Immediate solution of EM algorithm for non-blind image deconvolution

Seung-Gu Kim

Department of Big Data Science, Sangji University, Korea

Abstract

Due to the uniquely slow convergence speed of the EM algorithm, it suffers form a lot of processing time until the desired deconvolution image is obtained when the image is large. To cope with the problem, in this paper, an immediate solution of the EM algorithm is provided under the Gaussian image model. It is derived by finding the recurrent formular of the EM algorithm and then substituting the results repeatedly. In this paper, two types of immediate solution of image deconboution by EM algorithm are provided, and both methods have been shown to work well. It is expected that it free the processing time of image deconvolution because it no longer requires an iterative process. Based on this, we can find the statistical properties of the restored image at specific iterates. We demonstrate the effectiveness of the proposed method through a simple experiment, and discuss future concerns.

Keywords: image deconvolution, EM algorithm, immediate EM solution

1. Introduction

Image deconvolution refers to the operation of reconstructing a given blur image into a sharp image. In general, the relationship is assumed as the following model

\[ y = h \ast x + e, \]  

(1.1)

where \( \ast \) denotes finite discrete convolution operator, \( h \) is a psf (point spread function) blur kernel, \( x \) is an unknown true sharp 2D image, and \( e \) is a noise in which elements are independently followed by Gaussian distribution \( N(0, \sigma_e^2) \).

When \( h \) is also unknown, the simultaneous estimation task of \( x \) and \( h \) is called “blind deconvolution”, whereas when \( h \) is given as a known, it is called “non-blind deconvolution”. So the latter is treated as an easier problem than the former.

However, many blind deconvolution techniques, like in Fergus et al. (2006), Levin et al. (2011), Jin et al. (2018) among others, use a non-blind deconvoution because it is necessary to estimate \( x \) under a given \( x \) in the iterative process. Therefore, it is still important to develop an efficient non-blind deconvoution technique.

For non-blind deconvoution, the most popular algorithm is RL (Richardson, 1972 ; Lucy, 1974). This technique is based on the maximum likelihood estimation by the EM algorithm under the Poisson model. However, this technique requires a lot of time to obtain the desired result. Moreover, as shown
in the experiment in Section 5, it contains undesirable artifacts especially when the iteration is not sufficient.

In this paper, we provide non-blind deconvolution using the EM algorithm under the Gaussian model. However, due to the inherent slow convergence speed of the EM algorithm, the processing time is impractical when the image size is large. Therefore, in this study, an immediate restored image of the EM algorithm at any iterates is presented so that an iterative process is not required.

The following section presents terms and concepts necessary for this paper. Section 3 presents an iterative EM algorithm. In Section 4, an immediate solution of the iterative EM algorithm is derived. And based on this result, we examine the statistical properties of the restored image in the middle of iteration. In Section 5, we demonstrate the effectiveness of the proposed method through a simple experiment. Finally, in Section 6 conclusions and discussions are presented.

2. Preliminaries

Equation (1.1) can be expressed as the following matrix notation for convenience

\[ y = Hx + e, \]  

(2.1)

where \( y_{n \times 1} = \{y_i\}, \) \( x_{m \times 1} = \{x_j\}, \) and \( e_{n \times 1} = \{e_i\} \) are vectors reshaped by vectorizing 2D arrays \( y, x, \) and \( e \) in a column-stacking manner, respectively, and \( H_{n \times m} = \{h_{ij}\} \) is the corresponding convolution matrix (the blocked-Toelplitz matrix) for the filter \( h. \) Equivalently, it can be expressed as

\[ y_i = h_i^T x + e_i = \sum_{j=1}^{m} h_{ij} x_j + e_i, \quad i = 1, \ldots, n, \]  

(2.2)

where \( h_i \) is the \( i^{th} \) row of \( H, \) and thereafter the superscript \( T \) denotes the transpose of the matrix. Then \( y_i \) independently follows \( N(\sum_{j=1}^{m} h_{ij} x_j, \sigma_e^2). \) Our goal is to estimate the sharp image \( x \) given a blur image \( y. \)

The unbiased least squared estimator which maximizes the log-likelihood, \( L(x) \propto -1/2 \sigma_e^2 (y - Hx)^T (y - Hx) \)

\[ \hat{x}_{LS} = (H^T H)^{-1} H^T y = x + (H^T H)^{-1} H^T e, \]  

(2.3)

is a good choice if the noise \( e \) is negligible. However, it gives very undesirable results even for small noises since it amplifies the noise by the term \( (H^T H)^{-1} H^T e \) in systematic pattern.

To alleviate this problem, for a positive definite matrix \( W = \{w_{jk}\}, \) we need to regularize the solution by using a penalized log-likelihood

\[ L_\lambda(x) \propto -\frac{1}{2} (y - Hx)^T (y - Hx) - \frac{\lambda}{2} x^T W x, \]  

(2.4)

\( \lambda > 0 \) is a regularization constant. One can use \( W = K^T K \) where \( K \) is the convolution matrix of a kernel \( k \) (e.g. Laplacian or gradient etc.) in the way \( k \otimes x = Kx. \)

By maximizing the equation (2.4) we readily obtain the best solution as follows,

\[ \hat{x}_\lambda = (H^T H + \lambda W)^{-1} H^T y. \]  

(2.5)

It is well-known that \( x_\lambda \) is the ridge estimator when \( W = I, \) and in addition when \( \lambda = \sigma_e^2 / \sigma_x^2, \) where \( \sigma_x^2 \) is the variance of \( x, \) it is the Wiener deconvolution.
However, the optimal solution \( x_j \) gives an undesirable image for an inaccurately guessed \( \lambda \). Therefore, the method of stopping when a desired image is obtained is often used in the iterative process of the EM algorithm.

3. Iterative EM deconvolution

A natural setting for understanding our problem as a missing/observed data structure is to look at it as follows,

\[
y_i = \sum_{j=1}^{m} z_{ij}, \quad i = 1, \ldots, n, \quad (3.1)
\]

where \( z_{ij} \) are missing data which follows \( N(h_{ij}x_j, \sigma_z^2/m) \) independently, and therefore \((y, z)\) is complete data denoting \( z = \{z_{ij}\} \). It is same as in RL deconvolution but their \( z_{ij} \sim \text{Poisson}(h_{ij}x_j) \). The setting equation (3.1) allows us to estimate \( x_j \) individually without taking into account the superimposed relation between \( x_j, s \) due to convolution. However, since the parameter space is extended from \( \mathbb{R}^m \) to \( \mathbb{R}^{mn} \), the convergence speed of the EM algorithm will be extremely slow.

Noting that the density \( p(y, z) = p(z) \) from relationship of equation (3.1), the (penalized) log-likelihood for complete data is

\[
L^c(x) \propto -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left( z_{ij} - h_{ij}x_j \right)^2 - \frac{1}{2 \sigma^2} \sum_{k=1}^{m} x_j w_{jk} x_k \\
- \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left( z_{ij}^2 - 2h_{ij}x_jz_{ij} + h_{ij}^2x_j^2 \right) - \frac{\lambda}{2} \sum_{k=1}^{m} \sum_{j=1}^{m} x_j w_{jk} x_k, \quad (3.2)
\]

where \( \lambda = \sigma_y^2/\sigma_z^2 \). EM algorithm at the \( i^{th} \) iteration maximizes \( Q(x; x^{(i)}) = E[L^c(x)|y, x^{(i)}] \). In E-step, we need to compute two conditional expectation \( E[z_{ij}|y, x^{(i)}] \) and \( E[z^2_{ij}|y, x^{(i)}] \). The latter is just a necessary one for estimating \( \lambda \). But since it will be used as a given tuning constant in this paper, we only compute the former in order to estimate solely \( x \) as follows.

\[
\hat{z}_{ij}^{(i+1)} = E(z_{ij}|y_i, x_j^{(i)}) = E(z_{ij}|x_j^{(i)}) + \frac{\text{cov}(z_{ij}, y_i|x_j^{(i)})}{\text{var}(y_i|x_j^{(i)})} \left[ y_i - E(y_i|x_j^{(i)}) \right] \\
= h_{ij}x_j^{(i)} + 1/m \left( y_i - \sum_{k=1}^{m} h_{ik}x_k^{(i)} \right), \quad i = 1, \ldots, n, \quad j = 1, \ldots, m. \quad (3.3)
\]

In M-step, substituting \( \hat{z}_{ij}^{(i+1)} \) with the equation (3.3) in the normal equation \( 0 = \partial Q(x; x^{(i)})/\partial x_j \), we have

\[
0 = \sum_{i=1}^{n} \left( h_{ij}z_{ij}^{(i+1)} - h_{ij}^2x_j \right) - \frac{\lambda}{m} \sum_{k=1}^{m} w_{jk} x_k \\
= x_j^{(i)} \sum_{i=1}^{n} h_{ij}^2 + \frac{1}{m} \sum_{i=1}^{n} h_{ij} \left( y_i - \sum_{k=1}^{m} h_{ik}x_k^{(i)} \right) - x_j \sum_{i=1}^{n} h_{ij}^2 - \frac{\lambda}{m} \sum_{k=1}^{m} w_{jk} x_k, \quad j = 1, \ldots, m. \quad (3.4)
\]
where $\lambda = \sigma^2 / \sigma^2_x$ is used a user-specified tuning constant in this paper. Equivalently, equation (3.4) can be expressed in matrix form as

$$0 = x^{(t)} + vH^T(y - Hx^{(t)}) - x - \lambda vWx,$$

(3.5)

where $v^{-1} = m \sum_{i=1}^n h^2_{ij}$. Note that $\sum_{i=1}^n h^2_{ij}$ is the sum of square of all elements in the $j$th column of the convolution matrix $H$, and they are same for all columns, thus $v$ does not depend on $j$.

At hence, at $(t + 1)$th iteration, we obtain the EM deconvolution solution as

$$x^{(t+1)} = (I + \lambda vW)^{-1} \left( x^{(t)} + vH^T(y - Hx^{(t)}) \right)$$

$$= D^{-1}_\lambda \left( I - vH^T H \right) x^{(t)} + vD^{-1}_\lambda H^T y,$$

(3.6)

where $D_\lambda = I + \lambda vW$.

As an alternative, by applying the Green (1990)'s One-Step-Late (OSL) method, one can also use $\lambda vWx^{(t)}$ instead of $\lambda vWx$ in equation (3.5). Then we have

$$x^{(t+1)} = x^{(t)} + vH^T(y - Hx^{(t)}) - \lambda vWx^{(t)},$$

$$= \left[ I - vH^T H + \lambda W \right] x^{(t)} + vH^T y.$$  

(3.7)

By the essential property of EM algorithm, proceeding iteration monotonically increases the penalized log-likelihood $L_\lambda(x)$ for the incomplete data $y$.

If the image $y$ is very small, the matrix-typed algorithms (3.6) and (3.7) can be used as they are, but otherwise, the calculation is performed via the Fourier transform $F$ because the matrix operation time is quite burdensome. Let’s $\tilde{Y} = F(y)$, $\tilde{H} = F(h)$, $\tilde{X}_t = F(x_t)$, and $\tilde{K} = F(k)$. Calculate

$$\tilde{A} = \begin{cases} (1 + vH^T H) & \text{for (3.6)} \\
1 - vH^T H + \lambda \tilde{W} & \text{for (3.7)}, \end{cases}$$

(3.8)

$$\tilde{b} = \begin{cases} v\tilde{H}Y & \text{for (3.6)} \\
v\tilde{H} \tilde{Y} & \text{for (3.7)}, \end{cases}$$

(3.9)

$$\tilde{X}_{t+1} = \tilde{A} \tilde{X}_t + \tilde{b},$$

(3.10)

where $\tilde{W} = \tilde{K}^* \tilde{K}$, $\tilde{D} = 1 + \lambda \tilde{W}$, and $^*$ indicates the conjugation operator. Then, convert into the spatial domain as

$$x^{(t+1)} = \left| F^{-1} \left( \tilde{X}_{t+1} \right) \right|.$$  

(3.11)

As mentioned previously, however, algorithm (3.6) and (3.7) have very slow convergence speed. Moreover, if the image $y$ is large, the processing time is impractically expensive. If a very long iteration is required to obtain a desired image, the practicality of the EM algorithm will be reduced. In order to overcome this problem, in the next section, a method to obtain an immediate solution of the EM algorithm without performing iterations is suggested.

4. Immediate solution of EM algorithm

We can rewrite the equation (3.6) and (3.7) as follows,

$$x^{(t+1)} = A_t x^{(t)} + b_t,$$

(4.1)
saying that the current estimates \( x^{(t+1)} \) is a linear transformation of the just preceding estimates \( x^{(t)} \), where for the equation (3.6)

\[
A_\lambda = D_\lambda^{-1}(I - \nu H^T H),
\]

\[
b_\lambda = \nu D_\lambda^{-1} H^T y,
\]

(4.2)

and for the equation (3.7)

\[
A_\lambda = I - \nu \left(H^T H + \lambda W \right),
\]

\[
b_\lambda = \nu H^T y.
\]

(4.3)

For an initial estimates \( x^{(0)} \), one can show that

\[
x^{(t+1)} = A_\lambda x^{(t)} + b_\lambda
\]

\[
= A_\lambda^{t+1} x^{(0)} + \left( A_\lambda^t + A_\lambda^{t-1} + \cdots + I \right) b_\lambda,
\]

(4.4)

where \( A_\lambda^t = A_\lambda \cdots A_\lambda \) that is multiplied \( t \) times, and denote \( A_\lambda^0 = I \).

Note that if \( 0 < |A_\lambda| < 1 \), then the series \( \sum_{t=0}^{\infty} A_\lambda^t = (I - A_\lambda)^{-1} \), and \( A_\lambda^t \to 0 \) as \( t \to \infty \) (Searle, 1982). Thus \( \sum_{t=0}^{\infty} A_\lambda^t \to (I - A_\lambda)^{-1} \).

Accordingly, our immediate EM solution after \( t \) iteration is

\[
x_{\text{EM}}(t) = A_\lambda^t x^{(0)} + \left( I - A_\lambda \right)^{-1} b_\lambda
\]

(4.5)

In our case, it is straightforward to show that \(|A_\lambda| < 1\) is satisfied for a sufficiently small \( \nu \). For the equation (4.2), since \( D_\lambda = I + \lambda \nu W \) is positive definite, \(|A_\lambda| = |I - \nu H^T H|/|D_\lambda| < |I - \nu H^T H| \). Here, since \( \nu H^T H \) positive definite, its eigenvalues are within \([0, 1]\) for a small \( \nu \), therefore \(|I - \nu H^T H| \leq 1\), and then \(|A_\lambda| < 1\). In similar manner for the equation (4.3), since \( H^T H + \lambda W \) is also symmetric and positive definite, \(|A_\lambda| = |I - \nu \left(H^T H + \lambda W \right)| < 1\) for a sufficiently small \( \nu \). In this paper, it is recommended to choose \( 0 < \nu \leq 1\) because it worked well without divergence in various experiments. Actually, in our algorithm \( \nu \) can be used as a user-specified convergence step.

From (4.5), we find \( x_{\text{EM}}(\infty) = (I - A_\lambda)^{-1} b_\lambda \) because \( A_\lambda^\infty = 0 \), and easily show (for both algorithm (4.2) and (4.3))) that

\[
(I - A_\lambda)^{-1} b_\lambda = \left( H^T H + \lambda W \right)^{-1} H^T y = \hat{x}_\lambda
\]

(4.6)

which is just the solution (2.5). Consequently, the EM solution says that

\[
x_{\text{EM}}(t) = A_\lambda^t x^{(0)} + \left( I - A_\lambda \right) \hat{x}_\lambda,
\]

(4.7)

which is a mixture of the initial estimates and the limit estimates where the proportion rate is the “weight matrix” \( A_\lambda^t \) at the \( t^{th} \) iteration. As \( t \) increases, it moves away from the initial estimates \( x^{(0)} \) and approaches the limit estimates \( \hat{x}_\lambda \). Also, the convergency of this algorithm does not depend on initial estimates even when \( x^{(0)} = 0 \).

It is worth investigating the bias and variance of \( x_{\text{EM}}(t) \). For the true image \( x \)

\[
\text{Bias}(x_{\text{EM}}(t)) = x - A_\lambda^t E \left( x^{(0)} \right) - \left( I - A_\lambda \right) E(\hat{x}_\lambda)
\]

\[
= \left[ x - E(\hat{x}_\lambda) \right] + A_\lambda^t \left[ \left[ x - E \left( x^{(0)} \right) \right] - \left[ x - E(\hat{x}_\lambda) \right] \right]
\]

(4.8)

= bias of the limit estimates + \( A_\lambda^t \times \) bias increments from initials to limits.
Since the limit estimator \( \hat{x}_1 \) is always determined irrespective of the initial estimator \( x^{(0)} \), they are independent of each other. Therefore, the variance of \( x^{EM}(t) \) is

\[
\text{var}(x^{EM}(t)) = A_t' \text{var} \left( x^{(0)} \right) \left( A_t' \right)^T + (I - A_t') \text{var} \left( \hat{x}_1 \right) \left( I - A_t' \right)^T
\]

(4.9)

\[
= \text{a mixture of variance between initial and limit estimator.}
\]

From equation (4.8) and (4.9), it can be seen that the bias and variance of the initial estimator \( x^{(0)} \) is heavily reflected at the beginning of the algorithm, but eventually those of the limit estimator \( \hat{x}_1 \) are reached along with \( A_t' \to 0 \) as \( t \to \infty \).

Consider an initial estimator where the bias is greater than \( \hat{x}_1 \) and its variance is zero, for an example, an totally flat image like \( x^{(0)} = 0 \). Then, since the bias increments are positive, a restored image with a greater bias than \( \hat{x}_1 \) is always provided during iteration (i.e. more smooth image). On the other hand, its variance becomes gradually larger (i.e. more sharpen image).

From these opposite properties, it can be expected to obtain a visually more satisfactory restored image than the \( x^{(0)} \) or the \( \hat{x}_1 \) while compromising bias and variance during iteration. This explains why many users of EM algorithms for image deconvolution adopt a strategy of stopping in the middle of iterations. However, the problem is that it takes too much processing time to get to that point using iterative EM. The proposed immediate EM solves this problem.

Meanwhile, the Fourier version of \( x^{EM}(t) \) of equation (4.5) is

\[
\tilde{A}' = (\tilde{A})' \\
\tilde{X}_t = \tilde{A}' \tilde{X}_0 + \left( \frac{1 - \tilde{A}'}{1 - \tilde{A}} \right) \tilde{b},
\]

(4.10)

where \( \tilde{A}, \tilde{b} \) and \( \tilde{X}_0 \) are denoted in Section 3. Then we obtain the immediate EM restoration image as \( x^{EM}(t) = \left| \mathcal{F}^{-1}(\tilde{X}_t) \right| \).

It should be further noted that our \( x^{EM}(t) \) allows the number of iterations \( t \) to be used a real numbers, but also an integer. For example, it can be seen that a restoration image \( x^{EM}(2.5) \) provides about halfway between the 2nd and 3rd iteration. Another advantage of our method is that it allows us to examine the results of the restored image sequentially for the desired number of iterations. For example, you can get the restoration results: starting with 1000 iteration image, up to 2000 iteration image with increasing by 100 iterations.

5. Experiments

In this section, we use real images to show that immediate EM works properly. In Figure 1, a true image \( x \), a blur kernel \( h \), and the observed blur image \( y \) are provided. The \( x \) is a 512 \times 512 RGB image where pixels are scaled between \([0, 1]\). The \( h \) a 49 \times 49 uniform blur kernel with hight 0.0016 and its sum is 1. And the \( y \) is an observed image obtained by adding Gaussian noise with a standard deviation of \( \sigma_e = 0.001 \) to the result of \( x \otimes h \equiv Hx \).

Our goal is to reconstruct the distorted image \( y \) under a given \( H \) (that is, the matrix version of the blur filter \( h \)) to be close to the true image \( x \). Reconstruction was performed separately for each R, G, and B channel, and then the results were merged.

In this experiment, the penalty term in Equation (2.5) will be used a simple “roughness penalty”
Immediate EM solution for non-blind deconvolution

Figure 1: True image \( x \) (Left). Blur filter \( h \) (Middle), where all white area has a pixel value of 0.0016. Observed image \( y \) (Right) where \( x \) is convolved with a blur filter \( h \) and then added gaussian noise with \( \sigma \).

Figure 2: Penalized log-likelihood \( L_1(x^{(t)}) \) over the iteration. Original EM equation (3.6) (Blue), OSL–EM equation (3.7) (Red). ◯: \( \nu = 0.2 \), △: \( \nu = 0.5 \), and + : \( \nu = 1.0 \).

for the purpose of removing noise

\[
x^T W x = \sum_{j=1}^{m} (x_j - \bar{x}_0_j)^2 = \| k \otimes x \|^2,
\]

where \( \bar{x}_0_j \) denotes the average of neighbors at the pixel \( j \) of \( x \), and \( \partial_j \) denotes the 1st order neighbor-
hood,

\[
k = \frac{1}{4} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}.
\]

And the initial estimates of immediate EM is \( x^{(0)} = 0 \).

Figure 2 demonstrates the log-likelihood of the EM algorithm over iteration, where it monotonically increases in all cases. In particular, the original EM of equation (3.6) and OSL-EM of equation (3.7) showed very insignificant difference. And the larger the value of \( v \), the faster the convergence result at the beginning of iteration, but a similar convergence speed near the convergence point.

The first row of Figure 3 provides the RL deconvolution results for 50, 1500, and 2500 iterations (using function \texttt{richardson-lucy} in python module \texttt{skimage}). And similar results were obtained for more iterations. Although it worked well for deblur and denoising, "ringing artifacts", a problem unique to this technique, appear. First of all, the processing time was long, taking 342.71 seconds for 2500 iterations.

In the second row of Figure 3, Wiener deconvolution results are provided (using function \texttt{wiener} in python module \texttt{skimage}). The first and third ones are reconstructed using \( \lambda = 0.1 \) and \( \lambda = 0.17 \) in common for R, G, and B channel images, respectively. And the second one is with exact but unknown channel-wise different \( \lambda \),

\[
(\lambda_R, \lambda_G, \lambda_B) \approx (2.65, 2.27, 5.46) \times 10^{-5}
\]

which shows better result. If the correct \( \lambda \) is not specified, it gives undesirable results.

In the third row of Figure 3, our immediate EM deconvolution results are provided using \( \lambda = 0.17 \) that showed a bad result with Wiener deconvolution. The 1st and 3rd image are \( x_{EM}^{(1)} \) and \( x_{EM}^{(\infty)} \) where they are represent a high biased and high variance, respectively. A good restored image (the 2nd) was obtained by choosing tentatively \( t = 50000 \), although it not reached at maximum likelihood. As expected, it confirms that subjectively and visually satisfactory images can exist during the EM algorithm. However, since the processing time for one iteration is about 0.34 seconds, it is expected that it will take about 4.7 hours to obtain the image \( x_{EM}^{(50000)} \) with the iterative EM algorithm. On the other hand, the proposed immediate EM \( x_{EM}^{(50000)} \) was obtained in only 0.26 seconds.

The PC used in the experiment in this chapter is Intel(R) i7-8700 CPU (3.20GHz), and the program is implemented in python.

6. Conclusions and discussion

In this paper, a non-blind deconvolution by iterative EM algorithm under Gaussian model was provided. Then an immediate EM without iterative processing that provides the same result of iterative EM was developed. This significantly reduces the processing time of image deconvolution with EM algorithm, thereby increasing the practicality. The result makes it possible to reveal the statistical properties of bias and variance for the restored image. This has hitherto been vaguely recognized but not explicitly known.

The Richardson-Lucy method is an EM algorithm based on Poisson model, and it is necessary to study whether an immediate solution can be obtained for this method as well.
Immediate EM solution for non-blind deconvolution

Figure 3: The 1st row displays LR deconvolution: iter=50(left), iter=150(middle), and iter=2500(right). The 2nd row displays Wiener deconvolution: $\lambda = 0.1$(left) in common, accurate $\lambda = \sigma_e^2/(\sigma_{R}^2, \sigma_{G}^2, \sigma_{B}^2) = (2.65, 2.27, 5.46) \times 10^{-5}$, and $\lambda = 0.17$(right) in common. The 3rd row displays the immediate EM deconvolution: $EM_{0.17}(1)$(left), $EM_{0.17}(50,000)$(middle), and the limit estimates $EM_{0.17}(\infty) = \hat{x}_{0.17}$(right).

In this study, very limited experiments were provided and an excessively simple penalty term was used. The use of various initial images and edge preserving penalty is expected to provide more interesting results.

References

Received January 07, 2022; Revised February 09, 2022; Accepted March 10, 2022