TADI - Scale Space

Shubhamkumar Patel and Johan Pardo

January 26, 2022

1 2D Explicit Heat Equation - FTCS Scheme

The FTCS or Forward Time Centered Space Scheme is a finite difference method that we use for solving the heat equation.

In the 2D case it can be described as the following :

Define

$$r_x = \frac{\mu \Delta t}{\Delta x^2}, \qquad r_y = \frac{\mu \Delta t}{\Delta y^2}$$

Define undivided finite differences

$$\delta_x^2 u_{j,k}^n = u_{j-1,k}^n - 2u_{j,k}^n + u_{j+1,k}^n, \qquad \delta_y^2 u_{j,k}^n = u_{j,k-1}^n - 2u_{j,k}^n + u_{j,k+1}^n$$

FTCS scheme

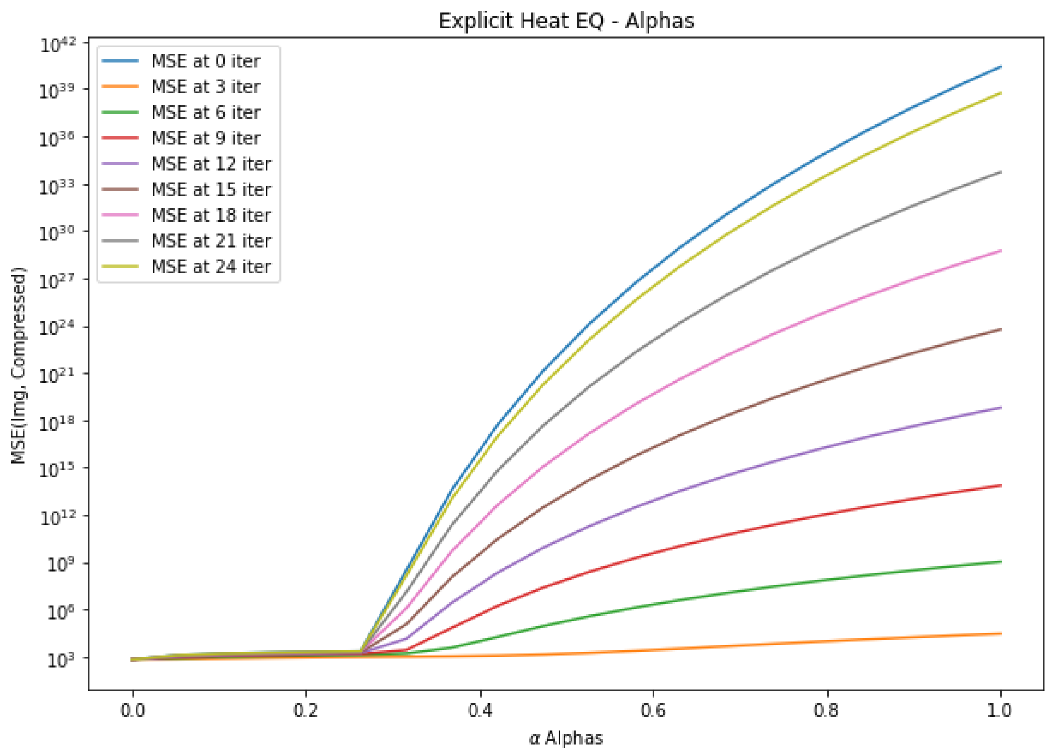
$$u_{j,k}^{n+1} = u_{j,k}^n + r_x \delta_x^2 u_{j,k}^n + r_y \delta_y^2 u_{j,k}^n$$

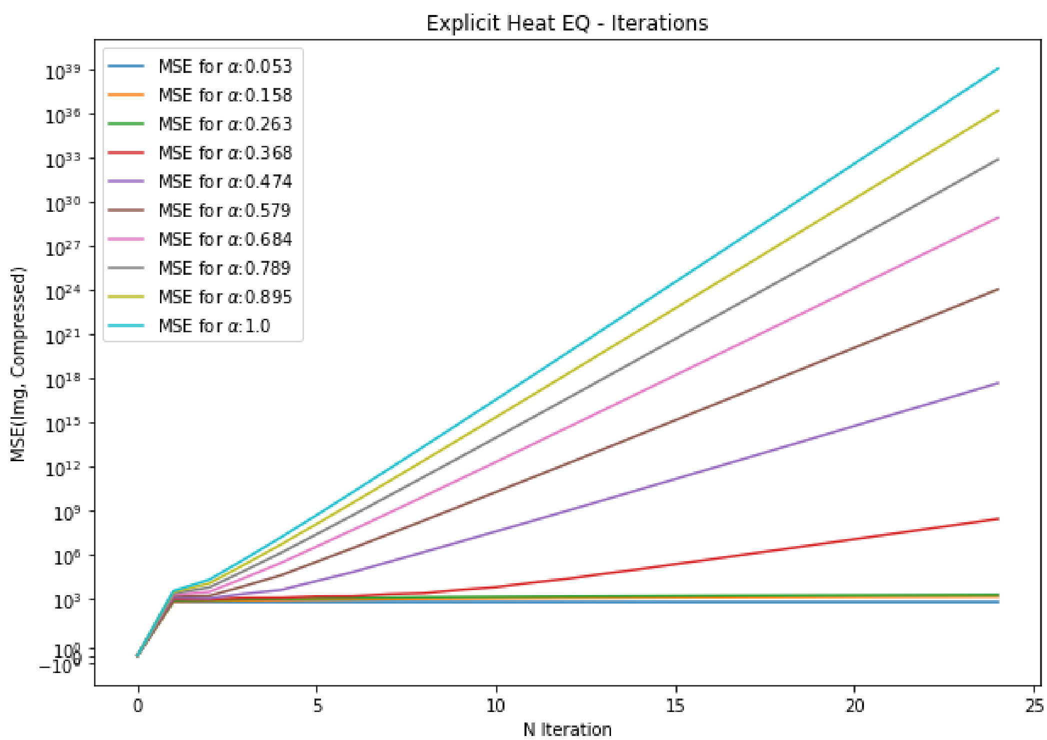
or

$$\boxed{u_{j,k}^{n+1} = (1 + r_x \delta_x^2 + r_y \delta_y^2) u_{j,k}^n}$$

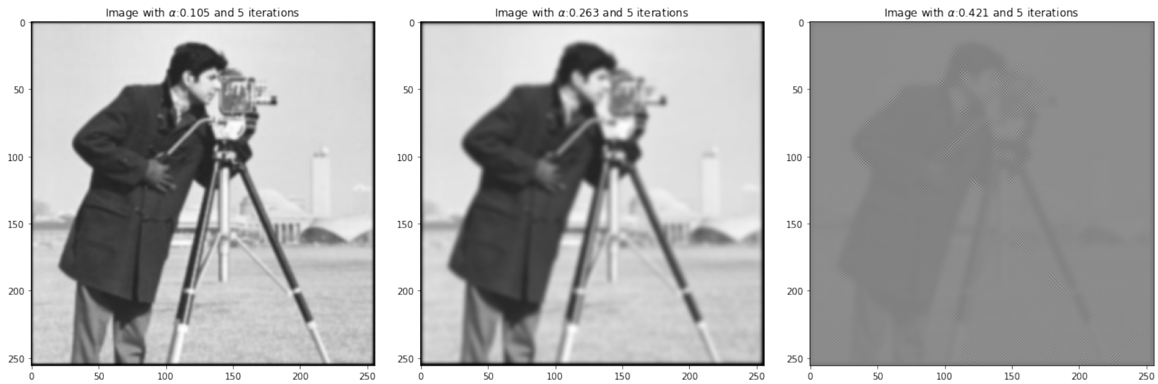
The stability of the scheme is provided by $r_x + r_y < \frac{1}{2}$.

1.1 Experimentation





In the graphs all plots with alpha smaller than CPL reach a ceiling in mse around 10e3 this indicates a clear stability zone and everywhere we see diverging behaviours.



The role of the α parameter is to set the diffusion speed, but it needs to be set in accordance to satisfy the CFL condition. In the last image the α is not set up in accordance with the CFL condition and therefore the result image tends to be gray. But the alpha is right like in the first example with α=0.1 and 5 iterations we have good a results

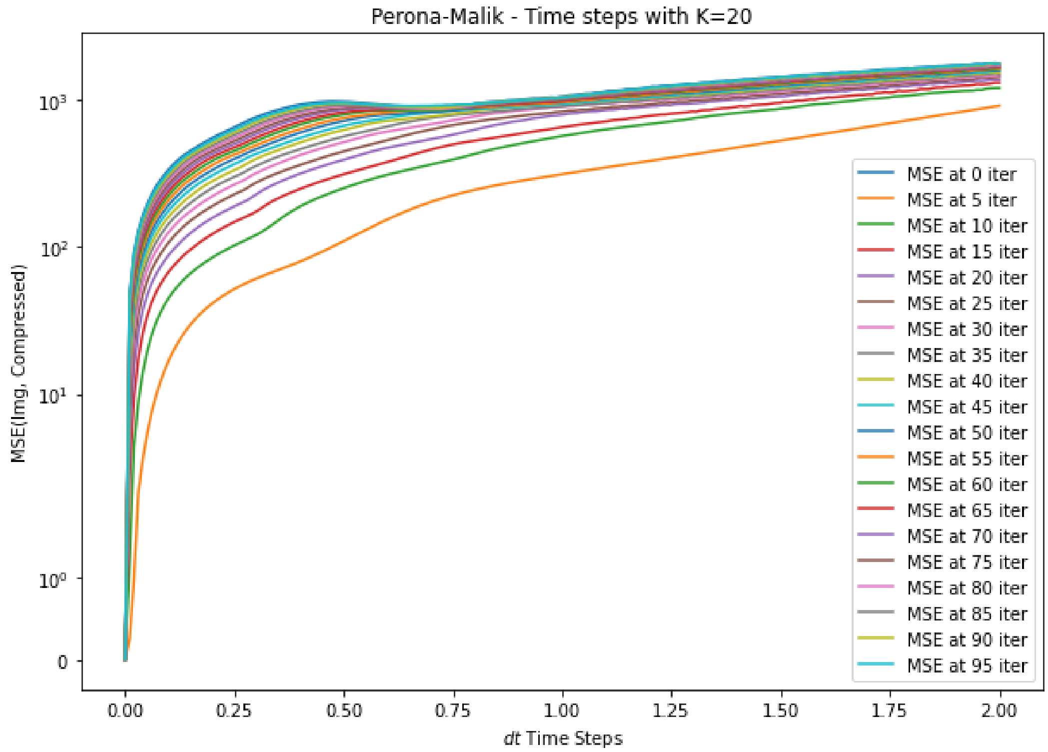
2 The Perona-Malik scheme

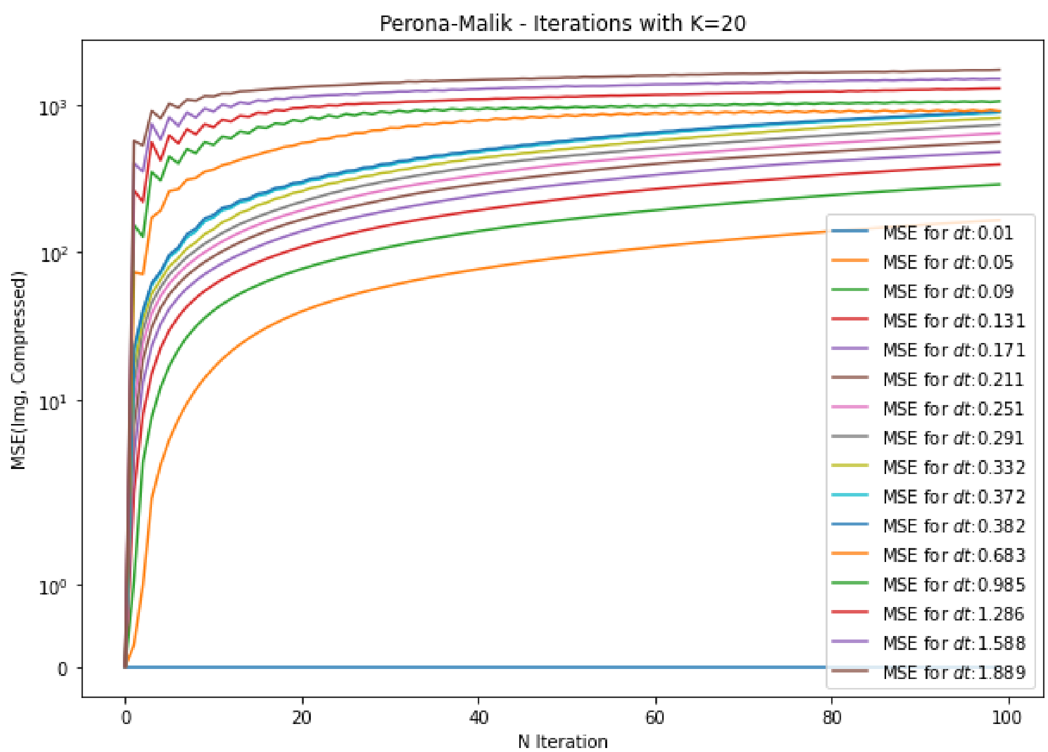
Here is the equation we are trying to implement:

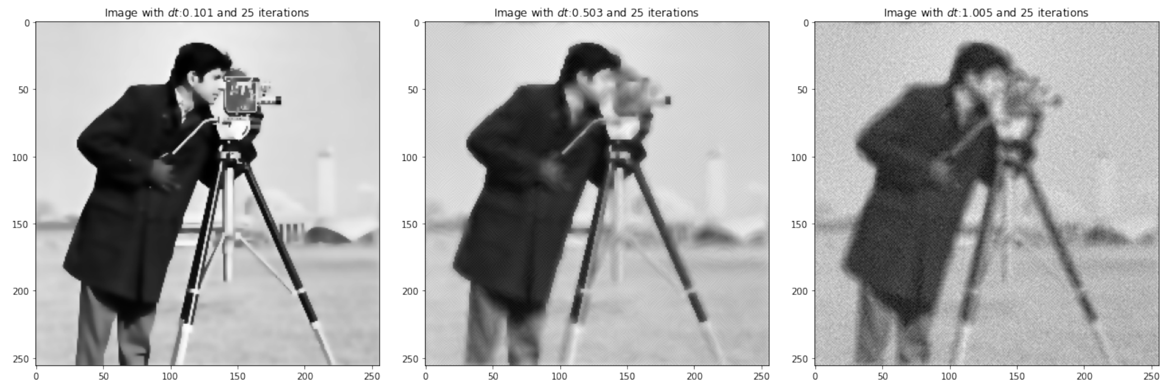
$$L_{i,j}^{k+1} = L_{i,j}^{k} + \triangle t[C_N.\nabla_N L + C_S.\nabla_S L + C_E.\nabla_E L + C_W.\nabla_W L]_{i,j}^{k}$$

The main advantage of the Perona-Malik is to have a diffusion which depend on local configuration of image values and therefore the results are more linear which we will try to show in this part.

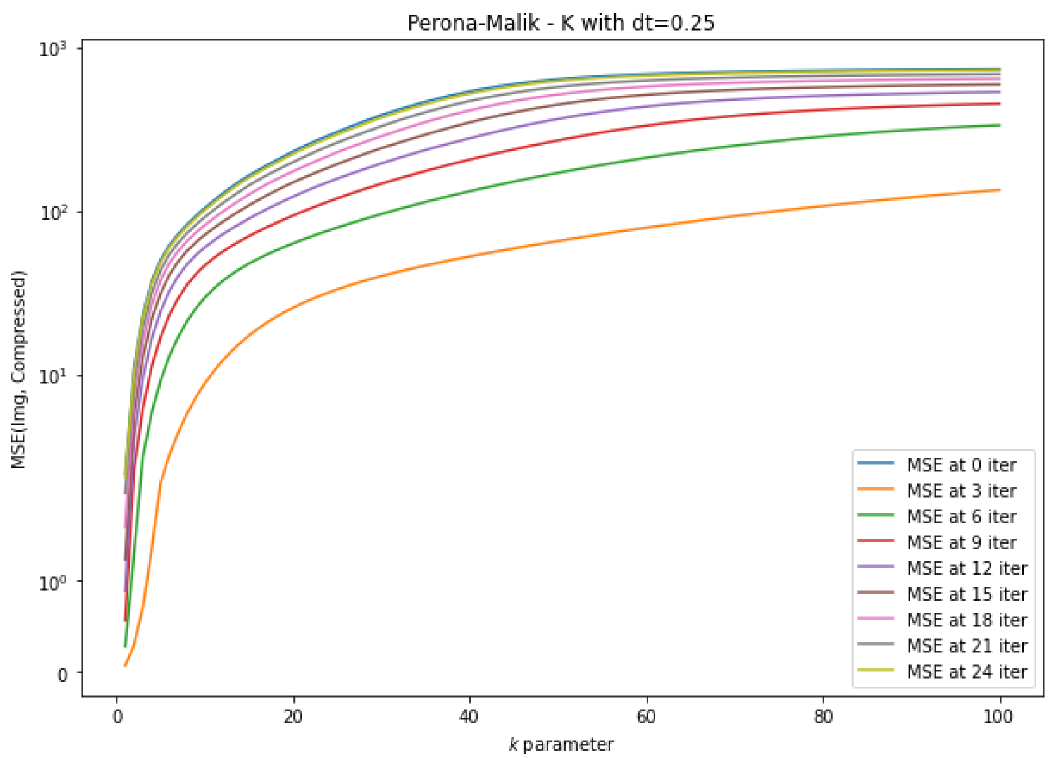
2.1 Experimentation

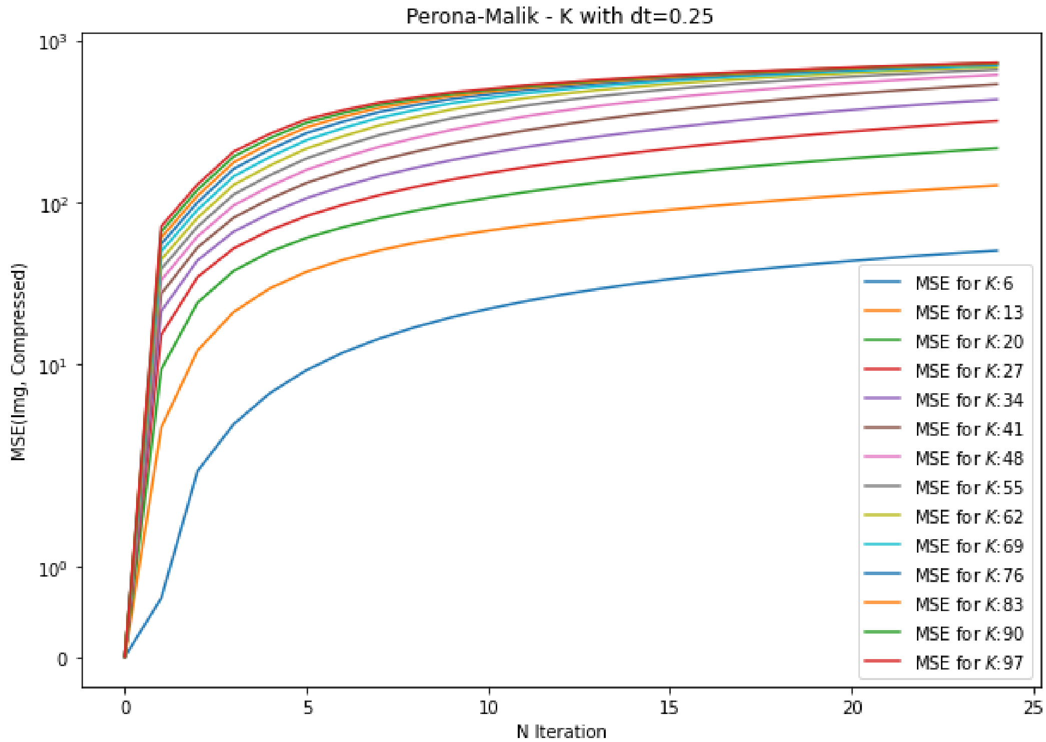


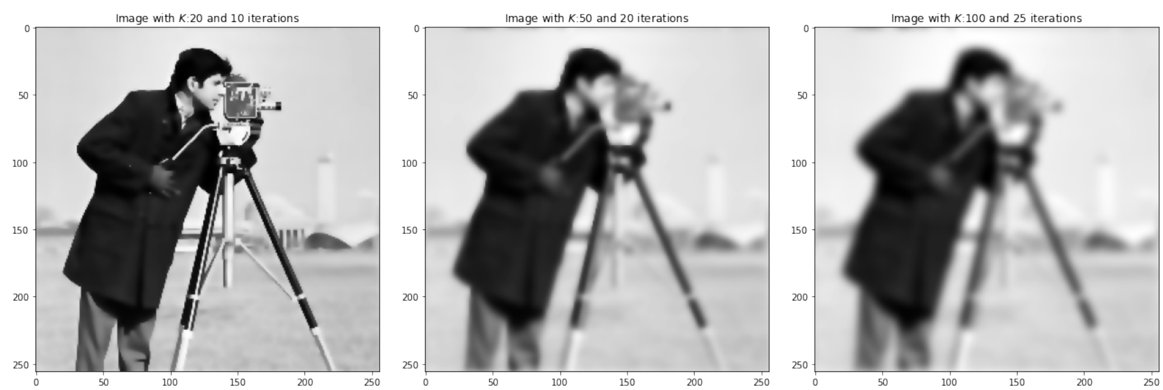




As previous algorithm the dt is used as a learning rate too low we might not learn enough, to high we might have some interferences that we can start to see in the second image. But when dt is way too high we start having some noise in image, like the third one. But when right, we have a sketch like the first image.







The K parameter is one of the parameter of the lorentz function, which for the Perona-Malik approach is a similitude weight on the neighbour of each pixel. We can see the role of this weight in the graph where when K is lower, so more similar on his neighbour, to emphasize edges, we tend to have better accuracy, due to less deviation from this threshold function. This result can be seen on the image above, with K increasing from left to right.

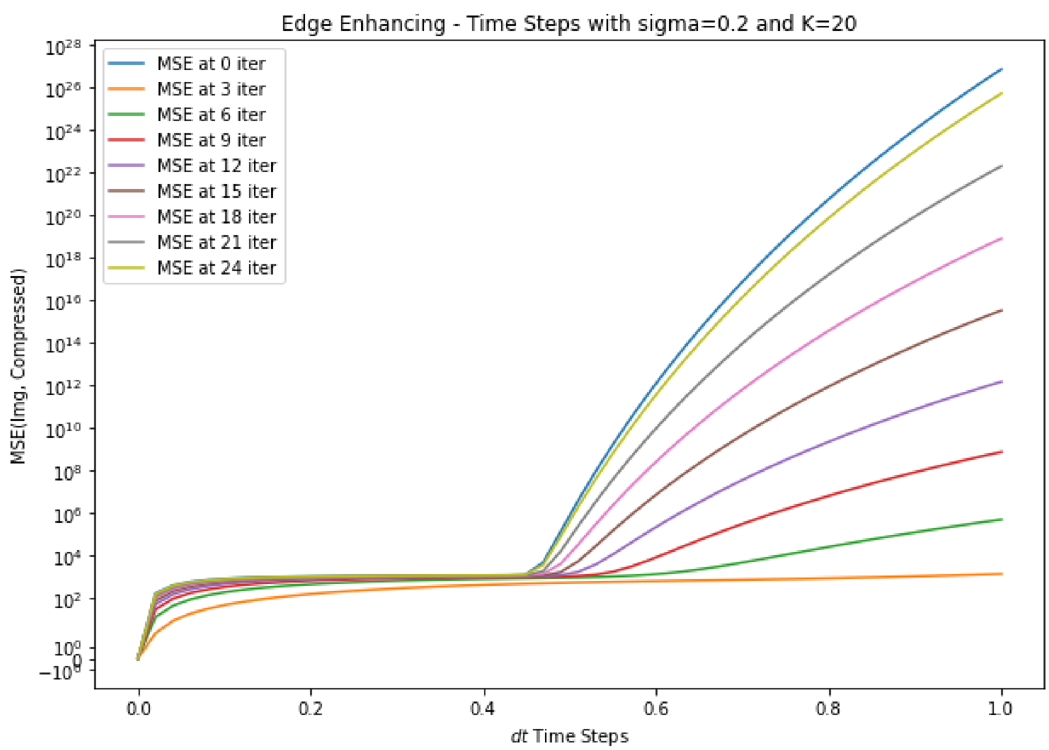
3 Edge Enhancing scheme

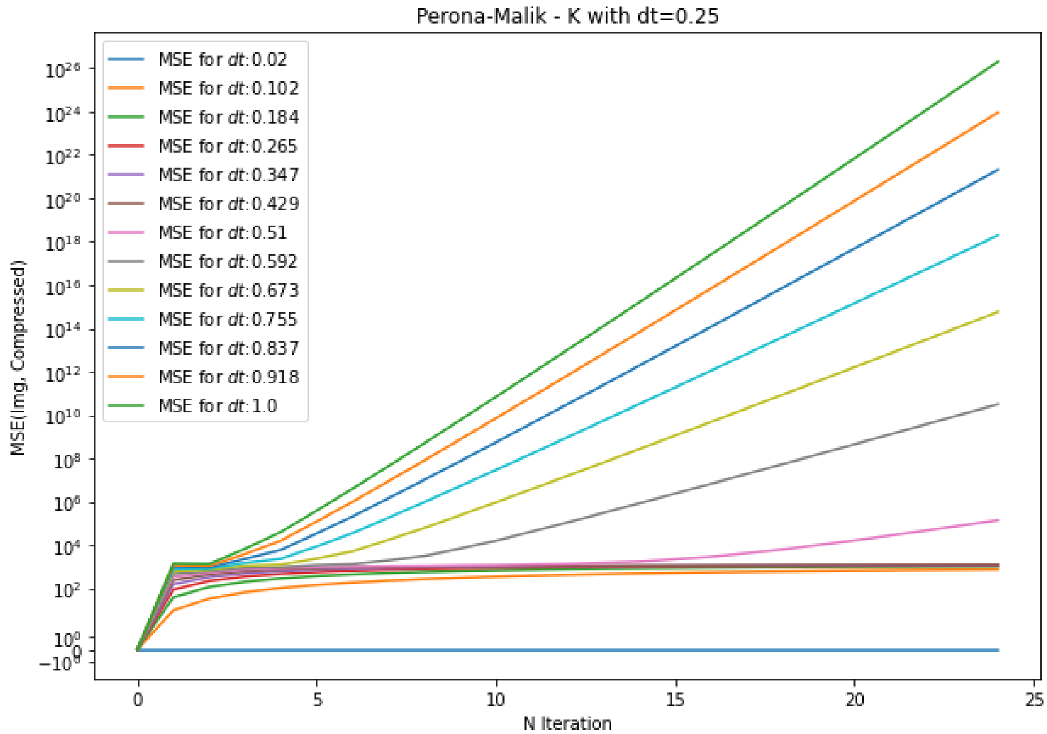
Here is the equation we are trying to implement:

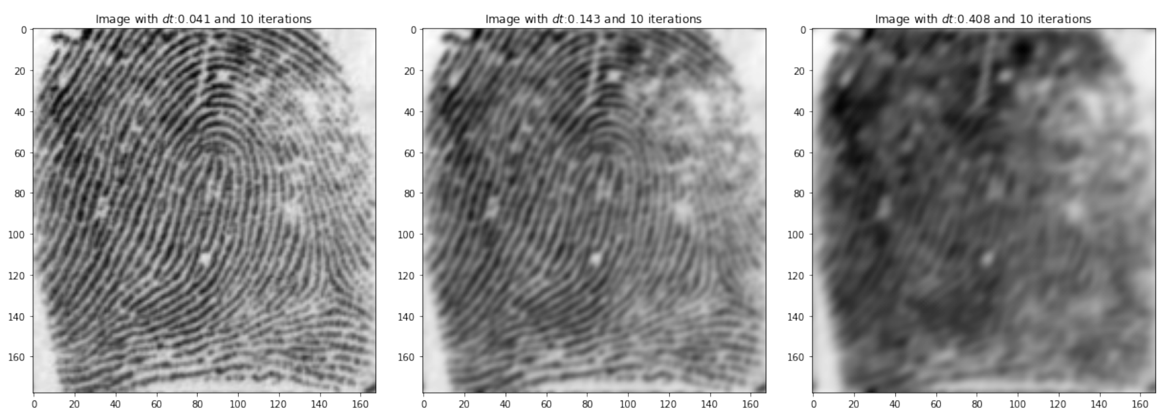
$$
\begin{aligned}
L_{i,j}^{k+1} = u_{i,j}^{k} + \triangle t \Bigg[& -\frac{b_{i-1,j}+b_{i,j+1}}{4} L_{i-1,j+1}^{k} + \frac{c_{i,j+1}+c_{i,j}}{2} L_{i,j+1}^{k} \\
& + \frac{b_{i+1,j}+b_{i,j+1}}{4} L_{i+1,j+1}^{k} + \frac{a_{i-1,j}+a_{i,j}}{2} L_{i-1,j}^{k} \\
& - \frac{a_{i-1,j}+2a_{i,j}+a_{i+1,j}+c_{i,j-1}+2c_{i,j}+c_{i,j+1}}{2} L_{i,j}^{k} \\
& + \frac{a_{i+1,j}+a_{i,j}}{2} L_{i+1,j}^{k} + \frac{b_{i-1,j}+b_{i,j-1}}{4} L_{i-1,j}^{k} \\
& + \frac{c_{i,j-1}+c_{i,j}}{2} L_{i,j-1}^{k} - \frac{b_{i+1,j}+b_{i,j-1}}{4} L_{i+1,j-1}^{k} \Bigg]
\end{aligned}
$$

The edge-enhancing method, as his name implies, smooth by taking into account the edges. This methods should better improve results by keeping edges in the final images

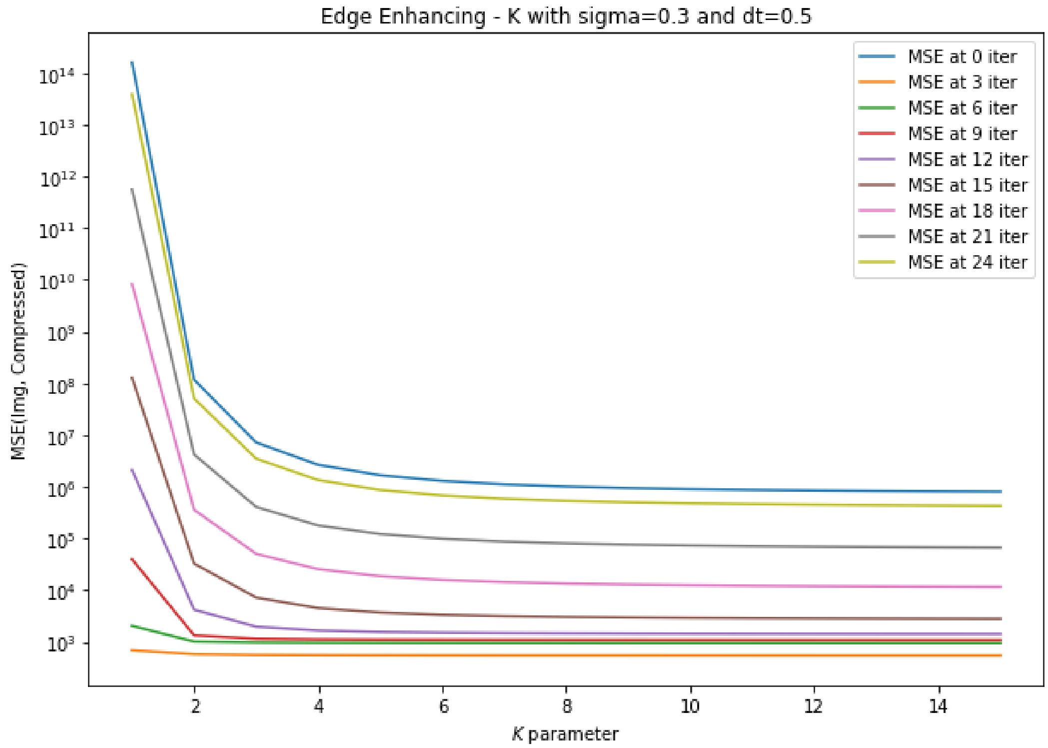
3.1 Experimentation

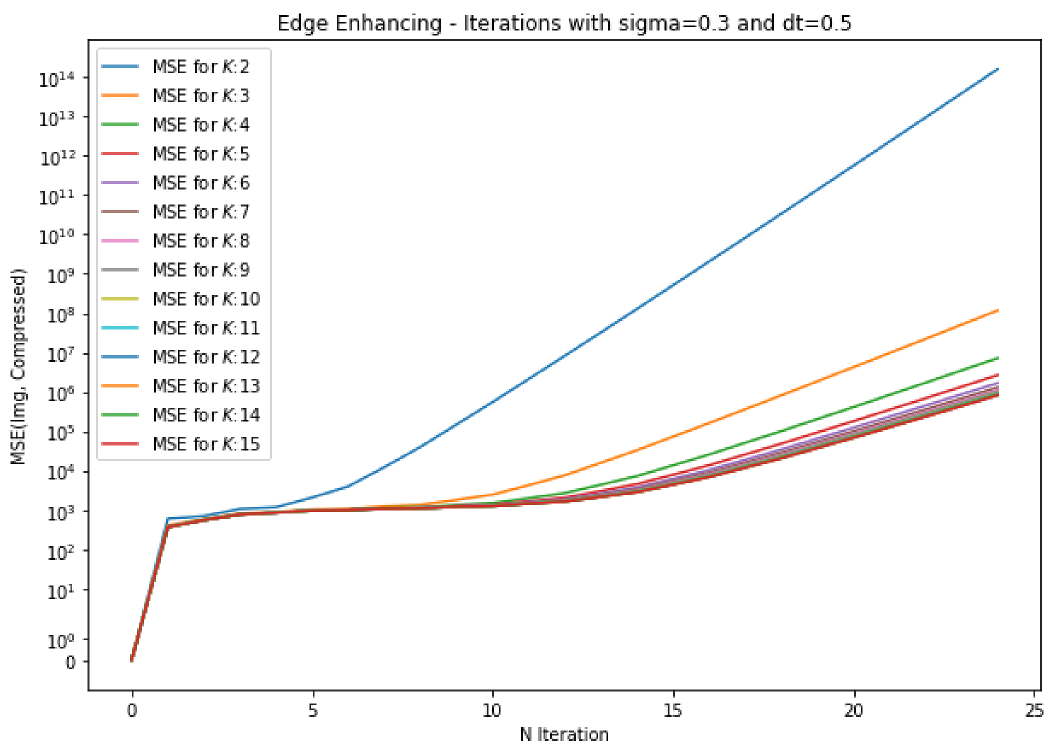


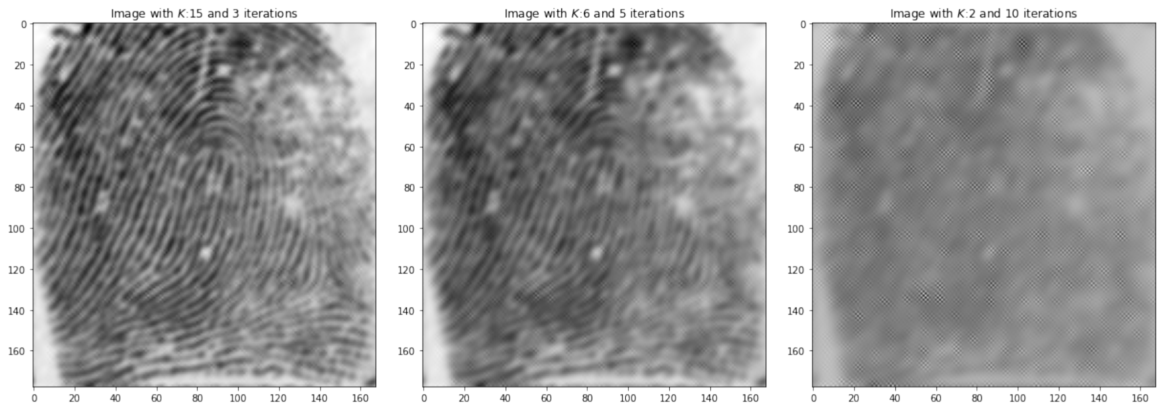




dt is used to set the time step which defines the rate at which we reach the result we want to have. So when this value is high the propagation of values in a neighborhood happens faster, this is why we get a very blurred image when dt is higher. We can see this effect in the images above, with the evolution of dt and the resulting image state.


Edge Enhancing - K with sigma=0.3 and dt=0.5
MSE at 0 iter
MSE at 3 iter
MSE at 6 iter
MSE at 9 iter
MSE at 12 iter
MSE at 15 iter
MSE at 18 iter
MSE at 21 iter
MSE at 24 iter
MSE(Img, Compressed)
K parameter


Edge Enhancing - Iterations with sigma=0.3 and dt=0.5
MSE for K:2
MSE for K:3
MSE for K:4
MSE for K:5
MSE for K:6
MSE for K:7
MSE for K:8
MSE for K:9
MSE for K:10
MSE for K:11
MSE for K:12
MSE for K:13
MSE for K:14
MSE for K:15
MSE(Img, Compressed)
N Iteration



The K is used to set up the λ_1 and λ_2 which in turn sets the weights in the a, b and c. The λ_1 term tunes the diffusion in the direction of $\nabla_\sigma L$ and λ_2 term tunes the diffusion in the orthogonal direction of $\nabla_\sigma L$, and using this terms we are tuning the diffusion along the edges. The graph shows us that the value of K is not important as long as it is not too close to 0. In the last image we see the effect of a high K where the image start having noises. This method tries to take in consideration the change of values in different directions (from the center towards the neighbours around), we can use this information for each direction to get diffusion along the edges and not where there is a brutal change in pixel values (going from a bright region to a darker one in a fingerprint).

4 Exercise 4

4.1 Implicit 2-D Heat Equation - BTCS Scheme

The FTCS or Forward Time Centered Space Scheme is a finite difference method that we use for solving the heat equation.

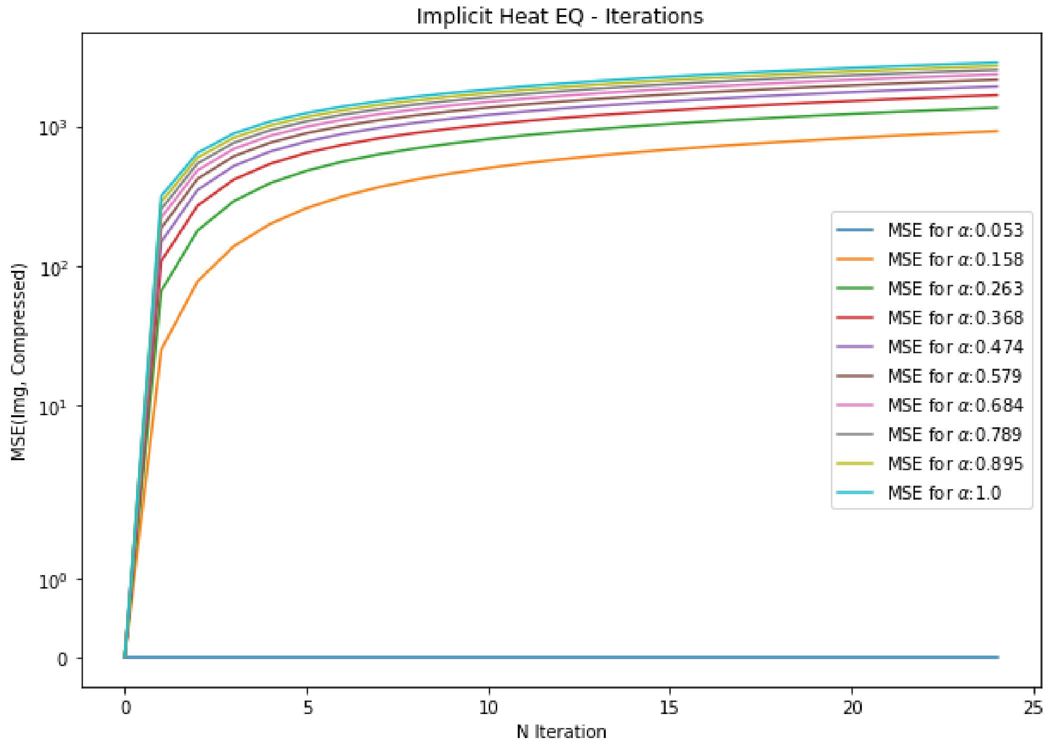
In the 2D case it can be described as the following :

$$u_{j,k}^{n+1} = u_{j,k}^{n} + r_x \delta_x^2 u_{j,k}^{n+1} + r_y \delta_y^2 u_{j,k}^{n+1}$$

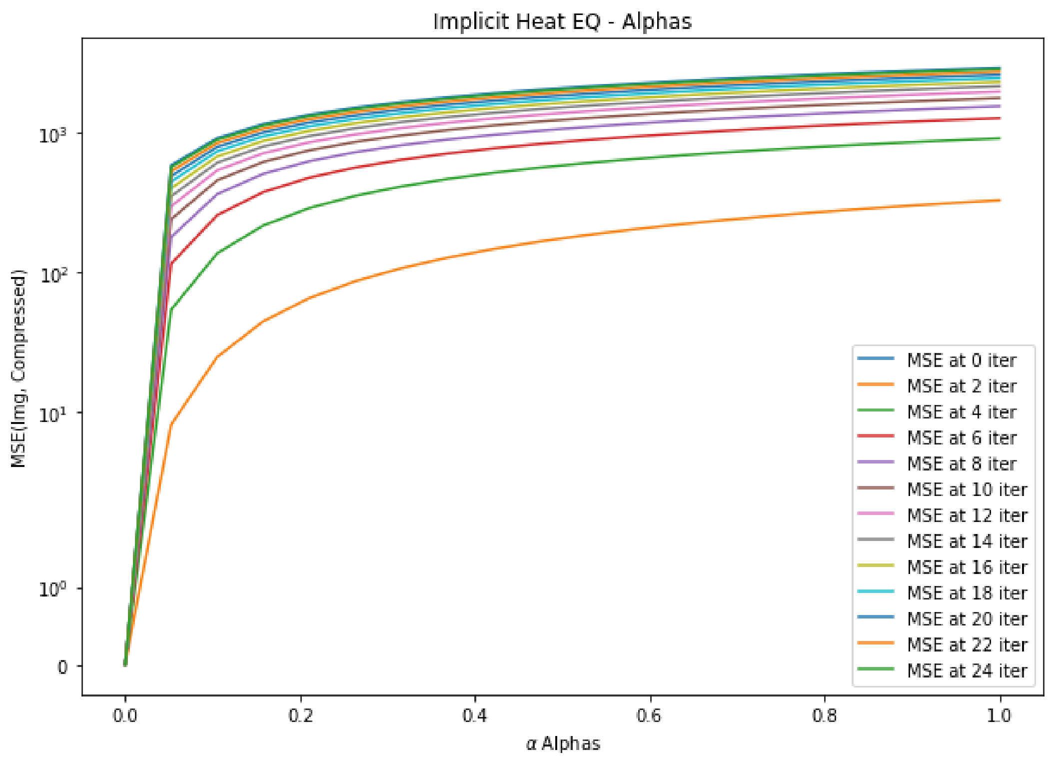
$$\boxed{(1 - r_x \delta_y^2 - r_y \delta_y^2) u_{j,k}^{n+1} = u_{j,k}^{n}}$$

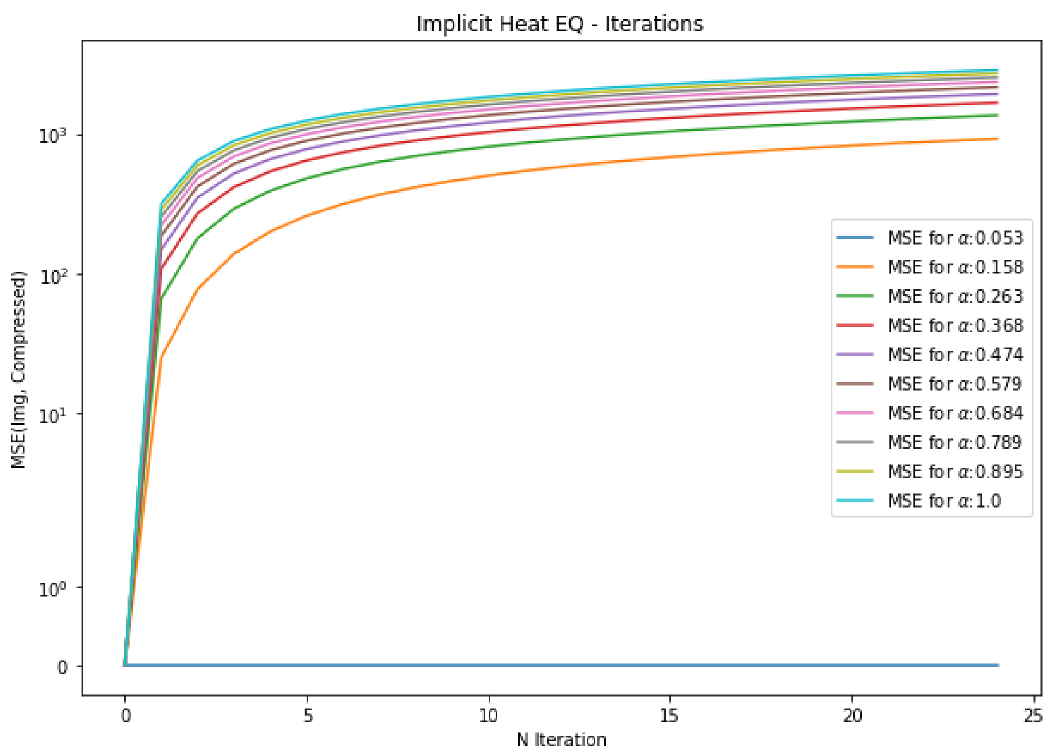
The BTCS is unconditionnally stable.

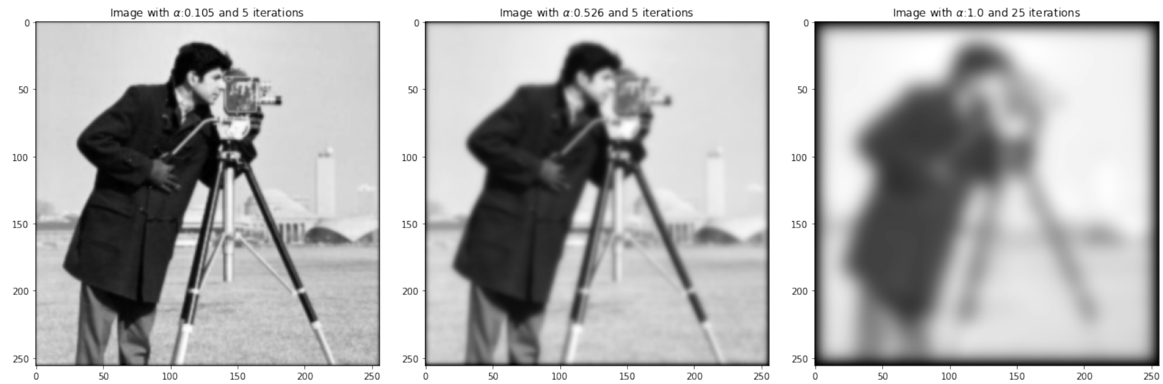
4.2 Experimentation



The iteration, as previous schemes, plays a role in divergence. As we go through the iteration the MSE increase, but this had to be expected.

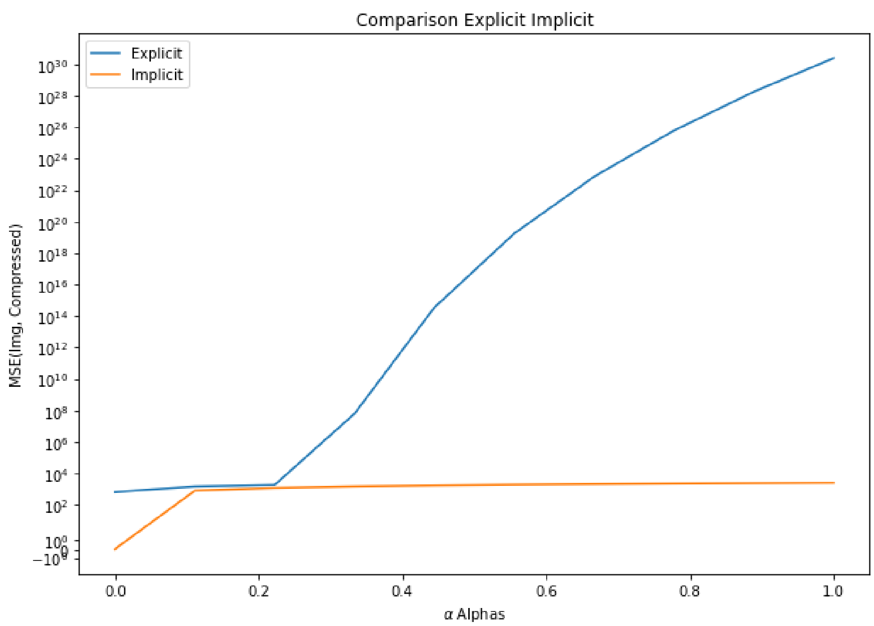

Implicit Heat EQ - Alphas
MSE(Img, Compressed)
α Alphas
MSE at 0 iter
MSE at 2 iter
MSE at 4 iter
MSE at 6 iter
MSE at 8 iter
MSE at 10 iter
MSE at 12 iter
MSE at 14 iter
MSE at 16 iter
MSE at 18 iter
MSE at 20 iter
MSE at 22 iter
MSE at 24 iter


Implicit Heat EQ - Iterations
MSE(Img, Compressed)
N Iteration
MSE for α:0.053
MSE for α:0.158
MSE for α:0.263
MSE for α:0.368
MSE for α:0.474
MSE for α:0.579
MSE for α:0.684
MSE for α:0.789
MSE for α:0.895
MSE for α:1.0



The first thing that we remark is that we darker edges as α increase, due to the unconditional stability of this scheme. Furthermore, compared to the implicit approach we don't have the divergence, which means we don't need to be constrained by the CPL condition we had in the FCTS scheme. But as we can see of the graph α also play a role in the speed at the final limit the errors reach.

4.3 Comparison



The BCTS scheme 2D Heat Equation is the implicit version of the first algorithm FCTS. From this graph we see that the on value smaller than CPL condition the implicit and explicit method behave in a similar way but as we approach higher values and surpass the CPL limit, the implict method stays stable while the other diverges This change in behaviour starts happening at around 0.3 where the error drift away at an increasing rate.

4.4 Conclusion