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

Lorena Pérez Gálvez 1535868

Artur Llabrés Brustenga 1528792

Xabier Oyanguren Asua 1456628

Block A

3. Identificad los diferentes tiempos de ejecución explicados en el Bloque 0. ¿Cuáles de estos tiempos presentan una reducción más significativa al aumentar el número de threads? ¿A qué es debido?

The different execution times are the following ones:

- **Mesh Time:** This is the time taken by the simulator to generate the grid of points over which the wind field values will be provided.
- **Initialization time:** The time taken to initialize the variables given the initial condition algorithm (possibly related to the generation of the initial iteration wind-field from the initial parameters relating the wind speed etc. and the orographic characteristics of the landscape).
- **Equation build time:** This is the time taken by the initialization of the pre-conditioner matrix and the rest of elements for the iterative algorithm.
- **Solver time:** The time taken by the conjugate gradient iterative algorithm to find the equation system solution.
- **Output write time:** Time taken by the dumping of the obtained data and results to the different generated files.
- **Total time:** The total sequential time taken by the execution of the case of study.

Let us now analyze per each time "type" the trend followed when the executions are made with more threads and the maps are made finer (higher resolution/more exhaustive simulations), which can be abstracted from Table 1:

- **Mesh Time:** The bigger the considered map, we see a clear increase in the time required to build the mesh, which is an immediate consequence of the size of the asked wind-field. Interestingly, the generation of the mesh seems to be highly parallelizable, since in all cases, when increasing the thread number, the mesh generation time monotonously decreases.
- **Initialization time:** The bigger the map, the field initialization time increases monotonously, which suggests that effectively, it is the process generating the initial windfield. This process seems to be hardly parallelizable, since the decrease in time for more threads is hardly noticeable.
- **Equation build time:** The same trend for increasing map size can be observed (time increases), which is a consequence of the bigger sizes of the matrices (the pre-conditioner etc.) which have a dimensionality proportional to the total number of nodes in the mesh. A clear reduction in time can be observed in all cases when the thread number is increased, which suggests that the matrices and equations in general are built (saved in memory) block wise, where each thread takes care for some parts of the initialization.

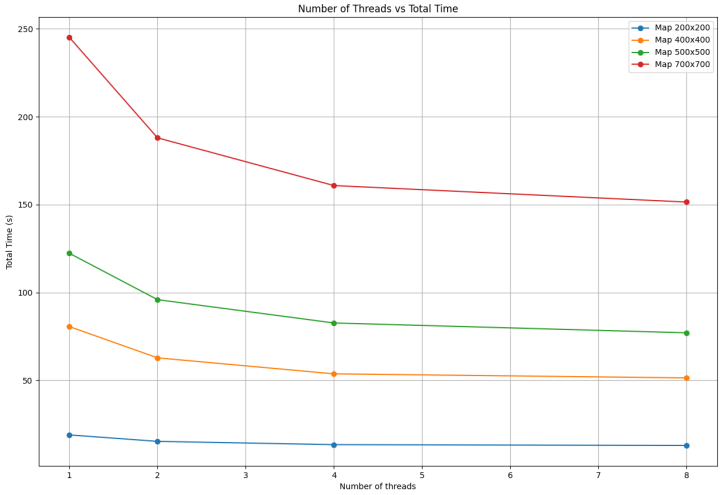
- **Solver time:** Comparing this time with the rest of times, we see that in all thread cases it is the bottleneck of the process, taking the biggest of the times. This was expected since it ts the core part of the computation (the iterative search of the solution). Since it involves iterative matrix-vector products, and these have a dimensionality proportional to the mesh size, the time of solving increases with the map size. What is more, it turns out that the reduction in solving time for increasing thread number is hardly noticeable, meaning that there is a sub-bottleneck in the solver that is not parallelized (the matrix vector product possibly).
- **Output write time:** Since for a bigger mesh, the output vector field concerns more nodes, it is natural that for bigger maps, the output writing time increases. We can also observe that the reduction in time for more threads is hardly significative, which is natural, since the output in the files must be generated sequentially within each of the individual files.

The trend for the total time is an immediate consequence of the conjunction of all the partitions we have commented separately.

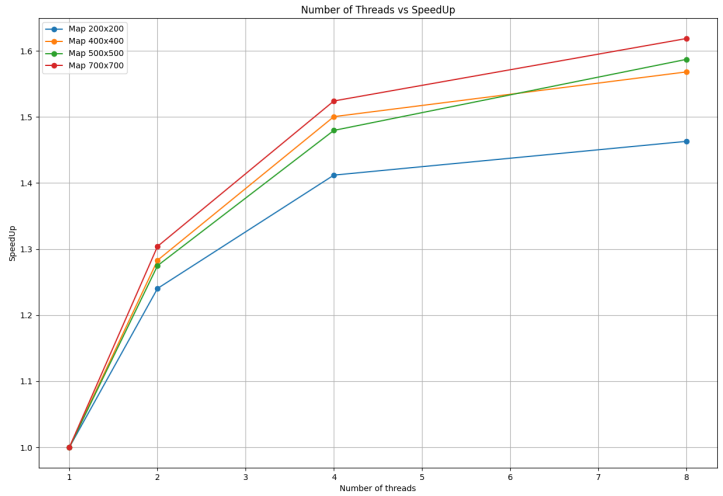
Table 1: Result table obtained from the simulations indicated in points 1 and 2 of Block A in the practice guide. All times are in seconds (s) and follow the notation described in the text.

Map	**Number of threads**	**Mesh time**	**Initialization time**	**Equation build time**	**Solver time**	**Output write time**	**Total time**
200x200	1	0.06	0.35	3.87	12.02	1.31	18.99
	2	0.04	0.35	2.21	10.96	1.00	15.31
	4	0.04	0.34	1.40	10.15	1.03	13.45
	8	0.02	0.33	0.98	10.13	1.01	12.98
400x400	1	0.26	1.58	15.83	51.54	5.60	80.60
	2	0.14	1.45	8.90	44.72	4.30	62.82
	4	0.10	1.42	5.45	40.86	3.95	53.72
	8	0.053	1.41	3.65	40.12	5.35	51.40
500x500	1	0.38	2.20	23.06	78.99	8.67	122.30
	2	0.20	2.11	13.35	69.04	6.32	95.93
	4	0.12	2.03	8.09	63.23	6.08	82.66
	8	0.094	2.33	5.65	60.62	6.00	77.06
700x700	1	0.67	4.29	47.08	158.97	16.41	245.12
	2	0.38	4.10	25.99	136.27	12.00	187.97
	4	0.25	4.05	15.77	123.48	11.52	160.83
	8	0.14	3.86	10.59	121.02	11.27	151.44

4. Realizad dos gráficas: una primera que muestre el tiempo total de ejecución para cada mapa y para cada número de threads, y una segunda que muestre el SpeedUp para cada uno de los mapas. Analizad las dos gráficas y extraed las conclusiones correspondientes.



(a) Total execution time



(b) SpeedUp

Figure 1: (a) Total execution time (s) per each of the maps and each thread number, as indicated in the legend. (b) SpeedUp for each of the maps, as indicated in the legend.

In the Number of threads vs Time plot in Figure 1 (a) we see a clear decrease of the total time as the number of threads increase, specially this decrease is more noticeable in the larger maps sizes. However the time decrease is not lineal, it tends asymptotically to a lower bound (a particular one for each map size).

In the Number of threads vs Speedup in Figure 1 (b) we see that speedup increases asymptotically in the way that it would be expected after seeing the previous figure. It suggests an upper bound for the parallelization speedup of around 1.65 (the computation time will not even be halved!).

This trends are due to the fact that not every part of the execution are equally parallelizable, in particular the sequential bottleneck which is the iterative solver is hardly parallelizable. This makes the overall process speedup be bounded as we were seeing in the plots.

Block B

3. Modificad el parámetro vegetation del archivo de configuración y ejecutad WindNinja con los valores de vegetación trees y brush utilizando el mapa 400x400.tif. ¿Hay cambios en el tiempo de ejecución al utilizar un tipo u otro de vegetación?

We can see in Table 2 the execution times for each of the vegetation cases. One could say the execution time for the "brush" mode is slightly bigger, however, paying attention to the overall orders of magnitude, we conclude there is no significant difference. This suggests that this parameter does not make a notable difference in the execution difficulty.

Table 2: Result table obtained from the simulations indicated in points 2 and 5 of Block B in the practice guide. All times are in seconds (s) and follow the notation described in the text.

Map	Number of threads	Mesh time	Initialization time	Equation build time	Solver time	Output write time	Total time
400x400 .tif *trees*	1	0.43	1.42	14.06	48.99	4.99	75.45
400x400 .tif *brush*	1	0.45	1.43	14.50	51.11	5.03	77.94
400x400 .lcp	1	0.43	1.42	14.85	32.74	4.75	59.55

4. Utilizando Qgis, buscad las diferencias que se observan entre los mapas obtenidos para el módulo de la velocidad del viento y de la dirección del viento. ¿Dónde se encuentran las diferencias? ¿Por qué?

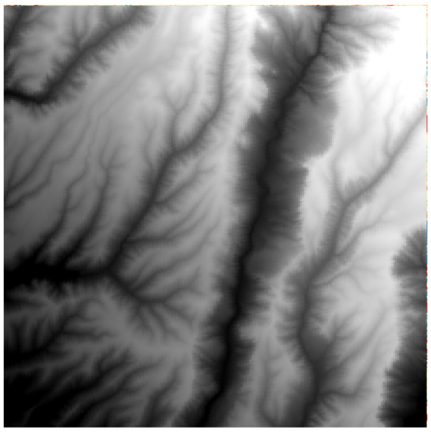
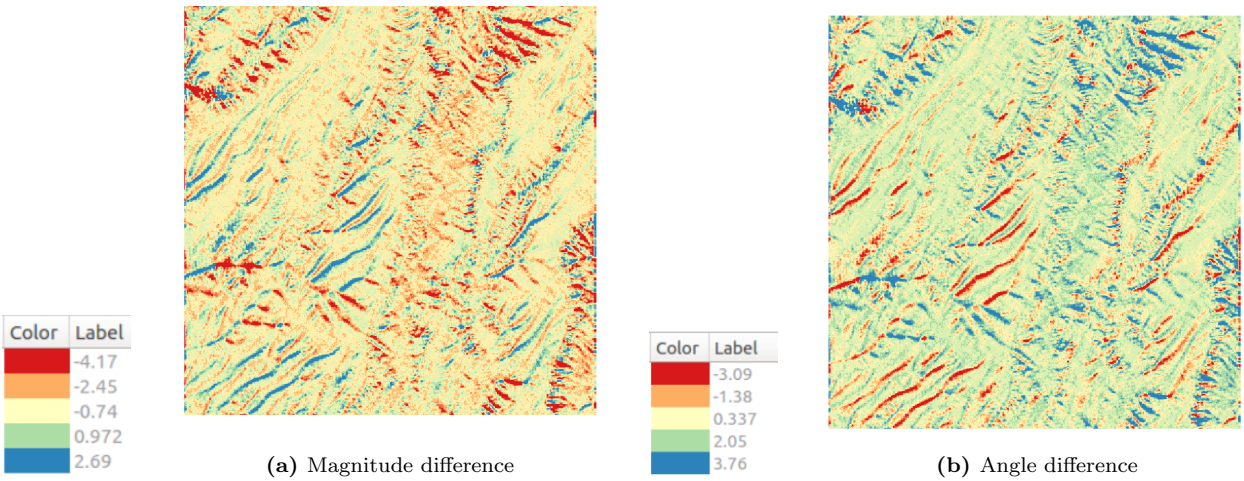


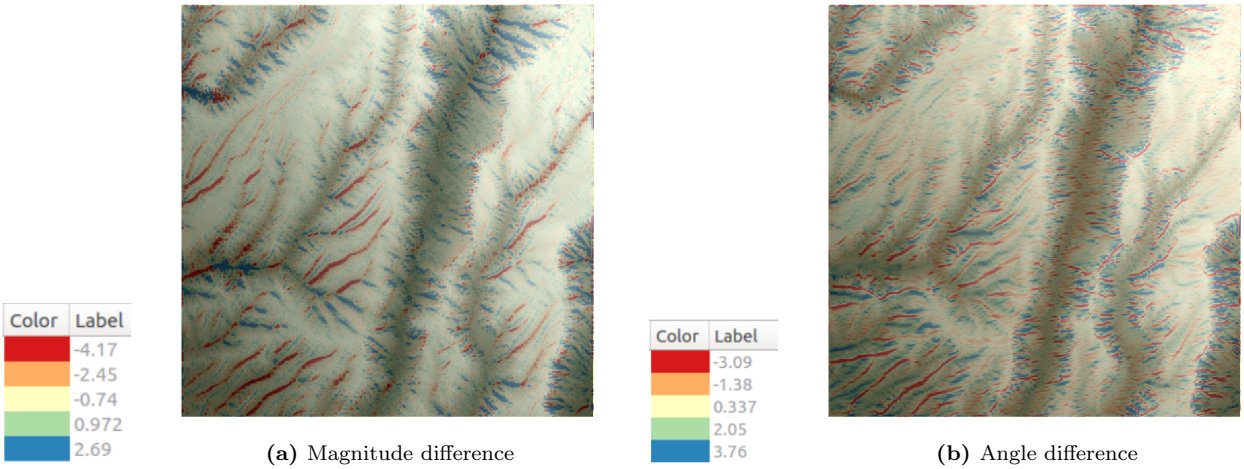
Figure 2: Color map of the topographic relief employed in Bolck B. Whiter indicates higher altitude while darker is a deeper altitude.

In Figure 3 (a) we have plotted the difference of the magnitudes of the velocity vectors (trees-brush), while in Figure 3 (b) we have plotted the relative angles indicating the direction of the wind vectors (trees-brush) computed as the difference of the angles.



(a) Magnitude difference **(b)** Angle difference

Figure 3: Differences computed from the output of the *.tiff trees* case and the *.tiff brush* case. (a) Difference in magnitude of wind velocity vectors. (b) Difference in angle for the wind velocity vectors. The colorbar of each plot is in the left of the plot.



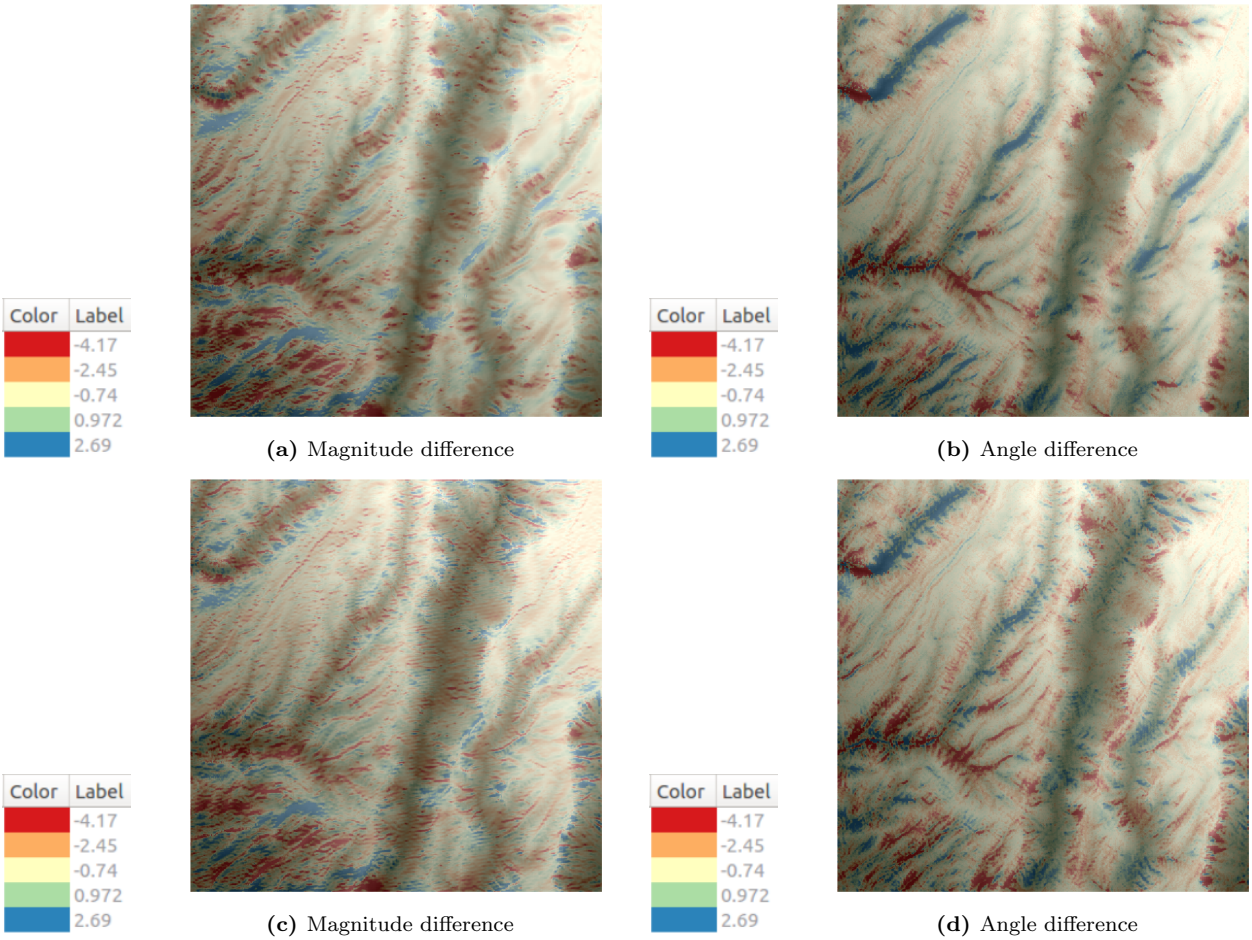
(a) Magnitude difference **(b)** Angle difference

Figure 4: Differences computed from the output of the *.tiff trees* case and the *.tiff brush* case, now with the topographic relief as a transparent layer over them. (a) Difference in magnitude of wind velocity vectors. (b) Difference in angle for the wind velocity vectors. The colorbar of each plot is in the left of the plot.

We can see most clearly in Figure 4, where we superpose the elevation map to the difference map, that in the abruptest altitude variation regions (the regions in which there is a fastest change of orography from white to dark) is precisely where the magnitude and angle differences are most notable (both positive or negative differences, red and blue) between varying vegetation modes. Contrarily, in the most orographicaly homogeneous regions (plain regions), both in high or low elevations, the differences are roughly negligible (yellow colors). This suggests that the main influence of the vegetation in the resulting wind field is in how the wind behaves in slopes (in regions where the magnitude of the elevation gradient is maximum in magnitude). This makes intuitive sense, since it is in these regions, in sudden elevation changes, in slopes, where the wind will "collide" with the stuff on the ground (like the vegetation).

6. ¿Varía el tiempo de ejecución con respecto al de la ejecución de los mapas tif? ¿En qué parte de la ejecución se observa un cambio en el tiempo de ejecución? ¿A qué pensáis que es debido?

In Table 2 we can also observe that there is a clear difference between tiff and lcp execution time. The difference in time is due to the reduction in the solver time as it decreases for the lcp. This suggests that the explicit indication of the vegetation type (as done in the tif) introduces extra computations in the resolution equations, which makes the computation more costly in time, in comparison with the implicit indication of the vegetation as a layer of the map (as done in the lcp). This must be noted, could be surprising since the degree of detail for the vegetation can be greater for an lcp file.



(a) Magnitude difference

(b) Angle difference

(c) Magnitude difference

(d) Angle difference

Figure 5: Differences computed from the output of the *.tiff trees* case and the *.tiff brush* case with the *.lcp* case, with the topographic relief as a transparent layer over them. (a) Difference in magnitude of wind velocity vectors. (b) Difference in angle for the wind velocity vectors. The colorbar of each plot is in the left of the plot.

When it comes to the differences of the computed wind fields plotted in Figure 5, we see the most significant differences are situated in the narrow river fissures when it comes to the angle differences and in the closest cliffs to these fissures for the magnitude of the wind. This comes together with the conclusion we took previously that the wind interacts most with the surface in the orographic slopes, just that now we also see differences in the lowest homogeneous elevations. This last difference could be due to the fact that the *.lcp* file has a varying type or detail for the vegetation at each point of the map.

Block C

4. ¿Se modifica el tiempo de ejecución al modificar la resolución? Representa gráficamente como varía el tiempo para cada resolución. ¿Qué dependencia se observa del tiempo de ejecución con respecto de la resolución del mapa?

In Table 3 we can see clear evidences that the variation of the execution time has a dependence on the resolution of the simulation. When using a resolution of 50 m, so there are less points in the mesh (bigger increment for the same map size) the total execution time is decreased up to 5 times compared with the usage of a finer resolution of 20 m (more points in the mesh).

Table 3: Execution times for several simulation cases of Block C. Times are in seconds.

Resolution	Angle (°)	Speed (mph) (mph)	Mesh time	Initialization time	Equation build t	Solver time	Output write	Total time
50	90	30	0.18	0.77	8.39	28.51	3.08	44.01
50	0	30	0.18	0.78	8.52	27.00	3.10	42.63
20	90	30	1.52	4.96	52.67	176.81	18.02	273.47
20	0	30	1.68	5.61	54.61	182.15	18.22	281.55
20	0	10	1.42	4.65	48.19	168.10	16.28	256.52
50	0	10	0.17	0.74	7.80	26.28	2.70	40.50

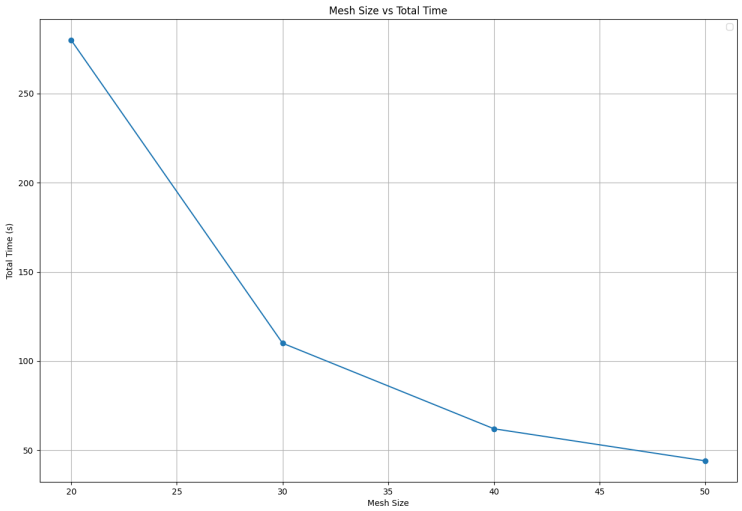
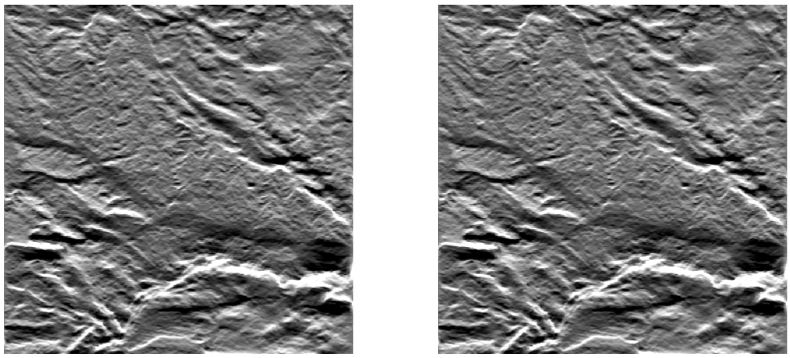


Figure 6: Execution time variation for each resolution

5. Utilizando Qgis, buscad las diferencias que se observan entre los mapas obtenidos para el módulo de la velocidad del viento y de la dirección del viento. ¿Dónde se encuentran las diferencias y por qué?

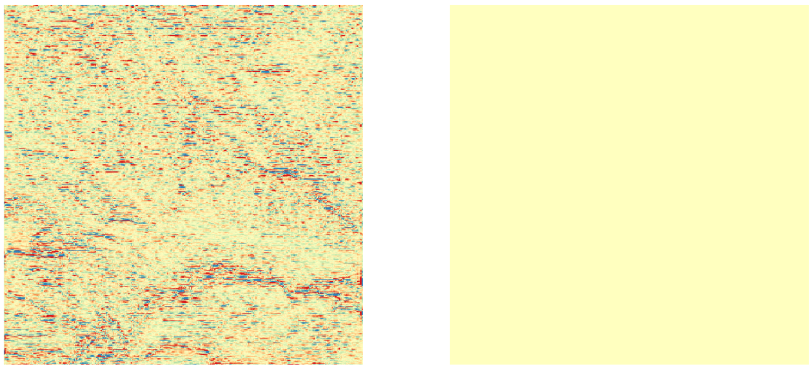
On the one hand we can see in Figure 7 that the simulations performed with a wind direction (say 0 rad) and the opposed direction (say π rad) have qualitatively similar wind maps. This was expected since the same orographic variations will be found by the wind (the same complications). Thus, factors like the resolution of the mesh will influence them in a same standing.



(a) 0 rad **(b)** π rad

Figure 7: Simulations computed with the same exact parameters of wind speed (30 mph) and resolution (50 mesh increment size) and same orograpgy. The difference is just the incidence angle of the wind. In (a) it is 0 radians with respect to the width axis, while (b) is π radians.

Therefore first of all we conclude that comparing only orthogonal angles shall be enough to find an interesting case to analyse the effect of the mesh resolution. In our case we will study 0 and $\pi/2$ radians. For each of them, we fix the wind speed to the maximum of 30 mph (presumably providing us the most clear complication detection, since it is the case where numerical problems could be most noticeable) and simulate with the finest and grossest grids (20 and 50 respectively). We then take and plot the differences between the resolutions within each direction case (0 and $\pi/2$ rad). This is plotted in Figure 8. We can see that there is a direction in which the wind finds more complications (the simulation is more prone to numerical problems), as evidenced by the big difference in the results when making the simulation with a poorer grid. That is, the simulation is more sensitive if the wind impinges in a direction with abrupt changes (more prone to numerical instabilities). This is the case for the angle of 0 rad. Thus presumably it is also the case for its opposed angle: π rad.



(a) 0 rad wind direction, difference between extreme resolutions (mesh_resolution = 20 and 50)

(b) $\pi/2$ rad wind direction, difference between extreme resolutions (mesh_resolution = 20 and 50)

Figure 8: Simulations performed with a wind speed of 30 mph, and wind direction 0 rad (a) and $\pi/2$ (b). The plots show differences between extreme resolutions (mesh_resolution = 20 and 50). Both plots are in the same color range. The yellow color indicates values close to 0. Clearly (a) shows a more notable variation.

Now that we have found the least trivial case (most sensitive to the resolution of the simulation), we proceed to compute its wind field for two extreme wind speeds and resolutions, and proceed to pay attention to the simulation times.

These are summarized in Table 3. In particular, we can see that as the mesh increment is made smaller the computation takes more time, which was expected for the matrices will get bigger. Additionally, we can also see that the variation of the speed and the angle do not affect the computation times significantly. This means that we will not take care of their particular values for the study case of the following section. However it is now clear that the direction of the wind does condition the error sensitivity, so it will be important to leave unchanged the fixed reference direction.

Study Case - Group 6

As we have seen in the previous sections changing the wind velocity, angle, or the type of vegetation, produce no changes in the execution time. Also we have seen that using an .lpc file is faster than using the .tif file and therefore we will be using the .lpc file that is provided (we also chose to use .lpc because the degree of detail of the vegetation can be higher in a .lpc -as its embedded vegetation layer-, making a more realistic simulation). Therefore the parameters we can change to control the execution time are: the number of threads, the mesh size and the output resolution.[1]

Table 4: Execution times for several simulation cases of the Study Case. Times are in seconds. Fixed parameters are wind speed 24.3 mph, wind height 20 ft, input direction 166°, output wind height 20 ft, out resolution 30 m.

Threads	Resolution (m)	Mesh time	Initialization build time	Equation time	Solver write t	Output time	Total
1	30	1.50	4.96	49.21	71.27	16.15	161.03
2	30	1.12	4.72	27.30	61.72	12.28	117.06
4	30	0.92	4.64	16.82	56.10	11.74	96.46
8	30	0.85	4.53	11.28	53.87	11.73	87.26
8	40	0.23	2.57	6.53	31.03	6.48	49.71
8	50	0.16	1.67	4.12	21.30	4.36	33.37

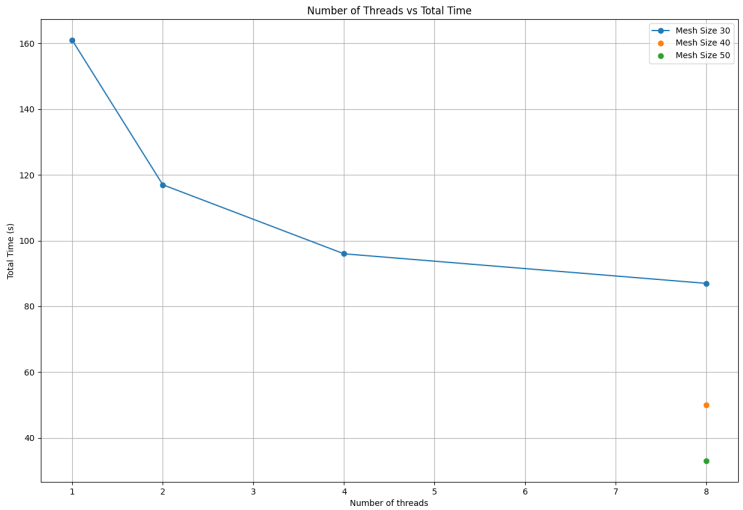


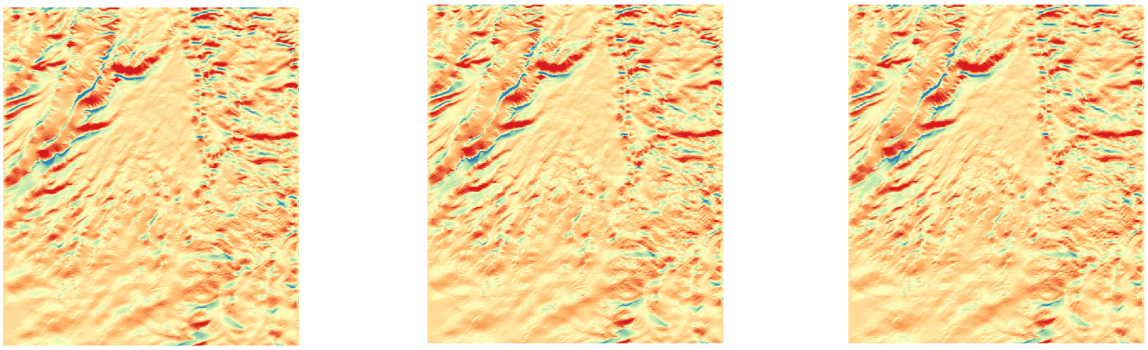
Figure 9: Total execution time in seconds, against the employed number of threads for different mesh sizes (see color legend). The numerical results are gathered in Table 4. Note the time reduction for an increasing thread number is monotonous, yet it presents an asymptote making almost pointless the increase in threads from 4 to 8. For the increase of the mesh increment there is also a clear time reduction.

As we can see in the first entry of Table 4, we have executed our file with the default parameters (1 thread, wind speed 24.3 mph, mesh resolution 30 m, wind height 20 ft, input direction 166°, output wind height 20 ft, out resolution 30 m) and got a time of 161.03 seconds as a reference. Our objective is to reduce the execution time until one third of this reference time (reduce it below 53.7 seconds) without significant result-quality loss.

[1]Technically, we could also change the wind heights, but we decided to leave them fixed since in all our study it has been a fixed given parameter.

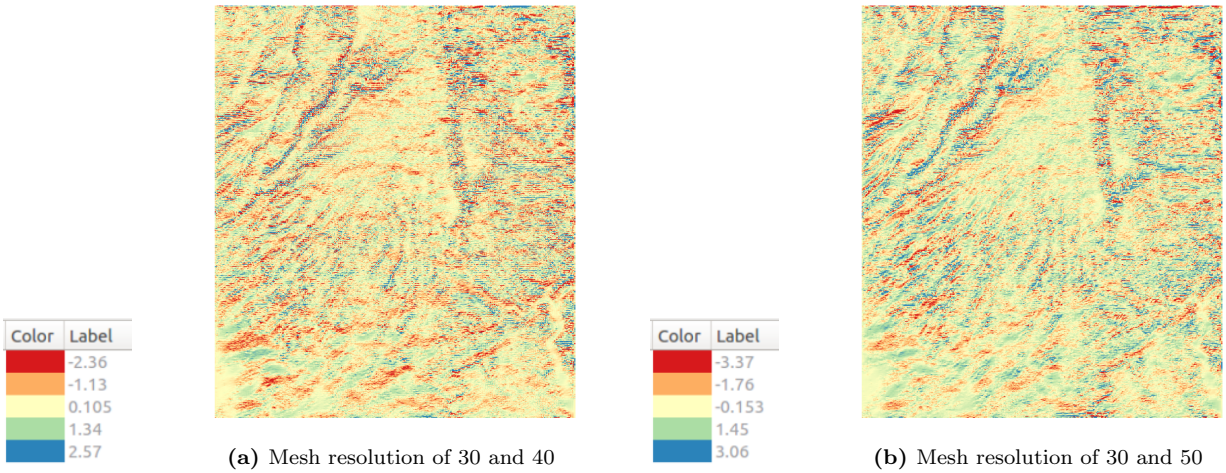
We have seen that if we only change the number of threads up to 8, the total execution time does not achieve even the half of the original time (in accordance with the speedup bound we found in Block A, which was smaller than 2 -about 1.6), as it can be seen in Table 4. So we clearly need to change another parameter, such as the mesh size. It must be noted that changing the number of threads does not worsen the quality of the results, so it is a preferred approach if we really have the required resources (we will not need to compare the results to assert the quality). However, as we have already mentioned, there is an upper bound on the speed-up we can obtain doing so, following the results of Block A. All in all, we leave the optimal number of threads to our maximum allowed 8.

Using a resolution of 40 we reduce the total execution time until 49.71 seconds. Moreover, if we use a resolution of 50, we reduce 5 times the original execution time. However, we will need to compare the resulting wind maps with the reference simulation (resolution 30) in order to be able to see if we are fine with the loss of quality of any of them.



(a) Mesh resolution of 30 **(b)** Mesh resolution of 40 **(c)** Mesh resolution of 50

Figure 10: Outputs obtained from executing the study case with different resolutions and fixing the rest of parameters to 8 threads and the ones specified in the header of Table 4.



(a) Mesh resolution of 30 and 40 **(b)** Mesh resolution of 30 and 50

Figure 11: Differences computed from comparing the outputs with different resolutions. (a) Difference between mesh_resolution of 30 and 40. (b) Difference between mesh_resolution of 30 and 50. The colorbar of each plot is in the left of the plot.

We can see in Figure 11 the difference in wind velocity between using the original mesh resolution of 30 and resolutions of 40 and 50 respectively. Paying attention to the color legend, we can see that the changes produced by the different resolutions are not very large in magnitude. For mesh resolution 40 the biggest absolute change in velocity is 2.57 mph and in the case of mesh resolution 50 the biggest change is 3.37 mph. Considering that the input velocity was 24.3 mph and the fact that most of the velocity values lay between 20 and 30 mph, as we can see in the histogram on Figure 12, the order of magnitude of the 2.57 and 3.37 mph changes is not really significant. Therefore the results obtained at mesh resolution 40 and 50 seem to have an equally valid quality.

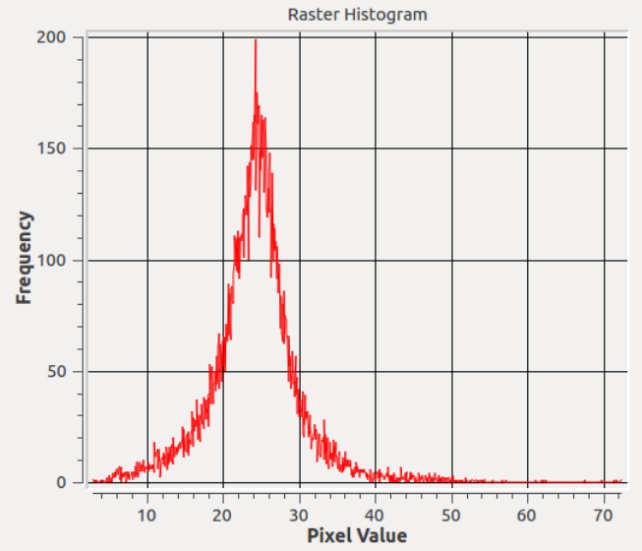


Figure 12: Histogram of the relative counts of wind speed values for a simulation using the fixed parameters of the caption of Table 4, 8 threads and mesh resolution 30.

It turns out that at mesh resolution 40 when using 8 threads, the total time to compute the results is 49 seconds, which is already less than 1/3 the time of the original run at mesh resolution 30 and 1 thread (53.7 s). Thus, we need not reduce the quality of the mesh until 50 m, and instead we propose using a mesh 40 m, for which the wind map preserves more the original results (as the max velocity difference will be 2.57 mph instead of 3.37 mph).

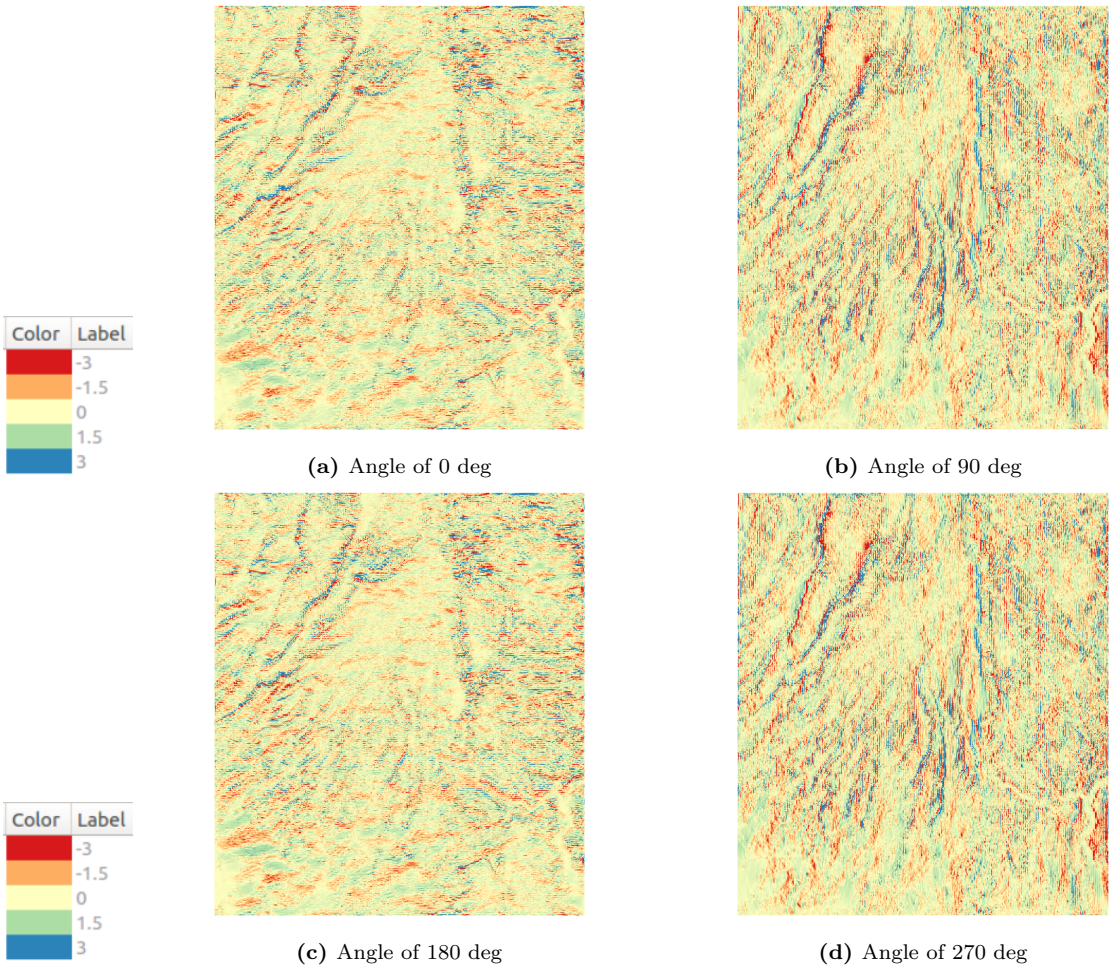
Before considering these grid conditions as the optimal ones, we need to assert that changing the initial wind speed and angle, the quality of the simulation is still adequate when compared with the reference simulation (even if we know the computation times *per se* will not change).

In particular, we can generate the results of simulations with the same mesh parameters but with a varying angle around the circle, say 0, 90, 180 and 270 degrees to see if changing the angle the quality is still preserved. We can see in Figure 13, the differences between the simulations with each angle at the optimal conditions (mesh 40) and the reference conditions (mesh 30). For all the angles, once again, the differences are bounded in absolute value by a maximum difference that is smaller than 3 mph in all cases. The same argument as before (comparing these 3 mph to the order of magnitude of the main speeds in histogram 12), would lead us to conclude the quality is roughly equivalent to the reference simulation. When it comes to the time, they all offer us less than a third of the simulation time using the reference conditions, since changing the wind speed and angle makes no significant change in the computation time as was proven in Block C.

Additionally, we also simulate a critical speed case of 50 mph (more than double the previously studied cases) for the optimal and the reference mesh conditions (mesh resolutions 40 and 30 respectively), as a benchmark for the numerical difficulties high speeds could introduce. In Figure 14, we can see the difference map between the simulation with mesh resolution 40 and the reference of 30. Again, paying attention to the legend's range, it has an absolute error bounded by a maximum error of about 6, which is not really significant if we pay attention to the orders of magnitude of the wind speeds at the histogram in Figure 12 (which was computed for the reference simulation of 24.3 mph). That is, we have doubled the wind speed and the error bound has only doubled (from about 3 mph to 6 mph), which indicates the quality loss due to the mesh resolution reduction is equivalent to the case we considered as optimal.

Thus, we conclude the conditions allowing us to answer the problem are a mesh resolution of 40 m, 8 threads, any angle and reasonable speed and the default other parameters listed in the caption of Table 4.

Finally, we could have also adjusted the output resolution (the parameter called ascii_out_resolution), for a grosser resolution will make the output generation easier. However, in order to be able to compare the results to the reference simulation, we need to be able to make a pixel-wise comparison, which suggests the usage of a constant output resolution (fixed at 30m).



(a) Angle of 0 deg

(b) Angle of 90 deg

(c) Angle of 180 deg

(d) Angle of 270 deg

Figure 13: Differences between simulations with each angle at the optimal conditions (40 mesh resolution) and the reference conditions (30 mesh resolution), each with a varying initial wind direction as indicated by the captions of the subplots.

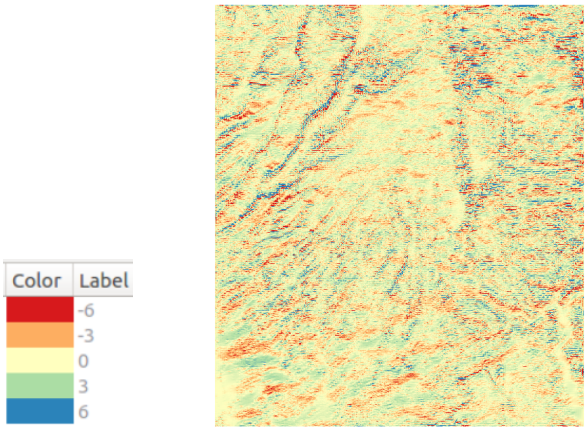


Figure 14: Simulation with high speed (50 mph) for optimal conditions (40 mesh resolution) and the reference conditions (30 mesh resolution).