African Journal of Science, Technology & Mathematics Education (AJSTME) www.ajstme.com.ng
ISSN: 2251-0141
Vol. 6, (1). April, 2021

Comparison Between the Convergence of Successive Over Relaxation (Sor) and L-U Decomposition Method for Solving System of 3×3 Linear Equations

Haruna, B., Manga, M. & Akpienbi, I. O.
[1]Department of Mathematics, Modibbo Adama Unversity of Technology, Yola.
[2]FECOET, Federal College of Education Potiskum, Yobe State
[3]Department of Mathematics/Statistics, Federal University Wukari, Taraba State
E-mail: engrbuhari@gmail.com,

Abstract

This research work considered the comparative study of the converging behaviors of successive over relaxation (SOR) and L-U decomposition methods. To investigate this, three test problems were carefully selected each from 3x3 matrices. The experiment was done with an initial guess to exact solutions. At the end of the research, L-U decomposition method provides a faster and a more accurate technique for solving problem involving a class of 3x3matrixes.

Keywords: System of linear equation, Successive over-relaxation (SOR), L-U decomposition and comparison

Introduction

In numerical analysis and linear algebra, LU decomposition (where 'LU' stands for 'lower upper', and also called LU factorization) factors a matrix as the product of a lower triangular matrix and an upper triangular matrix. The product sometimes includes a permutation matrix as well. The LU decomposition can be viewed as the matrix form of Gaussian elimination. Computers usually solve square systems of linear equations using the LU decomposition, and it is also a key step when inverting a matrix, or computing the determinant of a matrix (Schwarzenberg-Czerny, 1995). Stated that, the three main iterative methods for solving linear equation are Successive-Over Relaxation, the Gauss-Seidel and the Jacobi technique. A numerical experiment on 3 x 3 and 4 x 4 Systems of linear equations using a software which can accommodate up to 10 x 10 system of linear equations, shows that Jacobi method takes longer time of 0.82 seconds and 2.09 seconds for the 3 x 3 and 4 x 4 matrices respectively (Kalambi, 2008). It also takes about 40 and 48 iterations for the 3 x 3 and 4 x 4 linear equations to converge respectively within the same tolerance factor (Kalambi, 2008). Ruixia, (2011), Acknowledged Successive Over-relaxation Iteration Algorithm as the effective method of solving large-scale sparse matrix equation set and a first order linear stationary iterative method.

The judgment condition of SOR iterative algorithm, depend on the selection of convergence factor. When A is an irreducible matrix with weak diagonal dominance, obviously, both the coefficient of matrix A and the corresponding diagonal matrix D is nonsingular. Base on this case the theorem "If A is an irreducible matrix with weak diagonal dominance, then ASOR convergence for all $-1 \le r \le r$ and $\omega > 0$"holds (Shin-liang, and Yu-Jun, 2016). Richard and John, 9th ed. (2011), Stated that, the choice of ω with $0 < \omega < 1$, the procedure is called Under-relaxation method, while the choice of ω with $1 < \omega$, is called Over-relaxation method. The aim of the paper is to compare the convergence behavior of SOR and L-U decomposition on the system of 3×3 matrices and also to identify the best method among the two.

Methodology

Successive Over-relaxation method (SOR) is a method of solving a linear system of equation AX=b derived by extrapolating the Gauss-Seidel method. This extrapolation takes the form of a weighted average between the previous iterates and the computed Gauss-Seidel iterate successively for each component.

$$x_i^{(k)} = \omega x_i^{(k)} + (1-\omega) x_i^{(k-1)} \qquad (2.1)$$

Where $\bar{x}$ denote Gauss-Seidel iteration and ω is the extrapolation factor. The idea to choose a value of ω that will accelerate the rate of convergence of the iterates to the solution in matrix terms, the SOR algorithm can be written as;

$$x^{(k)} = (D-\omega l)^{-1}\left[\omega u + (1-\omega D) x^{(k+1)} + \omega (D-\omega l)^{-1} b\right] \qquad (2.2)$$

Where the matrix D, -L and –U represent diagonals, strictly lower-triangular, and strictly upper-triangular part of A, respectively.

As stated by Barrett, *et al.* 1994. That, If ω =1, then SOR method simplifies to Gauss-Seidel method. shows that SOR fails to converge if ω is outside the interval (0, 2). In general, it is not possible to compute in advance the value of ω that will minimize the rate of convergence of SOR. Frequently, some heuristic estimate is used, such as ω =2-0 (h) where h is the mesh spacing of the discretization of the underlying physical domain. Ahmad and David (2005), Acknowledged that in solution of a system of linear algebraic equations Ax=b with a large sparse coefficient matrix A, the LU-decomposition with iterative refinement (LUIR) is compared with the LU decomposition with direct solution (LUD), which is without refinement. We verify by numerical experiment that the use of sparse matrix technique with LUIR result in reduction of both the computing time and the storage requirements. For a system of linear equation of the form of Ax=b, one of the methods to solve the unknown is LU decomposition (Gilbert, 2010). Where an upper triangular, U is form by forward-elimination and then figure out the unknown by backward substitution.

The LU factorization is to decouple the factorization phase (Usually computationally expensive) from the actual solving phase. The factorization phase only needs the matrix A, while the actual solving phase makes use of the factored form of A and the right hand side to solve the linear system. Hence, once we have the factorization, we can make use of the factored form of A, to solve for different right hand side at a relatively moderate computational cost. In numerical analysis, LU decomposition factors a matrix as the product of lower triangular and an upper triangular matrix (AMS. 2009). The product sometimes includes a permutation matrix as well. The LU decomposition can be viewed as the matrix form of Gaussian elimination. With LU decomposition, system of linear equations could be solved simply with forward and backward substitution, providing scalability to solve large system of equation. The second terms on the right hand side of each of these equations represents the change, or correction, that is made to $x_1^{[m]}$ in order to calculate $x_i^{[m+1]}$. An LU factorization refers to the factorization of A, with proper row and/or column orderings or permutations, into two factors, a lower triangular matrix *L* and an upper triangular matrix *U*. Stated that SOR rate of convergence strongly depends on the choice of the relaxation factor, ω (Bailey, 2003). Burden and Faires (2011), Stated that, for $0<\omega<1$, the procedure is called under-relaxation method, while $\omega>1$:is called over-relaxation method for 3x3 SOR system of linear equation.

$$x_1^{(m+1)} = x_1^m + \frac{\omega}{a_{11}}(b_1 - a_{11}x_1^m - a_{12}x_2^m - a_{13}x_3^m) \quad (2.3)$$

$$x_2^{(m+1)} = x_2^m + \frac{\omega}{a_{21}}(b_2 - a_{21}x_2^m - a_{22}x_2^m - a_{23}x_3^m) \quad (2.4)$$

$$x_3^{(m+1)} = x_{31}^m + \frac{\omega}{a_{31}}(b_3 - a_{31}x_1^m - a_{32}x_2^m - a_{33}x_3^m) \quad (2.5)$$

Numerical Experiment

Successive Over-Relaxation (SOR)

Example: 3.1.1.1 Solve using SOR method; giving ω =1.025000

$$80x_1 - 20x_2 - 20x_3 = 20$$
$$-20x_1 + 40x_2 - 20x_3 = 20$$
$$-20x_1 - 20x_2 + 130x_3 = 20$$

Solution: using the formula,

$$x_1^{(m+1)} = x_1^m + \frac{\omega}{a_{11}}(b_1 - a_{11}x_1^m - a_{12}x_2^m - a_{13}x_3^m \quad (3.1)$$

$$x_2^{(m+1)} = x_2^m + \frac{\omega}{a_{21}}(b_2 - a_{21}x_2^m - a_{22}x_2^m - a_{23}x_3^m \quad (3.2)$$

$$x_3^{(m+1)} = x_{31}^m + \frac{\omega}{a_{31}}(b_3 - a_{31}x_1^m - a_{32}x_2^m - a_{33}x_3^m \quad (3.3)$$

Where ω =1.025000, m=0 and $x_1^0 = x_2^0 = x_3^0 = 0$

$$x_1^1 = 0 + \frac{1\cdot025000}{80}(20 - 80\times0 + 20\times0 + 20\times0) = 0\cdot2562500$$

$$x_2^1 = 0 + \frac{1\cdot02500}{40}(20 + 20\times0\cdot2562500 - 40\times0 + 20\times0) = 0\cdot643820$$

$$x_3^1 = 0 + \frac{1\cdot02500}{130}(20 + 20\times0\cdot2562500 + 20\times0\cdot643820 - 130\times0) = 0\cdot299626$$

$\therefore x_1^1 = 0\cdot256250, x_2^1 = 0\cdot643820, x_3^1 = 0\cdot299626$ These values are now becoming the initial for The iterations continue in this pattern and the following results are obtained on table 3.1

Table: 3.1 Iterative result at ω =1.025000

M	x_1^m	x_2^m	x_3^m
0	0·000000	0·000000	0·000000
1	0·256250	0·643820	0·299626
2	0·491602	0·901909	0·369948
3	0·569873	0·971611	0·391526
4	0·591307	0·052814	0·407169

5	0·615588	0·010343	0·403910
6	0·603263	0·003418	0·400956
7	0·601039	0·000937	0·400288
8	0·600288	0·000272	0·400081
9	0·600083	0·999493	0·399931
10	0·599850	0·999900	0·399962
11	0·599968	0·999967	0·399991
12	0·599990	0·999991	0·399997
13	0·599997	0·999997	0·399999
14	0·599999	0·999999	0·400000
15	0·600000	1·000000	0·400000
16	0·6000000	1·000000	0·400000

Example: 3.1.1.2 Solve using SOR method**;** giving ω =1.000030

$$80x_1 - 20x_2 - 20x_3 = 20$$
$$-20x_1 + 40x_2 - 20x_3 = 20$$
$$-20x_1 - 20x_2 + 130x_3 = 20$$

Solution:

where ω =1.000030, m=0 and $x_1^0 = x_2^0 = x_3^0 = 0$

$$x_1^1 = 0 + \frac{1\cdot 000010}{4}(7 - 4\times 0 - 1\times 0 - 1\times 0) = 1\cdot 750002$$

$$x_2^1 = 0 + \frac{1\cdot 000010}{-8}(-21 - 4\times 1\cdot 750002 + 8\times 0 - 3\times 0) = 3\cdot 500005$$

$$x_3^1 = 0 + \frac{1\cdot 000010}{5}(15 + 2\times 1\cdot 750002 - 1\times 3\cdot 500005 - 5\times 0) = 3\cdot 000003$$

$\therefore x_1^1 = 1\cdot 750002, x_2^1 = 3\cdot 500005, x_3^1 = 3\cdot 000003$ These values are now becoming the initial for The iterations continue in this pattern and the following results are obtained on table 3.3

Table: 3.2 iterative result at ω =1.000030

M	x_1^m	x_2^m	x_3^m
0	0·000000	0·000000	0·000000
1	1·750002	3·500005	3·000003
2	0·124996	3·812499	2·287498
3	0·225001	3·596062	2·370788
4	0·258288	3·643190	2·374677
5	0·245533	3·638270	2·370559
6	0·277793	3·652856	2·380546
7	0·241649	3·638529	2·368954
8	0·248129	3·637422	2·371767
9	0·247703	3·638264	2·371428
10	0·217577	3·623074	2·364216
11	0·253628	3·641720	2·373107

12	0·246293	3·634052	2·371707
13	0·248560	3·638670	2·371690
14	0·247410	3·638089	2·371346
15	0·247641	3·638075	2·371441
16	0·247621	3·638101	2·371428
17	0·247618	3·638094	2·371428
18	0·247620	3·638096	2·371429
19	0·247619	3·638095	2·371429
20	0·247619	3·638095	2·371429

Example: 3.1.2 Solve using SOR method; giving ω=1.000030

$$4x_1 + x_2 + x_3 = 7$$
$$4x_1 - 8x_2 + x_3 = -21$$
$$-2x_1 + x_2 + 5x_3 = 15$$

Solution: where ω=1.000030, m=0 and $x_1^0 = x_2^0 = x_3^0 = 0$

$$x_1^1 = 0 + \frac{1\cdot 000030}{4}(7 - 4\times 0 - 1\times 0 - 1\times 0) = 1\cdot 750053$$
$$x_2^1 = 0 + \frac{1\cdot 000030}{-8}(-21 - 4\times 1\cdot 750053 + 8\times 0 - 3\times 0) = 3\cdot 500132$$
$$x_3^1 = 0 + \frac{1\cdot 000030}{5}(15 + 2\times 1\cdot 750053 - 1\times 3\cdot 500132 - 5\times 0) = 3\cdot 500085$$

$\therefore x_1^1 = 1\cdot 750053, x_2^1 = 3\cdot 500132, x_3^1 = 3\cdot 500085$ These values are now becoming the initial for The iterations continue in this pattern and the following results are obtained on table 3.2

Table: 3.3 Iterative result at ω=1.000030

M	x_1^m	x_2^m	x_3^m
0	0·000000	0·000000	0·000000
1	1·750053	3·500132	3·000085
2	0·124897	3·812490	2·287439
3	0·225021	3·595294	2·370952
4	0·279318	3·653768	2·380994
5	0·241308	3·638526	2·368818
6	0·248164	3·637389	2·371788
7	0·277706	3·638274	2·371428
8	0·247574	3·638072	2·371415
9	0·247628	3·638097	2·371432
10	0·247618	3·638096	2·371428
11	0·247619	3·638095	2·371429
12	0·247619	3·638095	2·371429

Example: 3.1.3 Use SOR to solve

$$3x_1 - x_2 + x_3 = -1$$
$$-x_1 + 3x_2 - x_3 = 7$$
$$x_1 - x_2 + 3x_3 = -7$$

Solution: where ω=1.000300, m=0 and $x_1^0 = x_2^0 = x_3^0 = 0$

$$x_1^1 = 0 + \frac{1\cdot 000300}{3}(-1 - 3\times 0 + 1\times 0 - 1\times 0) = -0\cdot 333433$$
$$x_2^1 = 0 + \frac{1\cdot 000300}{3}(7 - 1\times 0\cdot 333433 - 3\times 0 - 1\times 0) = 2\cdot 222856$$
$$x_3^1 = 0 + \frac{1\cdot 000300}{3}(-7 + 1\times 0\cdot 333433 + 1\times 2\cdot 222856 - 3\times 0) = -1\cdot 481681$$

$\therefore x_1^1 = 0\cdot 333433, x_2^1 = 2\cdot 222856, x_3^1 = -1\cdot 481681$ These values are now becoming the initial The iterations continue in this pattern and the following results are obtained on table 3.4

Table 3.4: Iterative result at ω=1.000300

M	x_1^m	x_2^m	x_3^m
0	0·000000	0·000000	0·000000
1	-1·333433	2·222856	-1·481681
2	0·901883	2·140042	-1·920745
3	1·020298	2·033152	-1·995738
4	1·009627	2·004621	-2·001670
5	1·002095	2·000140	-2·000651
6	1·000263	1·999871	-2·000131
7	1·000001	1·999957	-2·000015
8	0·999991	1·999992	-2·000000
9	0·999997	1·999999	-1·999999
10	0·999999	2·000000	-2·000000
11	1·000000	2·000000	-2·000000
12	1·000000	2·000000	-2·000000

Lower and Upper (LU) Decomposition

This method consists of two parts, the lower diagonal and upper diagonal part. Below are some examples.

Example: 3.2.1 Solve using LU decomposition method

$$80x_1 - 20x_2 - 20x_3 = 20$$
$$-20x_1 + 40x_2 - 20x_3 = 20$$
$$-20x_1 - 20x_2 + 130x_3 = 20$$

Solution: Using the formula,

$$[L][Z] = C \quad (3.4)$$
$$[U][X] = Z \quad (3.5)$$

where

$$\begin{bmatrix} 80 & -20 & -20 \\ -20 & 40 & -20 \\ -20 & -20 & 130 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix}$$

$$R_{3n} - \frac{R_{31}}{R_{11}} R_{1n} \Rightarrow U_{3n}, \quad \text{For n} \geq 1 \tag{3.6}$$

$$\therefore \quad U = \begin{bmatrix} 80 & -20 & -20 \\ 0 & 35 & -25 \\ 0 & 0 & 107 \end{bmatrix} \text{ and L=} \begin{bmatrix} 80 & -20 & -20 \\ -20 & 40 & -20 \\ -20 & -20 & 130 \end{bmatrix}$$

$$[L] \qquad [Z] = [C]$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -0\cdot 25 & 1 & 0 \\ -0\cdot 25 & -0\cdot 71 & 1 \end{bmatrix}\begin{bmatrix} z_1^0 \\ z_2^0 \\ z_3^0 \end{bmatrix} = \begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix}$$

$$\therefore \begin{bmatrix} z_1^0 \\ z_2^0 \\ z_3^0 \end{bmatrix} = \begin{bmatrix} 20 \\ 25 \\ 42\cdot 75 \end{bmatrix}$$

Then from equation (3.5)

$$U = \begin{bmatrix} 80 & -20 & -20 \\ 0 & 35 & -25 \\ 0 & 0 & 107 \end{bmatrix}\begin{bmatrix} x_1^0 \\ x_2^0 \\ x_3^0 \end{bmatrix} = \begin{bmatrix} 20 \\ 25 \\ 42\cdot 75 \end{bmatrix}$$

$$\therefore \; x_1^0 = 0\cdot 599800, x_2^0 = 0\cdot 999666, x_3^0 = 0\cdot 399533$$

Now the value of $x_1^0, x_2^0, x_3^0 = C_1$

That is, $c_1 = \begin{bmatrix} 0\cdot 599800 \\ 0\cdot 999666 \\ 0\cdot 399533 \end{bmatrix}$

The iterations continue in this pattern and the following results are obtained on table 3.5

Table 3.5 iterative of 3.2.1

M	x_1^m	x_2^m	x_3^m
0	0·599800	0·999666	0·399533
1	0·021179	0·041963	0·012764
2	0·000809	0·001695	0·000482
3	0·000229	0·000068	0·000019
4	0·000005	0· 000005	0·000002
5	0·000000	0·000000	0·000000

Example: 2.2 Solve Using LU-Decomposition Method

$$4x_1 + x_2 + x_3 = 7$$
$$4x_1 - 8x_2 + x_3 = -21$$
$$-2x_1 + x_2 + 5x_3 = 15$$

Solution:

$$\therefore U = \begin{bmatrix} 4 & 1 & 1 \\ 0 & -9 & 2 \\ 0 & 0 & 5{\cdot}8 \end{bmatrix} \text{ and } L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -0{\cdot}5 & -0{\cdot}2 & 1 \end{bmatrix}$$

$$\therefore \begin{bmatrix} z_1^0 \\ z_2^0 \\ z_3^0 \end{bmatrix} = \begin{bmatrix} 7 \\ -28 \\ 12{\cdot}9 \end{bmatrix}$$

Then, using (3.5) equation above,

$$U = \begin{bmatrix} 4 & 1 & 1 \\ 0 & -9 & 2 \\ 0 & 0 & 5{\cdot}8 \end{bmatrix} \begin{bmatrix} x_1^0 \\ x_2^0 \\ x_3^0 \end{bmatrix} = \begin{bmatrix} 7 \\ -0{\cdot}28 \\ 12{\cdot}9 \end{bmatrix}$$

$$\therefore x_1^0 = 0{\cdot}287357, x_2^0 = 3{\cdot}609195, x_3^0 = 2{\cdot}241379$$

$$\text{That is, } c_1 = \begin{bmatrix} 0{\cdot}287357 \\ 3{\cdot}609195 \\ 2{\cdot}241379 \end{bmatrix}$$

The iterations continue in this pattern and the following results are obtained on table 3.6

Table 3.6 iterative of 3.2.2

M	x_1^m	x_2^m	x_3^m
0	0·287359	3·609195	2·241379
1	0·018477	-0·262522	0·475970
2	-0·025795	0·047659	0·073967
3	-0·008400	-0·005259	0·013062
4	0·001636	0·000015	-0·002513
5	0·001881	0·000103	-0·000348
6	0·001633	0·000207	0·000041
7	0·000339	0·000180	0·000099
8	0·000075	0·000022	0·000019
9	0·000015	0·000008	0·000008
10	0·000003	0·000001	0·000002
11	0·000001	0·000000	0·000001
12	0·000000	0·000000	0·000000

Example: 2.3 Solve Using LU-Decomposition Method

$$3x_1 - x_2 + x_3 = -1$$
$$-x_1 + 3x_2 - x_3 = 7$$
$$x_1 - x_2 + 3x_3 = -7$$

Solution:

$$U=\begin{bmatrix}3 & -1 & 1\\ 0 & 2\cdot 67 & -0\cdot 67\\ 0 & 0 & 2\cdot 89\end{bmatrix} \text{ and } L=\begin{bmatrix}1 & 0 & 0\\ -1\cdot 33 & 1 & 0\\ 1\cdot 33 & -0\cdot 25 & 1\end{bmatrix}$$

$$\begin{bmatrix}z_1^0\\ z_2^0\\ z_3^0\end{bmatrix}=\begin{bmatrix}-1\cdot 000000\\ 5\cdot 666667\\ -4\cdot 250000\end{bmatrix}$$

And from equation (3.5) above;

$$U=\begin{bmatrix}3 & 1 & 1\\ 0 & 2\cdot 67 & -0\cdot 67\\ 0 & 0 & 2\cdot 89\end{bmatrix}\begin{bmatrix}x_1^0\\ x_2^0\\ x_3^0\end{bmatrix}=\begin{bmatrix}-0\cdot 000000\\ 5\cdot 666667\\ -4\cdot 250000\end{bmatrix}$$

$$\therefore x_1^0=0\cdot 767308, x_2^0=1\cdot 732692, x_3^0=-1\cdot 569231$$

$$\text{That is, } c_1=\begin{bmatrix}0\cdot 767308\\ 1\cdot 732692\\ -1\cdot 569231\end{bmatrix}$$

The iterations continue in this pattern and the following results are obtained on table 3.7

Table 3.7 iterative of 3.2.3

M	x_1^m	x_2^m	x_3^m
0	0·767308	1·732692	−1·569231
1	0·702533	1·632111	1·321516
2	0·502121	0·595795	−0·271073
3	0·391234	0·450361	−0·200111
4	0·326105	0·391612	−0·160234
5	0·221440	0·201772	−0·200111
6	0·120456	0·149043	−0·160234
7	0·100021	0·120013	−0·113210
8	0·000126	0·100015	−0·100732
9	0·000018	0·000052	−0·000016
10	0·000000	0·000000	0·000000

Comparing SOR and L-U Decomposition Iterative Results

Table 3.8: Comparing the iterative results of examples 3.1.1.1 (SOR) and 3.2.1 (LU)

	LU	SOR	LU	SOR	LU	SOR
M	x_1^m	x_1^m	x_2^m	x_2^m	x_3^m	x_3^m
0	0·599800	0·000000	0·999666	0·000000	0·399533	0·000000

1	0·021179	0·256250	0·041963	0·643820	0·012764	0·299626
2	0·000809	0·491602	0·001695	0·901909	0·000482	0·369948
3	0·000229	0·569873	0·000068	0·971611	0·000019	0·391526
4	0·000005	0·591307	0· 000005	0·052814	0·000002	0·407169
5	0·000000	0·615588	0·000000	0·010343	0·000000	0·403910
6	0·599800	0·603263	0·999666	0·003418	0·399533	0·400956
7	0·021179	0·601039	0·041963	0·000937	0·012764	0·400288
8	0·000809	0·600288	0·001695	0·000272	0·000482	0·400081
9	0·000229	0·600083	0·000068	0·999493	0·000019	0·399931
10	0·000005	0·599850	0· 000005	0·999900	0·000002	0·399962
11	0·000000	0·599968	0·000000	0·999967	0·000000	0·399991
12		0·599990		0·999991		0·399997
13		0·599997		0·999997		0·399999
14		0·599999		0·999999		0·400000
15		0·600000		1·000000		0·400000
16		0·6000000		1·000000		0·400000

Table 3.9 Comparing the iterative results of examples 3.1.1.2 (SOR) and 3.2.1 (LU)

	LU	SOR	LU	SOR	LU	SOR
M	x_1^m	x_1^m	x_2^m	x_2^m	x_3^m	x_3^m
0	0·287359	0·000000	3·609195	0·000000	2·241379	0·000000
1	0·018477	1·750002	-0·262522	3·500005	0·475970	3·000003
2	-0·025795	0·124996	0·047659	3·812499	0·073967	2·287498
3	-0·008400	0·225001	-0·005259	3·596062	0·013062	2·370788
4	0·001636	0·258288	0·000015	3·643190	-0·002513	2·374677
5	0·001881	0·245533	0·000103	3·638270	-0·000348	2·370559
6	0·001633	0·277793	0·000207	3·652856	0·000041	2·380546
7	0·000339	0·241649	0·000180	3·638529	0·000099	2·368954
8	0·000075	0·248129	0·000022	3·637422	0·000019	2·371767
9	0·000015	0·247703	0·000008	3·638264	0·000008	2·371428
10	0·000003	0·217577	0·000001	3·623074	0·000002	2·364216
11	0·000001	0·253628	0·000000	3·641720	0·000001	2·373107
12	0·000000	0·246293	0·000000	3·634052	0·000000	2·371707
13		0·248560		3·638670		2·371690
14		0·247410		3·638089		2·371346
15		0·247641		3·638075		2·371441
16		0·247621		3·638101		2·371428
17		0·247618		3·638094		2·371428
18		0·247620		3·638096		2·371429
19		0·247619		3·638095		2·371429
20		0·247619		3·638095		2·371429

Table 3.10 Comparing the iterative results of examples 3.1.2 (SOR) and 3.2.2 (LU)

	LU	SOR	LU	SOR	LU	SOR
M	x_1^m	x_1^m	x_2^m	x_2^m	x_3^m	x_3^m
0	0·287359	0·000000	3·609195	0·000000	2·241379	0·000000
1	0·018477	1·750053	-0·262522	3·500132	0·475970	3·000085
2	-0·025795	0·124897	0·047659	3·812490	0·073967	2·287439
3	-0·008400	0·225021	-0·005259	3·595294	0·013062	2·370952
4	0·001636	0·279318	0·000015	3·653768	-0·002513	2·380994
5	0·001881	0·241308	0·000103	3·638526	-0·000348	2·368818
6	0·001633	0·248164	0·000207	3·637389	0·000041	2·371788
7	0·000339	0·277706	0·000180	3·638274	0·000099	2·371428
8	0·000075	0·247574	0·000022	3·638072	0·000019	2·371415
9	0·000015	0·247628	0·000008	3·638097	0·000008	2·371432
10	0·000003	0·247618	0·000001	3·638096	0·000002	2·371428
11	0·000001	0·247619	0·000000	3·638095	0·000001	2·371429
12	0·000000	0·247619	0·000000	3·638095	0·000000	2·371429

Table 3.11 Comparing the iterative results of examples 3.1.3 (SOR) and 3.2.3 (LU)

	LU	SOR	LU	SOR	LU	SOR
M	x_1^m	x_1^m	x_2^m	x_2^m	x_3^m	x_3^m
0	0·767308	0·000000	1·732692	0·000000	−1·569231	0·000000
1	0·702533	-1·333433	1·632111	2·222856	1·321516	-1·481681
2	0·502121	0·901883	0·595795	2·140042	−0·271073	-1·920745
3	0·391234	1·020298	0·450361	2·033152	−0·200111	-1·995738
4	0·326105	1·009627	0·391612	2·004621	−0·160234	-2·001670
5	0·221440	1·002095	0·201772	2·000140	−0·200111	-2·000651
6	0·120456	1·000263	0·149043	1·999871	−0·160234	-2·000131
7	0·100021	1·000001	0·120013	1·999957	−0·113210	-2·000015
8	0·000126	0·999991	0·100015	1·999992	−0·100732	-2·000000
9	0·000018	0·999997	0·000052	1·999999	−0·000016	-1·999999
10	0·000000	0·999999	0·000000	2·000000	0·000000	-2·000000
11		1·000000		2·000000		-2·000000
12		1·000000		2·000000		-2·000000

Result Discussion

SOR converges only when the system of a linear equation is strictly diagonal. And the diverged rate also depends on the value of ω as in example 3.1.1.1 and 3.1.1.2, the value of ω are 1·000030and 1·000001respectively, while the systems diverged at 12 and 20 iterations respectively. This signifies that, the more the value of ω approaches 2, the less the number of iterations required to converge and vice versa. The comparison shows that, on table 3.8, LU converged at 11 iterations, while SOR diverged at 16 iterations, on table 3.9; LU converged at 12 iterations, while SOR diverged at 22 iterations, on table 3.10, both LU and SOR converged and diverged at 12 iterations respectively. And on table 3.11, LU converged at 10 iterations, while SOR diverged at 12 iterations. Both LU and SOR converged at the same iterative point of 12,

because 12 is the minimum diverged point of SOR at the best relaxation of factor, ω of 1·000030.. We said SOR to diverge because it is starting from the initial value $x_1^0 = x_2^0 = x_3^0 = 0$ of to the final (diverging) value of $x_1^0, x_2^0, x_3^0 \neq 0$, while LU is said to converge because it starts from the initial value of $x_1^m, x_2^m, x_3^m \neq 0$ to the final (converging) value of $x_1^m = x_2^m = x_3^m = 0$

Conclusion

In this paper, we conclude that LU-decomposition converges faster than the SOR on the system of 3x3 matrices.

References

Ahmad, A., & David, R. K. (2005). *LU-Decomposition with Iterative Refinement for Solving Sparse Linear System. Department of Mathematics*, Faculty of Science, Teshreen University, Lattakia, SYRIA

Bailey W. (. 2003) "The Successive Over Relaxation Algorithm and its application to Numerical Solutions of Elliptic Partial Differential Equations". B.Sc. Project, Dublin Institute of Technology.

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R. Romine, C., and Van-der, V. H. (1994) *Numerical recipes in fortran: the art of scientific computing, 2,866-869.* Cambridge university press,

Burden, R. L., & Faires, J. D. (2011) *numerical Analysis.* (9th ed.) Gilbert, S. (2010) *(math.stackexchange.com/question)*

Kalambi, I.B. (2008). A comparison of three iterative methods for solution of linear equation *journal. Appl. Sci. environ. Manage. 12(4), 53-55.*

Richard, L. B., & J. D. F. (2011) *numerical Analysis.* (9th ed.)

Ruixia, C. (2011). *Research and Applications of Successive over-relaxation Iterative Algorithm. (pp. 1)*

Schwarzenberg-Czerny, A. (1995). *On matrix factorization and efficient least squares solution*. Astronomy and Astrophysics supplement series. 110:405. Bibcode:1995A&As. 110..405.

Shin-liang, W., & Yu-Jun, L. (2016) *A New Version of the Accelerated Over-relaxation Iterative method, 1, 45*

Xingechen, Y. (2015). *AMS 209 Final year project*, Department of Applied Mathematics and Statistics, University of California Santa Cruz.