Numerical problems in Lattice Quantum Chromodynamics

C. Bonati
claudio.bonati@unipi.it

Università di Pisa
Istituto Nazionale di Fisica Nucleare, sezione di Pisa.

Seminari di Analisi Numerica
Pisa, 25/03/2013

Outline

1. Quantum Field Theories and QCD
2. Hybrid Monte Carlo
3. The Dirac matrix linear problem
4. Bibliography

Quantum field theory

$\substack{\text{Quantum}\\\text{Mechanics}} + \substack{\text{Special}\\\text{Relativity}} \rightarrow$ Quantum Field Theories (QFT)

The fundamental variables are the fields $\hat{\phi}(x_\mu)$ which are operator on an Hilbert space.

QFT is an infinite dimensional generalization of QM: $[\hat{x}_a, \hat{p}_b] = i\delta_{ab}$ becomes $[\hat{\phi}(x_\mu), \hat{\pi}(y_\mu)] = i\delta^{(4)}(x_\mu - y_\mu)$.

The fundamental observables are the vacuum expectation values of the operator products: $O(x, y) = \langle 0| T(\hat{\phi}^\dagger(x)\hat{\phi}(y))|0\rangle$

The usual computation method for physical observables is perturbation theory. In some cases perturbation theory is not reliable, since the coupling constant is ≈ 1. Need for non-perturbative methods.

Numerical quantum field theory

Starting point: path-integral quantization (aka Feynman integral) in the Minkowski space-time

$$\langle 0|\, T(\hat{\phi}^\dagger(x)\hat{\phi}(y))|0\rangle = \frac{1}{Z}\int \mathscr{D}\phi\ \phi^*(x)\phi(y)e^{iS[\phi]}$$

$$S[\phi] = \int \mathrm{d}^4x \mathscr{L}[\phi, \partial_\mu\phi] \qquad Z = \int \mathscr{D}\phi e^{iS[\phi]}$$

Wick rotation: analytic continuation from real to imaginary time. Path-integral in euclidean space-time.

$$\langle \phi^*(x)\phi(y)\rangle = \frac{1}{Z}\int \mathscr{D}\phi\ \phi^*(x)\phi(y)e^{-S_E[\phi]} \qquad Z = \int \mathscr{D}\phi e^{-S_E[\phi]}$$

Formally identical to expressions typical of statistical physics!

Quantum Chromodynamics (QCD) in the continuum

$$A_\mu \in \mathfrak{su}(3) \qquad F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu - i[A_\mu, A_\nu] \qquad D_\mu = \partial_\mu - iA_\mu$$

$$\gamma^\mu \in GL(4) \qquad \{\gamma^\mu, \gamma^\nu\} = 2\delta^{\mu\nu}$$

$$\mathscr{L} = \frac{1}{2g^2}\mathrm{Tr}(F_{\mu\nu}F^{\mu\nu}) + \sum_f \bar{\psi}_f^a(\gamma^\mu D_\mu^{ab} + m_f\delta^{ab})\psi_f^b$$

Local gauge invariance: the Lagrangian density is invariant under the local transformation

$$A_\mu \to \Omega A_\mu \Omega^\dagger - i[\partial_\mu\Omega]\Omega^\dagger \qquad F_{\mu\nu} \to \Omega F_{\mu\nu}\Omega^\dagger \qquad \psi \to \Omega\psi$$

For practitioners: A is the connection $1-$form on a principal $SU(3)$ bundle, $F = \mathrm{d}A + A \wedge A$ is the curvature $2-$form, D is the covariant derivative and ψ is a section of the bundle, $\mathscr{L} = \mathrm{Tr}(F \wedge *F)$, second Chern class $\mathcal{C}_2 = \int \mathrm{Tr}(F \wedge F)$.

Quantum Chromodynamics on the lattice (LQCD)

In order to maintain the gauge invariance in the lattice discretized theory (hyper-cubic lattice) the elementary variables of the theory becomes the parallel transports $U_\mu \approx \exp(iaA_\mu)$ ($a =$lattice spacing) and the action can be written in the general form:

$$S = \underbrace{\frac{2N}{g^2}\sum_{\square}\left(1-\frac{1}{N}\mathrm{ReTr}\Pi_{\mu\nu}\right)}_{\text{gauge part}} + \overbrace{\bar{\psi}_i M_{ij}\psi_j}^{\text{fermion part}}$$

M_{ij} =Dirac matrix=sparse structured matrix, whose elements are special unitary matrices of dimension 3 × 3. The explicit form of M depends on the used discretization (Wilson, staggered, domain-wall, overlap).

Problem: in the functional integrals the fermionic variables ψ_i are Grassman variables, i.e. $\{\psi_i, \psi_j\} = 0$. How to perform a Monte Carlo on these variables?

The pseudofermions

The fermionic action is quadratic, so it can be formally integrated (by using the rules of the Grassman algebra) and the result rewritten by using ordinary (non grassmanian) variables (it can be shown that $\det M > 0$)

$$Z = \int \mathscr{D}U_\mu \mathscr{D}\bar{\psi}\mathscr{D}\psi e^{-S_g[U]-\bar{\psi}M[U]\psi} \propto \int \mathscr{D}U_\mu \det M[U] e^{-S_g[U]} \propto$$

$$\propto \int \mathscr{D}U_\mu \mathscr{D}\phi^* \mathscr{D}\phi e^{-S_g[U]-\phi^* \frac{1}{M[U]^\dagger M[U]}\phi} = \int \mathscr{D}U_\mu \mathscr{D}\phi^* \mathscr{D}\phi e^{-S_{\mathrm{eff}}[U,\phi]}$$

Starting from a local fermionic action we arrived to a non local action written by using the bosons ϕ (pseudofermions).
The Metropolis algorithm requires the computation of ΔE and typically M is a matrix of dimension $\mathcal{O}(10^6)$, so a local update (like the ones in statistical physics) is not numerically convenient.

Problem: how to perform a global update with a reasonably large acceptation probability.

Hybrid Monte Carlo

Let's start from the action $S[\phi]$ and introduce the momentum p conjugated to ϕ, from which we build the new action $\tilde{S}[\phi, p]$ as

$$\tilde{S}[\phi, p] = S[\phi] + \frac{1}{2}p^2$$

An Hybrid Monte Carlo update consists of the following steps:

1. the momentum p is generated with the distribution $P_G(p) \propto \exp\left(-\frac{1}{2}p^2\right)$
2. starting from the state (ϕ, p) we get the trial state (ϕ', p') by numerically solving the canonical equation of motion

$$P_C\big((\phi, p) \to (\phi', p')\big) = \delta\big((\phi(t), p(t)) - (\phi', p')\big)$$

3. the trial state is accepted with probability $P_A((\phi, p) \to (\phi', p')) = \min[1, \exp(-\Delta\tilde{S})]$

Hybrid Monte Carlo (proof)

The transition probability for $\phi \to \phi'$ is thus

$$P_M(\phi \to \phi') = \int \mathrm{d}p\mathrm{d}p' P_G(p) P_C\big((\phi, p) \to (\phi', p')\big) P_A\big((\phi, p) \to (\phi', p')\big)$$

From $e^{-S(\phi)} P_G(p) = e^{-\tilde{S}(\phi,p)}$, $\tilde{S}(\phi, p) = \tilde{S}(\phi, -p)$ and

$$e^{-\tilde{S}(\phi,p)} \min(1, e^{-\Delta\tilde{S}}) = e^{-\tilde{S}(\phi',p')} \min(e^{\Delta\tilde{S}}, 1)$$

we get

$$\begin{aligned} e^{-S(\phi)} P_G(p) P_A\Big((\phi, p) \to (\phi', p')\Big) = \\ \overset{(1)}{=} e^{-S(\phi')} P_G(p') P_A\Big((\phi', p') \to (\phi, p)\Big) = \\ = e^{-S(\phi')} P_G(p') P_A\Big((\phi', -p') \to (\phi, -p)\Big) \end{aligned}$$

Hybrid Monte Carlo (proof)

By multiplying this equation by

$$P_C\Big((\phi,p)\to(\phi',p')\Big)\overset{(2)}{=}P_C\Big((\phi',-p')\to(\phi,-p)\Big)$$

we get

$$\begin{aligned}&e^{-S(\phi)}\int \mathrm{d}p\mathrm{d}p' P_G(p)P_C\Big((\phi,p)\to(\phi',p')\Big)P_A\Big((\phi,p)\to(\phi',p')\Big)=\\&=e^{-S(\phi')}\int \mathrm{d}p\mathrm{d}p' P_G(p')P_C\Big((\phi',-p')\to(\phi,-p)\Big)\times\\&\qquad\qquad\qquad\qquad\times P_A\Big((\phi',-p')\to(\phi,-p)\Big)\end{aligned}$$

which after $p\to -p$ and $p'\to -p'$ becomes the detailed balance for ϕ:

$$e^{-S(\phi)}P_M(\phi\to\phi')=e^{-S(\phi')}P_M(\phi'\to\phi)$$

Hybrid Monte Carlo requirements

We have shown that the Markov chain for ϕ obtained by using the HMC satisfies the detailed balance if the following conditions hold true:

1. the temporal evolution does not change the measure on the phase space: $\frac{\partial(\phi,p)}{\partial(\phi',p')} = 1$
2. the time evolution is reversible: $(\phi, p) \to (\phi', p')$ if and only if $(\phi', -p') \to (\phi, -p)$

These properties are surely true in the continuum (Liouville theorem and unicity theorems for differential equations), but have to be satisfied also by the algorithm used to numerically integrate the equations of motion.

A large class of integrators exists that satisfies these conditions: the symmetric symplectic integrators.

The symmetric symplectic integrators

Let's assume to have an action $S = V(q) + T(p)$

$$U(\tau) = \exp\left(\tau \frac{\mathrm{d}}{\mathrm{d}t}\right) = \exp\left(\tau\left[T'(p)\frac{\partial}{\partial q} - V'(q)\frac{\partial}{\partial p}\right]\right)$$

$$\mathcal{Q} = T'(p)\frac{\partial}{\partial q} \qquad e^{\tau\mathcal{Q}} f(q,p) = f(q+\tau T'(p), p)$$

$$\mathcal{P} = -V'(q)\frac{\partial}{\partial p} \qquad e^{\tau\mathcal{P}} f(q,p) = f(q, p-\tau V'(q))$$

Algorithms with errors $O(\tau^2)$

1. leapfrog (or Verlet): $e^{\frac{1}{2}\delta\tau\mathcal{P}} e^{\delta\tau\mathcal{Q}} e^{\frac{1}{2}\delta\tau\mathcal{P}}$
2. second order minimum norm (or Omelyan): $e^{\lambda\delta\tau\mathcal{P}} e^{\frac{\delta\tau}{2}\mathcal{Q}} e^{(1-2\lambda)\delta\tau\mathcal{P}} e^{\frac{\delta\tau}{2}\mathcal{Q}} e^{\lambda\delta\tau\mathcal{P}}$ with $\lambda \approx 0.1931833275$

Multistep integrators

Since the total action is $V = V_g + V_f$ and the first term is "easy" while the second term is "difficult", it is convenient to use multistep integrators:

$$\mathcal{Q} = T'(p)\frac{\partial}{\partial q} \qquad e^{\tau\mathcal{Q}} f(q,p) = f(q+\tau T'(p), p)$$

$$\mathcal{P}_g = -V_g'(q)\frac{\partial}{\partial p} \quad e^{\tau\mathcal{P}_g} f(q,p) = f(q, p-\tau V_g'(q))$$

$$\mathcal{P}_f = -V_f'(q)\frac{\partial}{\partial p} \quad e^{\tau\mathcal{P}_f} f(q,p) = f(q, p-\tau V_f'(q))$$

For the leapfrog:

$$U_0(\delta\tau_0) = e^{\frac{1}{2}\delta\tau_0\mathcal{P}_g} e^{\delta\tau_0\mathcal{Q}} e^{\frac{1}{2}\delta\tau_0\mathcal{P}_g}$$

$$U_1(\delta\tau_1) = e^{\frac{1}{2}\delta\tau_1\mathcal{P}_f}\Big[U_0(\delta\tau_1/N)\Big]^N e^{\frac{1}{2}\delta\tau_1\mathcal{P}_f}$$

The Dirac matrix

Sparse structured matrix (block $SU(3)$) of typical dimension at least

$$4V \gtrsim 4 \times 20^4 = 6.4 \times 10^5 \qquad (\times 3 \text{ for block})$$

For the Wilson discretization ×4 more, for the domain-wall still ×30. The main algebraic properties of M are different depending on the discretization scheme used: For example

- in the staggered case $M_{st} = M_{st}^\dagger$ and M_{st} is positive defined
- in the Wislon case $M_W = \gamma_5 M_W^\dagger \gamma_5$

Typically $\lambda_{min} = m$ (quark's mass) and $\lambda_{max} \sim \mathcal{O}(1)$ and (very) often we are interested in the so called "chiral limit" $m \to 0$. Usual values for the condition number are: $\chi \sim 10^4 \div 10^6$.

In the computation of the fermionic component of the force (and of ΔS) we need to solve systems of the form $Mx = b$ with b a random vector (with gaussian distribution).

Solution algorithms used

The algorithms used in order to solve $Ax = b$ are the Krylov algorithm:

- staggered case: conjugate gradient (CG)
- other cases: typically bi-conjugate stabilized gradient (BiCGstab) or generalized minimal residual (GMRES)

The basic scheme of all the Krylov algorithms is the following:

1. we start from a test solution x_0 and its remainder r_0
2. from the partial solution x_j and its remainder r_j we get some auxiliary vectors $p_j^{(k)}$ obtained by multiplying M or $M^\dagger$ by x_j or r_j
3. we get x_{j+1} and r_{j+1} as a linear combination of x_j, r_j and of the auxiliary vectors $p_j^{(k)}$
4. we iterate from point (2) until $\|r_{j+1}\| < \epsilon$ (goal precision)

An example: the CG algorithm

The simplest Krylov solver is the conjugate gradient algorithm, which can be used to solve $A\boldsymbol{x} = \boldsymbol{b}$ when A is real, symmetric and positive definite (i.e. $(\boldsymbol{y}, A\boldsymbol{y}) > 0$ if $\|\boldsymbol{y}\| > 0$).

```
$\boldsymbol{x}_0$ arbitrary starting point, $\boldsymbol{r}_0 = \boldsymbol{p}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$;
$\epsilon > 0$ goal precision;
while $\|\boldsymbol{r}_n\| > \epsilon$ do
    $\beta_n = -\|\boldsymbol{r}_n\|^2/(\boldsymbol{p}_n, A\boldsymbol{p}_n)$
    /* it can be shown that $(\boldsymbol{p}_n, A\boldsymbol{p}_n) \neq 0$    */
    $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \beta_n \boldsymbol{p}_n$
    $\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + \beta_n A\boldsymbol{p}_n$
    $\alpha_{n+1} = \|\boldsymbol{r}_{n+1}\|^2/\|\boldsymbol{r}_n\|^2$
    $\boldsymbol{p}_{n+1} = \boldsymbol{r}_{n+1} + \alpha_{n+1}\boldsymbol{p}_n$
end
```

Computational limits of Krylov space methods in LQCD

In LQCD all these methods are bandwidth-limited.
There are several ways to reduce the bandwidth-pressure:

- Brute force: we store only the first two raws of the $SU(3)$ matrices and reconstruct the third on-fly when needed
- Residual replacement (aka as reliable updates)

Idea of the residual replacement strategy: we perform the computation in low precision, in this way the true residual diverges from the iteratively computed residual[1]. Sometimes we perform a restarting of the algorithms, with an accurate computation of the true residual[2].
In this way most of the computations are performed in "low precision" but the final result is in "high precision" (on GPUs also the half-precision is used).

[1] A. Greenbaum, Estimating the attainable accuracy of recursively computed residual methods, SIAM J. Matrix Anal. Appl. 18 (1997) 535.

[2] H. K. van der Vorst, Q. Ye, Residual replacement strategies for Krylov subspace iterative methods for the convergence of true residuals, SIAM J. Sci. Comput. 22 (2000) 835.

Rational Hybrid Monte Carlo

Some time (because it is needed by the discretization or in order to speed-up the algorithm) we need to solve

$$M^{\alpha} x = b \qquad \alpha \in \mathbb{Q} \qquad (\mathrm{let's\ assume}\ M = M^{\dagger})$$

- we compute a rational approximation $R(x)$ of x^{α} on $[\epsilon, 1]$ accurate with given precision (possibly machine precision)

$$R(x) = a_0 + \sum_i \frac{a_i}{x + \sigma_i}$$

- if $\lambda_{min}/\lambda_{max} > \epsilon$ we can use

$$M^{\alpha} \approx R(M) = \frac{a_0}{\lambda_{max}^{\alpha}} + \sum_i \frac{a_i/\lambda_{max}^{\alpha+1}}{M + \sigma_i/\lambda_{max}}$$

- so we need to solve $(M + \sigma_i) = x$ (luckily $\sigma_i > 0$)

Shift-inverters

In order to solve k systems

$$(M + \sigma_i)x = b$$

with a Krilov solvers we would need typically $\times\mathcal{O}(k)$ matrix×vector products with respect to the simple case $Mx = b$.
By using the shifted versions of the solvers we arrive to a $\mathcal{O}(1)$ difference, see[3].
No free lunch principle: the shift-solvers are much less versatile than the original Krylov algorithms:

- is much harder to use preconditioners
- for the algorithm to converge all the starting residuals have to be collinear, so we need to start from a vanishing test solution

In particular: it is not possible to restart the algorithm and to use the residual replacement strategy.

[3] B. Jegerlehner "Krylov space solvers for shifted linear systems" arXiv:hep-lat/9612014

The overlap discretization

The "overlap" discretization of the Dirac equation is the best one from the theoretical physics point of view (manifest chiral invariance at non-vanishing lattice spacing[4]), but the discretized Dirac matrix is given by

$$D_O = 1 + \frac{D_W - m}{\sqrt{(D_W - m)^\dagger (D_W - m)}}$$

and thus requires two nested Krilov solver: the "standar one" (the outer) and an inner one for the computation of the square root.

This discretization requires a number of computations $\approx \mathcal{O}(10^2)$ higher with respect to the others (less theoretically pleasant) discretizations. For this reason its use is still extremely limited.

[4] D. B. Kaplan, Chiral Symmetry and Lattice Fermions, arXiv:0912.2560

Computing power required

In order to generate enough data for a real simulation of LQCD thermodynamics typically $\mathcal{O}(10^7 \div 10^8)$ linear system has to be solved. (Each molecular dynamics trajectory ~ 20, number of trajectories for fixed β value $\sim 10^5$, number of βs ~ 10)
Need for dedicated massively parallel machines!

Past: APE machines (produced by INFN) since the late '80s up to a few years ago.

Today:

- PC clusters
- Blue Gene or similar
- Video Card (Graphic Processing Units, GPU)
- Heterogeneous clusters mixing the above ingredients

General bibliography

For QFT and QCD in particular

- S. Weinberg "The quantum theory of fields" vol. 1&2, Cambridge.
- A. Smilga "Lectures on quantum chromodynamics", World Scientific.

For non abelian gauge theories (intro. to formal aspects)

- A. Jaffe, E. Witten "Quantum Yang-Mills theory" official description of the millennium prize "Yang-Mills and Mass Gap" (Clay Mathematics Institute).

For lattice QCD

- I. Montvay, G. Munster "Quantum fields on a lattice", Cambridge.
- H. J. Rothe "Lattice gauge theories", World Scientific.
- T. DeGrand, C. DeTar "Lattice methods for quantum chromodynamics", World Scientific.

Something more specific

- M. Hasenbusch "Speeding up the Hybrid-Monte-Carlo algorithm for dynamical fermions" arXiv:hep-lat/0107019
- M. Luscher "Schwarz-preconditioned HMC algorithm for two-flavour lattice QCD" arXiv:hep-lat/0409106
- A. D. Kennedy "Algorithms for Dynamical Fermions" arXiv:hep-lat/0607038
- M. Luscher "Deflation acceleration of lattice QCD simulations" arXiv:0710.5417
- M. A. Clark et al. "Solving Lattice QCD systems of equations using mixed precision solvers on GPUs" arXiv:0911.3191
- R. Babich et al. "Adaptive multigrid algorithm for the lattice Wilson-Dirac operator" arXiv:1005.3043