Cross contamination from WP1

Mattia Bruno

WP5 meeting - 12.04.2023

Lattice field theories

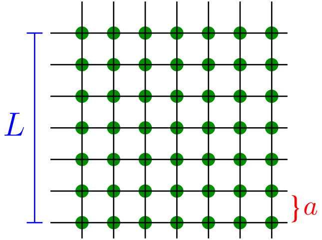
Due to confinement → non-perturbative formulation is necessary

lattice spacing a → regulate UV divergences

finite size L → infrared regulator

Continuum theory $a \to 0$, $L \to \infty$

Euclidean metric → Boltzman interpretation of path integral



$$\langle O \rangle = \mathcal{Z}^{-1} \int [DU] e^{-S[U]} O(U) \approx \frac{1}{N} \sum_{i=1}^{N} O[U_i]$$

Very high dimensional integral → Monte-Carlo methods

Markov Chain of gauge field configs $U_0 \to U_1 \to \cdots \to U_N$

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Lattice QCD workflow

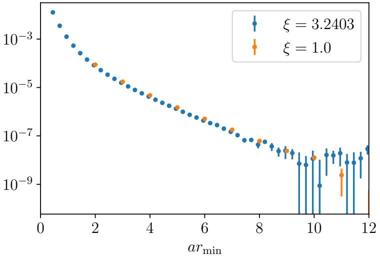
Typical case: $SU(3)$ + up,down,strange dynamical quarks
lattices from $32^3 \times 96$ up to $96^3 \times 192$ $[12 \cdot 10^9$ d.o.f.$]$

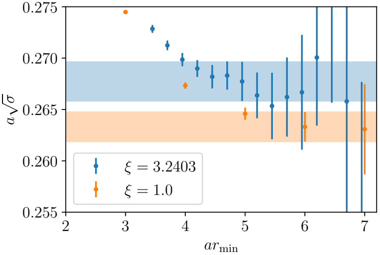
1. production
   generation of configurations U_i
   large supercomputers, $O(100)$ configs in months
   $96^3 \times 192 = 91$ GB $\times$ 300 configs $= 27$ TB

2. measurements
   calculate $O[U_i]$ for all U_i
   still needs supercomputers, usually 10-50% of cost of generating U_i
   $O[U_i]$ at most 1GB per config, usually 10-100 MB

3. analysis
   read $O[U_i]$ take averages, estimate errors
   compute functions of averages: physics

Example





correlator: every point is observable

points correlated
evaluated on same U_i

$y = ae^{-\sqrt{\sigma}r}$
perform fits to extract physical quantity

software library to identify common configs for correct error propagation

PYOBS

A library to analyse Markov Chain Monte Carlo data

- python: based on numpy, scipy with additional c++ modules
- single-core (unless numpy+BLAS), often RAM limited
  - on laptops hitting more and more the problem of out-of-RAM
  - dask interesting direction

WP1 - Lattice

1. development of algorithms for improved $O[U_i]$
2. development of algorithms for generations of U_i at finite temperature

Supporting WP1

- creation of "database/repository" of configs U_i
- creation of "database/repository" of observables $O[U_i]$
    - suitable for WP5?

Phase 1

Backend:

1. several PetaBytes on tapes w/ U_i
   different simulations, e.g. different quark masses etc..
2. a few TeraBytes on disks w/ $O[U_i]$
   observables so far measured, with space for new ones

Frontend:

1. webpage with description of ensembles
   "diagnosis" observables (numerical quality of Monte Carlo)
2. server for analysis of observables $O[U_i]$

On-going discussion w/ CNAF for backend+frontend

Frontend

1. Webpage

Quantities measured on-the-fly to diagnose quality/health of simulations

analysed in python w/ pyobs

displayed w/ html+java

Format expandable, easy to maitain (add new pages, new sections)

suggestions welcome, personnel required

Automatic deployment: configs uploaded on backend + observables uploaded on frontend trigger wepage upgrade

---

2. JupyterHub analysis server

controlled working environment w/ preinstalled+maitained libraries

access to $O[U_i]$ (not U_i) for advanced/physics analysis

possibility to test/develop algorithms for $O[U_i]$ as well

ILDG - PHASE 2

Part of backend as storage point for Italian Lattice Community

ILDG: International Lattice Data Grid

- community agreed file format + metadata schema
- 10-year effort currently being revitalized
- respecting FAIR data policies
- ongoing effort for deployable API (e.g. fetch metadata)

Possible phase 2 of WP1 data-effort:

- create metadata catalogue that "speaks" ILDG
- "connect" CNAF storage point to ILDG
- interface with ILDG auth system (gridFTP)

A python frontend for massively parallel Lattice QCD calculations

- based on C++ Grid parallel library (SIMD, SIMT, GPU)
- heavy-load on performant C++ library
- algorithms in python, object-oriented

User-friendly: code very similar to math equations

- fast test of new ideas
- w/o compromise on performances

Customize and adapt: machine-learning module + quantum computing simulators