Block Diagonal Preconditioners for an Image De-blurring Problem with Fractional Total Variation

Adel Al-Mahdi$^1$ and Faisal Fairag$^2$

$^1$PYP-Math, King Fahd University of Petroleum & Minerals, Dhahran, 31261 Saudi Arabia
$^2$Mathematics and Statistics Department, King Fahd University of Petroleum & Minerals, Dhahran, 31261 Saudi Arabia

E-mail: $^1$almahdi@kfupm.edu.sa, $^2$ffairag@kfupm.edu.sa

Abstract. Even though using the total fractional-order variation (TFOV) model reduces or eliminates the stair casing effect in image de-blurring problems, but in the same time it generates a dense and ill-conditioned linear system of equations. These properties lead to slow convergence of any iterative method such as Krylov subspace methods. One treatment of the slowness property is to apply the preconditioning technique. In this research work, we propose two block diagonal preconditioners to construct an image from a given blurred image using circulant matrices. These matrices allow us to use the fast Fourier transform (FFT) and the convolution theorem. Using FFT and the convolution theorem reduces the cost of the computations from $O(n^2)$ into $O(n \log n)$ operations in each iteration and also reduces the storages. Our proposed preconditioners are of Murphy, Golub, and Wathen type. Numerical examples are given to illustrate the efficiency of our preconditioners.

1. Introduction

To de-blur an image, we need a mathematical model of how it was blurred. The recorded image $z$ and the true (exact) image $u$ are related by the equation

$$z = Ku + \varepsilon,$$

where $K$ denotes the following blurring operator

$$(Ku)(x) = \int_{\Omega} k(x, x')u(x')dx', \quad x \in \Omega$$

with translation invariance kernel, $k(x, x') = k(x - x')$, and it is known as the point spread function (PSF). $\varepsilon$ is the additive noise function. $\Omega$ will denote a square in $\mathbb{R}^2$ on which the image intensity is defined. The operator $K$ is compact, so problem (1) is ill-posed [1] and then the matrix systems resulting from the discretization of this problem is highly ill-conditioned. In this case directly solving this problem is difficult. The most popular idea to get a well posed problem is to add a regularization term. In the literature, different regularization terms are used for example: Tikhonov regularization [2] is used to stable the problem (1). Although, this model is easy to use and simple to calculate, it cannot preserve image edges. One of the most commonly used regularization model is the total variation (TV). It was introduced in the first time [3] in edge preserving image denoising by Rudin, Osher and Fatemi (ROF) and it has
improved in the recent years for image de-noising, de-blurring, in-painting, blind de-convolution and processing, see for example, [4, 5, 6, 7]. When using the TV model, the problem is then to find a $u$ that minimize the functional

$$ T(u) = \frac{1}{2} \| Ku - z \|^2 + \lambda J_{TV}(u), \quad (3) $$

where

$$ J_{TV}(u) = \int_{\Omega} |\nabla u| \, dx. \quad (4) $$

where $|\cdot|$ denotes Euclidean norm, and $\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$. However, the Euclidean norm, $|\nabla u|$, is not differentiable at zero. Common modification is to add small positive parameter $\beta$ and then the modified functional is

$$ J_{TV \beta}(u) = \int_{\Omega} \sqrt{|\nabla u|^2 + \beta^2} \, dx. \quad (5) $$

The well-posedness of the above minimization problem (3) with the functional given in (5) is studied and analyzed in the literature see for example [4]. The success of using TV regularization is that TV gives a balance between the ability to describe piecewise smooth images and the complexity of the resulting algorithms. Moreover, TV performs very well for removing noise/blurry while preserving edges. Despite of the good contributions of the TV regularization mentioned above, it favors a piecewise constant solution in the bounded variation (BV) space which often leads to the staircase effect. So, stair casing remains one of the drawback of the TV regularization. To remove the stair case effects, two modifications to the TV regularization model are used in the literature. In the first one, the higher order of derivative in the TV regularization term, the mean curvature and a nonlinear combination of the first and second derivatives are used, see for example, [8, 9, 10, 11]. These modifications are used to remove/reduce the staircase effects and they are effective but the computations are very difficult due to the increasing of the order of the derivatives or due to the nonlinearity terms. In the second approach, the fractional-order derivatives in the TV regularization term are used. This approach is widely used in the literature [12, 13, 14, 15, 16]. When TFOV model is used, the problem is then to find a $u$ that minimize the functional

$$ T^\alpha(u) = \frac{1}{2} \| Ku - z \|^2 + \lambda J_{TV \beta}^\alpha(u) \quad (6) $$

where

$$ J_{TV \beta}^\alpha(u) = \int_{\Omega} \sqrt{|\nabla^\alpha u|^2 + \beta^2} \, dx, \quad (7) $$

and $|\nabla^\alpha u|^2 = (D_x^\alpha u)^2 + (D_y^\alpha u)^2$. The parameter $\alpha$ represents the order of the fractional derivatives and $D_x^\alpha, D_y^\alpha$ are the fractional derivative operators in the $x$ and $y$ directions respectively. Existence and uniqueness of a minimizer to the above problem (6) with the functional (7) is studied and analyzed in the literature see for example [16], [17]. The numerical results in [16], [17], showed that the staircase effect can be eliminated effectively by using the fractional-order derivative.

### 1.1. Fractional-order derivatives

Several definitions for fractional order derivative have been proposed to describe a fractional order derivative [18], [19] and [20]; we shall present three of them below. For a systematic presentation of the mathematics, a fractional order derivative is denoted as function operator $D_{[a,x]}^\alpha$, where $a$ and $x$ are the bounds of the integrals, and $\alpha$ is the order of the fractional derivative such that $0 < l := n - 1 < \alpha < n$ where $n = [\alpha] + 1$ and $[\cdot]$ is the greatest integer function.
1- Riemann-Liouville (RL) definitions: The left and right-sided RL derivatives of order $\alpha$ of a function $f(x)$ are given as follows:

$$D^\alpha_{[a,x]} f(x) = \frac{1}{\Gamma(n - \alpha)} \left( \frac{d}{dx} \right)^n \int_a^x (x-t)^{n-\alpha-1} f(t) dt$$

and

$$D^\alpha_{[x,b]} f(x) = \frac{1}{\Gamma(n - \alpha)} \left( -\frac{d}{dx} \right)^n \int_x^b (t-x)^{n-\alpha-1} f(t) dt$$

where $\Gamma(\cdot)$ is the gamma function and it is defined by

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt.$$ 

2- Grünwald Letnikov (GL) definitions: The left and right-sided GL derivatives are defined by

$$GD^\alpha_{[a,x]} f(x) = \lim_{\Sigma_j \to 0} \sum_{j=0}^\infty (-1)^j C_j^\alpha f(x - jh)/h^\alpha$$

and

$$GD^\alpha_{[x,b]} f(x) = \lim_{\Sigma_j \to 0} \sum_{j=0}^\infty (-1)^j C_j^\alpha f(x + jh)/h^\alpha$$

where

$$C_j^\alpha = \frac{\alpha(\alpha-1)...(\alpha-j+1)}{j!}.$$ 

3- Caputo (C) definitions: The left and right-sided Caputo derivatives are defined by

$$CD^\alpha_{[a,x]} f(x) = \frac{1}{\Gamma(n - \alpha)} \int_a^x (x-t)^{n-\alpha-1} f(t) dt$$

and

$$CD^\alpha_{[x,b]} f(x) = \frac{(-1)^n}{\Gamma(n - \alpha)} \int_x^b (t-x)^{n-\alpha-1} f(t) dt$$

where $f^{(n)}$ denotes the $n$th-order derivative of function $f(x)$. The spatial Riesz-type fractional $\alpha$-order derivative which is considered as a Riesz-type potential is defined as a half-sum of the left and right-sided derivatives of three definitions for function $f(x)$:

$$\gamma D^\alpha_{[a,b]} f(x) = \frac{1}{2} \left[ \gamma D^\alpha_{[a,x]} f(x) + (-1)^n \gamma D^\alpha_{[x,b]} f(x) \right]$$

where $\gamma$ takes $R, RG$ and $RC$ for Riesz-RL, Riesz-GL and Riesz-Caputo fractional derivative respectively. It is known that when a function is $n - 1$ times continuously differentiable and its $n - th$ derivative is integrable, the fractional derivatives by the above definitions are equivalent subject to homogeneous boundary conditions [20]. In our case, we impose the zero Dirichlet boundary condition and we mainly consider the discretization of the $\alpha$-order fractional derivative at the inner point along the $x$-direction by using the shifted Grünwald approximation approach as in [20], [21], [16], and [17].
1.2. Euler-Lagrange equations

In this section, we present the Euler-Lagrange equations associated with the TFOV in image de-blurring problem.

**Theorem 1** If $\alpha \in (1, 2)$, the Euler-Lagrange equations for the functional given in (6) are

\[
K^* (Ku - z) + \lambda L_\alpha(u)u = 0, \quad \text{in } \Omega
\]

\[
D^\alpha - 2 \left( \frac{\nabla^\alpha u}{\sqrt{|\nabla^\alpha u|^2 + \beta^2}} \right) \cdot \vec{n} = 0, \quad D^{\alpha - 1} \left( \frac{\nabla^\alpha u}{\sqrt{|\nabla^\alpha u|^2 + \beta^2}} \right) \cdot \vec{n} = 0, \quad \text{on } \partial \Omega,
\]

where $K^*$ is the adjoint operator of the integral operator $K$ and the differential operator $L_\alpha(u)$ is given by

\[
L_\alpha(u)w = (-1)^m \nabla^\alpha \cdot \left( \frac{\nabla^\alpha w}{\sqrt{|\nabla^\alpha u|^2 + \beta^2}} \right).
\]

where $m = [\alpha] + 1$.

Note that (16) is a nonlinear integro-differential equation of elliptic type. Equation (16) can be expressed as a nonlinear first order system [22]

\[
K' Ku - \lambda \nabla^\alpha \vec{v} = K^* z,
\]

\[
-\nabla^\alpha u + \sqrt{|\nabla^\alpha u|^2 + \beta^2} \vec{v} = 0,
\]

with the dual, or flux, variable

\[
\vec{v} = \frac{\nabla^\alpha u}{\sqrt{|\nabla^\alpha u|^2 + \beta^2}}.
\]

Applying Galerkin’s method to (18-19), the midpoint quadrature for the integral term, the cell center finite difference method (CCFDM) and the fractional discretization, one obtains the following dual-system

\[
V + K_h U = Z,
\]

\[
K_h^* V - \lambda L_\alpha h V = 0
\]

For the simplicity we eliminate the subscript $h$ equipped with the matrices in (21) to get

\[
\begin{bmatrix}
I_n & K \\
-K^* & \lambda L^\alpha
\end{bmatrix}
\begin{bmatrix}
V \\
U
\end{bmatrix} =
\begin{bmatrix}
Z \\
0
\end{bmatrix},
\]

where the matrix $K$ is symmetric positive semi definite and $L^\alpha$ is symmetric positive definite. Both $K$ and $L^\alpha$ are of size $n \times n$. The matrix $K$ has the Toeplitz structure (BTTB). Even though using the TFOV model reduce/eliminate the stair casing effect, but in the same time it generates a dense and ill-conditioned linear system of equations. These properties lead to slow convergence of any iterative method such as Krylov subspace methods. One treatment of the slowness property is to apply the preconditioning technique. Due to the properties of the coefficient matrix, the preconditioned generalized minimal residual (preconditioned GMRES) [23] can be used.
2. The Problem

In this research work, we consider:

\[
\begin{bmatrix}
I_n & K \\
-K^* & \lambda L^\alpha
\end{bmatrix}
\begin{bmatrix}
V \\
U
\end{bmatrix}
= 
\begin{bmatrix}
Z \\
0
\end{bmatrix},
\]

(23)

The coefficient matrix, \(A\), of the system (23) is huge i.e it is of size \(2n\) by \(2n\) where \(n = n_x^2\) and \(n_x\) is the number of pixels of the image. The reason for the huge size is that for example a small image of size 256 times 256 pixels will have a corresponding matrix of size \(256^2\) times \(256^2\). Moreover, \(A\) is a dense matrix. This is because both \(K\) and \(L\) are dense (in case \(\alpha = 1\), \(L^\alpha\) is a sparse matrix). Further, \(A\) is an ill-conditioned matrix. It has a large condition number and this condition number increases when the number of image pixels increases. It is clear form the structure of \(A\) that it is not symmetric but it is positive definite matrix. Regarding the above properties of the matrix \(A\), we apply the idea of the preconditioning technique to use the preconditioned generalized minimum residual (PGMRES) method [23] for the outer iterations and Preconditioned conjugate gradients method (PCG) for the inner iterations. This will help to get convergence in few iterations. Our main contribution in this work is: we develop two block diagonal preconditioners for the system (23). These preconditioners are of Murphy, Golub, and Wathen type [24] and they only differ in the approximation of the \((2,2)\) block matrix. We demonstrate the performance of our approach through calculating the peak signal-to-noise ratio (PSNR) to different de-blurred images and compare it with the PSNR of the blurred image. Furthermore, we show the effective of our preconditioners through calculating and plotting the residuals versus the iterations number.

3. The Preconditioners

In this section, we present our main preconditioner matrix

\[
P = \begin{bmatrix}
I_n & 0 \\
0 & -S
\end{bmatrix}
\]

(24)

where \(S = K^*K + \lambda L^\alpha\) is the Schur complement matrix. We notice that the Schur complement matrix contains the product \((K^*K)\) which is not a BTTB matrix. In this case, the computation of a matrix times vector of the form \(z = Pr\) will be very expensive when using the exact Schur complement. Hence, we use circulant preconditioners for the matrices that have the Toeplitz structures. Circulant preconditioning for Toeplitz systems was introduced by Strang [25] and extended by others to block Toeplitz systems [26]. Many researchers use a Toeplitz preconditioners and block Toeplitz preconditioners for Toeplitz systems; see, for instance, [27] and [28]. Band Toeplitz preconditioner and band BTTB preconditioner are proposed by Chan [29] and Serra [30]. In [31], BTTB preconditioners for BTTB systems are discussed.

In our preconditioner matrix \(P\) given in (24), we use two approximations. The first one is obtained by approximating the matrix \((K^*K)\) by a symmetric BTTB matrix \(T\) as in [32, 33] while in the second one, we approximate the matrix \((K^*K)\) by \(C^*C\) where \(C\) is the optimal circulant approximation [34] to the matrix \(K\).

In this case, we have the following two block diagonal preconditioners

\[
PT = \begin{bmatrix}
I_n & 0 \\
0 & -S_t
\end{bmatrix}, \quad PC = \begin{bmatrix}
I_n & 0 \\
0 & -S_c
\end{bmatrix},
\]

(25)

where \(S_t = (T + \lambda L^\alpha)\) and \(S_c = (C^*C + \lambda L^\alpha)\). The above two block diagonal preconditioners are mainly different by the approximations of the matrix \((K^*K)\). These circulant approximations
are very important to allow us to use the FFT and the convolution theorem. Moreover, all that is needed for computation is the first column or the first row of the circulant matrix, which decreases the amount of required storage. Also using FFT and the convolution theorem will reduce the cost of the computation from $O(n^2)$ into $O(n \log n)$. This reduction in the computations and storages leads to have efficient solvers for our problem (23). Triangular preconditioners for the same problem (23) are proposed and analyzed in [35].

4. Preconditioned GMRES Algorithms

In this section, we give detailed algorithms in using the two preconditioners $P_T$ and $P_C$. In both algorithms, GMRES method is used to solve the linear system (23). In the first algorithm, $P_T$ is used as a preconditioner while in the second algorithm, $P_C$ is used as a preconditioner.

**Algorithm 1a for the $P_T$-preconditioned system**

(i) Choose $x^0$ as the initial guess
(ii) Compute $\tilde{r}^0 = b - Ax^0$
(iii) Solve $P_T r^0 = \tilde{r}^0$
(iv) Let $\beta_0 = \|r^0\|$, and compute $v^{(1)} = r^0 / \beta_0$
(v) For $k = 1, 2, ...$ until $\beta_k < \tau \beta_0$

(vi) $\tilde{w}_0^{(k+1)} = A v^{(k)}$

(vii) Solve $P_T w_0^{(k+1)} = \tilde{w}_0^{(k+1)}$

(viii) For $l = 1$ to $k$

(ix) $h_{lk} = \langle w_l^{(k+1)}, v^{(l)} \rangle$

(x) $w_l^{(k+1)} = w_l^{(k+1)} - h_{lk} v^{(l)}$

(xi) end do

(xii) $h_{k+1,k} = w_{k+1}^{(k+1)} / h_{k+1,k}$

(xiii) Compute $y^{(k)}$ such that $\beta_k = \|\beta_0 e_1 - \tilde{H}_k y^{(k)}\|$ is minimized, where

$\tilde{H}_k = [h_{ij}]_{1 \leq i \leq k+1, 1 \leq j \leq k}$ and $e_1 = (1, 0, ..., 0)^T$

(xiv) end do

(xv) $x^{(k)} = x^0 + V_k y^{(k)}$

**Algorithm 1b for the $P_C$-preconditioned system**

(i) Choose $x^0$ as the initial guess
(ii) Compute $\tilde{r}^0 = b - Ax^0$
(iii) Solve $P_C r^0 = \tilde{r}^0$
(iv) Let $\beta_0 = \|r^0\|$, and compute $v^{(1)} = r^0 / \beta_0$
(v) For $k = 1, 2, ...$ until $\beta_k < \tau \beta_0$

(vi) $\tilde{w}_0^{(k+1)} = A v^{(k)}$

(vii) Solve $P_C w_0^{(k+1)} = \tilde{w}_0^{(k+1)}$

(viii) For $l = 1$ to $k$

(ix) $h_{lk} = \langle w_l^{(k+1)}, v^{(l)} \rangle$

(x) $w_l^{(k+1)} = w_l^{(k+1)} - h_{lk} v^{(l)}$

(xi) end do

(xii) $h_{k+1,k} = w_{k+1}^{(k+1)} / h_{k+1,k}$
(xiii) Compute $y^{(k)}$ such that $\beta_k = \|\beta_0 e_1 - \hat{H}_k y^{(k)}\|$ is minimized, where $\hat{H}_k = [h_{ij}]_{1 \leq i \leq k+1, 1 \leq j \leq k}$ and $e_1 = (1, 0, ..., 0)^T$

(xiv) end do

(xv) $x^{(k)} = x^0 + V_k y^{(k)}$

For more details in using GMRES see [36]. In the above algorithms, in the steps 3 and 7, we need to solve a matrix times a vector of the form

$\begin{bmatrix} I_n & 0 \\ 0 & -S_t \end{bmatrix}_P \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$. \hspace{1cm} (26)

and

$\begin{bmatrix} I_n & 0 \\ 0 & -S_c \end{bmatrix}_P \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$. \hspace{1cm} (27)

To do the above multiplications, we use the conjugate gradients method as in the following algorithms:

**Algorithm 2a for the $P_T$-conjugate gradient method**

(i) $x_1 = x(1 : n) = b(1 : n)$;
(ii) $S_t = P_T(n + 1 : 2n, n + 1 : 2n)$;
(iii) $b_2 = b(n + 1 : 2n)$;
(iv) $x_2 = x(n + 1 : 2n)$
(v) Solve for $x_2$ in the system $-S_t x_2 = b_2$ using conjugate gradient method.

**Algorithm 2b for the $P_C$-conjugate gradient method**

(i) $x_1 = x(1 : n) = b(1 : n)$;
(ii) $S_c = P_C(n + 1 : 2n, n + 1 : 2n)$;
(iii) $b_2 = b(n + 1 : 2n)$;
(iv) $x_2 = x(n + 1 : 2n)$;
(v) Solve for $x_2$ in the system $-S_c x_2 = b_2$ using conjugate gradient method.

5. **Numerical results**

In this section, we experimentally study the performance of the GMRES method with the proposed block diagonal preconditioners. In the following numerical experiments, we take the zero vector to be the initial guess. We stopped the outer iterations (GMRES) when the residuals is less than $10^{-7}$. We used the PCG for the inner iterations and it is stopped when the tolerance is $10^{-8}$. No restarting is used for GMRES algorithm. For this purpose, two famous $128 \times 128$ test images, called Cameraman, and Peppers images, are used in the experiments, as shown in Figures (1, 2) and they are blurred by the motion kernel shown in Figure (3).

In order to show the performance of the proposed preconditioners, we need to calculate the Peak Signal-to-Noise Ratio (PSNR) which is commonly used in signal processing field. Bigger PSNR means better de-blurring performance.

**Example 1** In this example, we show the impact of our preconditioners by comparing between GMRES with using the proposed preconditioners $P_T$ and $P_C$ and GMRES without preconditioners. We fix $nx = 128$ i.e $n = 16384$, the fractional order $\alpha = 1.6$, the parameter $\beta = 0.01$, and the regularization parameter $\lambda = 0.01$. No restarting is used for GMRES algorithm.
Figure 1. cameronman image

Figure 2. peppers image

Figure 3. Shape of the kernel

and it is stopped when the tolerance is $10^{-7}$. We use the test image “Camerman”. In each non-preconditioned GMRES (NPGMRES) iteration, we calculate the logarithm of $\|r^{(n)}\|_2$ where $r^{(n)}$ is the residual in the $n$-th iteration. Furthermore, in each preconditioned GMRES (PGMRES) iteration, we calculate the logarithm of $\|P^{-1}r^{(n)}\|_2$ where $P$ is the preconditioner $P_T$ and $P_C$. The results of these calculations are plotted in Figure (4). In Figure (4), NP stands for GMRES with out preconditioners, $P_T$ stands for GMRES with the diagonal preconditioner $P_T$ and $P_C$ stands for GMRES with the diagonal preconditioner $P_C$. From Figure (4), we observe that the convergence of NP is the slowest, followed by $P_C$ and $P_T$. Furthermore, we observe that NP needs about 78 iterations to reach tolerance $10^{-3}$ while our proposed preconditioner $P_C$ needs 42 iterations and $P_T$ needs 33 iterations to reach the same tolerance. From this example, we notice
that the block diagonal preconditioners are much faster than non-preconditioner and also the $P_T$ is a little bit better than $P_C$.

**Example 2** In this example, we show the performance of our proposed preconditioners. We use our proposed preconditioners to construct approximation images from the given blurred images. In other words, through this example, we show the improvement in the PSNR. To do that, we start using the two blurred images (of size $128 \times 128$) in Figures (5,6) and we use GMRES with our preconditioners $P_T$ and $P_C$ to de-blurred them. The results are shown in Figures (7-10).

From Figures (5-10), one notice the impact of our preconitioners. They improved the PSNR from 18.632 into 40.347 in the cameraman image. Also, they improved the PSNR from 21.434 into 43.748 in the peppers image. From these results, we conclude that our preconditioners produce highly improvements in the PSNR.

6. Conclusion
In this research work, we have proposed and investigated two block diagonal preconditioners for solving the generalized saddle point system which is derived from discretizing the Euler Lagrange equations associated with the total fractional-order variation (TFOV) in image de-blurring problems. Our experiments show that the block diagonal preconditioners are very
deblured image psnr = 40.347

Figure 7. de-blurred using $P_T$

deblured image psnr = 43.748

Figure 8. de-blurred using $P_T$

deblured image psnr = 38.964

Figure 9. de-blurred using $P_C$

deblured image psnr = 42.353

Figure 10. de-blurred using $P_C$

effective. We have also showed that our technique improves the quality of the reconstruction images via calculation the PSNR. We also show the performance of the preconditioned GMRES with our proposed preconditioners through calculating the residuals in each iteration. We also concluded that $P_T$ is slightly better than $P_C$.

Acknowledgments
We would like to acknowledge the support provided by the Deanship of Scientific Research (DSR) at King Fahd University of Petroleum and Minerals (KFUPM) for funding this work through project No. SR161018.

References
[1] Groetsch C W and Groetsch C 1993 Inverse problems in the mathematical sciences (Springer) vol 52
[2] Tikhonov A N 1963 Regularization of incorrectly posed problems Soviet Math. Dokl 4 1624–1627
[3] Rudin L I, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms Physica D: Nonlinear Phenomena 60(1) 259–268
[4] Acar R and Vogel C R 1994 Analysis of bounded variation penalty methods for ill-posed problems Inverse problems 10(6) 1217
[35] Al-Mahdi A and Fairag F (submitted) Preconditioning technique for the total fractional order variation model in image de-blurring problem

[36] Elman H C, Silvester D J and Wathen A J 2014 *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics* (Numerical Mathematics and Scientific Computation)