Applied Mathematical Modelling 39 (2015) 4494–4510

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A new *Walk on Equations* Monte Carlo method for solving systems of linear algebraic equations ☆

Ivan Dimov [a,*], Sylvain Maire [b], Jean Michel Sellier [a]

[a] Department of Parallel Algorithms, IICT, Bulgarian Academy of Sciences, Acad. G. Bonchev 25 A, 1113 Sofia, Bulgaria
[b] Laboratoire LSIS, UMR 7296 Equipe Signal et Image, Université de Toulon, AV. Georges Pompidou, BP 56 83162, La Valette du Var Cedex, France

ARTICLE INFO

Article history:
Received 20 January 2014
Received in revised form 25 November 2014
Accepted 10 December 2014
Available online 3 January 2015

Keywords:
Monte Carlo algorithms
Markov chain
Linear systems

ABSTRACT

A new *Walk on Equations* (WE) Monte Carlo algorithm for solving systems of linear algebraic (LA) equations is proposed and studied. This algorithm relies on a non-discounted sum of an absorbed random walk. It can be applied for either real or complex matrices. Several techniques like simultaneous scoring or the sequential Monte Carlo method are applied to improve the basic algorithm. Numerical tests are performed on examples with matrices of different size and on systems coming from various applications. Comparisons with standard deterministic or Monte Carlo algorithms are also done.

1. Introduction

In this paper a new *Walk on Equations* (WE) Monte Carlo algorithm is proposed. The applicability and efficiency of the algorithm is analyzed. The algorithm is based on a Markov chain Monte Carlo (MC) approach. Our consideration is focused on the real symmetric matrices, but a wider class of matrices can also be treated.

Many 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 linear algebraic system. For many LA problems it is important to be able to find a preconditioner with relatively small computational complexity. That is why, it is important to have relatively *cheap* algorithm for matrix inversion. The computation time for very large problems, or for finding solutions in real-time, can be prohibitive and this prevents the use of many established algorithms. Monte Carlo algorithms give statistical estimates of the required solution, by performing random sampling of a random variable, whose mathematical expectation is the desired solution [1–7].

An important advantage of the Monte Carlo algorithms is that often a slight modification of the algorithm for solving systems of LA equations allows to solve other linear algebra problems (LAP) like matrix inversion and computing the extremal eigenvalues. In this work algorithms for solving systems of LA equations are presented. It is also shown what some modifications are needed to be able to apply these algorithms for a wider class of LAP. In [8] a MC method for solving a class of linear problems (including systems of LA equations) is presented. It is proved that the method is *closed to optimality* in sense that for a special choice of the right-hand side the requirements of the Kahn theorem are fulfilled and the method is optimal.

☆ This work was partially supported by the European Commission under FP7 project AComIn (FP7 REGPOT-2012-2013-1), by the Bulgarian NSF Grant DCVP 02/1, as well as by the Université du Sud Toulon-Var.

* Corresponding author.
E-mail addresses: ivdimov@bas.bg (I. Dimov), sylvain.maire@univ-tln.fr (S. Maire), jeanmichel.sellier@parallel.bas.bg (J.M. Sellier).
URLs: http://parallel.bas.bg/dpa/BG/dimov/index.html (I. Dimov), http://maire.univ-tln.fr/ (S. Maire), http://www.bas.bg (J.M. Sellier).

http://dx.doi.org/10.1016/j.apm.2014.12.018
0307-904X/

This method is called almost optimal method (MAO). The MAO method was exploit by C. Jeng and K. Tan in [9] to build on the top of MAO the relaxed Monte Carlo method for systems of LA problems. The idea of the approach used in [9] is to use a relaxation parameter γ like in deterministic Gauss–Seidel algorithm, or Jacobi overrelaxation algorithm. In [10] an unbiased estimation of the solution of the system of linear algebraic equations is presented. The proposed estimator can be used to find one component of the solution. Some results concerning the quality and the properties of this estimator are presented. Using this estimator the author gives error bounds and constructs confidence intervals for the components of the solution.

In [11] a Monte Carlo algorithm for matrix inversion is proposed and studied. The algorithm is based on the solution of simultaneous linear equations. In our further consideration we will use some results from [11,3] to show how the proposed algorithm can be used for approximation of the inverse of a matrix. Several authors have presented works on MC algorithms for LA problems and on the estimation of computational complexity [12–17]. There are MC algorithms for computing components of the solution vector; evaluating linear functionals of the solution; matrix inversion, and computing the extremal eigenvalues. An overview of all these algorithms is given in [3]. The well-known Power method [18] gives an estimate for the dominant eigenvalue λ_1. This estimate uses the so-called *Rayleigh quotient*. In [19] the authors consider bilinear forms of matrix powers, which is used to formulate a solution for the eigenvalue problem. A Monte Carlo approach for computing extremal eigenvalues of real symmetric matrices as a special case of Markov chain stochastic method for computing bilinear forms of matrix polynomials is considered. In [19] the robustness and applicability of the Almost Optimal Monte Carlo algorithm for solving a class of linear algebra problems based on bilinear form of matrix powers (v, A^k) is analyzed. It is shown how one has to choose the acceleration parameter q in case of using Resolvent Power MC [14,20,21]. This approach is a discrete analogue of the resolvent analytical continuation method used in the functional analysis [22]. There are cases where the polynomial becomes the resolvent matrix [3,16,20]. It should be mentioned that the use of acceleration parameter based on the resolvent representation is one way to decrease the computational complexity. Another way is to apply a variance reduction technique [23] in order to get the required approximation of the solution with a smaller number of operations. The variance reduction technique for particle transport eigenvalue calculations proposed in [23] uses Monte Carlo estimates of the forward and adjoint fluxes.

The rest of the paper is organized as follows. In Section 2, we formulate the problem of solving linear systems using Monte Carlo methods based on Markov chains with or without absorbing states. In Section 3, we describe the new *Walk on Equations* method for solving linear systems and analyze its variance. In Section 4, we give algorithms to use this method for both approximating a single component of the system and all components simultaneously. We also explain how to accelerate the convergence of the algorithms thanks to the sequential Monte Carlo approach. In Section 5, we make numerical comparisons between our method and standard deterministic methods for the approximation of all components of large linear systems.

2. Formulation of the problem: solving linear systems and matrix inversion

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:

$$A = \{a_{ij}\}_{i,j=1}^{n} = (a_1, \ldots, a_i, \ldots, a_n)^t,$$

where $a_i = (a_{i1}, \ldots, a_{in}),\ i = 1, \ldots, n$ and the symbol t means *transposition*.

The following norms of vectors (l_1-norm):

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

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.

Consider a matrix $A \in \mathbb{R}^{n\times n}$ and a vector $b = (b_1, \ldots, b_n)^t \in \mathbb{R}^{n\times 1}$.The matrix A can be considered as a linear operator $A[\mathbb{R}^{n\times 1} \to \mathbb{R}^{n\times 1}]$ [3,22], so that the linear transformation

$$Ab \in \mathbb{R}^{n\times 1}, \tag{1}$$

defines a new vector in $\mathbb{R}^{n\times 1}$.

The linear transformation (1) is called *iteration*. The stochastic representation we are using actually mimics the iterative algorithms using the transformation (1). Actually, we build a random variable on a specially defined Markov chain with an expectation equal to the functional of the truncated Neumann series (biased Monte Carlo algorithms), or infinite Neumann series (unbiased Monte Carlo algorithms). As it will be shown later on, a jump from one stage to another stage of the Markov chain corresponds to one classical iteration. The algebraic transformation (1) plays a fundamental role for both unbiased and biased iterative Monte Carlo algorithms [3].