Chaotic behaviour of the lattice Yang-Mills on CUDA

Richard Forster
forceuse@inf.elte.hu
Eotvos Lorand University

Agnes Fulop
fulop@caesar.elte.hu
Eotvos Lorand University

We present an algorithm, which enables to study the chaotic behavior as complex Lyapunov spectrum of SU(2) Yang-Mills fields and the entropy-energy relation utilizing the Kolmogorov-Sinai entropy. It was determined by this numerical algorithm to apply CUDA to calculate the eigenvalues of the monodromy matrix, which is an fxf sparse matrix (f=24N).
We are using a hybrid block Hessenberg reduction system to compute the required eigenvalues, that makes us capable to achieve 2-3 times higher performance than the CPU only version.

Homogeneous Yang-Mills Fields

The non-Abelian gauge field theory was introduced to understand the strong interaction of elementary particles. The homogenous Yang-Mills equation contains the quadratic part of gauge field tensor.

The $F^a_{\mu\nu}$ form is a component of an antisymmetric gauge field tensor in Minkowski space:

$$F^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g f^{abc} A^b_\mu A^c_\nu$$

where $\mu,\nu = 0,1,2,3$ are space-time coordinates, the symmetry generators are labeled by $a,b,c = 1,2,3$ and g is the bare gauge coupling constant and f^{abc} are the structure constants of the continuous Lie group. The generators of this group fulfill the following relationship $[T^b, T^c] = if^{bcd}T^d$. The equation of motion can be expressed by covariant derivative in the adjoint representation:

$$\partial^\mu F^a_{\mu\nu} + g f^{abc} A^{b\mu} F^c_{\mu\nu} = 0$$

Lattice Yang-Mills Fields Theory

The real time differential equations are solved for basic variables, while the total energy is constrained value. In the 3 dimensional regular lattice with spacing a the basic variables are group elements, which related to the vector potential: $U_{x,i} = \exp(aA^a_i(x)T^c)$, where T^c is a group generator. For SU(2) these are given by the Pauli matrices $\tau_c T^c = -\left(\frac{ig}{2}\right)\tau^c$. The indices x,i denotes the link of the lattice starting at the 3 dimensional position x and pointing into the i-th direction.

SU(2) Hamiltonian

$$H = \sum_{x,i}\left(\frac{1}{2}\langle \dot{U}_{x,i}, \dot{U}_{x,i}\rangle + \left(1 - \frac{1}{4}\langle U_{x,i}, V_{x,i}\rangle\right)\right)$$

The $V_{x,i}$ is a complement link variable constructed of products of $U_{x,i}$'s along all link triples which close with given link an elementary plaquette. The canonical variable: $P_{x,i} = \dot{U}_{x,i}$. The non-Abelian gauge field is connected to the plaquette product: $U_{x,ij} = U_{x,j}U_{x+j,i}U^\dagger_{x+i,j}U^\dagger_{x,i}$

We will denote the single link $U_{x,i}$ with U.

Quaternion representation (for one link)

$$U = u_0 + i\tau^a u^a \qquad U = \begin{pmatrix} u_0 + iu_3 & iu_1 + u_2 \\ iu_1 - u_2 & u_0 - iu_3 \end{pmatrix}$$

The Hamiltonian equation of motion is solved with dt discrete time steps. This algorithm satisfies the Gauss law and the constraint of total energy.

Lattice EOMs

We will denote single link $U_{x,i,j}$ in time t with U_t.

$$U_{t+1} - U_{t-1} = 2\Delta t(P_t - \epsilon U_t)$$
$$P_{t+1} - P_{t-1} = 2\Delta t(V(U_t) - \mu U_t + \epsilon P_t)$$
$$\epsilon = \frac{\langle U_t, P_t\rangle}{\langle U_t, U_t\rangle}$$

The ϵ, μ means the Lagrange multipliers and the symmetry SU(N) is fulfilled by the next expressions: $\langle U_t, U_t\rangle = 1$ (unitarity) and $\langle U_t, P_t\rangle = 0$ (orthogonality).

Relation between Wilson action and lattice Hamiltonian

The Wilson action should be sum over all elementary squares of lattice $S = \Sigma_{P_{x,i,j}} S_{P_{x,i,j}}$. The action function of the gauge theory on lattice is written over plaquette sum to use the nearest neighbor pairs. The coupling on space-like and time-like plaquettes are no longer equal in the action:

$$S = \frac{2}{g^2}\sum_{P_t}(N - tr(U_{P_t})) - \frac{2}{g^2}\sum_{P_s}(N - tr(U_{P_s}))$$

The time like plaquette is denoted by U_{P_t} and space like U_{P_s}.

Consider the path is a closed contour i. e. Wilson loop, it is invariant under gauge changes and independent of the starting point. The product of such group elements along the closed line is a gauge covariant quantity, the trace over such products are invariant. Because the U_{P_t} can be series expansion by a_t:

$$U_{P_t} = U(t)U^\dagger(t + a_t) = UU^\dagger + a_t\dot{U}U^\dagger + \frac{a_t^2}{2}\ddot{U}U^\dagger + \cdots$$
$$N - trU_{P_t} = -\frac{a_t^2}{2}tr(\dot{U}\dot{U}^\dagger);\ \text{up to } O(a_t^3)\text{correction, where } UU^\dagger = \mathbf{1}.$$

Therefore the homogenous non-Abelian gauge action:

$$\Delta S_H = \frac{2}{g^2}\left(\frac{a_t^2}{2}\sum_{x,i,j} tr(\dot{U}_{x,i,j}\dot{U}^\dagger_{x,i,j}) - \sum_{x,i,j}\left(N - tr(U_{x,i,j})\right)\right)$$

The Scaled Hamiltonian was derived in the next form:

$$a_t H = \frac{2}{g^2}\left(\frac{a_t^2}{2}\sum_{x,i,j} tr(\dot{U}_{x,i,j}\dot{U}^\dagger_{x,i,j}) + \sum_{x,i,j}\left(N - tr(U_{x,i,j})\right)\right)$$

Implicit-Explicit-Endpoint Algorithm

$$P' = P_{t+1} \quad P = P_t$$
$$P' = P + (V - \mu V + \epsilon P')$$
$$U' = U + (P' - \epsilon U)$$

Determine: μ, ϵ:

$$(1 - \epsilon)P' = P + (V - \mu U)$$
$$U' = (1 - \epsilon)U + P' \qquad c = 1 - \epsilon$$

i. Obtain from orthogonality:

$$c\langle U', P'\rangle = \langle cU, P\rangle + \langle P', P\rangle + c\langle U, V - \mu U\rangle + \langle P', V - \mu U\rangle$$
$$0 = 0 + c(\langle U, V\rangle - \mu) + c\langle P', P'\rangle$$
$$\mu = \langle U, V\rangle + \langle P', P'\rangle$$

ii. Obtain c from unitarity:

$$\langle U', U'\rangle = c\langle U', U\rangle + \langle U', P'\rangle$$
$$1 = c\langle U', U\rangle = c(\langle cU, U\rangle + \langle P', U\rangle)$$
$$1 = c^2 + c\langle P', U\rangle$$
$$c\langle U, P'\rangle = \langle U, P\rangle + \langle U, V - \mu V\rangle = \langle U, V\rangle - \mu = -\langle P', P'\rangle$$
$$1 = c^2 - \langle P', P'\rangle \Rightarrow c = \sqrt{1 + \langle P', P'\rangle}$$
$$(c > 1, \epsilon < 0)$$

The nonlinearity of the Yang-Mills fields is described by the chaotic theory. Instead of the classical rescaling solution we apply the monodromy matrix method, which can describe the gauge field evolution, in this case the short– and long time behaviour. The full complex Lyapunov spectrum of the SU(2) Yang-Mills fields can be determined on a three dimensional lattice from classical dynamics using eigenvalues of the monodromy matrix. The question of ergodization is addressed via the Kolmogorov-Sinai entropy.

Chaos

The Lyapunov spectrum L_i is expressed in terms of the monodromy matrix's eigenvalues Λ_i:

$$L_i = \lim_{t\to\infty}\frac{\int_0^t \Lambda_i(t)dt}{T}, \qquad i = 1..f$$

where the $\Lambda_i(t)$'s are the solutions of the characteristic equation:

$$\det[\Lambda_i(t)I - M(t)] = 0$$

at a given time t. Here M is the linear stability matrix, and f is the number of degrees of freedom.

Conservative dynamics are fulfilling Liouville's theorem: $\sum_{i=0}^{f} L_i = 0$

The discrete definition of the Lyapunov spectrum:

$$\hat{L}_i = \langle\Lambda_i\rangle(n) = \frac{1}{n}\sum_{j=1}^{n}\Lambda_i(t_{j-1}), \qquad i = 1..f$$

where t_j's are subsequent times along an evolutionary path of the gauge field configurations.

The Kolmogorov-Sinai entropy by using Perin's formula: $h^{KS} = \sum L_i\Theta(L_i)$

$\Theta(x)$ being 1 for positive arguments and 0 otherwise.

The dimension of h^{KS} is a rate (1/time) estimating the entropy: $S = \frac{h^{KS}}{\mathrm{Re}(L_0)N^3}$

The monodromy matrix: $M = \begin{pmatrix} \frac{\delta\dot{U}}{\delta U} & \frac{\delta\dot{U}}{\delta P} \\ \frac{\delta\dot{P}}{\delta U} & \frac{\delta\dot{P}}{\delta P} \end{pmatrix}$

The elements are:

$$\frac{\delta\dot{U}^a}{\delta U^b} = 0 \qquad \frac{\delta\dot{U}^a}{\delta P^b} = \delta^{ab} \qquad \frac{\delta\dot{P}^a}{\delta P^b} = -2P^bU^a$$
$$\frac{\delta\dot{P}^a}{\delta U^b} = \frac{\delta V^a}{\delta U^b} - \left(U_c\frac{\delta V^c}{\delta U^b}\right)U^a - V^bU^a - (U_cV^c + P_cP^c)\delta^{ab}$$

They are providing information about the stability of trajectories in the neighborhood of any point of an orbit in the (U,P) phase space. A small perturbation $(\delta U, \delta P)$ evolves in time governed by the monodromy matrix M. The eigenvalues of this matrix can be classified as follows: for real and positive eigenvalues, neighboring trajectories part exponentially and the motion is unstable. In the limit of large time we obtain the Lyapunov components from these eigenvalues. The imaginary parts of the complex eigenvalues describe oscillatory frequencies of perturbations.

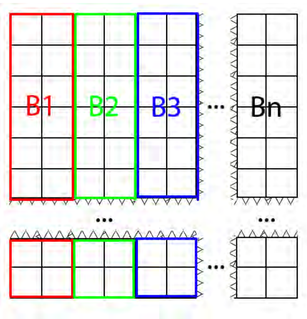
Hessenberg reduction

The most computing intensive part of the eigenvalue problem is to transform our initial matrix into its Hessenberg form. To achieve this the Block Hessenberg Algorithm is used. In this instead of taking the whole matrix as the input of the transformation process we devide it into smaller blocks. We take these blocks and calculate the Householder vector for each column in that block and with it update the consecutive columns. When finished we use the accumulated Householder transformations to update the rest of the whole matrix. We repeat this until we update all the blocks.

This way with the accumulated Householder transformations in overall less matrix multiplications will be used compared to the original Hessenberg Algorithm in which case we always have to update every column with the calculated Householder vector.

Householder vector, where vector x is the kth column of block Bk with the first k entries set to 0

$$v_k = \left(\frac{1}{A_{k+1,k}+\sigma}\right)\left(x + \sigma e_{k+1}\right)$$
$$\sigma = sign(A_{k+1,k})\|x\|_2$$
$$\tau_k = 2/\|v_k\|_2^2$$
$$V_k = [v_1, v_2, \ldots, v_k]$$



Compact-WY representation of the k Householder transformations

$$(I - v_1v_1^T)\ldots(I - v_kv_k^T) = I - V_kT_kV_k^T$$
$$A := A(I - V_LT_LV_L^T) = A - Y_LV_L^T$$
$$A := (I - V_LT_LV_L^T)A$$

$$V_1 = [v_1]$$
$$T_1 = [\tau_1]$$
$$Y_1 = AV_1T_1$$

1. $$T_k = \begin{bmatrix} T_{k-1} & -\tau_kT_{k-1}V_{k-1}^Tv_k \\ 0^T & \tau_k \end{bmatrix}$$

2. $$Y_k = AV_kT_k = \left[Y_{k-1} \quad \tau_k\left(-Y_{k-1}V_{k-1}^Tv_k + Av_k\right)\right]$$

Update formula for one column of a block

3. $$A_{*,k} := A_{*,k} - Y_k((V_k)_{k,*})^T$$
4. $$A_{*,k} := (I - V_kT_k^TV_k^T)A_{*,k}$$

Update formula for the rest of the matrix

5. $$A_{*,L+1:n} := A_{*,L+1:n} - Y_L((V_L)_{L+1:n,*})^T$$
6. $$A_{*,L+1:n} := (I - V_LT_L^TV_L^T)A_{*,L+1:n}$$

Hybrid Hessenberg reduction

With the hybrid implementation we extend the algorithm to the GPU as much as reasonably possible. The GPU will need a high amount of data to be able to achieve high parallelism and thus high performance, so the low on data parts of the calculation are kept on the CPU while the intensive matrix multiplications are feeded to the GPU.

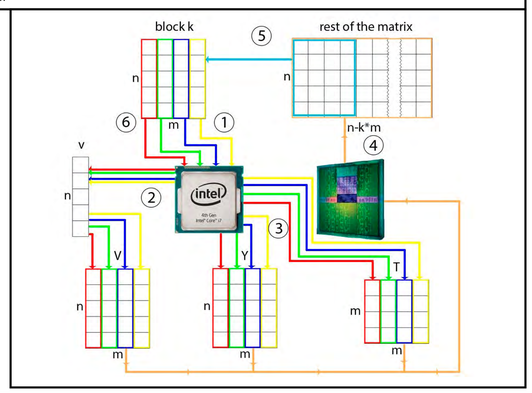
Before commencing any calculations we upload the matrix into the GPU's memory,, after that we follow the next steps for the kth block:

For every column (i) in the block:

1. Compute the Householder vector (v), the ith column of V (CPU)
2. Update T and Y matrices (CPU) [eq. 1,2]
3. Update the next column (CPU) [eq. 3,4]

After updating the block:

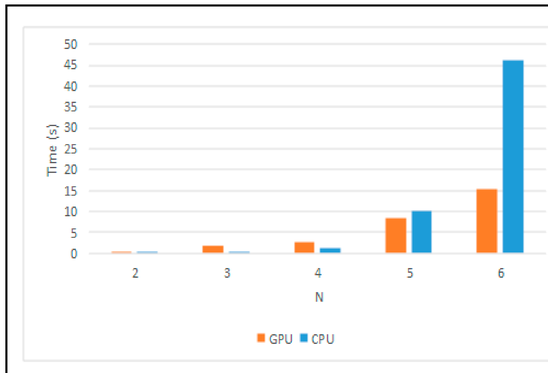
4. Update the rest of the matrix with V,Y,T (GPU) [eq. 5,6]
5. Copy over the next block to the CPU as it has been updated on the GPU
6. Continue the reduction



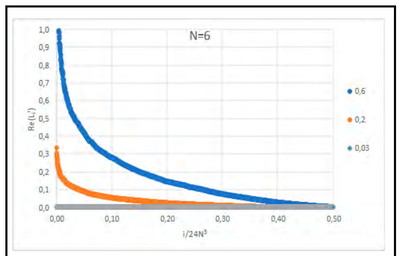
Results, conclusions

For our testing we have used the following system. As for the Hessenberg block we have set the size to be **32** columns. For these tests we have used lattices with **N=2,3,4,5,6**, which gives us a **192x192**, **648x648**, **1536x1536**, **3000x3000** and a **5184x5184** matrix resoectively. Hence our blocks are **192**, **648**, **1536**, **3000** and **5184** long and 32 wide.

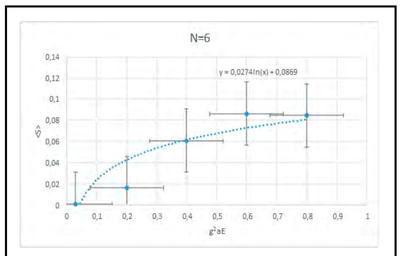
Test system			
CPU	GPU	Memory	OS
Core i7 4710 @2.5 GHz	GTX 980M 4 GB	24 GB	Windows 8.1 x64



Comparing the CPU and GPU runtimes we can see that the GPU is capable to achieve a 3 times faster processing speed, than the CPU. Due to resource limitations the evaluation was done on lattice size N= 2,3,4,5,6. It is visible, that on lower sizes the GPU cannot bring any speed up thanks to the required memory copies.



The positive real part of Lyapunov Spectrum.



The Kolmogorov-Sinai entropy.

1. Comparing the runtime of the CPU to the GPU, the GPU gives a reasonable 3 fold performance gain over the CPU variant.
2. The numerical results fulfill the physical principle, the constraint value of the physical quantity remains constant during the time evolution of the equation of motion.
3. The positive real part of Lyapunov Spectrum justifies the existence of the chaotic motion in the Yang-Mills fields. Kolmogorov-Sinai entropy is obtained from the evolution eigenvalues of the monodromy matrix as functions of the scaled energy. These results gives good approximation for an ideal gas (S~logE). This numerical computation can be extrapolated to larger sizes by parallel algorithms.
4. The behavior of the GPU makes it possible to solve the more complicated systems for example Yang-Mills-Higgs fields. High precision computations in thermo dynamical limit are mandatory, where the GPUs high double precision performance is needed.

Goals:

1. Due to the high memory requirement of the algorithm the current implementation cannot handle very big lattices. It should be considered how to improve the implementation to allow us to partition the matrix for multiplication, so even more dense lattices could be used.
2. It should be considered how to change the matrix structure to exclude the non valuable elements, thus decreasing the size of the matrix.

Reference

T. S. Biró, C. Gong, B. Müller, A. Trayanov, Int. Journ. of Modern Phys. C Vol. 5. (1994)113-14 9,

A. Fülöp, T. S. Biró, Phys. Rev. C 64 (2001)064902(5)

T. S. Biró, A. Fülöp, C. Gong, S. Matinyan, B. Müller, A. Trajanov, Lect. Notes in Phys. 494. (1997)164-176

R. Forster, A. Fülöp, Acta Univ. Sap., Inf., 5, 2 (2013) 184-211

J. Muramatsu, T. Fukaya, S. Zhang, Acceleration of Hessenberg Reduction for Nonsymmetric Eigenvalue Problems in a Hybrid CPU-GPU Computing Environment, International Journal of Networking and Computing, Vol. 1, Nr. 2 (2011), p. 132–143