

# A New Monte Carlo Algorithm for Linear Algebraic Systems Based on the “Walk on Equations” Algorithm

Venelin Todorov<sup>\*†</sup>, Nikolay Ikonov<sup>\*</sup>, Ivan Dimov<sup>†</sup>, Rayna Georgieva<sup>†</sup>

<sup>\*</sup>Institute of Mathematics and Informatics

Bulgarian Academy of Sciences

8 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria

Email: vtodorov@math.bas.bg, nikonov@math.bas.bg

<sup>†</sup>Institute of Information and Communication Technologies

Bulgarian Academy of Sciences

25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria

Email: venelin@parallel.bas.bg, ivdimov@bas.bg, rayna@parallel.bas.bg

**Abstract**—A new Monte Carlo algorithm for solving systems of Linear Algebraic (LA) equations is presented and studied. The algorithm is based on the “Walk on Equations” Monte Carlo method recently developed by Ivan Dimov, Sylvain Maire and Jean Michel Sellier [4]. The algorithm is optimized by choosing the appropriate values for the relaxation parameters which leads to dramatic reduction in time and lower relative errors for a given number of iterations. Numerical tests are performed for examples with matrices of different size and on a system coming from a finite element approximation of a problem describing a beam structure in constructive mechanics.

## I. INTRODUCTION

**M**ANY scientific and engineering applications are based on the problems of solving systems of LA equations. For some applications it is also important to compute directly the inner product of a given vector and the solution vector of a LA system. In Monte Carlo numerical algorithms we construct a Markov process and prove that the mathematical expectation of the process is equal to the unknown solution of the problem [3]. Iterative Monte Carlo algorithms can be defined as terminated Markov chains:

$$T = \{\alpha_{t_0} \rightarrow \alpha_{t_1} \rightarrow \alpha_{t_2} \dots \alpha_{t_k}\}, \quad (1)$$

where  $\alpha_{t_q}$ ,  $q = 1, \dots, i$  is one of the absorbing states. By  $A$  and  $B$  we denote matrices of size  $n \times n$ , i.e.,  $A, B \in \mathbb{R}^{n \times n}$ . We use the following presentation of matrices:

$$A = \{a_{ij}\}_{i,j=1}^n = (a_{11}, \dots, a_{i1}, \dots, a_n)^t,$$

where  $a_i = (a_{i1}, \dots, a_{in})$ ,  $i = 1, \dots, n$  and the symbol  $t$  means *transposition*. The following norms of vectors:

$$\|b\| = \|b\|_1 = \sum_{i=1}^n |b_i|, \quad \|a_i\| = \|a_i\|_1 = \sum_{j=1}^n |a_{ij}|$$

The first author is supported by the Bulgarian National Science Fund under Projects DN 12/5-2017 and DN 12/4-2017 and by the Bulgarian Academy of Sciences through the Program for Career Development of Young Scientists, Grant DFNP-17-88/28.07.2017.

and matrices

$$\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \quad \|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$$

are used, where  $b \in \mathbb{R}^n$ .

We consider a system of LA equations

$$Bx = f, \quad (2)$$

where  $B = \{b_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$  is a given matrix;  $f = (f_1, \dots, f_n)^t \in \mathbb{R}^{n \times 1}$  and  $v = (v_1, \dots, v_n) \in \mathbb{R}^{1 \times n}$  are given vectors.

We deal with the matrix  $A = \{a_{ij}\}_{i,j=1}^n$ , such that  $A = I - DB$ , where  $D$  is a diagonal matrix  $D = \text{diag}(d_1, \dots, d_n)$  and  $d_i = \frac{\gamma}{b_{ii}}$ ,  $i = 1, \dots, n$ , and  $\gamma \in (0, 1]$  is a parameter that can be used to accelerate the convergence. The system (2) can be presented in the form of equation

$$x = Ax + b, \quad (3)$$

where  $b = Df$ . Let us suppose that the matrix  $B$  is diagonally dominant. It easily follows that if  $B$  is a diagonally dominant matrix, then the elements of the matrix  $A$  must satisfy the following condition:  $\sum_{j=1}^n |a_{ij}| \leq 1$ ,  $i = 1, \dots, n$ .

A *stationary linear iterative algorithm* [3] can be used:

$$x_k = Ax_{k-1} + b, \quad k = 1, 2, \dots \quad (4)$$

and the solution  $x$  can be presented in a form of a Neumann series

$$x = \sum_{k=0}^{\infty} A^k b = b + Ab + A^2b + A^3b + \dots \quad (5)$$

The *stationary linear iterative Monte Carlo algorithm* is based on (5). As a result, the convergence of the Monte Carlo algorithm depends on the truncation error of the series (4) [3]. We are interested to evaluate the linear form  $W(x)$  of the solution  $x$  of the system (3), i.e.,  $W(x) \equiv (w, x) = \sum_{i=1}^n w_i x_i$ ,

where  $w \in \mathbb{R}^{n \times 1}$ . We shall define a random variable  $X[w]$ , which expectation is equal to the above defined linear form, i.e.,  $EX[w] = W(x)$  using a discrete Markov process with a finite set of states. Then the problem is to determine repeated realizations of  $X[w]$  and of connecting them into a proper statistical estimator of  $W(x)$ .

Consider an initial density vector  $p = \{p_i\}_{i=1}^n \in \mathbb{R}^n$ , such that  $p_i \geq 0, i = 1, \dots, n$  and  $\sum_{i=1}^n p_i = 1$ . Consider also a transition density matrix  $P = \{p_{ij}\}_{i,j=1}^n \in \mathbb{R}^{n \times n}$ , such that  $p_{ij} \geq 0, i, j = 1, \dots, n$  and  $\sum_{j=1}^n p_{ij} = 1$ , for any  $i = 1, \dots, n$ .

We will be dealing with *permissible* densities [3]  $\mathcal{P}_b$  and  $\mathcal{P}_A$ . It follows easily that in such a way the random trajectories constructed to solve the problems under consideration never visit zero elements of the matrix. Such an approach decreases the computational complexity of the algorithms [2]. It is also very convenient when large sparse matrices are used.

## II. PROBABILISTIC REPRESENTATION OF THE ALGORITHM

Consider a real linear system of the form  $x = Ax + b$  where the matrix  $A$  of size  $n$  is such that the convergence radius  $\varrho(A) < 1$ , its coefficients  $a_{ij}$  are real numbers and

$$\sum_{j=1}^n |a_{ij}| \leq 1, \forall 1 \leq i \leq n.$$

We now define a Markov chain  $T_k$  with  $n + 1$  states  $\alpha_1, \dots, \alpha_n, \alpha_{n+1}$ , such that

$$P(\alpha_{k+1} = j | \alpha_k = i) = |a_{ij}|$$

if  $i \neq n + 1$  and

$$P(\alpha_{k+1} = n + 1 | \alpha_k = n + 1) = 1.$$

We also define a vector  $c$  such that  $c(i) = b(i)$  if  $1 \leq i \leq n$  and  $c(n + 1) = 0$ . Denote by  $\tau = (\alpha_0, \alpha_1, \dots, \alpha_k, \alpha_{n+1})$  a random trajectory that starts at the initial state  $\alpha_0 < n + 1$  and passes through  $(\alpha_1, \dots, \alpha_k)$  until the absorbing state  $\alpha_{k+1} = n + 1$ . The probability to follow the trajectory  $\tau$  is  $P(\tau) = p_{\alpha_0} p_{\alpha_0 \alpha_1} \dots p_{\alpha_{k-1} \alpha_k} p_{\alpha_k}$ . We use the MAO algorithm (see [1], [5]) for the initial density vector  $p = \{p_\alpha\}_{\alpha=1}^n$  and for the transition density matrix  $P = \{p_{\alpha\beta}\}_{\alpha,\beta=1}^n$ , as well. The weights  $Q_\alpha$  are defined:

$$Q_m = Q_{m-1} \frac{a_{\alpha_{m-1}, \alpha_m}}{p_{\alpha_{m-1}, \alpha_m}}, \quad m = 1, \dots, k, \quad Q_0 = \frac{c_{\alpha_0}}{p_{\alpha_0}}. \quad (6)$$

The estimator  $X_\alpha(\tau)$  can be presented as  $X_\alpha(\tau) = c_\alpha + Q_k \frac{a_{\alpha_k, \alpha}}{p_{\alpha_k, \alpha}}$ ,  $\alpha = 1, \dots, n$  taken with a probability  $P(\tau) = p_{\alpha_0} p_{\alpha_0 \alpha_1} \dots p_{\alpha_{k-1}, k} p_{\alpha_k}$ .

For the convergence of the process we use that the random variable  $X_\alpha(\tau)$  is an unbiased estimator of  $x_\alpha$  [4], i.e.

$$E\{X_\alpha(\tau)\} = x_\alpha. \quad (7)$$

Consider the variance of the random variable  $X_\alpha(\tau)$  for evaluation the linear form for the solution  $W(x)$ . We use the following notations:  $\bar{A} = \{|a_{ij}|\}_{i,j=1}^n$ ,  $\hat{c} = \{c_i^2\}_{i=1}^{n+1}$ . The

special choice of the probability densities leads to the Markov chain:

$$c_{\alpha_0} \rightarrow a_{\alpha_0 \alpha_1} \rightarrow \dots \rightarrow a_{\alpha_{k-1} \alpha_k}. \quad (8)$$

For this finite chain we have that

$$A_c^k = c_{\alpha_0} \prod_{s=1}^k a_{\alpha_{s-1} \alpha_s}, \quad (9)$$

where  $c \in \mathbb{R}^{n \times 1}$  and  $c(i) = b(i)$  if  $1 \leq i \leq n$  and  $c(n + 1) = 0$ . The variance of the random variable  $X_\alpha^k(\tau)$  is defined as [4]

$$X_\alpha^k(\tau) = \frac{c_{\alpha_0}}{p_{\alpha_0}} \frac{a_{\alpha_0 \alpha_1}}{p_{\alpha_0 \alpha_1}} \frac{a_{\alpha_1 \alpha_2}}{p_{\alpha_1 \alpha_2}} \dots \frac{a_{\alpha_{k-1} \alpha_k}}{p_{\alpha_{k-1} \alpha_k}} \frac{c_{\alpha_k}}{p_{\alpha_k}} = \frac{A_c^k c_{\alpha_k}}{P^k(\tau)}. \quad (10)$$

The variance of the random variable  $X_\alpha^k(\tau)$  is very important for the quality of the algorithm. Smaller variance  $Var\{X_\alpha^k(\tau)\}$  leads to better convergence of the stochastic algorithm. It is proven that [4]:

$$Var\{X_\alpha^k(\tau)\} = \frac{c_{\alpha_0}}{p_{\alpha_0} p_\alpha} (\bar{A}_c^k \hat{c})_\alpha - (A_c^k c)_\alpha^2. \quad (11)$$

## III. AN IMPROVED MONTE CARLO ALGORITHM FOR LA SYSTEMS

We use the Sequential Monte Carlo (SMC) method for linear systems introduced by John Halton [6]. We introduce the new improved Monte Carlo algorithm for the computation a linear functional form  $W(x)$  of the solution of a linear system with real coefficients. The matrices  $B$  and the right-hand side  $f$  are normalized to accelerate the convergence rate of the stochastic process. Special values of the relaxation parameter  $\gamma_i = b_{ii}, i = 1, \dots, n$ , have been chosen, compared to the constant  $\gamma \in (0, 1]$  in [4]. Numerical experiments show that it leads to balancing of the iteration matrix  $A$ .

The initial equation is picked uniformly at random among the  $n$  equations. After that for each state  $i$  we define the total score  $S(i)$  and the total number of visits  $V(i)$  that are modified as soon as state  $i$  is visited during a walk.

The following algorithm describes the above:

Now we give a description of the improved Monte Carlo algorithm for computing all the components of the solution. We compute scores for all the states (seen as new starting states) that are visited during a given trajectory. The initialization and preprocessing are the same as in the previous algorithm.

## IV. NUMERICAL EXAMPLES AND RESULTS

In order to check the accuracy of a computed solution  $\hat{x}$ , we compute the residual  $r := B\hat{x} - f$  and “weighted residual” [8]:

$$\rho := \frac{\|r\|}{\|B\| \|\hat{x}\|}. \quad (12)$$

The number of SMC iteration is  $N$  and the computational time  $t$  is measured in seconds. In the Tables and Figures below we present the values of the weighted residual. We perform a comparison with the refined iterative Monte Carlo (RIMC) [3] and the original “walk on equations” (WE) method which is completely described in [4]. For our method we use the

**Algorithm 1** Computing one component  $x_{i_0}$  of the solution  $x_i, i = 1, \dots, n$ .

**Require:** Initialization with initial data: the matrix  $B$ , the vector  $f$ , the constants  $\gamma_i = b_{ii}, i = 1, \dots, n$ , and the number of random trajectories  $M$ .

**Ensure:** Preliminary calculations (preprocessing): compute the matrix  $A$  using the parameter  $\gamma \in (0, 1]$ :

$$\{a_{ij}\}_{i,j=1}^n = \begin{cases} 1 - b_{ii} & \text{when } i = j \\ -b_{ij} & \text{when } i \neq j. \end{cases}$$

Set  $S := 0$ .

**for**  $k=1$  **to**  $M$  **do**

**set**  $m := i_0$

**set**  $S := S + f(m)$

$test = 0, sign = 1$

**update**  $S := S + sign * f_m$ ;

**end for**

**return**  $x_{i_0} = \frac{S}{M}$ .

**Algorithm 2** Computing all components  $x_i, i = 1, \dots, n$  of the solution.

**Require:** Initialization.

**Ensure:** Preprocessing.

Set  $S(i) := 0; V(i) := 0$ .

**for**  $k = 1$  **to**  $M$  **do**

**set**  $m := rand(1 : n)$

**set**  $test := 0; m_1 := 0$

$V(m) := V(m) + 1; m_1 = m_1 + 1; l(m_1) = m$

**for**  $q = 1$  **to**  $m_1$  **do**

$S(l(q)) := S(l(q)) + f_{l(q)}$

**end for**

**end for**

**return**

**for**  $j = 1$  **to**  $n$  **do**

$V(j) := \max\{1, V(j)\}$

$x_j = \frac{S(j)}{V(j)}$

**end for**

notation improved “walk on equations” algorithm (IWE). We have done numerical experiments with different matrices with dimensions  $n = 7, 100, 5000$ , where the number of equations in the linear system is  $n$  and  $B \in \mathbb{R}^{n \times n}$ . The number of random trajectories for lower dimensions  $n = 7$  is  $10n$ . The number of random trajectories for  $n = 100$  is  $5n$  and for  $n = 5000$  is  $n$ . A few sequential steps for the improved IWE algorithm combined with SMC are necessary.

In the examples below we try to find the solution  $x$  defined by the linear systems of algebraic equations  $Bx = f$ , where the matrix  $B$  and the vector  $b$  are preliminary given.

*Example 4.1:* In the first example we deal with two solutions  $x_1$  and  $x_2$  of  $Bx = f$ , where the matrix  $B$  and the vectors  $b_1$  and  $b_2$  are given below. The matrix is:

$$B = \begin{pmatrix} 5 & -1 & -1 & 0 & 0 & -1 & -1 \\ -1 & 5 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & 5 & -1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 5 & -1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 5 & -1 & -1 \\ -1 & 0 & 0 & -1 & -1 & 5 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 & 5 \end{pmatrix}. \quad (13)$$

The vectors  $f_1$  and  $f_2$  are:

$$f_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad f_2 = \begin{pmatrix} 4 \\ -2 \\ -1 \\ 0 \\ -1 \\ -2 \\ 4 \end{pmatrix}. \quad (14)$$

The solutions are

$$x_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (15)$$

*Example 4.2:* Let  $B$  is the matrix NOS4 from the Harwell-Boeing Collection [7], and  $b \in \mathbb{R}^{100}, b_i = 1, i = 1, \dots, 100$ . This particular matrix is taken from an application connected to finite element approximation of a problem describing a beam structure in constructive mechanics [7].

*Example 4.3:* Let  $B$  is a dense matrix  $5000 \times 5000$  with elements in  $[0,1]$ , and  $f \in \mathbb{R}^{5000}, f_i = 1, i = 1, \dots, 5000$ .

TABLE I  
WEIGHTED RESIDUAL FOR THE MATRIX  $B \in \mathbb{R}^{7 \times 7}$  AND  $x_1$

N	RIMC	t	WE	t	IWE	t
2	2.28e-15	0.02	4.15e-02	0.11	1.00e-01	0.007
5	7.89e-16	0.07	1.39e-02	0.23	2.29e-03	0.026
10	8.07e-16	0.21	3.18e-06	0.68	1.24e-06	0.04
15	7.45e-16	0.35	3.94e-08	1.11	2.55e-10	0.1
20	6.66e-16	0.69	1.71e-10	2.53	7.78e-14	0.23
30	5.79e-16	1.14	8.12e-15	3.69	8.32e-17	0.49

TABLE II  
WEIGHTED RESIDUAL FOR THE MATRIX  $B \in \mathbb{R}^{7 \times 7}$  AND  $x_2$

N	RIMC	t	WE	t	IWE	t
2	1.54e-01	0.003	4.63e-01	0.11	8.21e-02	0.003
5	1.01e-01	0.01	9.93e-03	0.23	2.48e-03	0.008
10	3.55e-02	0.04	1.43e-06	0.68	1.42e-06	0.05
15	4.04e-02	0.08	6.17e-09	1.11	2.66e-10	0.09
20	5.00e-02	0.14	1.40e-09	2.53	6.56e-14	0.16
30	4.72e-02	0.24	1.53e-14	3.69	5.03e-17	0.29

TABLE III  
WEIGHTED RESIDUAL FOR THE MATRIX  $NOS4 \in \mathbb{R}^{100 \times 100}$ .

N	RIMC	t,s	WE	t,s	IWE	t,s
2	7.253e-02	0.05	4.178e-01	0.84	3.028e-03	0.08
5	5.449e-02	0.22	4.148e-01	2.37	3.071e-05	0.24
10	4.319e-02	0.56	5.943e-03	5.31	7.461e-08	0.61
15	3.520e-02	0.78	2.419e-06	9.1	1.217e-10	0.89
20	3.197e-02	1.11	3.336e-09	13.5	1.022e-13	1.13
30	1.835e-02	2.15	3.660e-12	18.6	1.109e-16	1.92

TABLE IV  
WEIGHTED RESIDUAL FOR THE MATRIX  $B \in \mathbb{R}^{5000 \times 5000}$ .

N	RIMC	t	WE	t	IWE	t
2	5.438e-03	10.05	4.304e-02	3.95	2.931e-02	0.15
5	3.875e-03	60.2	1.217e-01	13.3	1.816e-04	0.9
10	2.866e-03	130.5	2.301e-05	32.3	1.235e-07	2.4
15	2.367e-03	310.7	6.486e-09	67.8	1.833e-10	5.1
20	1.941e-03	811	3.205e-09	171.5	1.054e-14	11.1
30	1.701e-03	2135	1.126e-07	418.6	2.481e-16	25.2

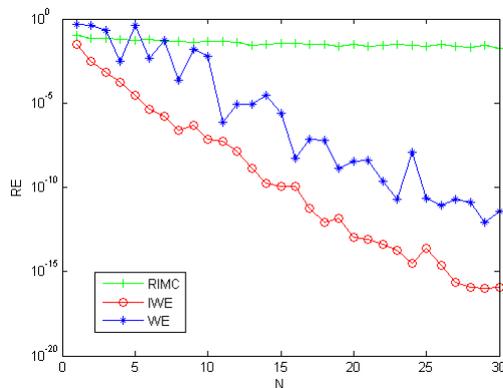


Fig. 1. Weighted residual for the matrix  $B \in \mathbb{R}^{100 \times 100}$ .

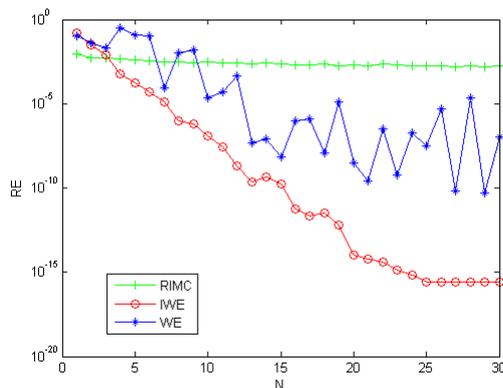


Fig. 2. Weighted residual for the matrix  $B \in \mathbb{R}^{5000 \times 5000}$ .

It can be seen that for the 7 dimensional case the difference in the accuracy between the WE and IWE for a given number of iterations is 2-3 order for  $N > 15$  – see Table I and Table II. It is worth mentioning that refined iterative Monte Carlo convergence is very slow except for the trivial solution  $x_1$  of Example 1. For the 100 dimensional case of the matrix NOS4 IWE produces much better results than WE and it is nearly 5-6 times faster – see Table III. The prior behavior of the proposed MC algorithm does not depend on the matrix density. The matrix NOS4 has only 5.9 average non-zeros per row and per column. The advantages of the algorithm hold for dense matrices. Also the difference in the accuracy for a fixed number of iterations is 3-5 order – see Figure 1. For larger dimensions the advantage of IWE over WE is even more pronounced. For the last example after 30 iterations WE has accuracy of  $10^{-9}$  while IWE gives accuracy of  $10^{-16}$  – see Figure 2. Also the time for the IWE is 15 times better than WE – see Table IV. The advantages of the proposed MC algorithm can be observed especially for larger matrix size. The special choice of relaxation parameters leads to the balancing of the iteration matrix and the experiments show that for larger dimensions the improvements leads to lower relative errors for small number of SMC iterations for IWE.

## V. CONCLUSION

A new improved Monte Carlo algorithm for solving linear algebra problems is presented and studied. It is used for evaluating all the components of the solution of real valued systems. Due to the optimization techniques it gives superior results to the standard “walk on equations” method and it is established as one of the fastest and accurate Monte Carlo algorithm for solving systems of LA equations.

## REFERENCES

- [1] Curtiss, J.H., *Monte Carlo methods for the iteration of linear operators*, J. Math Phys., Vol. 32(4), 209–232, 1954, DOI: 10.1002/japm1953321209.
- [2] Dimov, I., *Optimal Monte Carlo Algorithms*, *Proceedings IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing*, October 2006, Sofia, Bulgaria, IEEE, Los Alamitos, California, 125–131, 2006, DOI: 10.1109/JVA.2006.37.
- [3] Dimov, I., *Monte Carlo Methods for Applied Scientists*, New Jersey, London, Singapore, World Scientific, 291p, 2008.
- [4] Dimov, I.T., S. Maire, J.M. Sellier, *A New Walk on Equations Monte Carlo Method for Linear Algebraic Problems*, *Applied Mathematical Modelling*, Vol. 39(15), 4494–4510, 2015, DOI: 10.1016/j.apm.2014.12.018.
- [5] Golub, G.H., Van Loan C.F., *Matrix computations*, Third Edition, Johns Hopkins Univ. Press, Baltimore, 1996.
- [6] Halton, J., *Sequential Monte Carlo*, *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 58(1), 57–78, 1962, DOI: 10.1017/S0305004100036227.
- [7] NOS4: Lanczos with partial reorthogonalization. Finite element approximation to a beam structure. <http://math-nist.gov/MatrixMarket/data/Harwell-Boeing/lanpro/nos4.html>
- [8] Errors for Linear Systems: <http://www.math.umd.edu/petersd/466/linsystern.pdf>