

An Optimized Modified Walk on Equations Monte Carlo Algorithm for Large Sparse Linear Systems

Venelin Todorov^{*†}, Fatima Sapundzhi[‡], Metodi Popstoilov[‡]

^{*}Department of Parallel Algorithms and Machine Learning with a Laboratory in Neurotechnologies, Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria

[†]Centre of Excellence in Informatics and Information and Communication Technologies, Acad. G. Bonchev Str. Bl. 25A, 1113 Sofia, Bulgaria

[‡]Department of Communication and computer engineering, Faculty of Engineering, South-West University "Neofit Rilski", 2700, Blagoevgrad, Bulgaria
Email: venelin@parallel.bas.bg, sapundzhi@swu.bg

Abstract—This paper investigates stochastic *Walk on Equations* (WE) methods for the numerical solution of linear algebraic systems and proposes an optimized modified WE algorithm aimed at large sparse matrices. Building on the classical WE Monte Carlo framework and its more recent modified version, we reformulate the main computational procedures, compare their algorithmic structure, and discuss their suitability for large-scale problems. To the best of our knowledge, this is the first study in the present setting that explicitly directs the WE methodology toward large sparse linear systems and interprets the modified construction as the basis for a new optimized stochastic solver.

The paper considers the classical WE method, the modified WE method, and standard stationary iterative schemes, namely Jacobi and Gauss–Seidel. Their behavior is examined through numerical experiments for systems of dimensions $n = 500$ and $n = 1000$, where the relative error is analyzed as a function of the iteration number. The obtained results show that the modified WE algorithm consistently outperforms the classical WE approach and both deterministic benchmarks in terms of convergence speed and final accuracy. In particular, the new method exhibits a significantly faster decay of the relative error and preserves its favorable behavior as the dimension increases.

These findings indicate that the modified WE framework is not merely a reformulation of the original stochastic scheme, but a practically more effective Monte Carlo strategy. The reported results also suggest that it provides a promising foundation for the development of optimized algorithms for large sparse linear systems, where efficiency, scalability, and controlled trajectory behavior are of primary importance.

I. INTRODUCTION

Systems of linear algebraic equations are among the most important mathematical models in computational science and engineering. They arise naturally in a broad range of applications, including structural mechanics, signal processing, control, computational physics, image analysis, economics, and data-driven modeling. In many practical problems, complex continuous phenomena are transformed, after discretization or linearization, into algebraic systems whose efficient numerical solution is essential for reliable simulation and decision making [1], [2], [4], [10].

The need for efficient solvers becomes even more pronounced in the case of large sparse systems. Such systems occur routinely in finite difference, finite element, and finite

volume discretizations of partial differential equations, as well as in optimization, network analysis, and inverse problems. Although classical deterministic techniques remain fundamental tools for this class of problems, their computational cost may become substantial when the dimension is very high or when repeated approximate solutions are required. For this reason, stochastic numerical methods provide an attractive complementary direction, especially when one seeks scalable procedures, reduced memory demands, or efficient estimation of selected components of the solution vector.

Among stochastic approaches, Monte Carlo methods occupy a distinguished position due to their flexibility, probabilistic interpretation, and suitability for large-scale computation [1], [2], [10]. In the context of linear algebraic systems, the *Walk on Equations* (WE) methodology offers a particularly interesting probabilistic framework. The classical WE approach and its later modification provide Monte Carlo estimators for one or all components of the solution vector by interpreting the underlying algebraic structure through random trajectories [3]. These ideas open the possibility of developing solvers that remain effective even when deterministic iterative methods become expensive or insufficiently flexible.

The present paper is devoted to a revised and more application-oriented study of WE-type Monte Carlo methods, with emphasis on large sparse matrices. More precisely, we reconsider the classical WE construction and the modified WE formulation, and we use this comparison as a basis for proposing an optimized modified algorithm designed for large sparse linear systems. The main motivation is that sparse problems require not only good convergence properties, but also algorithmic structures that can exploit reduced connectivity and controlled trajectory behavior. In this setting, the modified WE strategy appears especially promising.

The contribution of the paper is therefore twofold. First, we provide a unified presentation of the classical and modified WE methods in a form suitable for large sparse systems. Second, we show through numerical experiments that the modified WE approach has a clear computational advantage over the classical WE scheme and over standard stationary

iterative methods such as Jacobi and Gauss–Seidel. This supports the view that the modified formulation is not merely a variant of the original algorithm, but a more efficient stochastic framework that can be further optimized for sparse matrix computations.

The remainder of the paper is organized as follows. The next section presents the classical and optimized WE algorithms in a unified pseudocode form. After that, numerical experiments are reported in order to compare their convergence behavior with that of standard deterministic iterative methods. Finally, the last section summarizes the main conclusions and outlines possible directions for the further development of optimized stochastic solvers for large sparse linear systems.

II. THE DESCRIPTION OF THE WE MONTE CARLO ALGORITHMS

This section summarizes the three stochastic schemes considered in the paper: the classical *Walk on Equations* (WE) procedure and the new optimized modified WE formulation. The presentation covers estimators for a single component of the solution vector as well as procedures intended to approximate all components simultaneously.

A. Classical WE Monte Carlo Estimator for a Single Component

We first recall the original WE Monte Carlo method for approximating one prescribed component of the solution of a linear system, following the framework in [3]. The input consists of the system matrix B , the right-hand side vector f , a relaxation parameter $\gamma \in (0, 1]$, and the number N of random trajectories.

The method begins by constructing an auxiliary matrix $A = \{a_{ij}\}_{i,j=1}^n$ according to

$$a_{ij} = \begin{cases} 1 - \gamma, & i = j, \\ -\gamma \frac{b_{ij}}{b_{ii}}, & i \neq j. \end{cases}$$

For each row, one computes the quantity

$$\text{asum}(i) = \sum_{j=1}^n |a_{ij}|, \quad i = 1, \dots, n,$$

and then forms the transition probabilities

$$p_{ij} = \frac{|a_{ij}|}{\text{asum}(i)}, \quad i, j = 1, \dots, n.$$

To estimate the component x_{i_0} , each random walk starts from the state i_0 . The score of the trajectory is initialized with the corresponding right-hand side contribution. At every subsequent step, the trajectory either terminates, with probability determined by the row sum $\text{asum}(m)$, or moves to a new state selected according to the transition probabilities $\{p_{mj}\}_{j=1}^n$. Since the coefficients may have different signs, a sign variable is propagated along the path and is used to update the cumulative contribution of the walk. After

generating N independent trajectories, the estimator is taken as the arithmetic mean of the accumulated scores:

$$x_{i_0} \approx \frac{S}{N}.$$

Thus, the classical one-component WE method approximates the desired entry of the solution by averaging signed contributions collected along random transitions associated with the transformed matrix.

B. Classical WE Monte Carlo Estimator for the Full Solution Vector

The original WE methodology can also be adapted to approximate all solution components simultaneously. In this setting, one assumes, without loss of generality for the algorithmic construction, that the relevant coefficients are non-negative. The preprocessing stage remains the same, namely, the construction of the auxiliary matrix and the corresponding transition mechanism.

For each component, two auxiliary arrays are maintained: a score accumulator $S(i)$ and a visit counter $V(i)$, for $i = 1, \dots, n$. During each Monte Carlo realization, the procedure starts from a randomly selected equation, generates a trajectory through the state space, and records all visited states. Every time a state is encountered, its visit counter is increased, while the score contributions associated with the visited path are added to the corresponding accumulators.

After all random trajectories have been generated, the estimate of the j th component is obtained by normalizing the accumulated score by the number of visits:

$$x_j \approx \frac{S(j)}{V(j)}, \quad j = 1, \dots, n,$$

with the convention that $V(j)$ is replaced by $\max\{1, V(j)\}$ in order to avoid division by zero. In this way, the full solution vector is recovered from local path information gathered during the random walks.

The pseudocode version of the algorithms is given below.

C. New Optimized WE Monte Carlo Estimator for a Single Component

We next describe the newer WE-type algorithm for approximating a single component of the solution vector. In contrast to the classical procedure, this formulation is built directly in terms of the matrix A , the vector b , and the number of random trajectories N .

The preprocessing stage requires the computation of the row sums

$$\text{asum}(i) = \sum_{j=1}^n a_{ij}, \quad i = 1, \dots, n,$$

as well as the total sum of the right-hand side entries,

$$s_b = \sum_{i=1}^n b_i.$$

The estimator is generated by a sequence of random trajectories. At the beginning of each trajectory, an index is selected

Algorithm 1 Classical WE estimator for one component

Require: Matrix $B = (b_{ij})_{i,j=1}^n$, right-hand side vector b , relaxation parameter $\gamma \in (0, 1]$, target index i_0 , number of walks N

Ensure: Approximation of x_{i_0}

```

1: for  $i = 1$  to  $n$  do
2:   for  $j = 1$  to  $n$  do
3:     if  $i = j$  then
4:        $a_{ij} \leftarrow 1 - \gamma$ 
5:     else
6:        $a_{ij} \leftarrow -\gamma b_{ij}/b_{ii}$ 
7:     end if
8:   end for
9:    $r_i \leftarrow \sum_{j=1}^n |a_{ij}|$ 
10:  for  $j = 1$  to  $n$  do
11:     $p_{ij} \leftarrow |a_{ij}|/r_i$ 
12:  end for
13: end for
14:  $S \leftarrow 0$ 
15: for  $k = 1$  to  $N$  do
16:    $m \leftarrow i_0, \quad \sigma \leftarrow 1, \quad Y \leftarrow b_m$ 
17:   while true do
18:     Draw  $u \sim U(0, 1)$ 
19:     if  $u \geq r_m$  then
20:       break
21:     else
22:       Draw  $j$  from the discrete law  $(p_{m1}, \dots, p_{mn})$ 
23:        $\sigma \leftarrow \sigma \cdot \text{sgn}(a_{mj})$ 
24:        $m \leftarrow j$ 
25:        $Y \leftarrow Y + \sigma b_m$ 
26:     end if
27:   end while
28:    $S \leftarrow S + Y$ 
29: end for
30: return  $S/N$ 

```

according to a probability law induced by the entries of b . If the current state is denoted by p_j , then the next move is chosen from the augmented probability distribution

$$(a_{p_j,1}, a_{p_j,2}, \dots, a_{p_j,n}, 1 - \text{asum}(p_j)).$$

Hence, at each step the trajectory either jumps to one of the equations or enters an absorbing state. The contribution to the score is not updated at every intermediate step; instead, it is computed only when absorption occurs. In that case, the walk contributes

$$b(i_0) + s_b \frac{a_{p_j,i_0}}{1 - \text{asum}(p_j)}$$

to the estimator of the target component x_{i_0} . Averaging over all N independent realizations yields

$$x_{i_0} \approx \frac{S}{N}.$$

Algorithm 2 Classical WE estimator for the full solution vector

Require: Matrix $B = (b_{ij})_{i,j=1}^n$, right-hand side vector b , relaxation parameter $\gamma \in (0, 1]$, number of walks N

Ensure: Approximation of $x = (x_1, \dots, x_n)^\top$

```

1: for  $i = 1$  to  $n$  do
2:   for  $j = 1$  to  $n$  do
3:     if  $i = j$  then
4:        $a_{ij} \leftarrow 1 - \gamma$ 
5:     else
6:        $a_{ij} \leftarrow -\gamma b_{ij}/b_{ii}$ 
7:     end if
8:   end for
9:    $r_i \leftarrow \sum_{j=1}^n |a_{ij}|$ 
10:  for  $j = 1$  to  $n$  do
11:     $p_{ij} \leftarrow |a_{ij}|/r_i$ 
12:  end for
13: end for
14: for  $i = 1$  to  $n$  do
15:    $S_i \leftarrow 0, \quad V_i \leftarrow 0$ 
16: end for
17: for  $k = 1$  to  $N$  do
18:   Draw starting state  $m$  uniformly from  $\{1, \dots, n\}$ 
19:    $\mathcal{P} \leftarrow []$  ▷ current path
20:   while true do
21:     append  $m$  to  $\mathcal{P}$ 
22:      $V_m \leftarrow V_m + 1$ 
23:     for all  $q \in \mathcal{P}$  do
24:        $S_q \leftarrow S_q + b_q$ 
25:     end for
26:     Draw  $u \sim U(0, 1)$ 
27:     if  $u \geq r_m$  then
28:       break
29:     else
30:       Draw next state  $j$  from  $(p_{m1}, \dots, p_{mn})$ 
31:        $m \leftarrow j$ 
32:     end if
33:   end while
34: end for
35: for  $i = 1$  to  $n$  do
36:    $V_i \leftarrow \max\{1, V_i\}$ 
37:    $x_i \leftarrow S_i/V_i$ 
38: end for
39: return  $(x_1, \dots, x_n)^\top$ 

```

The key distinction from the classical approach is that the new estimator is evaluated at the end of the trajectory, rather than being incrementally updated along the entire path.

D. New Optimized WE Monte Carlo Estimator for All Components

The previous idea can be extended to the simultaneous approximation of the full solution vector. The main principle is to treat visited states as potential starting locations for component estimation and to accumulate information accordingly. For

each component i , the algorithm keeps a score variable $S(i)$ and a counter $cc(i)$ recording how many trajectories contribute to the approximation of x_i .

As before, the row sums $\text{asum}(i)$ and the total sum s_b are computed in advance. Then, during each Monte Carlo realization, one selects uniformly at random a component index $i_0 \in \{1, \dots, n\}$ whose value is to be estimated in that realization. The counter $cc(i_0)$ is increased by one. A trajectory is then generated exactly as in the one-component version: first, an index is sampled according to the distribution induced by b , and then transitions are performed according to the augmented probability vector

$$(a_{p_j,1}, a_{p_j,2}, \dots, a_{p_j,n}, 1 - \text{asum}(p_j)).$$

When the absorbing state is reached, the quantity

$$b(i_0) + s_b \frac{a_{p_j,i_0}}{1 - \text{asum}(p_j)}$$

is added to $S(i_0)$. At the end of the simulation, the estimator for each component is computed as

$$x_i \approx \frac{S(i)}{cc(i)}, \quad i = 1, \dots, n.$$

This version is closely related to the Sequential Monte Carlo philosophy introduced by Halton [5], [6], [7], [8] and can be interpreted as using a control-variates mechanism in the sense of [9].

Algorithm 3 New Optimized WE estimator for one component

Require: Matrix $A = (a_{ij})_{i,j=1}^n$, vector b , target index i_0 , number of trajectories N

Ensure: Approximation of x_{i_0}

```

1: for  $i = 1$  to  $n$  do
2:    $r_i \leftarrow \sum_{j=1}^n a_{ij}$ 
3: end for
4:  $s_b \leftarrow \sum_{i=1}^n b_i$ 
5:  $S \leftarrow 0$ 
6: for  $k = 1$  to  $N$  do
7:   Draw initial state  $m$  from the discrete law induced by
      $(b_1, \dots, b_n)$ 
8:   while true do
9:     Draw  $j$  from the augmented law
         $(a_{m1}, a_{m2}, \dots, a_{mn}, 1 - r_m)$ 
10:    if  $j = n + 1$  then
11:       $Y \leftarrow b_{i_0} + s_b \frac{a_{mi_0}}{1 - r_m}$ 
12:       $S \leftarrow S + Y$ 
13:      break
14:    else
15:       $m \leftarrow j$ 
16:    end if
17:  end while
18: end for
19: return  $S/N$ 

```

Algorithm 4 New Optimized WE estimator for the full solution vector

Require: Matrix $A = (a_{ij})_{i,j=1}^n$, vector b , number of trajectories N

Ensure: Approximation of $x = (x_1, \dots, x_n)^\top$

```

1: for  $i = 1$  to  $n$  do
2:    $r_i \leftarrow \sum_{j=1}^n a_{ij}$ 
3: end for
4:  $s_b \leftarrow \sum_{i=1}^n b_i$ 
5: for  $i = 1$  to  $n$  do
6:    $S_i \leftarrow 0, \quad c_i \leftarrow 0$ 
7: end for
8: for  $k = 1$  to  $N$  do
9:   Draw target component  $i_0$  uniformly from  $\{1, \dots, n\}$ 
10:   $c_{i_0} \leftarrow c_{i_0} + 1$ 
11:  Draw initial state  $m$  from the discrete law induced by
      $(b_1, \dots, b_n)$ 
12:  while true do
13:    Draw  $j$  from the augmented law
         $(a_{m1}, a_{m2}, \dots, a_{mn}, 1 - r_m)$ 
14:    if  $j = n + 1$  then
15:       $S_{i_0} \leftarrow S_{i_0} + b_{i_0} + s_b \frac{a_{mi_0}}{1 - r_m}$ 
16:      break
17:    else
18:       $m \leftarrow j$ 
19:    end if
20:  end while
21: end for
22: for  $i = 1$  to  $n$  do
23:    $c_i \leftarrow \max\{1, c_i\}$ 
24:    $x_i \leftarrow S_i / c_i$ 
25: end for
26: return  $(x_1, \dots, x_n)^\top$ 

```

III. COMPARISON WITH THE JACOBI AND GAUSS–SEIDEL METHODS

The two Monte Carlo procedures considered above differ substantially from classical deterministic iterative schemes such as the Jacobi and Gauss–Seidel methods. While Jacobi and Gauss–Seidel generate successive approximations to the entire solution vector, the Monte Carlo algorithms are designed to estimate a prescribed component x_{i_0} through random sampling. This distinction affects their convergence behavior, computational cost, and practical range of applicability.

The Jacobi method is based on the decomposition

$$A = D - (L + U),$$

where D is the diagonal part of A , while L and U are its strictly lower and strictly upper triangular parts. Its iteration has the form

$$x^{(k+1)} = D^{-1}(L + U)x^{(k)} + D^{-1}b.$$

The Gauss–Seidel method uses the splitting

$$A = (D - L) - U$$

and updates the iterates according to

$$x^{(k+1)} = (D - L)^{-1}Ux^{(k)} + (D - L)^{-1}b.$$

In both cases, convergence is guaranteed under standard assumptions such as strict diagonal dominance, while Gauss–Seidel typically converges faster than Jacobi because newly computed components are used immediately within each iteration.

In contrast, the old and the new Monte Carlo algorithms do not construct the full solution vector iteratively. Instead, they approximate the desired component by averaging the scores of randomly generated trajectories. Their convergence is therefore statistical rather than deterministic: the error decreases proportionally to $N^{-1/2}$, where N is the number of simulated walks. Consequently, the Monte Carlo estimators are unbiased or asymptotically unbiased, depending on the exact formulation, but they are affected by sampling variance.

A major advantage of the Jacobi and Gauss–Seidel methods is that, when convergence is ensured, they usually provide more accurate approximations of the full solution vector with considerably fewer operations than Monte Carlo simulation. For moderately sized systems, or when all components of the solution are required, deterministic iterative methods are therefore generally preferable. Moreover, Gauss–Seidel often achieves a better convergence rate than Jacobi, especially for strongly diagonally dominant systems.

On the other hand, the Monte Carlo algorithms possess several features that make them attractive in large-scale settings. First, they are naturally suited for estimating only one or a few components of the solution, without computing the entire vector. Second, the simulation of random trajectories can be parallelized very efficiently, since the walks are independent. Third, their performance is less sensitive to the global dimension of the system, which may be advantageous for very large sparse matrices. In such cases, the stochastic approach may become competitive, especially when a componentwise estimate is sufficient.

The difference between the old and the new Monte Carlo schemes is also relevant in this comparison. The old algorithm accumulates signed contributions throughout the walk and relies on transition probabilities derived from the absolute values of the transformed coefficients. The new algorithm, by contrast, assigns the contribution at the absorbing step and incorporates the coefficients more directly into the trajectory scoring rule. As a result, the new method may exhibit improved numerical stability or reduced variance for suitable classes of matrices, making it a more effective stochastic alternative to the classical iterative schemes in problems where random-walk-based estimation is appropriate.

In summary, Jacobi and Gauss–Seidel are usually the methods of choice when one seeks the full solution of a convergent system with moderate computational cost. The Monte Carlo methods are more suitable when only selected components

are needed, when massive parallelism is available, or when the problem size makes deterministic full-vector iteration less attractive.

IV. NUMERICAL EXAMPLE AND DISCUSSION

To illustrate the comparative behavior of the considered methods, we present two numerical experiments for sparse linear systems of dimensions $n = 500$ and $n = 1000$. The corresponding results are shown in Figures 1 and 2, where the relative error is plotted against the iteration number on a logarithmic scale. In both cases, four methods are compared: the classical WE algorithm (Old WE), the modified WE algorithm (New WE), the Jacobi method, and the Gauss–Seidel method.

A first important observation is that the modified WE algorithm exhibits the fastest decay of the relative error in both experiments. In the case $n = 500$, the error decreases almost linearly on the logarithmic scale, dropping from approximately 10^{-1} at the first iteration to about 10^{-12} at the fifteenth iteration. A very similar pattern is observed for $n = 1000$, where the final error is again close to machine-precision level. This behavior indicates a very stable and efficient convergence process of the new stochastic formulation.

The classical WE method also demonstrates clear convergence, but at a substantially slower rate. In both figures, its error decreases by several orders of magnitude, reaching approximately 10^{-8} after 15 iterations. Thus, although the old WE procedure performs significantly better than the classical stationary iterative methods, it is consistently outperformed by the modified WE algorithm throughout the entire iteration range.

The deterministic benchmarks, namely the Jacobi and Gauss–Seidel methods, converge much more slowly. Among them, Gauss–Seidel is superior to Jacobi, which is in agreement with the well-known theoretical and practical behavior of these two schemes. Nevertheless, even Gauss–Seidel remains clearly less accurate than both Monte Carlo WE variants in the present experiments. At the final iteration, its relative error is still around 10^{-4} – 10^{-3} , whereas the Jacobi method remains near 10^{-2} .

Another noteworthy feature is the close similarity between the two plots for $n = 500$ and $n = 1000$. The convergence curves for each method nearly coincide across the two system sizes. This suggests that, for the tested class of matrices, the relative performance of the algorithms is not significantly affected by doubling the dimension from 500 to 1000. In particular, the modified WE method preserves its strong advantage as the problem size increases, which is an encouraging sign for its applicability to larger-scale problems.

From an interpretative viewpoint, the results support the idea that the new WE algorithm benefits from a more favorable trajectory structure than the classical WE method. The much steeper slope of the New WE curve indicates that its estimator captures the relevant solution information more efficiently at each iteration. In contrast, the Old WE method improves steadily but requires many more iterations to achieve the same

level of accuracy. The much flatter Jacobi and Gauss–Seidel curves show that, in this example, the deterministic stationary iterations are considerably less competitive.

Overall, the two numerical experiments demonstrate a clear hierarchy of performance:

$$\text{New WE} > \text{Old WE} > \text{Gauss–Seidel} > \text{Jacobi}.$$

Therefore, the modified WE algorithm appears to be the most effective method among those tested, combining rapid error reduction with stable behavior across both matrix dimensions. The results strongly suggest that the proposed stochastic improvement is not merely marginal, but leads to a substantial acceleration of convergence in comparison with both the original WE scheme and the standard iterative solvers.

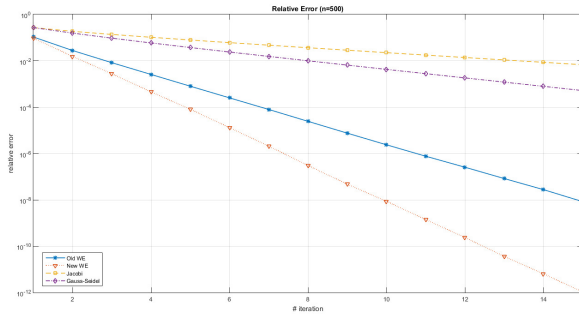


Fig. 1: Relative error versus iteration number for a sparse linear system of size $n = 500$.

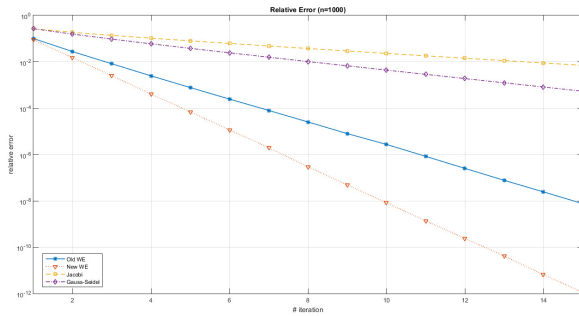


Fig. 2: Relative error versus iteration number for a sparse linear system of size $n = 1000$.

Figure 3 provides a compact summary of the convergence behavior observed in the two numerical experiments. The plot clearly shows that the modified WE algorithm achieves the fastest reduction of the relative error, exhibiting an almost linear decay on the logarithmic scale and reaching the lowest final error among all compared methods. The classical WE algorithm also converges steadily and significantly outperforms the deterministic stationary schemes, but its rate of error decay remains visibly slower than that of the modified WE method. At the same time, Gauss–Seidel performs better than Jacobi, which is consistent with its usual iterative advantage, although

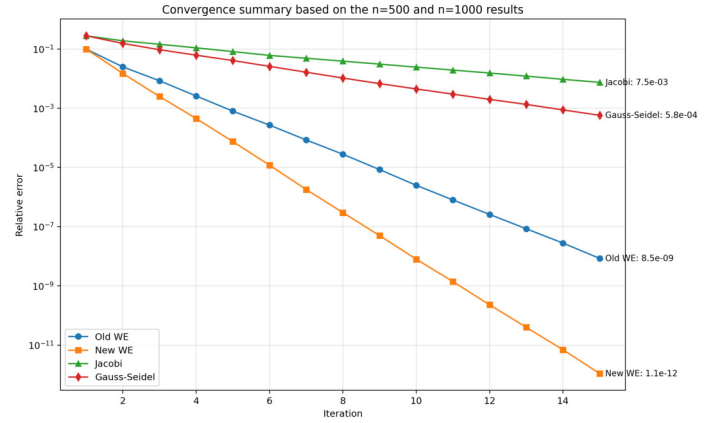


Fig. 3: Summary of the convergence behavior observed for the considered methods based on the experiments for $n = 500$ and $n = 1000$.

both classical deterministic methods remain substantially less efficient in the present setting. Overall, the summarized convergence curves confirm a stable hierarchy of performance, with the modified WE method showing the most favorable behavior, followed by the classical WE algorithm, Gauss–Seidel, and Jacobi.

V. DISCUSSION

The numerical results presented in Figures 1 and 2 reveal a clear and consistent advantage of the modified WE algorithm over the other methods considered in this study. In both test problems, the new WE scheme achieves the fastest reduction of the relative error, with an almost linear decay on the logarithmic scale. This behavior indicates a highly stable convergence pattern and suggests that the modified stochastic construction uses the trajectory information more efficiently than the classical WE formulation.

The old WE algorithm also demonstrates good convergence properties and performs substantially better than the two deterministic stationary methods, Jacobi and Gauss–Seidel. However, its convergence rate is noticeably slower than that of the new WE method. This confirms that the modification introduced in the new algorithm is not only a formal reformulation, but leads to a practically significant improvement in accuracy. In particular, the gap between the two WE curves becomes increasingly pronounced as the number of iterations grows, which shows that the benefit of the modified estimator accumulates throughout the computational process.

The comparison with Jacobi and Gauss–Seidel is also instructive. As expected, Gauss–Seidel performs better than Jacobi, but both deterministic methods remain clearly less competitive than the stochastic WE approaches in the present experiments. Their error curves decrease much more slowly and remain several orders of magnitude above the error levels reached by the new WE algorithm after the same number of iterations. This suggests that, at least for the class of test systems

considered here, the stochastic WE methodology provides a much more effective framework for iterative approximation.

Another important point is the similarity of the results for $n = 500$ and $n = 1000$. The overall shape and ordering of the convergence curves are almost unchanged when the system dimension is doubled. This indicates that the relative performance of the algorithms is stable with respect to problem size and supports the idea that the modified WE scheme has good scalability properties. Such behavior is particularly promising when one has in mind applications to large sparse linear systems, where both computational efficiency and robustness are essential.

From a broader perspective, the results suggest that the modified WE approach may serve as the basis for a new optimized stochastic algorithm for large sparse matrices. The faster error decay observed in the experiments points to a more favorable trajectory mechanism, which is likely to become even more important in high-dimensional sparse settings. In this sense, the present study not only compares existing schemes, but also motivates further development of optimized WE-type solvers tailored to sparse matrix structures.

VI. CONCLUSION

In this paper, we revisited stochastic *Walk on Equations* techniques for the numerical solution of linear algebraic systems and presented a reformulated perspective on the classical and modified WE algorithms. Particular attention was given to the comparison between the original Monte Carlo estimator and the newer stochastic construction, whose trajectory structure is based on an absorbing-state mechanism. This study was motivated by the need for more efficient randomized solvers for large-scale linear systems, especially in settings where the system dimension makes standard approaches less attractive.

A central contribution of the present work is the extension of the WE methodology toward large sparse matrices. To the best of our knowledge, this is the first time that this stochastic WE-type framework is explicitly considered and discussed in the context of large sparse linear systems within the present formulation. In this sense, the paper does not only revisit existing Monte Carlo ideas, but also broadens their scope toward an important class of problems arising in scientific computing, engineering, and large computational models.

The numerical experiments reported for systems of size $n = 500$ and $n = 1000$ clearly indicate that the modified WE method achieves the best performance among the tested approaches. In both examples, its relative error decreases much faster than that of the classical WE method and also significantly faster than that of the deterministic stationary schemes Jacobi and Gauss–Seidel. The old WE algorithm also exhibits stable convergence and remains more effective than the two standard iterative benchmarks, but it is consistently inferior to the new WE formulation in terms of convergence speed and final accuracy.

An important conclusion of the study is that the modified WE approach provides the basis for a new optimized stochastic algorithm intended for large sparse matrices. The numerical

behavior suggests that the absorbing-state construction improves the efficiency of the estimator and makes the method especially promising for high-dimensional sparse problems, where computational savings and trajectory control are of essential importance. Thus, beyond the comparative analysis itself, the paper proposes a new optimized modified algorithmic direction for the approximate solution of large sparse linear algebraic systems.

Another noteworthy outcome is the stability of the observed behavior across the two tested dimensions. The convergence patterns remain essentially unchanged when the system size increases from 500 to 1000, indicating that the modified WE method preserves its efficiency as the problem dimension grows. This supports its potential applicability in larger sparse settings, where scalability is a key requirement.

Overall, the obtained results confirm that the modified WE methodology constitutes a substantial improvement over the classical stochastic formulation for the considered class of problems. The observed reduction in relative error, together with the robust behavior across different matrix sizes, demonstrates that the new construction is a promising tool for the efficient approximate solution of large linear systems and, in particular, opens a new perspective for treating large sparse matrices by Monte Carlo techniques.

Future work may proceed in several directions. A natural continuation is the development of a fully optimized implementation tailored specifically to sparse matrix structures, including storage-aware trajectory generation and reduced-cost transition mechanisms. It would also be valuable to investigate the hybrid WE approach in more detail, especially in regimes where neither the classical nor the modified method is uniformly dominant. In addition, further theoretical analysis of convergence, variance, and computational complexity for sparse systems would help to establish the full potential of the proposed optimized modified WE algorithm in practical large-scale applications.

ACKNOWLEDGMENT

Venelin Todorov was partially supported by the Centre of Excellence in Informatics and ICT under the Grant No BG16RFPR002-1.014-0018, financed by the Research, Innovation and Digitalization for Smart Transformation Programme 2021-2027 and co-financed by the European Union and by BNSF under Projects KP-06-N82/4 and KP-06-N62/6.

REFERENCES

- [1] Curtiss, J.H.: Monte Carlo methods for the iteration of linear operators. *Journal of Mathematics and Physics* **32**(4), 209–232 (1954)
- [2] Curtiss, J.H.: A theoretical comparison of the efficiencies of two classical methods and a Monte Carlo method for computing one component of the solution of a set of linear algebraic equations. In: *Proceedings of the Symposium on Monte Carlo Methods*, John Wiley and Sons, 191–233 (1956)
- [3] Dimov, I.T., Maire, S., Sellier, J.M.: A New Walk on Equations Monte Carlo Method for Linear Algebraic Problems. *Applied Mathematical Modelling* **39**(15), 4494–4510 (2015)

- [4] Dimov, I., Tonev, O.: Performance Analysis of Monte Carlo Algorithms for Some Models of Computer Architectures. In: *International Youth Workshop on Monte Carlo Methods and Parallel Algorithms – Primorsko*, eds. Bl. Sendov, I. Dimov, World Scientific, Singapore, 91–95 (1990)
- [5] Halton, J.: Sequential Monte Carlo. *Proceedings of the Cambridge Philosophical Society* **58**, 57–78 (1962)
- [6] Halton, J.: Sequential Monte Carlo. University of Wisconsin, Madison, Mathematics Research Center Technical Summary Report No. 816 (1967)
- [7] Halton, J., Zeidman, E.A.: Monte Carlo integration with sequential stratification. University of Wisconsin, Madison, Computer Science Department Technical Report No. 61 (1969)
- [8] Halton, J.: Sequential Monte Carlo for linear systems – a practical summary. *Monte Carlo Methods and Applications* **14**, 1–27 (2008)
- [9] Maire, S.: Reducing variance using iterated control variates. *Journal of Statistical Computation and Simulation* **73**(1), 1–29 (2003)
- [10] Spanier, J., Gelbard, E.: *Monte Carlo Principles and Neutron Transport Problem*. Addison–Wesley (1969)