Journal of the Association of Arab Universities for Basic and Applied Sciences (2014) **16**, 27–33

University of Bahrain

Journal of the Association of Arab Universities for Basic and Applied Sciences

www.elsevier.com/locate/jaaubas
www.sciencedirect.com

طريقة تفكيك جديدة لحل مجموعة من المعادلات الخطية

Muhammad Aslam Noor[1,*], Khalida Inayat Noor[1], Muhammad Waseem[2]

[1]Department of Mathematics, COMSATS Institute of Information Technology, Park Road, Islamabad, Pakistan
[2]Department of Mathematics, COMSATS Institute of Information Technology, Sahiwal, Pakista

الملخص:

في هذا البحث تم استخدام اسلوب تفكيك جديد لاقتراح واعتبار بعض الطرق التكرارية الجديدة لحل مجموعة من المعادلات الخطية. لقد قمنا باثبات أن هذه الطرق التكرارية هي مشابه لتلك الطرق التكرارية المتحصل عليها باستخدام طريقة اضطراب هوموتوبي (Homotopy) وطريقة تفكك ادوميان (Adomian). لقد تم توضيح اداء وكفاءة النتائج باستخدام معادلة التفاضل الجزئي القطع ناقصة وأمثلة أخرى عددية. ان نتائج هذه الدراسة يمكن ان ينظر اليها بانها عبارة عن تحسين وامتداد للنتائج المعروفة في السابق.

Journal of the Association of Arab Universities for Basic and Applied Sciences (2014) 16, 27–33

University of Bahrain
Journal of the Association of Arab Universities for Basic and Applied Sciences
www.elsevier.com/locate/jaaubas
www.sciencedirect.com

ORIGINAL ARTICLE

A new decomposition technique for solving a system of linear equations

Muhammad Aslam Noor [a,*], Khalida Inayat Noor [a], Muhammad Waseem [b]

[a] *Department of Mathematics, COMSATS Institute of Information Technology, Park Road, Islamabad, Pakistan*
[b] *Department of Mathematics, COMSATS Institute of Information Technology, Sahiwal, Pakistan*

Received 10 September 2012; revised 27 June 2013; accepted 1 July 2013
Available online 27 July 2013

KEYWORDS

Splitting matrix;
Iterative method;
Decomposition technique;
Auxiliary parameter;
System of equations

Abstract In this paper, we use a new decomposition technique to suggest and consider some new iterative methods for solving system of linear equations. We prove that these iterative methods are similar to the iterative methods derived by using homotopy perturbation method and Adomian decomposition method. We consider the elliptic partial differential equation along with other several numerical examples to illustrate the efficiency and performance of our results. Our results can be viewed as an improvement and extensions of the previously known results.

1. Introduction

In recent years, several methods and techniques have been developed to solve system of linear equations. Liu (2011), Keramati (2009), Noor (2010) and Noor et al. (2013a) have used homotopy perturbation method to derive iterative methods for solving linear (nonlinear) equations. Noor et al. (2013b) have used Adomian decomposition method to develop iterative methods for system of linear equations. Babolian et al. (2004) have used Adomian decomposition method to derive an iterative method similar to the Jacobi iterative method for solving system of linear equations. Allahviranloo (2005) used Adomian decomposition method for fuzzy system of linear equations. There are many publication in the field of analytical surveys using homotopy perturbation methods and other techniques, see, for example, Ganji (2006, 2012), Ganji and Sadighi (2006), Jalal and Ganji (2010, 2011) and Jalal et al. (2010, 2012). In the implementation of the Adomian (1989, 1994) decomposition method, one has to calculate the derivatives of the so-called Adomian polynomials, which is itself a difficult problem. To overcome this drawback, we use a different type of decomposition which is essentially due to Daftardar-Gejji and Jafari (2006), to develop the iterative methods for solving the system of linear equations. Noor (2006,2007), Noor and Noor (2006a,b), Noor et al. (2006c) and Noor et al. (2010a,b) have used the same decomposition technique for solving nonlinear equations. This decomposition method does not involve the high-order differentials of the function and is very simple as compared with Adomian decomposition technique. In this paper, we use this new decomposition method to develop iterative methods for solving system of linear equations. We show that our results obtained by using new decomposition technique are the same as derived by Liu (2011) and

* Corresponding author. Tel.: +92 3454027532.
E-mail addresses: noormaslam@hotmail.com (M.A. Noor), khalidanoor@hotmail.com (K.I. Noor), sattarmwaseem@gmail.com (M. Waseem).
Peer review under responsibility of University of Bahrain.

1815-3852 © 2013 Production and hosting by Elsevier B.V. on behalf of University of Bahrain.
http://dx.doi.org/10.1016/j.jaubas.2013.07.001

Noor et al. (2013b) by using the homotopy perturbation method and Adomian decomposition method, respectively. This is the main motivation of this paper. By using new iterative methods we solve the elliptic partial differential equation and it is well known that the elliptic partial differential equations have applications in almost all areas of mathematics and frequent applications in engineering and physics. We also give several numerical examples to demonstrate the efficiency and performance of our results.

2. Iterative methods

Consider the system of linear equations

$$AX = b, \tag{2.1}$$

where

$A = [a_{ij}], \quad X = [x_j] \quad \text{and } b = [b_j], \quad i = 1, 2, \cdots, n,$
$j = 1, 2, \cdots, n.$

It is well-known that systems of linear Eq. (2.1) arise in studies in many areas such as engineering, industrial science and so on. For example, in digital image and signal processing, especially in compressed sensing, biomedical engineering, systems and control science, machine learning and so on. For the formulation and applications of the system of linear Eq. (2.1), see Burden and Faires (2001) and the references therein. We decompose the system of linear Eq. (2.1) in such way, which is useful in developing the iterative methods. For an auxiliary parameter $\hbar \neq 0$, any splitting matrix Q and an auxiliary matrix H, we can decompose the system of linear Eq. (2.1) as follows:

$$QX + (\hbar HA - Q)X = \hbar Hb. \tag{2.2}$$

Let W_0 be the initial approximation of X, then, Eq. (2.2) can be written as:

$$QX = W_0 + [(Q - \hbar HA)X + \hbar Hb - W_0] \tag{2.3}$$

Eq. (2.3) can be written as:

$$L(X) = C + M(X), \tag{2.4}$$

where

$$L(X) = QX, \tag{2.5}$$

$$C = W_0, \tag{2.6}$$

$$M(X) = [(Q - \hbar HA)X + \hbar Hb - W_0] \tag{2.7}$$

Here we use a new decomposition technique, which is mainly due to Daftardar-Gejji and Jafari (2006), to construct a family of iterative methods. In this technique, the decomposition of the operator $M(X)$ is quite different than that of Adomian (1989, 1994) decomposition. See also He (1999), Babolian et al. (2004) and Yusufoglu (2009) for other techniques.

The main idea of this technique is to look for a solution of Eq. (2.4) having the series form

$$X = \sum_{i=0}^{\infty} X_i. \tag{2.8}$$

The operator M is decomposed (Daftardar-Gejji and Jafari (2006)) as

$$M(X) = M(X_0) + \sum_{i=1}^{\infty}\left\{M\left(\sum_{j=0}^{i} X_j\right) - M\left(\sum_{j=0}^{i-1} X_j\right)\right\} \tag{2.9}$$

Combining (2.4), (2.8), and (2.9), we have

$$L\left(\sum_{i=0}^{\infty} X_i\right) = C + M(X_0) + \sum_{i=1}^{\infty}\left\{M\left(\sum_{j=0}^{i} X_j\right) - M\left(\sum_{j=0}^{i-1} X_j\right)\right\} \tag{2.10}$$

By using (2.5) and (2.10), we have

$$Q\left(\sum_{i=0}^{\infty} X_i\right) = C + M(X_0) + \sum_{i=1}^{\infty}\left\{M\left(\sum_{j=0}^{i} X_j\right) - M\left(\sum_{j=0}^{i-1} X_j\right)\right\} \tag{2.11}$$

Thus, we have the following iterative scheme:

$$\begin{aligned} Q(X_0) &= C, \\ Q(X_1) &= M(X_0), \\ Q(X_2) &= M(X_0 + X_1) - M(X_0), \\ &\vdots \\ Q(X_{m+1}) &= M\left(\sum_{j=0}^{m} X_j\right) - M\left(\sum_{j=0}^{m-1} X_j\right). \quad m = 1, 2, \cdots. \end{aligned} \tag{2.12}$$

From (2.7) and (2.12), we have

$$\begin{aligned} &Q(X_0) = C, \\ &Q(X_1) = (Q - \hbar HA)X_0 + \hbar Hb - W_0, \\ &Q(X_2) = (Q - \hbar HA)X_1, \\ &\vdots \\ &Q(X_{m+1}) = (Q - \hbar HA)X_m, \quad m = 1, 2, \cdots. \end{aligned} \tag{2.13}$$

From (2.13), we get

$$\begin{cases} X_0 = Q^{-1}W_0, \\ X_1 = (I - \hbar Q^{-1}HA)X_0 + Q^{-1}(\hbar Hb - W_0), \\ X_m = (I - \hbar Q^{-1}HA)X_{m-1}, \quad m = 2, 3, 4, \cdots. \end{cases} \tag{2.14}$$

Taking initial approximation $W_0 = \hbar Hb$, we have

$$\begin{cases} X_0 = \hbar(Q^{-1}H)b, \\ X_m = (I - \hbar Q^{-1}HA)^m \hbar(Q^{-1}H)b, \quad m = 1, 2, 3, \cdots. \end{cases} \tag{2.15}$$

Thus, from (2.8) and (2.15), we have the series solution

$$X = \sum_{k=0}^{\infty} X_k = \sum_{k=0}^{\infty} (I - \hbar Q^{-1}HA)^k \hbar(Q^{-1}H)b. \tag{2.16}$$

Formula (2.16) gives exactly the same series solution obtained by using homotopy perturbation technique in Liu (2011) and Adomian decomposition method Noor et al. (2013b). However, our technique of the derivation of the series solution is quite easy and natural one. This technique does not involve the computation of the Adomian polynomials, which is itself a difficult problem. For the convergence analysis of series (2.16), see Liu (2011). The series (2.16) converges if and only if $\rho(I - \hbar Q^{-1}HA) < 1$, see Liu (2011). The auxiliary parameter $\hbar \neq 0$, and the auxiliary matrix H are chosen properly so that the series (2.16) converges.

Using (2.16), we suggest the following iterative scheme:

Algorithm 2.1. For an initial value $X^{(0)} = \hbar(Q^{-1}H)b$, compute the approximate solution $X^{(k)}$ by the following iterative scheme:

$$X^{(k)} = \sum_{m=1}^{k}(I - \hbar Q^{-1}HA)^{m}\hbar(Q^{-1}H)b, \quad k = 1, 2, 3, \cdots.$$

We now discuss some special cases, which can be obtained from our new Algorithm 2.1.I. If $Q = D$, where

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{n,n} \end{bmatrix},$$

then, Algorithm 2.1 reduces to the following method:

Algorithm 2.2. For an initial value $X^{(0)} = \hbar(D^{-1}H)b$, compute the approximate solution $X^{(k)}$ by the following iterative scheme:

$$X^{(k)} = \sum_{m=1}^{k}(I - \hbar D^{-1}HA)^{m}\hbar(D^{-1}H)b, \quad k = 1, 2, 3, \cdots.$$

II. If $Q = D - \hbar L$, where

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{n,n} \end{bmatrix} \text{ and } L = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix},$$

then, Algorithm 2.1 reduces to the following method:

Algorithm 2.3. For an initial value $X^{(0)} = \hbar((D - \hbar L)^{-1}H)b$, compute the approximate solution $X^{(k)}$ by the following iterative scheme:

$$X^{(k)} = \sum_{m=1}^{k}(I - \hbar(D - \hbar L)^{-1}HA)^{m}\hbar(D - \hbar L)^{-1}Hb,$$

$k = 1, 2, 3, \cdots$.

3. Numerical examples

In this section, we consider the elliptic partial differential equation and some other numerical examples to illustrate the efficiency of the newly developed methods in this paper. We compare the Jacobi method (JC) and Gauss–Seidel method (GS) (Burden and Faires (2001)), Algorithm 2.2 and Algorithm 2.3 for different values of ħ and taking $H = I$. For each example, we calculate number of iterations, error estimate, spectral radius $\rho(D^{-1}(L + U))$ (for Jacobi method), $\rho((D - L)^{-1}U)$ (for Gauss–Seidel method), $\rho(I - \hbar D^{-1}HA)$ (for Algorithm 2.2) and $\rho(I - \hbar(D - \hbar L)^{-1}HA)$ (for Algorithm 2.3). All computations are done on MATLAB. We use $\varepsilon = 10^{-15}$. The following stopping criteria is used for computer programs:

$$\frac{\|X^{(k)} - X^{(k-1)}\|_{\infty}}{\|X^{(k)}\|_{\infty}} < \varepsilon.$$

We consider the following examples to illustrate the implementation of the iterative methods.

Example 3.1. (Burden and Faires (2001)) Consider the following system of linear equations $AX = b$, such that

$$\begin{bmatrix} -1 & 0 & 0 & \frac{\sqrt{2}}{2} & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & 0 & -1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & 0 & 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\sqrt{3}}{2} & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10,000 \\ 0 \\ 0 \end{bmatrix}.$$

The numerical results for this example are presented in Table 3.1. It is clear from the Table 3.1 that Algorithm 2.3 is more efficient as compared to all other methods in the table. The optimal value of ħ for Algorithm 2.3 is 1.21 and for Algorithm 2.2 is 0.99. Residual fall of different methods is also shown by Fig. 3.1.

Example 3.2. (Burden and Faires (2001)) Consider the following system of linear equations $AX = b$, such that

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,40} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,40} \\ \vdots & \vdots & \ddots & \vdots \\ a_{40,1} & a_{40,2} & \cdots & a_{40,40} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{40} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{40} \end{bmatrix}$$

where the entries of matrix A are

$$a_{ij} = \begin{cases} 2i, & \text{when } j = i \text{ and } i = 1, 2, \cdots, 40, \\ -1, & \text{when} \begin{cases} j = i + 1 \text{ and } , i = 1, 2, \cdots, 39, \\ j = i - 1 \text{ and } i = 2, 3, \cdots, 40, \end{cases} \\ 0, & \text{otherwise,} \end{cases}$$

and the entries column matrix b are $b_i = 1.5i - 6$, foreach, $i = 1, 2, \cdots, 40$.

Table 3.1 (Numerical results for Example 3.1).

Method	ħ	ρ	IT	Error
JC		0.7598	127	8.6979e–016
GS		0.5774	64	8.6979e–016
Alg. 2.2	ħ = 1	0.7598	127	5.3264e–016
	ħ = 0.99	0.7622	124	9.3211e–016
	ħ = 0.98	0.7646	126	7.9896e–016
Alg. 2.3	ħ = 1	0.5774	63	6.1504e–016
	ħ = 1.2	0.2965	30	4.6128e–016
	ħ = 1.21	0.2461	29	7.3930e–016
	ħ = 1.22	0.2200	30	6.8395e–016
	ħ = 1.25	0.2500	33	6.3323e–016

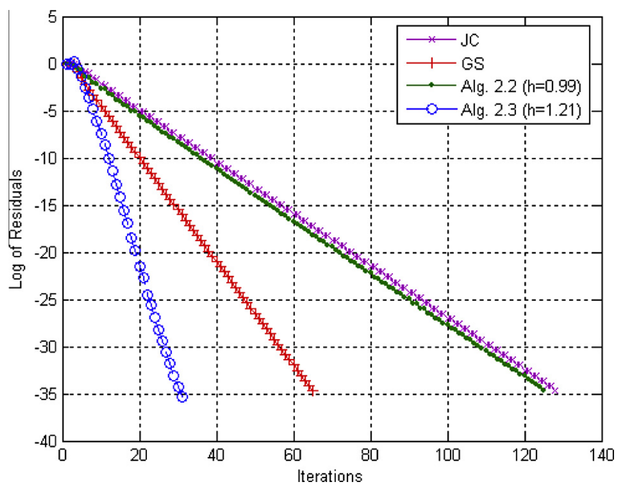

Figure 3.1 Residual fall for Example 3.1.

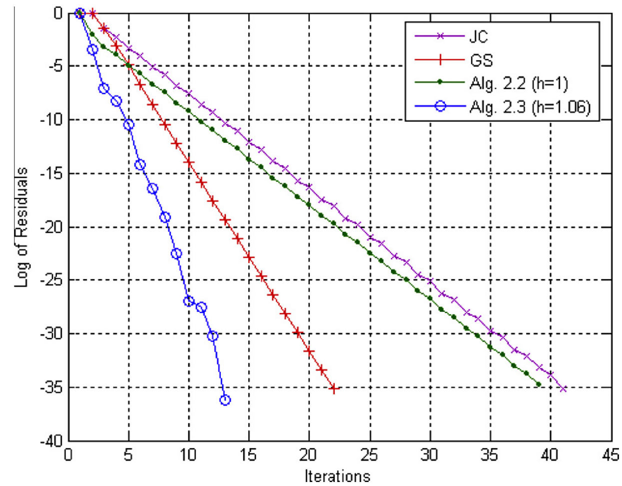

Figure 3.2 Residual fall for Example 3.2.

More precisely the above system of linear equations is given by

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 6 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 8 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & -1 & 10 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 12 & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & -1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & -1 & 76 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 78 & -1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 & 80 \end{bmatrix}_{(40\times 40)} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \vdots \\ x_{38} \\ x_{39} \\ x_{40} \end{bmatrix}_{(40\times 1)} = \begin{bmatrix} -4.5 \\ -3 \\ -1.5 \\ 0 \\ 1.5 \\ 3 \\ \vdots \\ 51 \\ 52.5 \\ 54 \end{bmatrix}_{(40\times 1)}.$$

The numerical results for the above system of linear equations are presented in Table 3.2. Table 3.2 shows that the spectral radius ρ for Algorithm 2.3 is 0.06 at $\hbar = 1.06$ which is less than the spectral radius ρ of all other methods presented in the Table 3.2. And hence Algorithm 2.3 is more efficient as compared to all other methods. The optimal value of $\hbar$ for Algorithm 2.3 is 1.06 and for Algorithm 2.2 is 1. Fig. 3.2 shows the residual fall of different method for this problem.

Table 3.2 (Numerical results for Example 3.2).

Method	$\hbar$	ρ	IT	Error
JC		0.4158	40	5.7747e–016
GS		0.1729	21	5.7747e–016
Alg. 2.2	$\hbar = 1$	0.4158	38	7.3984e–016
	$\hbar = 0.99$	0.4217	38	9.9190e–016
	$\hbar = 0.98$	0.4275	39	8.3105e–016
Alg. 2.3	$\hbar = 1$	0.1729	19	7.3407e–016
	$\hbar = 1.06$	0.0600	12	1.8031e–016
	$\hbar = 1.07$	0.0700	13	8.9869e–016
	$\hbar = 1.1$	0.1000	15	5.3386e–016
	$\hbar = 1.15$	0.1500	18	1.7480e–016

Example 3.3. Consider the following system of linear equations $AX = b$, such that

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,16} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,16} \\ \vdots & \vdots & \ddots & \vdots \\ a_{16,1} & a_{16,2} & \cdots & a_{16,16} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{16} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{16} \end{bmatrix}.$$

where the entries of matrix A are

$$a_{ij} = \begin{cases} 4, & \text{when } j = i \text{ and } i = 1,2,\cdots,16, \\ -1, & \text{when} \begin{cases} j = i+1 \text{ and } i = 1,2,3,5,6,7,9,10,11,13,14,15, \\ j = i-1 \text{ and } i = 2,3,4,6,7,8,10,11,12,14,15,16, \\ j = i+4 \text{ and } i = 1,2,\cdots,12, \\ j = i-4 \text{ and } i = 5,6,\cdots,16, \end{cases} \\ 0, & \text{otherwise,} \end{cases}$$

and the entries column matrix b are

$$b = (1.902207, 1.051143, 1.175689, 3.480083, 0.819600, -0.264419, -0.412789, 1.175689, 0.913337, -0.150209, -0.264419, 1.051143, 1.966694, 0.913337, 0.819600, 1.902207)^t.$$

The numerical results for this problem are presented in Table 3.3. Residual fall of different method for this problem

Table 3.3 (Numerical results for Example 3.3).

Method	ħ	ρ	IT	Error
JC		0.8090	156	9.3593e–016
GS		0.6545	81	8.5793e–016
Alg. 2.2	ħ = 1	0.8090	155	8.6973e–016
	ħ = 0.99	0.8109	156	9.6098e–016
	ħ = 0.98	0.8128	158	8.7745e–016
Alg. 2.3	ħ = 1	0.6545	80	6.3921e–016
	ħ = 1.25	0.3375	33	4.8150e–016
	ħ = 1.26	0.2600	28	8.7725e–016
	ħ = 1.28	0.2800	29	2.3622e–016
	ħ = 1.31	0.3100	31	4.6328e–016

Table 3.4 (Numerical results for elliptic partial differential equation).

Method	ħ	ρ	IT	Error
JC		0.9380	495	9.6128e–016
GS		0.8807	255	9.6128e–016
Alg. 2.2	ħ = 1	0.9380	493	8.8029e–016
	ħ = 0.99	0.9386	498	9.6817e–016
	ħ = 0.98	0.9392	503	9.6801e–016
Alg. 2.3	ħ = 1	0.8807	254	8.3853e–016
	ħ = 1.48	0.6247	72	6.0639e–016
	ħ = 1.49	0.6098	69	6.0164e–016
	ħ = 1.5	0.6175	68	4.6898e–016
	ħ = 1.51	0.6252	69	8.0360e–016

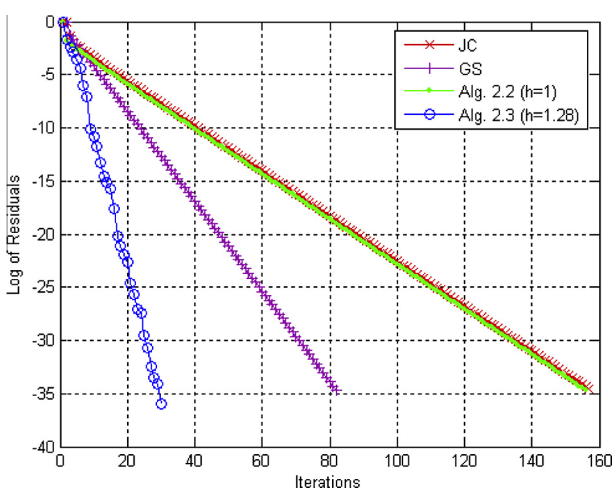

Figure 3.3 Residual fall for Example 3.3.

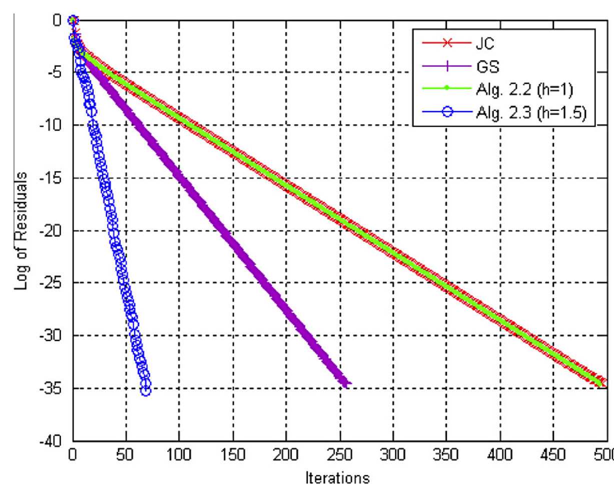

Figure 3.4 Residual fall for Example 3.4.

is also shown in Fig. 3.3. It is clear from Table 3.3 and Fig. 3.3 that Algorithm 2.3 is much efficient at optimal value ħ = 1.26.

Example 3.4. Consider the following elliptic partial differential equation

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=\frac{x}{y}+\frac{y}{x},\quad 1\leqslant x\leqslant 2,\quad 1\leqslant y\leqslant 2, \tag{3.1}$$

with

$$u(x,1)=x\ln x, u(x,2)=x\ln(4x^2), 1\leqslant x\leqslant 2,$$
$$u(1,y)=y\ln y, u(2,y)=2y\ln(2y), 1\leqslant y\leqslant 2.$$

Consider the following partition of the intervals:

$$x_0=1<x_1<x_2<\cdots<x_n=2,\quad x_i=x_0+ih,\quad h=\frac{1}{n},$$

and

$$y_0=1<y_1<y_2<\cdots<y_m=2,\quad y_j=y_0+jk,\quad k=\frac{1}{m}.$$

For each mesh point in the interior of the grid, (x_i, y_j), for each $i = 1,2,\cdots,n-1$ and $j = 1,2,\cdots,m-1$, we use the Taylor series in the variable x about x_i to generate the centered-difference formula:

$$\frac{\partial^2 u(x_i,y_j)}{\partial x^2}=\frac{u(x_{i+1},y_j)-2u(x_i,y_j)+u(x_{i-1},y_j)}{h^2}. \tag{3.2}$$

We also use the Taylor series in the variable y about y_j to generate the centered-difference formula

$$\frac{\partial^2 u(x_i,y_j)}{\partial y^2}=\frac{u(x_i,y_{j+1})-2u(x_i,y_j)+u(x_i,y_{j-1})}{k^2}, \tag{3.3}$$

for each $i = 1,2,\cdots,n-1$ and $j = 1,2,\cdots,m-1$.

The boundary conditions are as under

$$u(x_0,y_j)=u(1,y_j)=y_j\ln y_j,\quad\text{and}\quad u(x_n,y_j)=u(2,y_j)$$
$$=2y_j\ln(2y_j),\quad j=0,1,2,\cdots,m,$$

and

$$u(x_i,y_0)=u(x_i,1)=x_i\ln x_i,\quad\text{and}\quad u(x_i,y_m)=u(x_i,2)$$
$$=x_i\ln(4x_i^2),\quad i=1,2,\cdots,n-1.$$

By using the formulas (3.2) and (3.3) in equation (3.1), we have

$$2\left(\left(\frac{h}{k}\right)^2+1\right)w_{i,j}-(w_{i+1,j}+w_{i-1,j})-\left(\frac{h}{k}\right)^2(w_{i,j+1}+w_{i,j-1})=-h^2\left(\frac{x_i}{y_j}+\frac{y_j}{x_i}\right). \tag{3.4}$$

The linear system involving the unknowns $w_{i,j}$ is expressed for matrix calculations more efficiently if a re-labeling of the interior mesh points is introduced, as under:

$P_l = (x_i, y_j)$ and $w_l = w_{i,j},$

where $l = i + (m-1-j)(n-1)$, for each $i = 1,2,\cdots,n-1$ and $j = 1,2,\cdots,m-1$.

In particular, we use $n = m = 10$ and obtain the following system of linear equations $AX = b$, where the entries of A are

$$a_{ij}=\begin{cases}4, & \text{when } j=i \text{ and } i=1,2,\cdots,81,\\ -1, & \text{when}\begin{cases} j=i+1 \text{ and } i=1,2,\cdots,8,10,11,\cdots,17,19,20,\cdots 26,\\ \quad ,28,29,\cdots,35,37,38,\cdots,44,46,47,\cdots,53,\\ \quad ,55,56,\cdots,62,64,65,\cdots,71,73,74,\cdots,80,\\ j=i-1 \text{ and } i=2,3,\cdots,9,11,12,\cdots,18,20,21,\cdots 27,\\ \quad ,29,30,\cdots,36,38,39,\cdots,45,47,48,\cdots,54,\\ \quad 56,57,\cdots,63,65,66,\cdots,72,74,75,\cdots,81,\\ j=i+9 \text{ and } i=1,2,\cdots,72,\\ j=i-9 \text{ and } i=10,11,\cdots,81,\end{cases}\\ 0, & \text{otherwise},\end{cases}$$

and

$$b=(2.931066376, 2.078975848, 2.462872267, 2.861994518, 3.275275463, 3.701786539, 4.140712628, 4.591332604, 10.12600811, 1.035541250, -0.02166666667, -0.02106837607, -0.02063492063, -0.02033333333, -0.02013888889, -0.02003267974, -0.02000000000, 4.591332602, 0.8801428932, -0.02122549020, -0.02072398190, -0.02037815126, -0.02015686275, -0.02003676471, -0.02000000000, -0.02003267974, 4.140712630, 0.7305853522, -0.02083333333, -0.02043269231, -0.02017857143, -0.02004166667, -0.02000000000, -0.02003676471, -0.02013888889, 3.701786539, 0.5872279653, -0.02050000000, -0.02020512821, -0.02004761905, -0.02000000000, -0.02004166667, -0.02015686275, -0.02033333333, 3.275275463, 0.4504767156, -0.02023809524, -0.02005494505, -0.02000000000, -0.02004761905, -0.02017857143, -0.02037815126, -0.02063492063, 2.861994518, 0.3207938235, -0.02006410256, -0.02000000000, -0.02005494505, -0.02020512821, -0.02043269231, -0.02072398190, -0.02106837607, 2.462872267, 0.1987101106, -0.02000000000, -0.02006410256, -0.02023809524, -0.02050000000, -0.02083333333, -0.02122549020, -0.02166666667, 2.078975848, 0.1896823956, 0.1987101106, 0.3207938235, 0.4504767156, 0.5872279653, 0.7305853522, 0.8801428932, 1.035541250, 2.931066376)^t.$$

The numerical results for above system of linear equations are presented in Table 3.4 and the Fig. 3.4 shows the residual fall of different method for this problem. It is clear from the Table 3.4 and Fig. 3.4 that Algorithm 2.3 provides results in very small number of iterations as compared to the other methods.

4. Conclusions

In this paper, we have used a new decomposition technique to derive iterative methods for solving system of linear equations. Our method of derivation of the iterative methods is very simple as compared with the Adomian decomposition technique and homotopy perturbation method. This new technique does not involve the differentiation and easy to implement. From the Tables 3.1–3.4 and Figs. 3.1–3.4, it is clear that the new iterative methods obtained in this paper perform much better than the previously known methods. The technique and ideas of this paper may be extended for solving the system of nonlinear equations, see, for example, Noor et al. (2010a,b).

Acknowledgements

The authors would like to thank Dr. S.M. Junaid Zaidi, Rector, COMSATS Institute of Information Technology, Pakistan, for providing excellent research facilities and environment. The authors are grateful to the referees for their valuable and constructive comments, which helped us to improve this paper.

References

Adomian, G., 1989. Nonlinear Stochastic Systems and Applications to Physics. Kluwer Academic Publishers, Dordrecht, Holland.

Adomian, G., 1994. Solving Frontier Problems of Physics: The Decomposition Method. Kluwer Academic Publishers, Dordrecht, Holland.

Allahviranloo, T., 2005. The Adomian decomposition method for fuzzy system of linear equations. Applied Mathematics and Computation 163, 553–563.

Babolian, E., Biazar, J., Vahidi, A.R., 2004. On the decomposition method for system of linear equations and system of linear Volterra integral equations. Applied Mathematics and Computation 147, 19–27.

Burden, R.I., Faires, J.D., 2001. Numerical Analysis. PWS Publishing Company, Boston, USA.

Daftardar-Gejji, V., Jafari, H., 2006. An iterative method for solving nonlinear functional equations. Journal of Mathematical Analysis and Applications 316, 753–763.

Ganji, D.D., 2006. The application of He's homotopy perturbation method to nonlinear equations arising in heat transfer. Physics Letters A 355 (4–5), 337–341.

Ganji, D.D., 2012. A semi-analytic technique for non-linear settling particle equation of motion original. Journal of Hydro-Environment Research 6 (4), 323–327.

Ganji, D.D., Sadighi, A., 2006. Application of He's homotopy-perturbation method to nonlinear coupled systems of reaction-diffusion equations. International Journal of Nonlinear Sciences and Numerical Simulation 7 (4), 411–418.

He, J.H., 1999. Homotopy perturbation technique. Computational Methods in Applied Mechanics and Engineering 178, 257–262.

Jalal, M., Ganji, D.D., 2010. An analytical study on motion of a sphere rolling down an inclined plane submerges in a Newtonian fluid. Journal of Powder Technology 198 (1), 82–92.

Jalal, M., Ganji, D.D., 2011. On unsteady rolling motion of spheres in inclined tubes filled with incompressible Newtonian fluid. Advanced Powder Technology 22 (1), 58–67.

Jalal, M., Ganji, D.D., Ahmadi, G., 2010. Analytical investigation on acceleration motion of a vertical falling spherical particle in incompressible Newtonian media. Advanced Powder Technology 21 (3), 298–304.

Jalal, M., Ganji, D.D., Ahmadi, G., 2012. An analytical study on settling of non-spherical particles. Asia-Pacific Journal of Chemical Engineering 7 (1), 63–72.

Keramati, B., 2009. An approach to the solution of linear system of equations by He's homotopy perturbation method. Chaos, Solitons and Fractals 41, 152–156.

Liu, H.K., 2011. Application of homotopy perturbation methods for solving systems of linear equations. Applied Mathematics and Computation 61, 2555–2561.

Noor, M.A., 2006. New iterative schemes for nonlinear equations. Applied Mathematics and Computation 187, 937–943.

Noor, M.A., 2007. New family of iterative methods for nonlinear equations. Applied Mathematics and Computation 190, 553–558.

Noor, M.A., 2010. Iterative methods for nonlinear equations using homotopy perturbation technique. Applied Mathematics and Information Science 4, 227–235.

Noor, M.A., Noor, K.I., 2006a. Three-step iterative methods for nonlinear equations. Applied Mathematics and Computation 183, 322–327.

Noor, M.A., Noor, K.I., 2006b. Some iterative schemes for nonlinear equations. Applied Mathematics and Computation 183, 774–779.

Noor, M.A., Noor, K.I., Mohyud-Din, S.T., Shabbir, A., 2006c. An iterative method with cubic convergence for nonlinear equations. Applied Mathematics and Computation 183, 1249–1255.

Noor, M.A., Noor, K.I., Waseem, M., 2010a. Fourth-order iterative methods for solving nonlinear equations. International Journal of Mathematics and Engineering Science 4, 43–52.

Noor, M.A., Noor, K.I., Al-Said, E., Waeem, M., 2010b. Some new iterative methods for nonlinear equations. Mathematical Problems in Engineering 2010, 1–12. http://dx.doi.org/10.1155/2010/198943.

Noor, M.A., Noor, K.I., Khan, S., Waseem, M., 2013a. Modifies homotopy perturbation method for solving system of linear equations. Journal of the Association of the Arab Universities for Basic and Applied Sciences 13, 35–37.

Noor, M.A., Noor, K.I., Waseem, M., 2013b. Decomposition method for solving system of linear equations. Engineering Mathematics Letters 2, 34–41.