IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 1, Ser. 4 (Jan. – Feb. 2024), PP 14-24
www.iosrjournals.org

Toning Up Images By Smoothening Edges.

Rukia Nasimiyu Fwamba[1], Prof.James Gatoto[2]
[1]department Of Mathematics & Actuarial Science Kenyatta University,Kenya
[2]department Of Mathematics & Actuarial Science Kenyatta University,Kenya

Abstract

Background: *Under Partial Differential Equations, an image is a function* $f(x,y), X \in \mathbb{R}^2$. *These equations are used in smoothening of images. The necessity to have an excellent image quality is increasingly required in the current world. Most of the obtained images are not as smooth as we would want hence they are blurred. Use of the nonlinear diffusion equation is essential in the current day smoothening of images. This model inputs smoothness in the denoising process. This research improved the quality of images through the use of non-linear PDEs of parabolic nature i.e. the heat equation.*

Materials and Methods: *Numerical schemes of ADI (Alternating Direction Implicit method) and 2-EGSOR(2-Point Explicit Group Successive Over Relaxation) were used to solve the equations in MATLAB subject to an initial condition of a noisy image, generating various smoothened images.*

Results: *Comparatively output from ADI is very close to the original image in terms of alignment, smoothness i.e. refined texture, well outlined contours and overall detail without stair casing. .This method is characterized with a smaller residue and much time lapse. An error analysis too was carried out using Root Mean Square Method. ADI & the blocked (ADI and EGSOR) register a comparatively lower RMSE.*

Conclusion: *The most suitable algorithm for image smoothening is Alternating Direction Implicit Method.*

Keywords*: Non-linear diffusion equation, Alternating Direction Implicit method,2-point Explicit Group Over-Relaxation*

Date of Submission: 12-02-2024 Date of Acceptance: 22-02-2024

I. INTRODUCTION

If the Gamma function is used to give the derivative of $z = w^m$ then;

$$z^n = \frac{\Gamma(m+1)}{\Gamma(m+n)} w^n, m \geq n, n \in \mathbb{N} \tag{1.1}$$

Consider a function $h(x)$, its Fourier transform,

$$\hat{h}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(\alpha) e^{-ix\alpha} d\alpha \tag{1.2}$$

Whose inverse,

$$h(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{h}(\alpha) e^{-ix\alpha} d\alpha \tag{1.3}$$

Integrating by parts,

$$h^n(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (\mathrm{i}\alpha)^n \hat{h}(x)(\alpha) e^{-ix\alpha} d\alpha$$

(1.4)

The Description of $\frac{1}{2}$ fractional integral of a function $h(t)$ can be given by,

$$P = \int_0^y \left(y-t\right)^{-\frac{1}{2}} g(t)dt = \sqrt{\pi} \int_0^y \frac{1}{\Gamma(\frac{1}{2})} (y-t)^{-\frac{1}{2}} g(t)dt \tag{1.5}$$

P , is the velocity of a sliding mass,

$$P = \sqrt{\pi}\,\frac{d^{-\frac{1}{2}}g(x)}{dt^{-\frac{1}{2}}}$$

A look at the Fitzhugh-Nagumo equation for a dynamical system,

$$\left.\begin{aligned}\frac{du}{dt} &= d(-u(u-1)(u-e)-\eta+I_{ext}\\ \frac{d\eta}{dt} &= u-f\eta\end{aligned}\right\}$$

(1.6)

u,layer potential

η , resurgence variable

d & f , scaling factors

I_{ext}, stimulant current

e , nonstable balance that is coherent to the limit between electrical silence and firing

When I_{ext} external stimulus (like fluctuations in the weather , time of the day i.e. at night a photo can't be as clear as during the day) goes beyond a certain value, the system will experience a characteristic excursion in phase space, before u and η relax back.

This research smoothened noisy images producing desirable output.

II. MATERIALS AND METHODS

The non-linear heat equation that was solved in this research was of the form;

$$\frac{\partial u}{\partial t} = c_v\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+c_f u(1-u)(u-b)$$

(2.1)

$$u_x(0,y,t) = u_x(p,y,t) = u_y(x,0,t) = u_y(x,q,t) = 0$$

$$u(x,y,0) = \operatorname{Im} g(x,y)$$

p,q are image resolution i.e the image is of p pixels by q pixels , these pixels have a relationship with the spatial discretization i.e

$$i = \frac{x_i}{\Delta x}, i = 0,1,2,3......................., p$$

$$j = \frac{y_j}{\Delta y}, j = 0,1,2,3....................., q$$

The initial state is attributed to the input image, $u(x,y,0) = \operatorname{Im} g(x,y)$

The pixel flux c_v determines how vigorous the diffusive property of the model is and allows one to select the degree to which the image is denoised. c_f on the other hand tweaks the aggressiveness of the diffusion process. The direction assumed by this 'force' is based on the pixel's concentration in a given region versus the threshold parameter b .The cubic nature of the source term helps to stabilize pixels very quickly.

Methodology

Numerical approaches

Alternating Direction Implicit (ADI) Method

The idea of the ADI-method (alternating direction implicit) is to alternate direction and thus solve two one-dimensional problem at each time step. The first step keeps y-direction fixed:

$$\frac{u_{i,j}^{n+\frac{1}{2}} - u_{i,j}^{n}}{\Delta t/2} = DC_v\left(\frac{u_{i+1,j}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2}\right) + C_f u_{i,j}^n (1-u_{i,j}^n)(u_{i,j}^n - b)$$

(2.2)

In the second step x-direction is fixed:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n+\frac{1}{2}}}{\Delta t/2} = DC_v\left(\frac{u_{i+1,j}^{n+\frac{1}{2}} - 2u_{i,j}^{n+\frac{1}{2}} + u_{i-1,j}^{n+\frac{1}{2}}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2}\right) + C_f u_{i,j}^n (1-u_{i,j}^n)(u_{i,j}^n - b)$$

(2.3)

(2.2) & (2.3) can be written in tridiagonal form,

$$\eta = \frac{DC_v \Delta t}{2\Delta x^2}, \varphi = \frac{DC_v \Delta t}{2\Delta y^2}$$

(2.4)

Resulting in;

$$\eta u_{i+1,j}^{n+\frac{1}{2}} + (1+2\eta) u_{i,j}^{n+\frac{1}{2}} - \eta u_{i-1,j}^{n+\frac{1}{2}} = \varphi u_{i,j+1}^{n} + (1-2\varphi) u_{i,j}^{n} + \varphi u_{i,j-1}^{n} + C_f u_{i,j}^n (1-u_{i,j}^n)(u_{i,j}^n - b)$$

$$-\varphi u_{i,j+1}^{n+1} + (1+2\varphi) u_{i,j}^{n+1} - \varphi u_{i,j-1}^{n+1} = \eta u_{i+1,j}^{n+\frac{1}{2}} + (1-2\eta) u_{i,j}^{n+\frac{1}{2}} + \eta u_{i+1,j}^{n+\frac{1}{2}} + C_f u_{i,j}^n (1-u_{i,j}^n)(u_{i,j}^n - b)$$

(2.5)

Two-point explicit group successive over-relaxation (2-egsor)

This method can be formulated for equation (4.1.1) as follows;

$$u_{i,j}^{n+1} = \frac{1}{\alpha}\left\{u_{i,j}^{n} + C_v (u_{i,j+1}^{n} + u_{i,j-1}^{n+1} + u_{i+1,j}^{n} + u_{i-1,j}^{n+1})\right\} + C_f u_{i,j}^n (1-u_{i,j}^n)(u_{i,j}^n - b)$$

(2.6)

$$u_{i+1,j}^{n+1} = \frac{1}{\alpha_1}\left\{u_{i+1,j}^{n} + C_v (u_{i+1,j+1}^{k} + u_{i+1,j-1}^{k+1} + u_{i+2,j}^{k} + u_{i,j}^{k+1})\right\} + C_f u_{i,j}^n (1-u_{i,j}^n)(u_{i,j}^n - b) \qquad (2.7)$$

The equations (2.6) and (2.7) above can be formulated into matrix form which constructed as:

$$\begin{pmatrix} \alpha & C_v \\ C_v & \alpha_1 \end{pmatrix}\begin{pmatrix} u_{i,j}^{n+1} \\ u_{i+1,j}^{n+1} \end{pmatrix} = \begin{pmatrix} u_{i,j}^{n} + C_v (u_{i,j+1}^{n} + u_{i,j-1}^{n+1} + u_{i-1,j}^{n+1}) \\ u_{i+1,j}^{n} + C_v (u_{i+1,j+1}^{n} + u_{i+1,j-1}^{n+1} + u_{i+2,j}^{n}) \end{pmatrix} + C_f u_{i,j}^n (1-u_{i,j}^n)(u_{i,j}^n - b)$$

(2.8)

Where;

$$\alpha = 1 + \tau C_v$$

$$\tau = \frac{\Delta t}{h^2}, h = \Delta x = \Delta y$$

$\alpha_1 = \frac{\lambda}{\alpha}, \lambda$ is a weighted parameter

Lsqcurvefit(solve nonlinear curve-fitting/data-fitting)

This is a solver that uses nonlinear least squares.From this solver that is inbuilt in MATLAB,coefficients of x that solve the problem (2.9) are established ;

$$\min_x \|F(x, xdata) - ydata\|_2^2 = \min_x \sum_i (F(x, xdata_i) - ydata_i)^2$$
(2.9)

Given input data $xdata$ and $ydata$ are matrices or vectors and $F(x, xdata)$ is a matrix-valued or vector-valued function of the same size as $ydata$.Optionally the components of x are subject to the constraints;

$$lb \le x$$
$$x \le ub$$
$$Ax \le b$$
$$Aeqx = beq$$
$$c(x) \le 0$$
$$ceq(x) = 0$$
(2.10)

The arguments x, lb, ub can be vectors ,matrices .The lsqcurvefit function uses the same algorithm as lsqnonlin.Rather than compute the sum of squares lsqcurvefit requires the user-defined function to compute the vector-valued function;

$$F(x, xdata) = \begin{bmatrix} F(x, xdata)(1) \\ F(x, xdata)(2) \\ . \\ . \\ . \\ . \\ F(x, xdata)(k) \end{bmatrix}$$

(2.11)

$x = lsqcurvefit(fun, x_0, xdata, ydata)$ starts at x_0 and finds coefficients x to best fit the nonlinear function $fun(x, xdata)$ to the data $ydata$ (in the least-squares sense) $ydata$ must be the same size as the vector (or matrix) returned by fun.

III. RESULTS

This section presents the output of the various numerical algorithms from MATLAB. The output compares the initially noisy image with the final processed image to establish how effective the algorithm is as far as denoising of images.

Output due to the two-point explicit group successive over relaxation (2-egsor) method

Implementing (2.8) onto the noisy image termed ‘CCTV image’ in Matlab produces the image labeled EGSOR

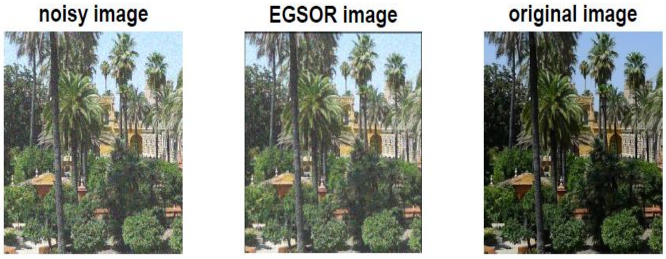

Fig 3.1, 2-point Explicit Group Successive over Relaxation output from MATLAB for the CCTV image

From the output we note that there isn't much difference between the noisy image and the output one.

Implementing (2.8)) on another noisy image of MRI nature;

Fig 3.2, 2-point Explicit Group Successive over Relaxation output from MATLAB for the MRI image

From this output, we realize that the contours/edges in the processed-MRI image are clearer as compared to the noisy image. There's slight recovery of lost details hence slight smoothening.

Implementing (2.8) on another noisy image of Galaxy nature;

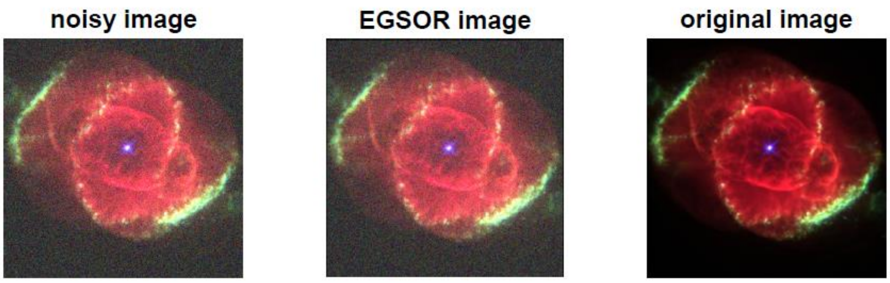

Fig 3.3, 2-point Explicit Group Successive over Relaxation output from MATLAB for the Galaxy image

We note that the output for the galaxy here is much brighter than the noisy image, the image is slightly aligned, details of the image output much clearer.

In conclusion: The 2-EGSOR performs fairly on the edges and texture of the image.

Output due to alternating direction implicit (adi) method from matlab

Implementing (2.5) onto the noisy image termed 'CCTV image' in Matlab produces the following output;

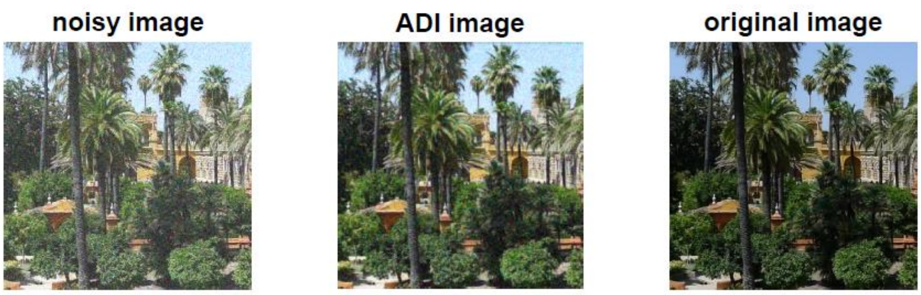

Fig 3.4, Alternating Direction Implicit method output from MATLAB for the CCTV image.

For this image output we realize much toning up, edges are so well outlined ,the noisy image was much faint but the output from ADI is very close to the original image in terms of alignment, texture and overall detail.

Implementing (2.5) on another noisy image of Galaxy nature;

Fig 3.5: Alternating Direction Implicit method output from MATLAB for the Galaxy image.

Once more, for this output by ADI, the edges are well outlined, background of the image is clear. Image is toned up.

Implementing (2.5)) on another noisy image of MRI nature;

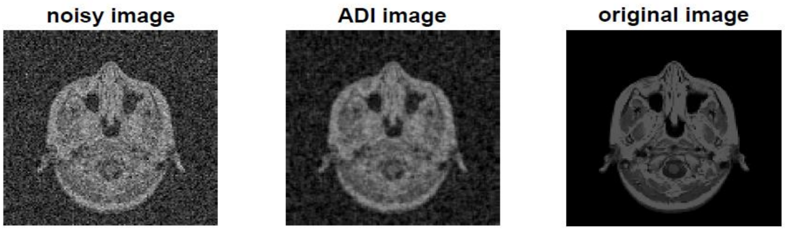

Fig 3.6, Alternating Direction Implicit method output from MATLAB for the MRI image.

From this output, ADI denoises the noisy image including its background, the edges are well outlined in the output, the features in the image are clear and well distinguished.
In conclusion; The ADI algorithm performs excellently on the contours and texture of the image

Output from running 2-egsor and adi successively in matlab

Subjecting the noisy galaxy-image, CCTV-image and MRI-image to both equations (2.8) and (2.5) successively;

Fig 3.7: Output from MATLAB for running the noisy galaxy image through 2-EGSOR and ADI successively

The contours are well defined, texture improved, details well brought out

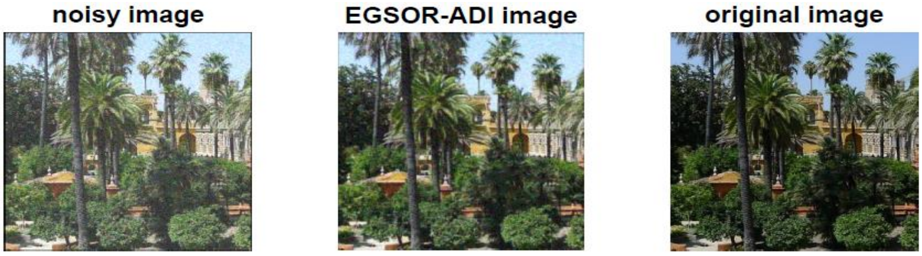

Fig 3.8: Output from MATLAB for running the noisy CCTV image through 2-EGSOR and ADI successively

Image well toned up ,aligned characterized by recovery of lost details

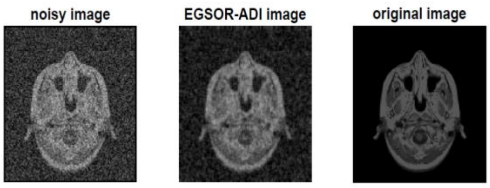

Fig 3.9: Output from MATLAB for running the noisy MRI image through 2-EGSOR and ADI successively

From this output the texture is restored, smooth contours recovered and details well brought out.
In conclusion, blocking two algorithms makes the output smoother. Hence the approach performs excellently.

IV. Discussion

The analysis will include comparison of both the output from the various algorithms ,time-lapse and establishing of residue (difference between noisy and restored images)

Time lapse

Table 4.0 Time lapse in MATLAB for each algorithm

Image	ADI	EGSOR	BLOCKING ADI & EGSOR
CCTV-output	389.54	125.79	337.35
MRI-output	91.44	81.13	81.99
Galaxy	496.12	125.84	389.58

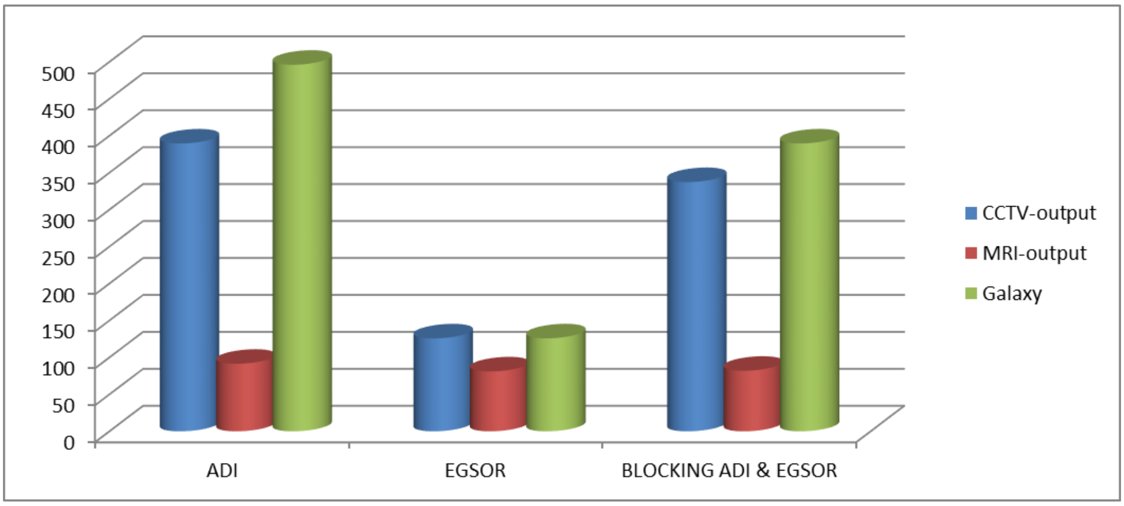

Fig 4.0 Chart of time-lapse vs algorithms for various images generated from MATLAB

It is worthwhile to note that ADI records the highest computational time in MATLAB , followed by the 'blocked' and finally 2-EGSOR .This implies that the more efficient the algorithm the more time it takes to produce results when implemented on the computer.

Residue

Residue is the difference between the original image and the processed image.

Table 4.1: Residues for CCTV-image in ADI

t	C_v	C_f	b	residue
0	0.000008	0.011478	6.000003	4058.687
0.25	0.004774	0.014856	6.000065	4058.687
0.5	0.004774	0.014856	6.000065	4058.687
0.75	0.004774	0.014856	6.000065	4058.687
1	0.004774	0.014856	6.000065	4058.687

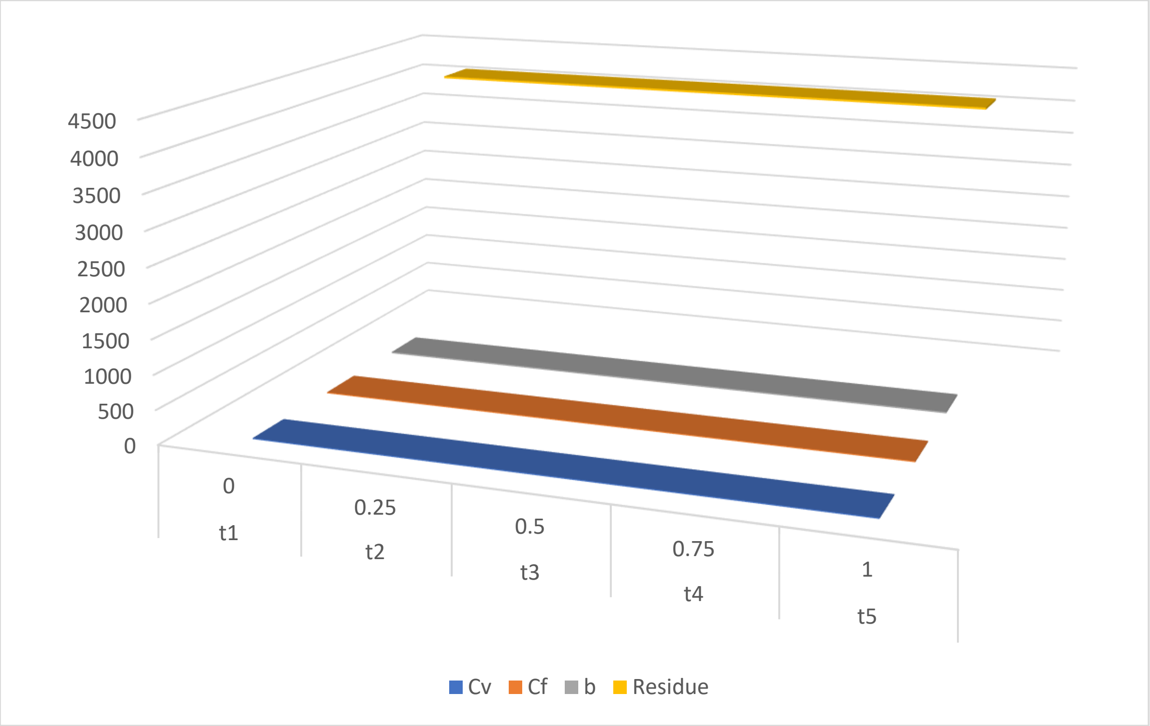

Fig. 4. 1: Residues for CCTV-image in ADI

Table 4.2 Residues for MRI image under ADI

t	C_v	C_f	b	residue
0	0.000125	0.005	4	110.5447
0.111111	0.00000008029	0.013126	4.0027213	110.6497
0.222222	0.00000012929	0.013245	4.002854	110.7410
0.333333	0.00000010089	0.013183	4.00271063	112.3212
0.444444	0.0000001009	0.013181	4.002816	111.9578
0.555555	0.000003031	0.013108	4.012603121	111.0898
0.666666	0.0000096083	0.013138	4.00291909	110.2448
0.777777	0.00000042498	0.013353	4.00002579	110.0330
0.888888	0.000000078885	0.013239	4.00278988	111.0296
1	0.00000001458	0.013227	4.00028693	109.1559

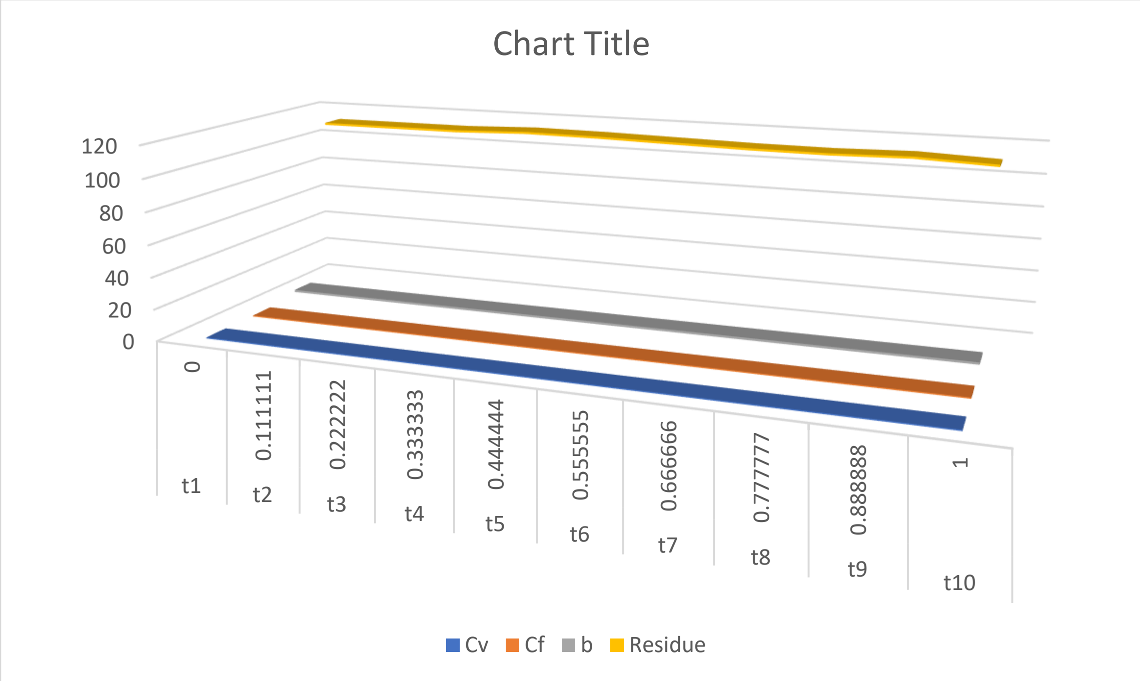

Fig. 4. 2: Residues for MRI image under ADI

Table 4.3 Residues for MRI image under 2-EGSOR

t	C_v	C_f	b	residue
0	0.000125	0.05	4	139.9288
0.111111	0.000000122726	0.006905	4.624991	137.8658
0.222222	0.000000240211	0.006963	4.624997	137.7496
0.333333	0.00000700246028	0.006166	4.937500	138.0863
0.444444	0.000000270080	0.006903	4.624978	137.5437
0.555555	0.000000446883	0.006923	4.624998	138.0859
0.666666	0.00000011264785	0.006189	4.937498	137.2743
0.777777	0.00000003510537	0.007040	4.624993	137.2596
0.888888	0.000003648087	0.007011	4.6249932	138.5581
1	0.0000001083779	0.006954	4.624995	139.0520

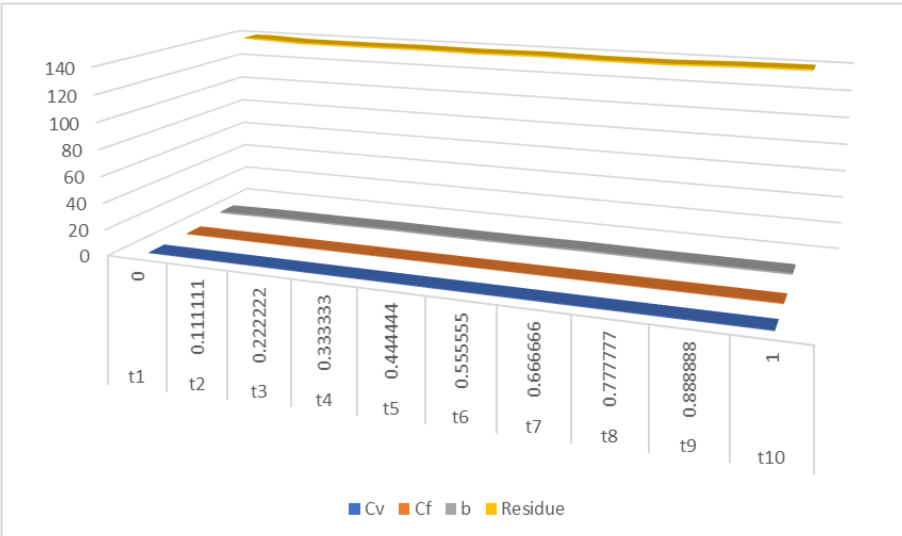

Fig. 4. 3: Table 4.2.3 Residues for MRI image under 2-EGSOR

Table 4.4: Residue for CCTV image under EGSOR

t	C_v	C_f	b	residue
0	0.000008	0.0005	6	6176.827
0.25	0.000008	0.011478	6.000003	6170.404
0.5	0.000033	0.011481	5.999779	6173.353
0.75	0.000002	0.011486	6.0000003	6174.455
1	0.000013	0.011491	6.000002	6178.815

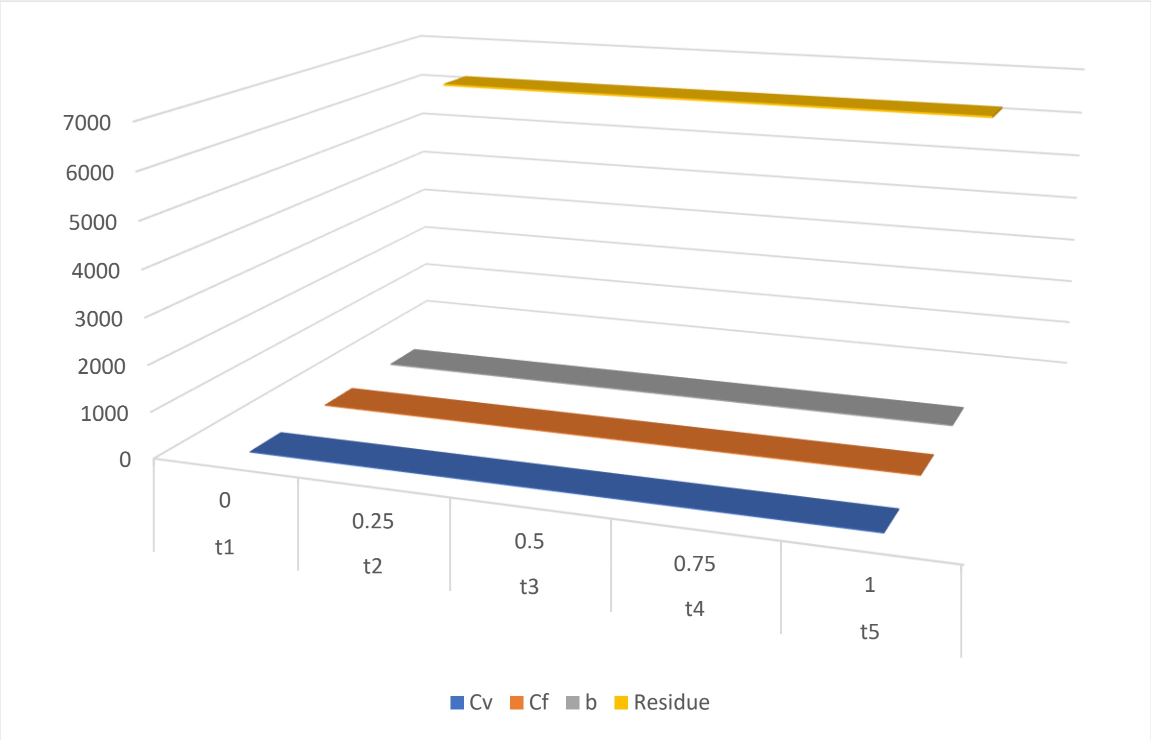

Fig. 4.4: Residue for CCTV image under EGSOR

The bigger the residue the less refined the image. The above tables indicate that ADI does more of the refinement as compared to 2-EGSOR. This is in agreement with the processed images under results above.

Error analysis

The aim of the denoising process is to improve the visual quality of the image and ideally the recovery of the original image. Mathematically we would like to minimize the error over between the restored and original image .This error is written as a distance, we use for simplicity the Euclidean distance. In order to optimize the restoration process the definition of such an error should incorporate the priori information about the original image and the noise when available.

Root Mean Square Error (RMSE)

This is a standard metric used in model evaluation. For a sample of n observations $y(y_i, i=1,2,........n)$,and n corresponding model predictions $\hat{y}$,the RMSE is;

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2} \qquad (4.0)$$

RMSE represents the typical or 'standard' error for normally distributed data. This error is the averaged form of L-2 norm that is common in mathematics and statistics.

Table 4.5 Root mean square for various images across the numerical algorithms from MATLAB

Image	ADI	EGSOR	BLOCKING ADI & EGSOR
CCTV-output	0.2062	0.2219	0.2060
MRI-output	0.2070	0.2198	0.2052
Galaxy	0.2035	0.2195	0.2031

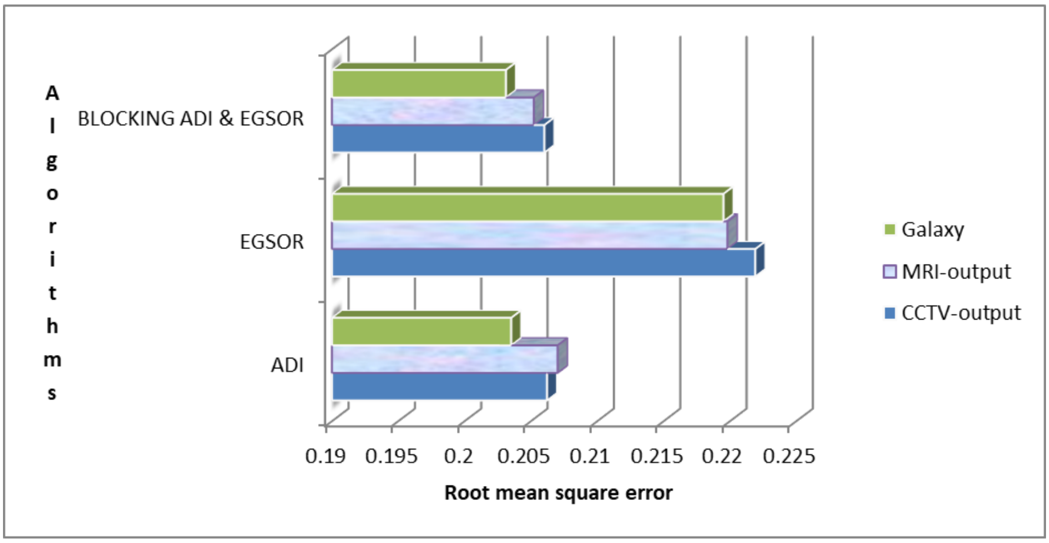

Fig. 4. 4: Root mean square for various images from MATLAB

A smaller error indicates that the estimate (output of a given numerical method) is closer to the original image. Indeed all algorithms perform some denoising though. But ADI & the blocked register a comparatively lower RMSE.

V. CONCLUSION

The most suitable algorithm for image smoothening is Alternating Direction Implicit Method.

References

[1] Basran,N.A.,Saudi,J.H, And Suleiman,J.(2018) Implementation Of The Explicit Group Iterativemethod For Solving Image Blurring Problem Using Nonlinear Diffusion Equations.Iop Conf.Series .Journal Of Physics :Conf Series 1123(2018) 012027

[2] Chew,J.V.L., And Sulaiman, J.(2017) Application Of Four –Point Newton –Egsor Iteration For The Numerical Solution Of 2d Porous Medium Equations, Journal Of Physics :Conference Series 890.

[3] Osher,S.,And Rudin,L.(1990) Feature Oriented Image Enhancement Using Shock Filters.Siam J.Numerical Analysis, 27:919–940.

[4] Ozkan ,M.K., Sezan ,M.I., And Tekalp ,A.M. (1993) Adaptive Motion Compensated Filtering Of Noisy Image Sequences. Ieee Trans. Circuits And Systems For Video Technology, 3:277–290.

[5] Evans,D.J.(1986) Group Explicit Iterative Methods For Solving Large Linear Systems International Journal Of Computer Mathematics 17(1) 81-108

[6] Peaceman, D.W.,And Rachford Jr. H. H.(1955).The Numerical Solution Of Parabolic And Elliptic Differential Equations. Journal Of The Society For Industrial And Applied Mathematics, 3(1):28 – 41.

[7] Perona,P.,And Malik,J.(1990)."Scale-Space And Edge Detection Using Anisotropic Diffusion," Ieee Transactions On Pattern Analysis And Machine Intelligence, 12(7), Pp.629-639.

[8] Rohit,K(2023).Image Processing :Techniques,Types And Applications.University Of California ,Riverside

[9] Roza,D.,Mohsen,S.(2021) Advanced Image Processing Systems. International Journal Of Imaging And Robotics.

[10] Rudin,L.S.,Osher,And Fatemi,E.(1992).“Nonlinear Total Variation Based Noise Removal Algorithms,” In Proc. Mod´Eliz. Mat. Traitement D’images, Inria, Pp.149–179.

[11] Saudi,A., And Sulaiman ,J.(2017) “Numerical Assessment For Poisson Image Blending Problem Using Msor Iterative Via Five-Point Laplacian Operator”,Journal Of Physics :Conference Series ,Vol 890,No.1,Pp.012010.

[12] Sirmaceck,B.,Shen,Y.,Lindenbergh,R.,Zlatanova,S.,And Diakite,A.(2016).Comparison Of Zebi And Leica C10 Indoor Laser Scanning Point Clouds .Xxiii, Isprs Congress,Ispr Annals ,Prague,Czech Republic

[13] Tannenbaum,A. Kimia,B.B., And Zucker, S.W. (1992). The Evolution Of Curves Via A Function Of Curvature, I: The Classical Case.

[14] Weicket,J.,Romeny ,B. M. T. H., And Viergever,M.A(1998) Efficient And Reliable Schemes For Nonlinear Diffusion Filtering. Ieee Transactions On Image Processing 7(3) 398-410.

[15] Witkin,A. P. (1983). “Scale-Space Filtering,” In Proc. Int. Joint Conf. Artif. Intel. Pp. 1019–1021.

[16] Xu, C. And Prince, J. L. (1998). “Snakes, Shapes, And Gradient Vector Flow,” Ieee Transactions On Image Processing, Vol. 7(3), Pp. 359-369.