomsday probability results were plotted for different total repeat values on top of each other.

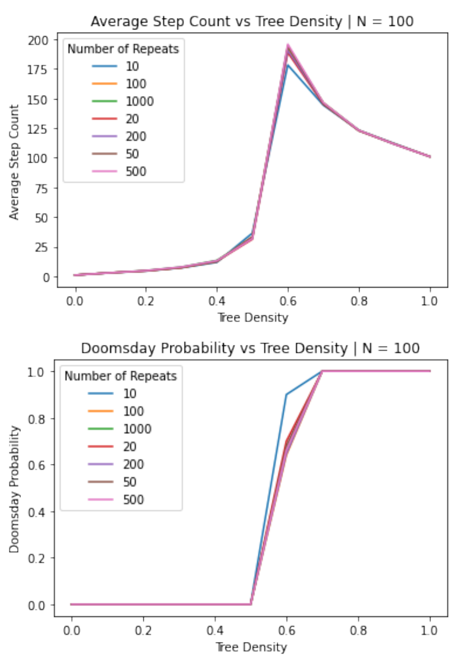


Figure 5: The Step Count and Doomsday Probability vs Tree Density for a Range of Repeat Values for N = 100

The plots show a converging relationship for most repeat counts, expect for the lowest value of 10. It is expected that the lower values will not reach this convergence as the influence of each result in the set of repeats has a more significant effect on the mean. For N = 100, a repeat count of 20 is able to converge to the trend line. However, since simulations of grids with different N values are to be carried out, the convergence should be investigated for more grid sizes.

Plotting the graphs for N=50, N=200 and N=500, it is seen that the larger the value of N, the higher the repeat count needs to be in order to reach convergence. This is shown in the N = 200 grid, where the lower repeat counts stray further from the convergence than in N =50, and N =100.

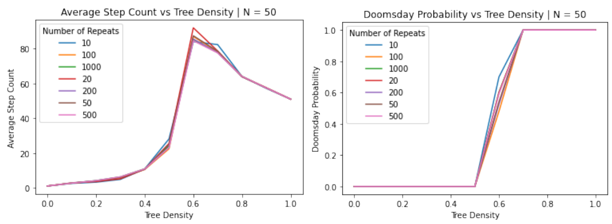
For values of N up to 500, a repeat count of 50 is sufficient in meeting the convergence. From now on the repeat count for the rest of the investigations are using a repeat count of 50, in order to speed up the time it takes to perform the simulations. However for much much larger values of N, a larger repeat value will be required. It is therefore worth investigating at what point N converges, and then find a repeat value for that N.

2.1.4 Varying the Grid Size N to Reach Convergence

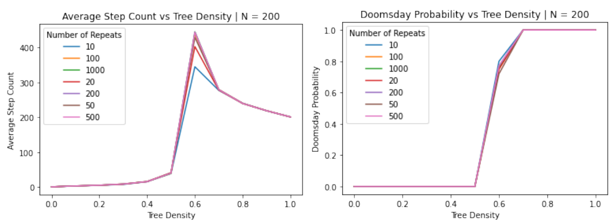
A basic step-count vs density plot for a range of different Ns is not very useful as the scale of the step-count varies with N, as the larger the grid is the further the fire can travel. This plot is shown below, and does show that the step-count vs tree density has the same relationship in all the value of N with a peak around 0.6.

A more useful way to visualise this data is to divide the step count for each N by N, this gives a relative step count and means that the data can be compared between different values of N.

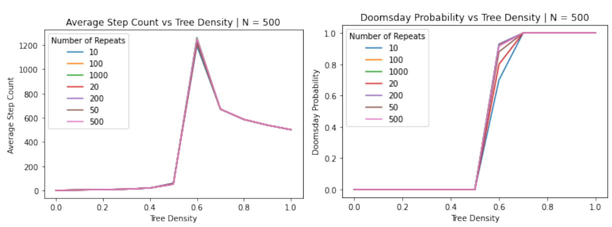
This shows an interesting behaviour between the different sizes of grids. For the smaller sizes, the relative step-count is higher between the density ranges of 0.0 to 0.5 and 0.7-1.0, but the larger grids have a greater relative step count between 0.5-0.7. At 0.6 density, the lower value of N, the lower the Step Count/N. Showing that the larger the grid, the more



(a) N=50



(b) N=200



(c) N=500

Figure 6: The Step Count and Doomsday Probability vs Tree Density for a Range of Repeat Values for N = 50, N = 200 and N = 500

relative steps it will do. This is due to the large amount of tiles in the grid, which causes ”runways” of tree tiles that the fire can burn down. These runways are the most prominent at densities around 0.6 as they rely on both having large enough gaps such that they are isolated from the rest of the grid, as well as having a high enough density in order to create them. As the density increases, the amount of runways falls and therefore so does the relative step count. The doomsday probability graph gives an insight into this behaviour.

This plot shows that at lower grid sizes and lower densities, the doomsday probability is higher than

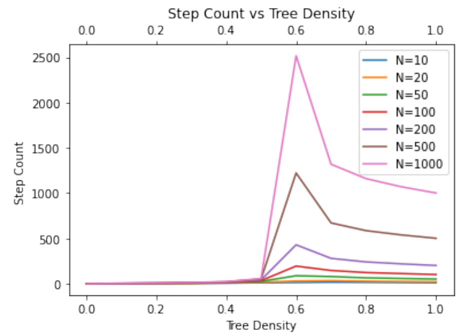


Figure 7: Average Step Count vs Tree Density for a Range of N Values

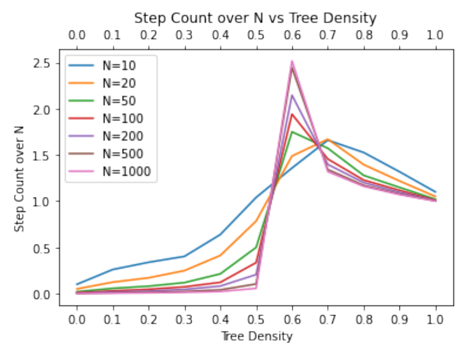


Figure 8: Average Step Count Over N vs Tree Density for a Range of N Values

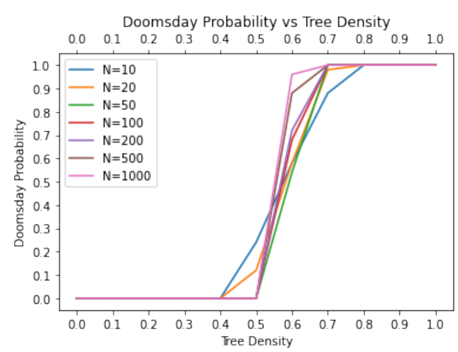


Figure 9: Doomsday Probability vs Tree Density for a Range of N Values

the larger grid size. This is most clearly seen at tree density 0.5, therefore the doomsday probability for a large range of N for density = 0.5 were calculated and a model of Doomsday probability vs N was plotted.

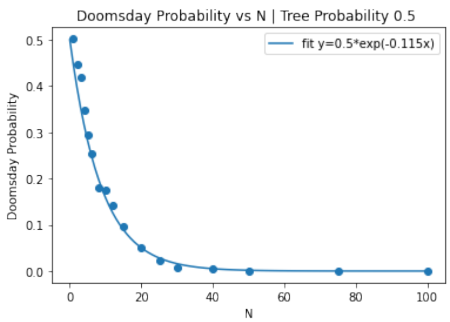


Figure 10: Doomsday Probability vs N — Tree Probability = 0.5

This exponential relationship shows that for grids with N above 40, there is almost no chance of a doomsday fire for densities of 0.5 and less. This exponential relationship can be represented for the densities less than this by changing the 0.5 in the fit equation.

2.2 Adding Wind

2.2.1 Introduction

One way to expand the forest fire simulation is to add wind, this adds a new complexity to the simulation and should result in a different behaviour. The rules implemented for the wind are that it can only travel in the four directions, North, West, South and East. The wind allows the fire to travel twice as far in a single time step, including jumping over a tile which doesn't contain a tree. The wind only affects the fire in the direction specified, i.e. when the wind is travelling South, the interactions in the other North, East and West directions carry on as the normal neighbours interaction. It is important to note that in the simulation, only the most Northern trees start on fire, this affects the results of the wind simulations and therefore the following data only applies to the scenario where the top row of trees starts on fire.

2.2.2 Results of Wind

Once again the plots of Step Count and Doomsday probability vs Density are plotted. Both graphs show that the wind direction has a significant affect on the data. An immediate note should be that when the wind and the direction the fire is spreading is the same, the probability of a doomsday becomes much higher than seen before; even for low tree densities.

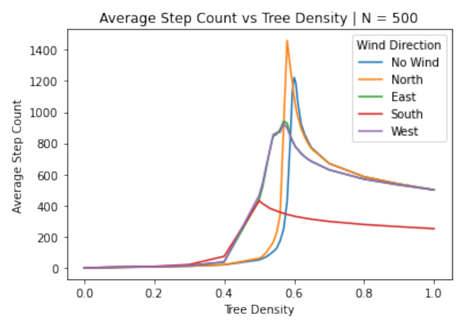


Figure 11: Step Count vs Tree Density — N = 500

The "No Wind" line is the same shape as the data seen above for the N=100 investigation, however with a larger grid size. The plot shows that there are significant changes to the amount of steps required to reach the end due to the wind direction. The data changes in three ways, the maximum average step-count, the tree density of that maximum and the convergence value of the average step-count. For the case where there is no wind, the maximum average-step count is roughly 1200 at a tree density of 0.6 and the step-count convergence is at the total grid size, N which for this investigation is 500. This windless line plot will be considered the base case for which the other wind directions can be compared to. When the wind is North facing, it is travelling against the direction of the fire spread; however, since the wind does not affect the spreading of the fire in any other direction other than the wind direction, in interesting behaviour is brought out. For a North wind, the maximum average step-count increases. This is due to the wind allowing the fire to backtrack and jump over empty tiles in the grid, allowing it to find more tree tiles to burn in a given run therefore increases the step-count. The tree density value of the maximum marginally decreases from 0.6 to 0.59; this is expected as if the wind is allowed to backtrack even further; it is able to find more routes to burn all the way through to the end of the grid and therefore lowering the tree density value that the grid needs in order to burn further and therefore longer. The relative differences between the North wind and the non-wind case become smaller and less significant as the tree density increases. This can be seen by a convergence of the data at larger densities. The East and South wind can be grouped together, for large enough repeat counts, these two scenarios should be identical as the for a sideward motion of fire on the grid, it should be irrelevant as to which sideward direction it is coming from. A good picture to understand this concept is to imagine a 2d grid in which the fire is travelling from left to right, this will have identical properties to when the grid is flipped onto its back and the fire now travels from right to left. From the graph it shows that these two data sets are almost identical as predicted. The slight difference between them comes from the random error in where the trees are generated when the grids are setup. The sideward winds have a lower maximum step-count than both the North and the Windless grids at 900, positioned at a tree density of roughly 0.57. The average step-count converges at 500 along with the North and windless models, emphasising the fact that an increase in density reduces the amount of fire spreading gain that the wind provides. The most significant wind direction is South as it is in the direction of that the fire is spreading. A South wind should effectively double the speed for a maximum density grid as the fire is able to spread twice as fast. It should also lead to even larger gains for lower densities as not only is the fire twice as fast, but it is able to jump over some the inefficiencies cause by holes in the grid. The data shows this exact behaviour, with a step-count convergence at 250 rather than 500, exactly half what the other winds were doing. And a maximum step-count value at 420, which is almost three times less than for the windless model, and demonstrates that the it doesn't just travel twice as fast which would lead to a speed up of 2, it instead increases its efficiency by jumping over the gaps and manages to get a speed up of 3.

The doomsday probabilities should coincide with the observations from the average step-count. Such as a lower tree density value required to reach a probability of 1 for a Southward wind.

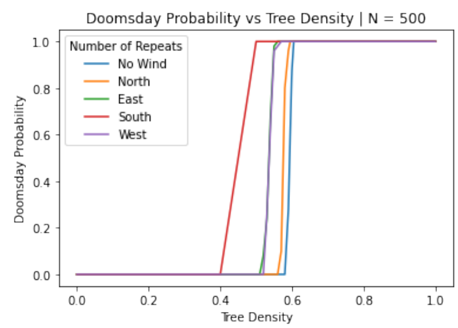


Figure 12: Doomsday Probability vs Tree Density — N = 500

This plot is able to reinforce the points from above, such that any wind decreases the tree density required in order to reach a doomsday probability of 1. The largest increase is in the South direction, where the density lies between 0.4 and 0.5. However further investigation is required in order to find the exact point between these densities.

2.3 Performance

2.3.1 Introduction

When performing large simulations, it is often important to consider the most efficient way to run the code. Different parallelisation methods can be used to significantly speed up calculations, however.

2.3.2 Using Walltime to Investigate Performance vs N

Wall time is a simple metric of performance, which records the amount of time it takes to run the calculations in a program. In an ideal world, a greater number of OpenMP threads leads to a lower walltime. However in reality, increasing the thread count can eventually lead to an increase in walltime.

The walltimes for various grid sizes N, were recorded for a small sample of thread counts and using this data, a Walltime vs N graph was plotted.

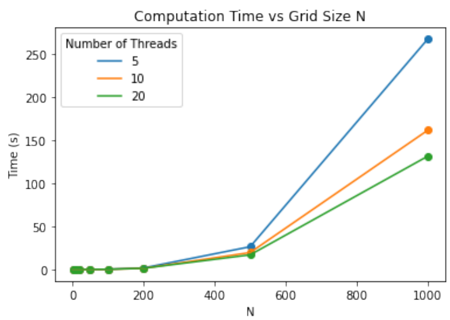


Figure 13: Walltime vs Grid Size N

This graph first proves that the code is being parallelised. This is shown by a difference in the walltimes for the different thread counts. The graph shows a parabolic shape, however this is because the number of computations per grid is equal to $N * N$ and therefore the amount of computation time varies parabolically with N. As expected, a larger number of threads leads to a lower computation time, with the lowest thread count 5, being twice as slow as the greatest 20. It is important to note that even though the thread count has been quadrupled, the wall time as only halved. This suggests that there is a diminishing return and a reduction in efficiency when adding more threads.

This implies that for this simulation, adding more threads leads to larger computational costs. Sometimes the walltime of the total simulation is not the most important factor. For example, if the simulation is being run on a third party cluster where each node hour has a monetary cost associated with it, it may be beneficial to reduce the amount of node-hours in order to reduce the price of using the systems. There are also environmental factors associated with wanting to reduce the computation cost, for example, reducing the energy usage in order to minimise the carbon footprint caused by a simulation.

It is therefore also important to investigate the computation costs in order to weigh up the wall speed over the simulations other impacts. In this investigation, the computational cost is calculated using the wall time of the simulation multiplied by the number of nodes used.

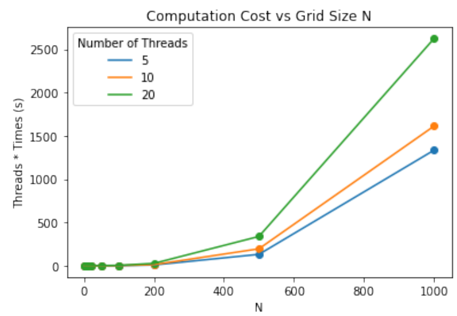


Figure 14: Computational Cost (Thread Seconds) vs Grid size N

The graph of computation cost vs grid size is almost identical to the walltime vs grid size, however with one large significance in that the greatest value of threads 20, is no the top line indicating that the greater the number of threads, the greater the computational cost associated. This suggests that the most cost efficient way to run a simulation is on a single thread, however for larger scale simulations, the amount of time required to do the computations on one thread is far too great to that in reality, there needs to be a balance between the time taken and the computational costs.

2.3.3 Computational Cost for Various Tree Densities

When looking into the amount of time that a problem takes to compute, it is worth looking into which parts of the simulation are causing the most strain. The grid size is inevitably going to increase the amount of time as that purely increases the amount of calculations required, however the tree density is another important factor that can have an effect on the computation time.

The walltime vs tree density and the computation

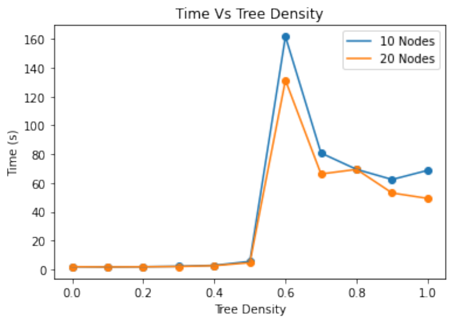


Figure 15: Walltime vs Tree Density — N = 500

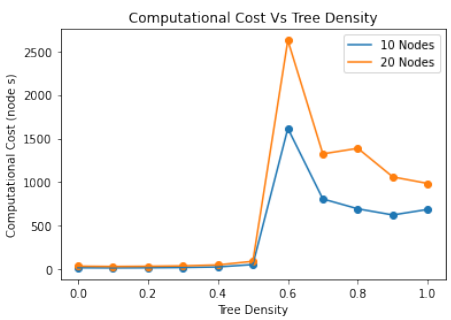


Figure 16: Computational Cost vs Tree Density — N = 500

cost vs tree density plots show that the density of 0.6 is indeed the most expensive part of the simulations. Up to the value of 0.5 the walltime and computational cost is so low that it would not affect the overall runtime significantly if the precision of the data points between 0.0 and 0.5 was greater.

2.3.4 Measuring the Speedup and Efficiency

The speedup is a metric used to see how much faster the parallel computations get per thread. The ideal speedup is calculated using $\frac{N}{N_0}$. Where N is the number of threads and N_0 is the smallest number of threads which is usually 1. However the real speedup is $\frac{t_0}{t}$ where t = time for N threads and t_0 = time for N_0 threads.

These two metrics can be combined into the Efficiency metric, calculated using $\frac{Speedup}{IdealSpeedup}(\%)$.

The walltime vs the number of threads plotted below use a tree density of 0.6 and a grid size of 500. The value of 0.6 was chosen as it is the most computational time expensive density, leading to more defined differences between the thread count walltimes.

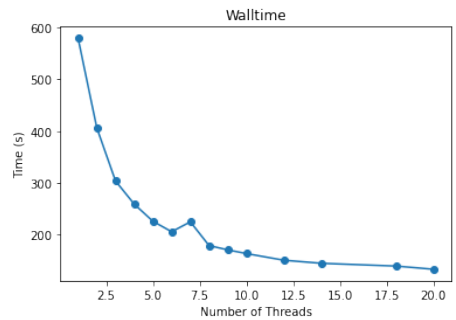


Figure 17: Walltime vs Number of Threads

As expected, the lower the thread count, the greater the walltime. However the shape of this graph is important as it shows that the relationship between walltime and thread count is not linear. The relationship is instead negatively exponential, meaning that a there is a point where adding more threads doesn't lead to an decrease in the computation time.

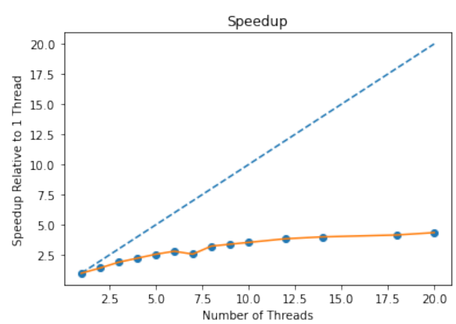


Figure 18: Relative Speedup vs Number of Threads

The speedup has been calculated and plotted along with the ideal speedup. This plot shows that the real speedup for this parallelisation is much lower than it theoretically could be and that for each thread added to the system, the speed in which the calculations are performed speeds up by less than one threads power. This relationship is better represented as the efficiency of the threads. The efficiency of the threads is plotted below. The efficiencies are compared to

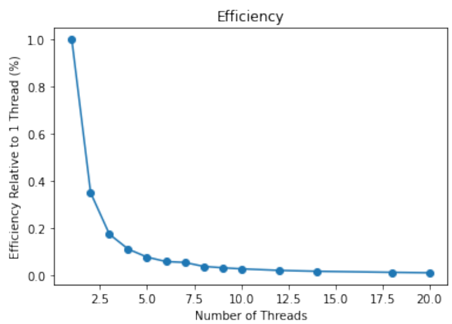


Figure 19: Efficiency vs Number of Threads

the computational power of the first thread. The relationship between efficiency and thread count is a negative exponential, mirroring what was shown in the wall time plot. This curve eventually converges to a point that adding more threads is negligible to the computational time.

These three metrics can be used to identify the maximum amount of threads a system actually uses.

3 Conclusion

In conclusion, the forest fire model was first investigated to find the values of the simulation inputs for which the model starts to converge. It was found that a repeat count size of 50 was more than enough for grid sizes up to 200, however the repeat count size turned out to depend on the grid size, meaning that an increase in grid size lead to an increase needed in the repeat counts and it therefore didn't converge. However the convergence of N was then investigated, and the values of step count and doomsday probability starts to converge at N = 500. All the following data was performed at this grid size N =100, with a repeat count of 100 in order to ensure the points will have fully converged and been filtered out of random errors. An extension to the model was then built to simulate wind in various different directions. The simulation was run and showed that the direction of the wind had significant effects on the model. With the most significant effect being when wind was traveleign in the same direction as the fire was spreading, leading to a huge increase in travel speed across the grid, and an increase the in fire spreading efficiency. The computation times of the simulation at different grid sizes was investigated in order to see how the thread choice affected the walltime. The expectation was that an increase in threads would decrease the amount of time the simulation would take to run, and this was shown in various plots. The downside to running a larger number of threads was shown through the computational cost, where more threads leads to a larger cost which in some situations leads to more money or more emissions need to be spent in order to investigate a model. The performance vs the tree density was looked into and revealed that the lower densities are practically neglible when compared to the larger ones, the data points for these values can therefore be increased without a larger computational cost associated. And finally the walltime, speedup, and efficiency were compared against the number of threads, revealing that there is a limit to the gain when adding more threads to a simulation and that in order to be efficient in terms of time to threads, the lower the thread count the better.