Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2022, 9(1):116-123

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

The Decomposition and Aggregation Algorithmic Numerical Iterative Solution Methods for the Stationary Distribution of Markov Chain

Agboola, Sunday Olanrewaju

Department of Mathematics, Nigerian Army University Biu, Biu, Nigeria
Corresponding Author: e-mail: agboolasunday70@gmail.com, Phone: 08058222949

Abstract The evolution of a system is represented by transitions from one state to the next, and the system's physical or mathematical behavior can also be depicted by defining all of the numerous states it can be in and demonstrating how it moves between them. In this study, the iterative solution methods for the stationary distribution of Markov chains which start with an initial estimate of the solution vector and then alter it in such a way that it gets closer and closer to the genuine solution with each step or iteration., and also involved the matrices operation such as multiplication with one or more vectors, which leaves the transition matrices unchanged and saves time has been investigated, in order to provide some insight into the solutions of stationary distribution of Markov chain. Our quest is to compute the solutions using decomposition and aggregation algorithmic numerical iterative methods by considering the following three steps: the left-hand eigenvector u_i of length n_i corresponding to the eigenvalue closest to 1, λ_{i_1}, in each block i, $1 \leq i \leq N$, is computed; the weights ξ_i, an approximate solution to the stationary probability vector π is estimated and the global solution $\pi^* = \xi_i u_i = (\xi_1 u_1, \ \xi_2 u_2, \ \xi_3 u_3, \ \ldots, \ \xi_N u_N)$ is computed. Concept of eigen values and vectors, Matrix operations such as Lower, upper and diagonal matrices and normalization principle are used with the help of some existing laws, theorems and formulas of Markov chain. The global solution π^* are obtained for the illustrative examples in each block i

Keywords *eigen value, eigen vector, Courtois matrix, nearly completely decomposable (NCD), Perron root, iterative aggregation/disaggregation (IAD)*

Introduction

A de-compositional approach to solving Markov chains is intuitively very attractive since it appeals to the principle of divide and conquer: if the model is too large or complex to analyze in toto, it is divided into subsystems, each of which is analyzed separately, and a global solution is then constructed from the partial solutions. Ideally the problem is broken into subproblems that can be solved independently and the global solution is obtained by "pasting" together the subproblem solutions. Although it is rare to find Markov chains that can be divided into independent sub-chains, it is not unusual to have Markov chains in which this condition almost holds. An important class of problems that frequently arise in Markov modeling are those in which the state space may be partitioned into disjoint subsets with strong interactions among the states of a subset but with weak interactions among the subsets themselves. Such problems are sometimes referred to as nearly completely decomposable (NCD), nearly uncoupled, or nearly separable. It is apparent that the assumption that the subsystems are independent and can therefore be solved separately does not hold. Consequently, an error arises. This error will be small if the assumption is approximately true. In the discipline of numerical analysis, there are two types of solution methods: iterative solution methods and direct solution methods. Iterative approaches start with an initial estimate of the solution vector and then alter it in such a way that it gets closer and closer to the genuine solution with each step or iteration. It eventually converges on the true solution. If there is no known initial approximation, a guess is performed or an arbitrary initial vector is used instead. The solution must be computed when a specified number of well-defined stages have been completed. The most widely utilized methods for deriving the stationary probability vector from either the stochastic transition probability matrix or the infinitesimal generator are iterative methods of one form or another. This decision was made for a variety of reasons. First, a look at the conventional iterative approaches reveals that the matrices are only involved in one operation: Romanovsky [1] established the application and simulation of discrete Markov Chains, which was followed by Stewart [2-3] with the development of Numerical Solutions of Markov Chains, while Pesch *et al*. [4] demonstrated the appropriateness of the Markov chain technique in the wind feed in Germany (2015). Uzun and Kiral [5] used the Markov chain model of fuzzy state to anticipate the direction of gold price movement and to estimate the probabilistic transition matrix of gold price closing returns, whereas Aziza *et al*. [6] used the Markov chain model of fuzzy state to predict monthly rainfall data (2019). Clemence [7] demonstrated the application of Markov chain to the spread of disease infection, demonstrating that Hepatitis B became more infectious over time than tuberculosis and HIV, while Vermeer and Trilling [8] demonstrated the application of Markov chain to journalism. However, in this study, the decomposition and aggregation numerical iterative solution methods and algorithms for computing the stationary distribution of Markov chain are considered.

Notation

$B = (B)_{ij}$ irreducible nearly completely decomposable (NCD) stochastic matrix, i.e., probability of leaving (any state of) block i to enter (any state of) block j

$D = (D)_{ij}$ coupling matrix

$n_i,\ i = 1, 2, 3, \dots\ , N$ length of eigen vector in each block i

u_i the eigen vector of length n_i

$\pi_i,\ i = 1, 2, 3, \dots\ , N$ stationary probability distribution matrix in each block i

$\lambda_{i_1},\ 1 \le i \le N$ the eigenvalue closest to 1 in each block i

ξ_i is the proportion of time spend in block i

$\emptyset_i,\ i = 1, 2, 3, \dots\ , N$ normalization of π_i

Materials and Methods

Let an irreducible nearly completely decomposable (NCD) stochastic matrix B be written as

$$B = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & B_{22} & \cdots & B_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ B_{n1} & B_{n1} & \cdots & B_{nn} \end{pmatrix},$$

Where

$$\|B_{ii}\| = O(1), \quad i = 1, 2, 3, \dots\ , N,$$

And

$$\left\|B_{ij}\right\| = O(\epsilon), \quad i \neq j.$$

Such a matrix has $(N - 1)$ eigenvalues that are extremely near to 1, and none of the previous point iterative approaches are useful in dealing with this circumstance. Block and de-compositional approaches, on the other hand, can be quite effective. We'll start by looking at what happens when all of the off-diagonal blocks are zero. i.e.,

$$(\pi_1 \quad \pi_2, \quad \cdots \quad , \pi_N) \begin{pmatrix} B_{11} & 0 & \cdots & 0 & 0 \\ 0 & B_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & \cdots & B_{n-1(n-1)} & 0 \\ 0 & 0 & 0 & 0 & B_{nn} \end{pmatrix} (\pi_1 \quad \pi_2, \quad \cdots \quad , \pi_N)$$

Each B_{ii} is a stochastic matrix in this case. The Markov chain can be segmented into N irreducible classes, each with its own stationary distribution. Each π_i can be found on its own page.

$$\pi_i B_{ii} = \pi_i, \quad i = 1, 2, \cdots, N$$

When the Markov chain is nearly completely decomposable (NCD) rather than completely decomposable, we can use the following approximate solution procedure:

Results and Discussion

Step (a): Solve blocks as if independent

Assume that the system is completely decomposable and compute the stationary probability distribution for each component as the first step in approximating the solution of $\pi B = \pi$ when $B_{ij} \neq 0$. The fact that the B_{ii} are strictly substochastic rather than stochastic is the first issue that arises. One simple solution is to ignore the problem entirely and deal directly with the substochastic matrices B_{ii}. In other words, we can use the normalized eigenvector corresponding to the Perron root (the eigenvalue closest to 1) of block B_{ii} as the probability vector, with the elements denoting the probabilities of being in each of the block's states conditioned on being in this block.

Illustrative example 1: the procedure is examined using the 8 × 8 Courtois matrix, an examination of which reveals that it is nearly completely decomposable (NCD) into a block of order 3, a second of order 2, and a third of order 3:

B

$$= \begin{pmatrix} 0.85 & 0.0 & 0.149 & 0.0009 & 0.0 & 0.00005 & 0.0 & 0.00005 \\ 0.1 & 0.65 & 0.249 & 0.0 & 0.0009 & 0.00005 & 0.0 & 0.00005 \\ 0.1 & 0.8 & 0.0996 & 0.0003 & 0.0 & 0.0 & 0.0001 & 0.0 \\ 0.0 & 0.0004 & 0.0 & 0.7 & 0.2995 & 0.0 & 0.0001 & 0.0 \\ 0.0005 & 0.0 & 0.0004 & 0.399 & 0.6 & 0.0001 & 0.0 & 0.0 \\ 0.0 & 0.00005 & 0.0 & 0.0 & 0.00005 & 0.6 & 0.2499 & 0.15 \\ 0.00003 & 0.0 & 0.00003 & 0.00004 & 0.0 & 0.1 & 0.8 & 0.0999 \\ 0.0 & 0.00005 & 0.0 & 0.0 & .00005 & 0.1999 & 0.25 & 0.55 \end{pmatrix}$$

For this matrix, we have the following blocks, Perron roots, and corresponding left-hand eigenvectors:

$$B_{11} = \begin{pmatrix} 0.85 & 0.0 & 0.149 \\ 0.1 & 0.65 & 0.249 \\ 0.1 & 0.8 & 0.0996 \end{pmatrix}, \quad \lambda_{1_1} = 0.9991, \qquad u_1 = (0.4014 \quad 0.4167 \quad 0.1819),$$

$$B_{22} = \begin{pmatrix} 0.7 & 0.2995 \\ 0.399 & 0.6 \end{pmatrix}, \quad \lambda_{2_1} = 0.99929, \quad u_2 = (0.5714 \quad 0.4286),$$

$$B_{33} = \begin{pmatrix} 0.6 & 0.2499 & 0.15 \\ 0.1 & 0.8 & 0.0999 \\ 0.1999 & 0.25 & 0.55 \end{pmatrix}, \quad \lambda_{3_1} = 0.999, \qquad u_1 = (0.24074 \quad 0.55563 \quad 0.20364),$$

To summarize, in part (a), the left-hand eigenvector u_i of length n_i corresponding to the eigenvalue closest to 1, λ_{i_1}, in each block i, $1 \leq i \leq N$, is computed. In other words, we solve the N eigenvalue problems

$$u_i B_{ii} = \lambda_{i_1} u_i, \quad u_i e = 1, \quad i = 1, 2, \dots, N.$$

Step (b): Estimate block probabilities

The second issue is that just concatenating the stationary probability vectors for each block will not provide a probability vector. Each sub-elements vector's add up to one. Each sub-vector must still be weighted by the chance of being in its sub-block of states. The B_{ii} probability distributions are conditional probabilities in the sense that they describe the probability of being in one of the states of subset $i, i = 1, 2, \dots, N,$ conditioned on the fact that the Markov chain is in one of those states. That condition must be removed. To calculate the probability of being in a certain block of states, we need to create a matrix with the element i, j corresponding to the probability of transitioning from block i to block j. This is a $(N \times N)$-dimensional stochastic matrix that describes the interactions between blocks. To construct this matrix, also known as the coupling matrix, we must reduce each block B_{ij} of B to a single element. The block probabilities, or weights, required to build the global approximation are provided by the stationary distribution of the coupling matrix. In the case of the running example, we must determine weights ξ_1, ξ_2, and ξ_3 such that

$$(\xi_1 u_1, \ \xi_2 u_2, \ \xi_3 u_3 \ )$$

is a close approximation to π_i. Here ξ_i represents the percentage of time we spend in block i. Our original (8×8) stochastic matrix must be reduced to a (3×3) stochastic matrix. This is done by first replacing each block's row with the sum of its elements. The probability of leaving state k of block i and entering (one of the states of) block j is given by the sum of the elements of row k of block $i\,j$. It doesn't matter which state of block j is the target state any longer. The procedure executed for each block is $B_{ij}e$.

Illustrative example 2: Summing across the block rows of the Courtois matrix gives

$$\left(\begin{array}{c|c|c}
0.999 & 0.0009 & 0.0001 \\
0.999 & 0.0009 & 0.0001 \\
0.9996 & 0.0003 & 0.0001 \\
\hline
0.0004 & 0.9995 & 0.0001 \\
0.0009 & 0.999 & 0.0001 \\
\hline
0.00005 & 0.00005 & 0.9999 \\
0.00006 & 0.00004 & 0.9999 \\
0.00005 & 0.00005 & 0.9999
\end{array}\right).$$

The probability of leaving (any state of) block i and entering (any state of) block j is then calculated using these results. This means that each column sub-vector, $B_{ij}e$, must be reduced to a scalar. The k^{th} element of $B_{ij}e$ is the probability of leaving state k of block i and entering into block j. We must sum the elements of this vector after each element has been weighed by the chance of existing in that state to obtain the total probability of leaving (any state of) block i and entering (any state of) block j. (given that the Markov chain is in one of the states of that block). The elements of the stationary probability vector can be used to calculate these weighing factors. They are the components of $\frac{\pi_i}{\|\pi_i\|_1}$. The $(ij)^{th}$ element of the reduced $(N \times N)$ matrix is therefore given by

$$D_{ij} = \frac{\pi_i}{\|\pi_i\|_1} B_{ij}e = \emptyset_i B_{ij}e,$$

Where

$$\emptyset_i = \frac{\pi_i}{\|\pi_i\|_1}.$$

If B is an irreducible stochastic matrix, then D also is irreducible and stochastic. Let ξ denote its left eigenvector, i.e., $\xi D = \xi$ and $\xi e = 1$. The i^{th} component of ξ is the stationary probability of being in (one of the states of) block i. It is easy to show that

$$\xi = (\|\pi_1\|_1, \|\pi_2\|_1 \ , \ \dots, \ \|\pi_N\|_1).$$

Of course, because the vector is unknown, it is impossible to determine the weights $\|\pi_i\|_1$. They can, however, be approximated by setting $\emptyset_i = u_i$ and utilizing the probability vector generated from each of the individual B_{ii}. As a result, the weights ξ_i may be calculated, and an approximate solution to the stationary probability vector π can be found.

Illustrative example 2: Performing these operations, we obtain the following coupling matrix for the Courtois example:

$$\begin{pmatrix} 0.40143 & 0.41672 & 0.18185 & 0.5714 & 0.4286 & 0.24074 & 0.55563 & 0.20364 \end{pmatrix} \times$$

$$\begin{pmatrix} 0.999 & 0.0009 & 0.0001 \\ 0.999 & 0.0009 & 0.0001 \\ 0.9996 & 0.0003 & 0.0001 \\ 0.0004 & 0.9995 & 0.0001 \\ 0.0009 & 0.999 & 0.0001 \\ 0.00005 & 0.00005 & 0.9999 \\ 0.00006 & 0.00004 & 0.9999 \\ 0.00005 & 0.00005 & 0.9999 \end{pmatrix} = \begin{pmatrix} 0.99911 & 0.00079 & 0.0001 \\ 0.00061 & 0.99929 & 0.0001 \\ 0.00006 & 0.00004 & 0.9999 \end{pmatrix} = D$$

Its eigenvalues are $(1.0,\ 0.9998,\ 0.9985)$ and its stationary probability vector is

$$\xi = (0.22252 \quad 0.27748 \quad 0.5000)$$

Step (c): Compute global approximation

We are now in a position to form the stationary probability vector's final estimate. The approximation gives

$$\pi \approx \ (\xi_1 u_1,\ \xi_2 u_2,\ \xi_3 u_3)$$

where the u_i are approximations to $\frac{\pi_i}{\|\pi_i\|_1}$.

Illustrative example 3: In the running example we obtain the approximate solution

$$\pi^* = (0.08932,\ 0.09273,\ 0.04046,\ 0.15855,\ 0.11893,\ 0.12037,\ 0.27781,\ 0.10182),$$

which may be compared to the exact solution

$$\pi = (0.08928,\ 0.09276,\ 0.04049,\ 0.15853,\ 0.11894,\ 0.12039,\ 0.27780,\ 0.10182).$$

To recapitulate, an approximation to the stationary probability vector of an NCD Markov chain may be achieved by first solving each of the blocks separately, then constructing and solving the coupling matrix, and then putting the parts together to form the approximate solution. The states have been organized in such a way that the transition probability matrix has the necessary NCD block structure. The entire operation might be written as an algorithm.

Algorithm 2: NCD Decomposition Approximation

1. Solve the individual blocks: $u_i B_{ii} = \lambda_{i_1} u_i, \quad u_i e = 1, \quad i = 1, 2, \ldots, N.$

2. (a) Form the coupling matrix: $(D)_{ij} = u_i B_{ij} e.$

(b) Solve the coupling problem: $\xi D = \xi,\ \xi e = 1.$

3. Construct the approximate solution: $\pi^* = \ (\xi_1 u_1,\ \xi_2 u_2,\ \xi_3 u_3).$

The question now is whether we can feed this approximation back into the decomposition procedure to improve the approximation even further. The u_i are utilized to build the coupling matrix, as you can see. We get the same solution to the coupling matrix as before if we replace them with the new approximations $\xi_i u_i$ As a result, the computed approximation will remain unchanged. However, it was discovered that giving the approximation a power step before plugging it back into the decomposition process had a highly beneficial effect. Later, this power step was replaced by a block Gauss–Seidel step, which became known as a disaggregation step; the aggregation step was formed and solved the matrix D. Iterative aggregation/disaggregation is the name given to the entire process (IAD). The algorithm is shown in the diagram below. A superscript in parentheses on the appropriate variable names indicates the iteration number. The initial vector is designated as $\pi(0)$. This could be a random selection or an approximation obtained from the simple decomposition strategy. Many of the steps in the decomposition algorithm have related steps. Many of the steps in the decomposition algorithm have related steps. The construction of the coupling matrix (Step 3) is, for example, similar in both. Step 4(a) relates to forming the computed approximation in the decomposition algorithm, while Step 4(b) corresponds to solving the different blocks, with the exception of the block Gauss–Seidel step in the iterative aggregation/disaggregation (IAD) process.

Algorithm 2: Iterative Aggregation/Disaggregation

1. Let $\pi^{(0)} = \left(\pi_1^{(0)},\ \pi_2^{(0)},\ \ldots\ ,\ \pi_N^{(0)}\right)$ be a given initial approximation to the solution to the π, and set $m = 1$.

2. Compute $\emptyset^{(m-1)} = \left(\emptyset_1^{(m-1)},\ \emptyset_2^{(m-1)},\ \ldots\ ,\ \emptyset_N^{(m-1)}\right),$ where

$$\emptyset_i^{(m-1)} = \frac{\pi_i^{(m-1)}}{\left\|\pi_i^{(m-1)}\right\|_1}\ ,\quad i = 1, 2, \ldots\ , N.$$

3. (a) Form the coupling matrix: $D_{ij}^{(m-1)} = \emptyset_i^{(m-1)} B_{ij} e,\quad for\ \ i,j = 1, 2, \ldots\ , N.$

(b) solve the coupling problem $\xi^{m-1} D^{m-1} = \xi^{m-1},\quad \xi^{m-1} e = 1.$

4. (a) construct the row vector

$$z^m = \xi_1^{(m-1)} \emptyset_1^{(m-1)},\ \xi_2^{(m-1)} \emptyset_2^{(m-1)},\ \ \ldots\ ,\ \xi_N^{(m-1)} \emptyset_N^{(m-1)}.$$

(b) solve the following N system of equations to find $\pi^{(m)}$:

$$\pi_k^{(m)} = \pi_k^{(m)} B_{kk} + \textstyle\sum_{j>k} z_j^{(m)} B_{jk} + \sum_{j<k} \pi_j^{(m)} B_{jk}\,,\quad k = 1, 2, \ldots\ , N. \qquad (10)$$

5. Normalize and conduct a test for convergence. Then stop and take $\pi^{(m)}$ as the required solution vector. Otherwise set $m = m + 1$ and go to step 2.

It is essential in these methods that the matrix has the block structure that the algorithms require, and it may be required to reorganize the states to achieve this. We can only guarantee that the resulting transition matrix has a property that directly represents the structural properties of the nearly completely decomposable (NCD) system after reordering the states. The convergence behavior may be substantially less satisfying if the partitioning provided to the algorithm does not match the decomposability features of the matrix. The states can be reordered by considering the Markov chain as a directed graph and removing the edges with small weights (probabilities). The connected components of $\hat{B} + \hat{B}^T$ must then be found using a graph algorithm, where $\hat{B}$ is the modified transition probability matrix. The algorithm's complexity is $O(|V| + |E|)$, where $|V|$is the number of vertices in the graph and $|E|$ is the number of edges.

Illustrative example 4: Both the iterative aggregation and disaggregation (IAD) and the block Gauss–Seidel (BGS) methods are particularly effective when used to the Courtois matrix. The table below illustrates that with the IAD approach, full machine precision is obtained in only four iterations, while with the BGS method, it takes nine iterations. The diagonal block equations are solved using LU decomposition in both cases.

Table 1: Iterative Aggregation/Disaggregation (IAD) and Block Gauss–Seidel (BGS) Residual for Courtois Matrix

IAD and BGS residuals for the Courtois NCD matrix		
Iteration	**IAD residual**	**BGS residual**
	$1.0E-5\ \times$	$1.0E-5\ \times$
1	0.93581293961421	0.94805408435419
2	0.00052482104506	0.01093707688215
3	0.00000000280606	0.00046904081241
4	0.00000000000498	0.00002012500900
5	0.00000000000412	0.00000086349742
6	0.00000000000351	0.00000003705098
7	0.00000000000397	0.00000000158929
8	0.00000000000529	0.00000000006641
9	0.00000000000408	0.00000000000596
10	0.00000000000379	0.00000000000395

The critical points are Steps 3 and 4(b). In Step 3, it is more efficient to compute $B_{ij}e$ only once for each block and to store it somewhere for use in all future iterations. This is only possible if sufficient memory is available; otherwise it is necessary to compute it each time it is needed. To obtain the vector ξ in Step 3(b), any of the methods of numerical iteration may be used, since the vector ξ is simply the stationary probability vector of an irreducible stochastic matrix *C*. In Step 4(b), each of the N systems of equation (10) can be written as $Px = r$ where $P = (I - B_{kk})^T$ and

$$r^T = \sum_{j>k} z_j B_{jk} + \sum_{j<k} \pi_j B_{jk}\,, \quad k = 1, 2, \ldots\ , N.$$

B_{kk} is a completely substochastic matrix in all instances, making B nonsingular. If the system is nearly completely decomposable, the vector r will have a small norm (NCD). If a direct technique is employed, the LU decomposition of $(I - B_{kk}),\ \ k = 1, 2, \ldots\ , N$ only needs to be done once because it is constant from iteration to iteration. We have an iteration algorithm within an iteration algorithm when we employ an iterative method. In this scenario, it is preferable to conduct only a few iterations (e.g., 8–12 for the Gauss–Seidel method) each time a solution of $(I - B_{kk})^T x = r$ is required, but to use the final estimate obtained at one step as the initial approximation the next time the same subsystem is required.

Illustrative example 5: Returning to the Courtois matrix, the table below shows the number of Iterations required to obtain full machine accuracy using the Gauss–Seidel iterative approach when the diagonal block equations in the IAD method are solved. It is clear that another iteration is now required.

Table 2: Iterative Aggregation/Disaggregation (IAD): Gauss–Seidel for Block solutions

IAD: Gauss–Seidel for Block solutions	
Iteration	**Residual: $\hat{\boldsymbol{\pi}} = (\mathbf{1} - \boldsymbol{B})$**
	$1.0E\ -03\ \times$
1	0.14117911369086
2	0.00016634452597
3	0.00000017031189
4	0.00000000015278
5	0.00000000000014
6	0.00000000000007
7	0.00000000000006
8	0.00000000000003
9	0.00000000000003
10	0.00000000000006

Also, when we look at the convergence of the inner Gauss–Seidel technique for diagonal blocks—as shown in the table below for the first diagonal block of size (3 × 3)—we can see that after only a few steps, the iterations become stuck. Progress slows to a crawl after around six iterations in each global iteration. As a result, when using iterative methods to solve the (inner) block equations, only a small number of iterations should be employed.

Table 3: Global Iteration of Stationary Probability Distribution

Inner Iteration	**1**	**2**	**3**	**4**
1	0.013160745	0.000009106488	0.00000002197727	0.00000000002345
2	0.0032775892	0.000002280232	0.00000000554827	0.00000000000593
3	0.0008932908	0.000000605958	0.00000000142318	0.00000000000151
4	0.0002001278	0.000000136332	0.00000000034441	0.00000000000037
5	0.0001468896	0.000000077107	0.00000000010961	0.00000000000011
6	0.0001124823	0.000000051518	0.00000000003470	0.00000000000003
7	0.0001178683	0.000000055123	0.00000000003872	0.00000000000002
8	0.0001156634	0.000000053697	0.00000000003543	0.00000000000002
9	0.0001155802	0.000000053752	0.00000000003596	0.00000000000002
10	0.0001149744	0.000000053446	0.00000000003562	0.00000000000002
11	0.0001145044	0.000000053234	0.00000000003552	0.00000000000002
12	0.0001140028	0.000000052999	0.00000000003535	0.00000000000002
13	0.0001135119	0.000000052772	0.00000000003520	0.00000000000002
14	0.0001130210	0.000000052543	0.00000000003505	0.00000000000002
15	0.0001125327	0.000000052316	0.00000000003490	0.00000000000002
16	0.0001120464	0.000000052090	0.00000000003475	0.00000000000002

Conclusion

The solutions of stationary distribution of Markov chain is investigated and computed using the decomposition and aggregation algorithmic numerical iterative methods by considering the following three steps: the left-hand eigenvector u_i of length n_i corresponding to the eigenvalue closest to 1, λ_{i_1}, in each block i, $1 \leq i \leq N$, is computed; the weights ξ_i, an approximate solution to the stationary probability vector π is estimated and the global solution $\pi^* = \xi_i u_i = (\xi_1 u_1, \ \xi_2 u_2, \ \xi_3 u_3, \ \ldots, \ \xi_N u_N)$ is computed. Concept of eigen values and vectors, Matrix operations such as Lower, upper and diagonal matrices and normalization principle are used with the help of some existing laws, theorems and formulas of Markov chain. The global solution $\pi^*$are obtained for the illustrative examples in each block i.

References

[1]. Romanovsky, V.I. (1970). *Discrete Markov Chains*, Wolters-Noord off, Groningen, Netherlands. pp. 23 – 44.

[2]. Stewart, W. J. (1994). *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, N.J. pp. 14 – 38.

[3]. Stewart, W. J. (2009). *Probability, Markov Chain, Queues and Simulation*, Princeton University Press, United Kingdom. pp. 1- 42.

[4]. Pesch, T., Schroder, S., Allelein, H. and Hake, J. (2015). A new Markov chain related statistical approach for modelling synthetic wind power time series, *New Journal of Physics*, *Dentsche Physikalishe,* **35**(2)**:** 64 – 85.

[5]. Uzun, B. and Kiral, E. (2017). Application of Markov chain-fuzzy states to gold price, *Science Direct. ELSEVIER*, **120**(1): 365 – 371.

[6]. Azizah, A., Welastica, R., Nur, F., Ruchjana, B. and Abdullah, A. (2019). An application of Markov chain for predicting rainfall data at West Java using data mining approach, *Earth and Environmental Science*, **303**(1): 203 – 216.

[7]. Clemence. T. (2019). Markov chain modelling of HIV, Tuberculosis, and Hepatitis B transmission in Ghana, *Hindawi, Interdisciplinary Perspective on Infectious Disease*, **27**(1)**:** 204 – 214.

[8]. Vermeer, S. And Trilling, D. (2020): Toward a better understanding of a new user journeys: A Markov chain approach. *Journalism Journal,* **21**(1): 879 – 894.