PROCEEDINGS OF SCIENCE

Ensemble Quasi-Newton HMC

Xiao-Yong Jin* **and James C. Osborn**
Computational Science Division
Argonne National Laboratory
9700 S. Cass Ave.
Lemont, IL 60439, USA
E-mail: xjin@anl.gov, osborn@alcf.anl.gov

We present a modification of the Hybrid Monte Carlo algorithm for tackling the critical slowing down of generating Markov chains of lattice gauge configurations towards the continuum limit. We propose a new method to exchange information within an ensemble of Markov chains, and use it to construct an approximate inverse Hessian matrix of the action inspired from quasi-Newton algorithms for optimization. The kinetic term of the molecular dynamics evolution includes the approximate Hessian for long distance couplings among the momenta. We show the result of applying the new algorithm to the $U(1)$ gauge theory in two dimensions, and discuss our future plans.

The 36th Annual International Symposium on Lattice Field Theory - LATTICE2018
22-28 July, 2018
Michigan State University, East Lansing, Michigan, USA.

*Speaker.

PoS(LATTICE2018)027

1. Introduction

In generating a Markov chain, we aim at speeding up Monte Carlo simulations, making proposal configurations far from the current configuration in phase space, with relative low cost. Molecular Dynamics (MD) evolution in fictitious time using random momenta naturally extends and mitigates the Langevin-like random walk behavior. This Hybrid Monte Carlo (HMC) algorithm [1] works well in high dimensional systems, such as lattice QCD. Approaching the continuum limit of the lattice theory, some physical modes in MD slows down exponentially, leading to research in Fourier acceleration [2 – 4] as a possible remedy. The analogous Riemannian manifold HMC [5] claims success for some probability density functions in guiding the MD evolution through the phase space. Recent efforts [6 – 8] surge in analyzing and applying similar techniques to lattice QCD. We focus on employing, as the acceleration kernel, a numerically cheaper approximation of the Hessian matrix from the Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm [9, 10] (L-BFGS), a common quasi-Newton optimization method. This approximation applies to not only the gauge fields but also the pseudo-fermion fields. The HMC, nevertheless, requires changes to adopt such approximation that uses information from multiple configurations.

In this paper we present the ensemble quasi-Newton HMC (QNHMC) method, discuss the characteristics of the method on two-dimensional $U(1)$ lattice gauge theory, and show preliminary results of its effect on the autocorrelation of the average plaquette value and topological charge.

2. Markov chain for assisted MD evolution

The MD evolution in the heart of the HMC algorithm follows from Hamiltonian dynamics,

$$H(x,p) = S(x) + \frac{1}{2} p^{\dagger} G^{-1} p, \quad \dot{x} = G^{-1} p, \; \dot{p} = -\nabla S, \tag{2.1}$$

where S is the action, p the fictitious momenta, and G a fixed MD mass matrix. A symplectic and reversible discrete integrator advances the state of the Markov chain from (x, p) to (x', p') over a fictitious time period, τ, the trajectory length. Using the Hamiltonian as the negative log probability of the enlarged phase space including x and p, the correctness of the HMC demands a positive definite G.

The choice of G affects the performance of HMC. For a general action, a MD mass matrix containing the local information of the Riemann curvature can bring considerable speedups [5] in the efficiency of Markov Chain Monte Carlo methods. The article recommends the Fisher information matrix as G. Fourier acceleration [3, 4] suggests the field Laplacian operator as G. We are interested in using a fixed G during one MD trajectory, for its simplicity and efficiency. Any explicit symplectic reversible integrator for equation (2.1) would still be applicable. However, this also means that G cannot depend on any configurations from this one whole MD trajectory.

In general we want a proposal of (x', p') for the next state of the Markov chain from (x, p) following a symplectic and reversible discretization of the MD evolution (2.1) where G^{-1} comes from our choice of a function, $\mathscr{G}^{-1}$,

$$G^{-1} = \mathscr{G}^{-1}\big(\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}\big). \tag{2.2}$$

whose argument is a list of $\mathscr{N}$ configurations $\mathscr{X}_i$, which are all different from x, x', or any configuration along the discretized path of this particular MD evolution. Assuming a particular choice of G^{-1} could help through out the Markov chain generation, we can fix $\mathscr{G}^{-1}$ and $\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}$, optimally picked from available configurations, and the HMC procedure remains the same except for the additional mass matrix G.

In this paper, we focus on building G^{-1} that is fixed during one MD evolution but changes after each trajectory. We use $\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}$ from a set of parallel streams of Markov chains. We update one of the streams using information from the others, suggested in references [11, 12]. We can have multiple ways to generate such Markov chains, and obtain $\mathscr{G}$ of $\mathscr{X}$ from neighboring streams. The following is the base case with provable reversibility. We use an arbitrary information exchange kernel $\mathscr{F}$ to generalize the ensemble assisted Markov chain.

Let $\mathbb{N}$ be the number of coupled parallel streams, each labeled $\mathbb{X}_j$, for $j = 0$ to $\mathbb{N}-1$. Let $\mathscr{F}$ be a function on a *unordered* set of configurations, $\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}$, with $\mathscr{N} = \mathbb{N}-1$. Let $\mathscr{U}$ be a symplectic and reversible mapping that generates the next state of one Markov chain, from $\mathbb{X}_j$ to $\mathbb{X}'_j = \mathscr{U}\left(\mathscr{F}\left(\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}\right)\right)\mathbb{X}_j$. Given a fixed set of $\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}$, $\mathscr{U}$ depends on the value of $\mathscr{F}$, and satisfies the detailed balance, $\pi(\mathbb{X}_j)P(\mathbb{X}_j|\mathbb{X}'_j) = \pi(\mathbb{X}'_j)P(\mathbb{X}'_j|\mathbb{X}_j)$, where π is probability density we want to simulate and P the transition probability. We give the definition of $\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}$ within the steps of the ensemble assisted Markov chain described in the following.

1. When updating $\mathbb{X}_k$ for each k from 0 to $\mathbb{N}-1$:
   (a) Setting the list, $\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}$ from the list $\{\mathbb{X}_j\}_{j\neq k}$, with $\mathscr{N} = \mathbb{N}-1$.
   (b) Evolve $\mathbb{X}_k$ according to $\mathscr{U}\left(\mathscr{F}\left(\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}\right)\right)$.
2. After generating one trajectory for each $\mathbb{N}$ parallel streams, Set the new $\{\mathbb{X}'_j\}$ to the reversed sequence of it, $\mathbb{X}'_j \leftarrow \mathbb{X}'_{\mathbb{N}-1-j}$. This is for the purpose of reversibility.

Figure 1 illustrates an example of one update with $\mathbb{N} = 4$ streams. The current configurations from these four streams are labeled with 0, 1, 2, and 3. We update those successively, each time with information obtained from the function $\mathscr{F}$ applied to configurations from other streams. After updating each stream to $0'$, $1'$, $2'$, and $3'$, we reverse the ordering of those, and complete this one update. We can see that when we reverse the Markov chain, we would update $3'$ first with the information from $\mathscr{F}(\{2',1',0'\})$, because $\mathscr{F}$ does not depend on the ordering of $\mathscr{X}_i$, we can reproduce the reversed Markov chain.

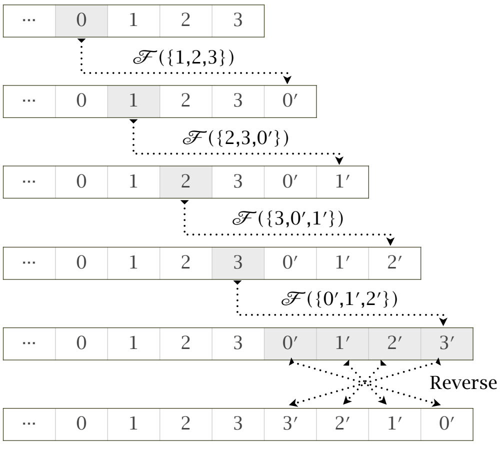

Figure 1: One update for $\mathbb{N} = 4$.

For the purpose of assisting MD evolution, where $\mathscr{U}$ represents the procedure of refreshing p, integrating the equation (2.1), and finishing with a Metropolis-Hastings accept/reject step, we use

$$G^{-1} = \mathscr{F}\left(\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}\right) = \mathscr{G}^{-1}\left(\text{sorted}\left(\{\mathscr{X}_i\}_{i=0}^{\mathscr{N}-1}\right)\right), \tag{2.3}$$

where a special routine sorted sort the set of $\mathscr{X}_i$ before applying the function $\mathscr{G}^{-1}$, because our choice of $\mathscr{G}$ in equation (2.2) depends on the ordering of $\mathscr{X}_i$, and sorting guarantees the same

G^{-1} with a reversed procedure. In addition to the simple case presented here, we can build more involved Markov chains by using more states per parallel stream for the exchange kernel $\mathscr{F}$, or by decoupling some of the parallel streams for parallel evolution.

3. L-BFGS approximated Hessian

Among $\mathbb{N}$ coupled parallel HMC streams, for each stream, we use the latest configurations from the other streams to construct the approximate Hessian with the L-BFGS algorithm. For these $\mathbb{N}-1$ configurations, we compute the site-wise finite differences of lattice gauge fields, $\mathscr{U}_k$,

$$s_k = \log \mathscr{U}_k \mathscr{U}_{k-1}^{-1} = \log \mathscr{U}_k \mathscr{U}_{k-1}^{\dagger}, \quad y_k = \nabla S(\mathscr{U}_k) - \nabla S(\mathscr{U}_{k-1}), \quad k = 1\ldots L \le \mathbb{N}-2, \tag{3.1}$$

where s_k and y_k are fields of the elements of the Lie algebra, L is the length of the L-BFGS memory, and the inequality comes from selecting only those field pairs with $s_k^{\dagger} y_k = y_k^{\dagger} s_k > 0$ (the inner product implicitly traces over the color indices and sums over lattice) for the positive definiteness of the approximated Hessian. As required in equation (2.3) we sort the field pairs according to $s_k^{\dagger} y_k$.

The L-BFGS algorithm gives the inverse Hessian as a recursively defined operator,

$$\mathscr{G}_k^{-1} = (I - \rho_k s_k y_k^{\dagger}) \mathscr{G}_{k-1}^{-1} (I - \rho_k y_k s_k^{\dagger}) + \rho_k s_k s_k^{\dagger}, \quad \mathscr{G}_0^{-1} = 1/(2\beta), \tag{3.2}$$

where 2β as an initial value comes from the diagonal term of the Hessian matrix of the 2-D $U(1)$ action in the weak coupling limit. The associated L-BFGS Hessian matrix can be expressed as,

$$\mathscr{G}_k = \mathscr{G}_{k-1} + \frac{y_k y_k^{\dagger}}{y_k^{\dagger} s_k} - \frac{\mathscr{G}_{k-1} s_k s_k^{\dagger} \mathscr{G}_{k-1}}{s_k^{\dagger} \mathscr{G}_{k-1} s_k}. \tag{3.3}$$

This rank-2 update has a symmetric product form [13], showing the explicit positive definiteness,

$$\begin{aligned}
\mathscr{G}_k &= A_k A_k^{\dagger}, & \mathscr{G}_0 &= A_0 A_0^{\dagger}, & A_k &= (I - \gamma_k v_k s_k^{\dagger}) A_{k-1}, & \gamma_k &= \rho_k \beta_k, \\
\mathscr{G}_k^{-1} &= B_k B_k^{\dagger}, & \mathscr{G}_0^{-1} &= B_0 B_0^{\dagger}, & B_k &= (I - \rho_k s_k v_k^{\dagger}) B_{k-1}, & \rho_k &= 1/(s_k^{\dagger} y_k), \\
\alpha_k &= 1/(s_k^{\dagger} \mathscr{G}_{k-1} s_k), & \beta_k &= \pm\sqrt{\alpha_k/\rho_k}, & v_k &= y_k + \beta_k \mathscr{G}_{k-1} s_k.
\end{aligned} \tag{3.4}$$

In our test with the unmodified L-BFGS algorithm, the low eigenvalues of the approximated Hessian matrix decreases rapidly as the L-BFGS memory length increases, even after removing all the exact zero modes from the theory described in the next section. While the largest eigenvalues are stable, the condition number increases and the Hessian matrix becomes singular with a modest L-BFGS memory length, because the approximate action surface spanned by the samples we draw for the L-BFGS algorithm has zero modes and may even be concave. Since the MD evolution involves the inverse of the Hessian matrix, the evolution becomes unstable with near zero modes. A straightforward method to regulate the approximated Hessian would be to add a small term to the diagonal of $\mathscr{G}_k$. It nevertheless breaks the rank-2 update iteration, invalidates the simple inversion formula and the symmetric decomposition for the square root of $\mathscr{G}$. This would require a conjugate gradient inversion and a rational approximation of the square root of $\mathscr{G}$.

Investigating the determinant behavior from the symmetric product form (3.4) leads us to one solution: adding a small term to one of the rank-1 updates in equation (3.3),

$$\mathscr{G}_k = \mathscr{G}_{k-1} + \frac{y_k y_k^{\dagger}}{y_k^{\dagger} s_k} - \left(1 - \lambda \frac{s_k^{\dagger} s_k}{s_k^{\dagger} \mathscr{G}_{k-1} s_k}\right) \frac{\mathscr{G}_{k-1} s_k s_k^{\dagger} \mathscr{G}_{k-1}}{s_k^{\dagger} \mathscr{G}_{k-1} s_k}. \tag{3.5}$$

This still invalidates the iteration formula (3.2). The symmetric product form (3.4), however, remains applicable with minimal changes. The complexity of iterating the symmetric product form is always linear in lattice volume and, in terms of L, $O(L)$ in space and $O(L^2)$ in time.

4. $U(1)$ gauge theory on a 2-D lattice

We use the Wilson plaquette action for the $U(1)$ gauge theory on a two-dimensional lattice with periodic boundary conditions. As the QNHMC algorithm uses approximated Hessian, we first need to remove all the exact zero modes of the Hessian from the theory, in order to improve the stability of the MD evolution.

The gauge degrees of freedom form the exact zero modes of the Hessian. With periodic boundary conditions, that is $N_s \times N_t - 1$ zero modes for a lattice with spatial and temporal extent N_s and N_t. We fix the gauge using a maximal tree of links which we set the gauge variables $U_{x,\mu}$ to unity. The maximal tree includes lattice sites $x = (x_0, x_1)$ and directions $\mu \in \{\hat{0}, \hat{1}\}$ satisfying

$$\begin{cases} 0 \le x_0 < N_t - 1 & \text{with } \mu = \hat{0} \text{ for temporal links,} \\ x_0 = 0;\ 0 \le x_1 < N_s - 1 & \text{with } \mu = \hat{1} \text{ for spatial links.} \end{cases} \tag{4.1}$$

There are two global gauge degrees of freedom, due to the abelian nature of the theory,

$$\begin{cases} U_{x,\hat{0}} \to U_{x,\hat{0}}\Lambda_0 & \text{for } x_0 = N_t - 1, \\ U_{x,\hat{1}} \to U_{x,\hat{1}}\Lambda_1 & \text{for } x_1 = N_s - 1, \end{cases} \tag{4.2}$$

where Λ_0 and Λ_1 are elements of the $U(1)$ group. Thus we fix two more gauge links, $U_{(N_t-1,0),\hat{0}} = U_{(0,N_s-1),\hat{1}} = 1$, during the MD evolution of QNHMC to remove these two zero modes.

We are interested in observables that are slow to evolve in the Markov chain, particularly the topological charge. We use the definition of the topological charge [14, 15], $Q = (\sum_x \operatorname{Arg} P_x)/(2\pi)$, where the complex argument Arg takes the principle value of $(-\pi, \pi)$. This definition does not apply to exceptional configurations (with no contribution to the partition function in the continuum limit) where $P_x = -1$ for some x. On a two dimensional lattice with periodic boundary conditions, this definition of topological charge gives exact integer values.

5. Current status, and future plans

We implement the $U(1)$ gauge theory in the QEX framework [16]. We use PRIMME [17] to study the eigenmodes of the exact Hessian matrix and the L-BFGS approximated one.

The results below come from the 2-D $U(1)$ theory at $\beta = 4.5$ on a lattice of size 24×24, with the number of coupled parallel HMC streams, $\mathbb{N} = 10$ and 20, using the number of configurations from 8192 to 65536, depending on the trajectory length. The MD evolution uses the Omelyan's second order minimum norm integrator [18]. We keep the number of steps per MD trajectory fixed at 8, 16, 32, or 64, while tuning for optimal trajectory length separately for conventional HMC and QNHMC.

Figure 2 shows the integrated autocorrelation length of topologic charge squared (left) and the average plaquette (right). To include the cost of generating the Markov chain, we multiply the integrated autocorrelation length by the number of MD steps in an HMC trajectory, converting the

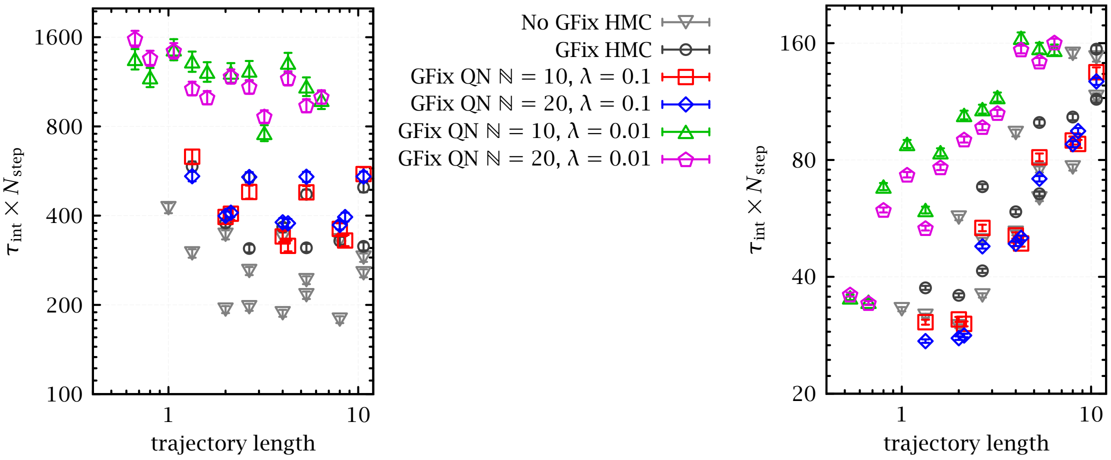

Figure 2: Integrated autocorrelation length of the topologic charge squared (left) and the average plaquette (right) in units of discrete MD steps. GFix denotes gauge fixing during MD. QN stands for QNHMC.

correlation length from the unit of configuration to the unit of MD steps. We refer to this quantity as the *cost* of generating configurations for uncorrelated measurable quantities, in terms of the force evaluations, and we tune simulation parameters to lower the cost. The apparent increase of the cost for the average plaquette after the trajectory length grows longer than three is due to the fact that the autocorrelation becomes minimal between successive configurations and the cost here becomes linearly proportional to the MD steps.

Comparing the conventional HMC with and without gauge fixing, we see that the topological quantity shows about a factor of two increased cost, going from no gauge fixing to gauge fixing. The autocorrelation of the average plaquette value however depends less on the gauge fixing. Using QNHMC with $\lambda = 0.1$ shows no improvement for the topological quantity, and the cost worsens with $\lambda = 0.01$. On the other hand, QNHMC reduces the cost for the average plaquette with $\lambda = 0.1$, and more so with increased number of coupled Markov chains, from $\mathbb{N} = 10$ to 20.

Moving forward, we will do more tuning and testing with the QNHMC algorithm, studying the scaling behavior toward the continuum limit. On the other hand, we will also look for other approaches to approximate the Hessian. L-BFGS is designed for its efficiency in iterative optimizations. Since we have an ensemble of Markov chains, we will look for other ways to approximate the Hessian matrix [19, 20].

Acknowledgments

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. We gratefully acknowledge the computing resources provided and operated by the Joint Laboratory for System Evaluation (JLSE) at Argonne National Laboratory.

References

[1] S. Duane, A. Kennedy, B. Pendleton and D. Roweth, *Hybrid Monte Carlo*, *Phys.Lett.* **B195** (1987) 216.

[2] G. G. Batrouni, G. R. Katz, A. S. Kronfeld, G. P. Lepage, B. Svetitsky and K. G. Wilson, *Langevin Simulations of Lattice Field Theories*, *Phys. Rev.* **D32** (1985) 2736.

[3] S. Duane, R. Kenway, B. J. Pendleton and D. Roweth, *Acceleration of Gauge Field Dynamics*, *Phys. Lett.* **B176** (1986) 143.

[4] S. Duane and B. J. Pendleton, *Gauge Invariant Fourier Acceleration*, *Phys. Lett.* **B206** (1988) 101.

[5] M. Girolami and B. Calderhead, *Riemann manifold langevin and hamiltonian monte carlo methods*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73** (2011) 123.

[6] G. Cossu, P. Boyle, N. Christ, C. Jung, A. Jüttner and F. Sanfilippo, *Testing algorithms for critical slowing down*, *EPJ Web Conf.* **175** (2018) 02008 [1710.07036].

[7] N. H. Christ and E. W. Wickenden, *Fourier acceleration, the HMC algorithm and renormalizability*, *PoS* **LATTICE2018** (2018) 025 [1812.05281].

[8] Y. Zhao, *Numerical Implementation of Gauge-Fixed Fourier Acceleration*, *PoS* **LATTICE2018** (2018) 026 [1812.05790].

[9] R. Fletcher, *A new approach to variable metric algorithms*, *The Computer Journal* **13** (1970) 317.

[10] J. Nocedal, *Updating Quasi-Newton Matrices with Limited Storage*, *Mathematics of Computation* **35** (1980) 773 [https://doi.org/10.1090/S0025-5718-1980-0572855-7].

[11] Y. Zhang and C. A. Sutton, *Quasi-newton methods for markov chain monte carlo*, in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira and K. Q. Weinberger, eds., pp. 2393–2401, Curran Associates, Inc., (2011), http://papers.nips.cc/paper/4464-quasi-newton-methods-for-markov-chain-monte-carlo.pdf.

[12] C. Matthews, J. Weare and B. Leimkuhler, *Ensemble preconditioning for markov chain monte carlo simulation*, 1607.03954.

[13] K. W. Brodlie, A. R. Gourlay and J. Greenstadt, *Rank-one and rank-two corrections to positive definite matrices expressed in product form*, *IMA Journal of Applied Mathematics* **11** (1973) 73 [http://imamat.oxfordjournals.org/content/11/1/73.full.pdf+html].

[14] R. Flume and D. Wyler, *Topological Charge in the Lattice Schwinger Model*, *Phys. Lett.* **108B** (1982) 317.

[15] C. Panagiotakopoulos, *Topology of Two-dimensional Lattice Gauge Fields*, *Nucl. Phys.* **B251** (1985) 61.

[16] J. Osborn and X.-Y. Jin, *Introduction to the Quantum EXpressions (QEX) framework*, *PoS* **LATTICE2016** (2017) 271.

[17] A. Stathopoulos and J. R. McCombs, *PRIMME: PReconditioned Iterative MultiMethod Eigensolver: Methods and software description*, *ACM Transactions on Mathematical Software* **37** (2010) 21:1.

[18] I. Omelyan, I. Mryglod and R. Folk, *Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations*, *Computer Physics Communications* **151** (2003) 272 .

[19] J. Martens, I. Sutskever and K. Swersky, *Estimating the Hessian by Back-propagating Curvature*, *ArXiv e-prints* (2012) [1206.6464].

[20] M. Mathieu and Y. LeCun, *Fast approximation of rotations and hessians matrices*, *CoRR* **abs/1404.7195** (2014) [1404.7195].