Setting up Chroma

For the numerical exercises we will use the Chroma software suite https://github.com/JeffersonLab/chroma. Chroma can be compiled with different libraries to run calculations both on CPUs and GPUs. We will use precompiled CPU only version that runs on google colab. Run the next 2 cells to download and set the environment for the exercises.

```
#%rm -r /home/software /home/tests /home/chroma_gcolab/ /home/run*
%mkdir /home/software
%cd /home/software
!gdown https://drive.google.com/uc?id=16woEAYY0VPJqufF2574Q1AM2NFh13qH1
!unzip chroma.zip
!rm chroma.zip
!chmod u+rwx /home/software/chroma/install/chroma/bin/*
%cd /home
!git clone https://github.com/henrymonge/chroma_gcolab.git
!mv chroma_gcolab/tests .
%cd /home/chroma_gcolab
!git pull
! pip install mpi4py
```

Show hidden output

```
%cd /home/software/chroma
%env TOPDIR_HIP=/home/software/chroma
%env INSTALLROOT=${TOPDIR_HIP}/install
%env LD_LIBRARY_PATH=/home/software/chroma/install/chroma/lib:/home/software/chroma/install/qdpxx/lib:/home/software/chroma

#Load python modules
%cd /home/chroma_gcolab/modules
import importlib
import defs
import xml_input
importlib.reload(xml_input)
importlib.reload(defs)
```

```
/home/software/chroma
env: TOPDIR_HIP=/home/software/chroma
env: INSTALLROOT=${TOPDIR_HIP}/install
env: LD_LIBRARY_PATH=/home/software/chroma/install/chroma/lib:/home/software/chroma/install/qdpxx/lib:/home/software/chr
/home/chroma_gcolab/modules
<module 'defs' from '/home/chroma_gcolab/modules/defs.py'>
```

Lattice QCD Numerical Calculations

Lattice QCD calculations are commonly performed in three stages:

1. Gauge field generation
2. Quark propagator computation/Constructing observables from quark propagators
3. Extracting physical quantities

Our first exercise will be to generate a gauge field configuration using Chroma.

Stage 1: Generating an ensemble of gauge field configurations

Observables in LQCD are obtained by solving the Feynman path integral:

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \mathcal{D}\psi \mathcal{D}\bar{\psi} e^{-S[U,\psi,\bar{\psi}]} \mathcal{O}[U,\psi,\bar{\psi}]$$

To evaluate numerically this integral, Monte Carlo methods and importance sampling are used, such that:

$$\langle \mathcal{O} \rangle \approx \sum_{U} \mathcal{O}[U,\psi,\bar{\psi}]$$

Our goal is to generate and ensemble of gauge field configurations according to the distribution e^{-S} . Given a gauge field configuration U_0, an N-configuration ensemble can be generated as Markov chain starting at U_0 and ending at U_{N-1} . An element U_n in a Markov chain can be generated using the previous element, U_{n-1} and the metropolis algorithm.

$$U_0 \rightarrow U_1 \rightarrow U_2 \ldots \rightarrow U_N$$

The index n is called the computer time and the process of changing a field to a new one is referred as an ***update*** or ***Monte Carlo step***.

The transition from configurations is characterized by a transition probability:

$$P(U_n = U´|U_{n-1} = U) = T(U´|U).$$

and must satisfy the condition that the probability for moving from $U \rightarrow U´$ is the same as $U´ \rightarrow U$. This condition is know as ***detailed balanced*** can be written as:

$$T(U'|U)P(U) = T(U|U')P(U')$$

Metropolis algorithm

This algorithm is designed to advance a markov chain from a step n-1 to a step n. For a gauge field U given a distribution $P(U) \propto exp(-S[U])$ the steps are the following:

1. Generate a candidate configuration U' according to some a priori selection probability $T_0(U'|U)$, where $U = U_{n-1}$
2. Accept the candidate configuration U' as the new configuration Un with the acceptance probability $T_A(U'|U) = min(1, \frac{T_0(U|U')exp(-S[U'])}{T_0(U'|U)exp(-S[U])})$.

3.Repeat the previous step.

Exercise: verify that the detailed balance conditioned is satisty for the Metropolis algorith with $P(U) = e^{-S[U]}$.

Hybrid Monte Carlo

In order generate the gauge field configurations, we need to generate propose gauge field configurations $U´$. Molecular dynamics is commonly used to evolve a current gauge field configuration to a new gauge field configuration $U´$ that we can used as a proposed gauge field configuration.

In molecular dynamics, a system described by the Hamiltonian $H(Q, P)$ is constant under time evolution. Q and P and are the canonical coordinate and canonical momentum respectively. If we consider the following Hamiltonian:

$$H(P, Q) = \frac{1}{2}P^2 + V(Q) ,$$

Q and P obey the following equations of motions:

$$\dot{Q} = \frac{\partial H}{\partial P} , \qquad \dot{P} = -\frac{\partial H}{\partial Q} .$$

If we can integrate the above equations we can obtain a $Q´$ from an initial $Q(0)$. In our case $Q = U$ and there is no analytical formula for this and the integration must be performed numerically.

While there are multiple schemes to numerically implement this integration, not all choices satistify the required conditions for generating Markov chains. Lets call the integration scheme and integrator.

A good integrator is the Leapfrog integrator. In each iteration, Leapfrog evolves P in 2 steps and Q in one steps as follows:

$$\begin{aligned} P(t + dt/2) &= P(t) - V'(Q(t))dt/2 , \\ Q(t + dt) &= Q(t) + P(t + dt/2)dt , \\ P(t + dt) &= P(t + dt/2) - V'(Q(t + dt))dt/2 \end{aligned}$$

at a later time $d\tau$, we can solve for Q and P:

$$\begin{aligned} P(d\tau) &= -V'(Q(d\tau))d\tau , \\ Q(d\tau) &= P(d\tau)d\tau . \end{aligned}$$

Every term in the action must be evolve, the only choices we need to make for the LeapFrog integrator are the number of interation steps and the total time τ

HMC with Chroma

The first exercise is to generate a gauge field configuration using Chroma and to explore how different parameters affect this calculation. For this we need to choose the action and some HMC parameters.

Choosing the action

$$\mathcal{S}_{\text{QCD}} = \int \sum_{f=1}^{N} \bar{\psi}_f (i\gamma^\mu \partial_\mu - m_f)\psi_f + \frac{1}{4}\text{Tr}\, F_{\mu\nu}F^{\mu\nu}$$

The action shown above has one pure gauge term only and two fermion terms for each flavor. LQCD codes, like Chroma, are flexible and allow you to choose what terms to include to simulate the physics you want.

For instance you could choose a pure gauge action:

$$\mathcal{S}_{\text{QCD}} = \int \frac{1}{4}\text{Tr}\, F_{\mu\nu}F^{\mu\nu}$$

or an action with 3 dynamical fermions

$$\mathcal{S}_{\text{QCD}} = \int \sum_{f=1}^{3} \bar{\psi}_f (i\gamma^\mu \partial_\mu - m_f)\psi_f + \frac{1}{4}\text{Tr}\, F_{\mu\nu}F^{\mu\nu}$$

Therefore, the first decision to make is how many fermions we want in our simulation.

Every term in the actions is called a monomial, therefore the action is given by:

$$\mathcal{S}_{\text{QCD}} = \mathcal{S}_{\text{g}} + \mathcal{S}_{\text{f1}} + S_{\text{f2}} \ldots$$

Choosing the number and type of quarks is the first part to define the action and it is not enough to set up a simulation we need to choose the action discretization. The monomial implementations are discretization dependent, therefore the second step is to choose an gauge and fermion actions:

1. Wilson Gauge Action
2. Wilson Fermions
3. Clover Fermions

Finally, we choose the numerical techniques to compute the monomials.

1. Unpreconditioned
2. Even-Odd preconditioned

The following are some choices available on Chroma:

1. GAUGE_MONOMIAL
$$\int \frac{1}{4}\text{Tr}\, F_{\mu\nu}F^{\mu\nu}$$
2. TWO_FLAVOR_EOPREC (N_FLAVOR_LOGDET_EVEN_EVEN_FERM_MONOMIAL,TWO_FLAVOR_EOPREC_CONSTDET_FERM_MONOMIAL)
$$\int \sum_{f=1}^{2} \bar{\psi}_f (i\gamma^\mu \partial_\mu - m_f)\psi_f$$
$m_1 = m_2 = m_f$
3. ONE_FLAVOR_EOPREC_CONSTDET_FERM_RAT_MONOMIAL
$$\int \bar{\psi}_f (i\gamma^\mu \partial_\mu - m_f)\psi_f$$

In the next cell we select the parameters that define our action.

```
#Monomials
#Gauge Monomials
#Fermion Monomials
params={}
params['Monomials']={}

params['Monomials']['GAUGE_MONOMIAL']={'ACTION':'WILSON_GAUGEACT',
                                       'BETA':'3.4','ID':'HMC::gauge'}
params['Monomials']['ONE_FLAVOR_EOPREC']={'ACTION':'WILSON',
                                          'MASS':'-0.04','CLOV_COEFF':'1.2','ID':'HMC::strange'}
#params['Monomials']['TWO_FLAVOR_EOPREC']={'ACTION':'WILSON','MASS':'-0.02',
#                                          'CLOV_COEFF':'1.2','ID':'HMC::light'}
params['Monomials']['TWO_FLAVOR_EOPREC_OO']={'ACTION':'WILSON','MASS':'-0.02',
```

```
                                        'CLOV_COEFF':'1.2','ID':'HMC::light'}
defs.setMonomialIds(params)
defs.display_action(params['Monomials'])
```

```
**************************************************
              My chosen action is:
**************************************************
```

$$\mathcal{S}_{\rm QCD} = \int \tfrac{1}{4}{\rm Tr}\, F_{\mu\nu}F^{\mu\nu} + \int \bar{\psi}_f(i\gamma^\mu\partial_\mu - m_f)\psi_f + \int \sum_{f=1}^{2} \bar{\psi}_f(i\gamma^\mu\partial_\mu - m_f)\psi_f$$

```
#MD integrator
#Choose the integrator parameters
params['INTGRTOR']='LCM_STS_LEAPFROG'
params['INTGRTOR_STEPS']='64'
params['INTGRTOR_TAU']='1'

#HMC updates
params['WARMUPS']='1'
params['PROD_UPDATES']='1'
params['NUPDATES']='1'

#Lattice and gauge field
params['VOL']=' 4 4 4 8'
params['CFG']='WEAK_FIELD'


outdir='/home/run_0'
%mkdir $outdir
ini,out,log,stdout=defs.writeInputFile(params,outdir) #update input file
!echo "Input file is {ini}"
```

```
Input file is /home/run_0/hmc.ini.xml
```

Running Chroma

A useful LQCD calculation must use multiple computing resources to run. Chroma is designed to run in parallel on multiple resources using mpi. Therefore, running chroma requires at least the following parameters:

1. Number of processes: N
2. 4D Geometry: -geom GX GY GZ GT
3. Input fiile: $ini
4. Optional: output and log filenames $out, $log

For example: srun -N 8 path/to/chroma -geom 1 1 2 4 -i $ini -o $out -l $log

For a Lattice of volume $L_x L_y L_z L_T$, L_i/G_i must be an integer.

```
#!/home/software/chroma/install/chroma/bin/hmc -geom 1 1 1 1 -i $ini -o $out -l $log > $stdout 2>&1
! mpiexec --allow-run-as-root --oversubscribe -np 1 /home/software/chroma/install/chroma/bin/hmc -geom 1 1 1 1 -i $ini -o $
```

Computational cost volume dependence

The computational cost of HMC is $\propto V^{\frac{5}{4}}$.

Perform 4 runs with different volumes. Does the run time increases as $\propto V^{\frac{5}{4}}$?

```
#Run 1
params['VOL']= ' 4 4 4 16'#,' 4 4 4 16',' 4 4 4 32',' 8 8 8 16'
outdir='/home/run_VOL_1'
%mkdir $outdir
ini,out,log,stdout=defs.writeInputFile(params,outdir) #update input file
! mpiexec --allow-run-as-root --oversubscribe -np 1 /home/software/chroma/install/chroma/bin/hmc -geom 1 1 1 1 -i $ini -o $
```

```
!echo 'The stdout file is $stdout' # Look for HMC: total time in this files
!grep -r "HMC: total time" $stdout #
```

```
The stdout file is $stdout
HMC: total time = 46.232859 secs
```

```
#Run 2
#To fill in copy above and change VOL and outdir number


#Run 3
#To fill in


#Run 4
#To fill in
```

Acceptance rate

Every molecular dynamics trajectory generates a new gauge field configuration. This new configuration will be accepted depending on the acceptance probability.

Perform 4 calculations changing the trajectory length and the number of steps. How does the acceptance rate changes?

```
#Run 1
params['VOL']= ' 4 4 4 8'
params['INTGRTOR_STEPS']= '64'
params['INTGRTOR_TAU']= '1.0'
outdir='/home/run_ACP_0'
%mkdir $outdir
ini,out,log,stdout=defs.writeInputFile(params,outdir) #update input file
! mpiexec --allow-run-as-root --oversubscribe -np 1 /home/software/chroma/install/chroma/bin/hmc -geom 1 1 1 1 -i $ini -o $

!echo 'The stdout file is $stdout' # Look for AccProb =  in this files
!grep -r "Delta H = " $stdout
!grep -r "AccProb = " $stdout
!grep -r "AcceptP = " $stdout
```

```
mkdir: cannot create directory ‘/home/run_ACP_0’: File exists
The stdout file is $stdout
Delta H = 1.12279775234674
AccProb = 0.325368220336095
AcceptP = 1
```

```
#Run 2
#To fill in copy above and change VOL and outdir number


#Run 3
#To fill in copy above and change VOL and outdir number


#Run 4
#To fill in copy above and change VOL and outdir number
```

Preconditioning

Whe computing the fermion monomials, the algorithms must solve equations such as:

$$D\chi = \psi$$

The time it takes to solve this equations depends on the eigenvalues of D. Alternative, we can solve a slightly different equation and extract the solution in an additional step. For example:

$$M^{-1} D\chi = M^{-1} \psi$$

In many cases, the eigenvalues of $M^{-1} D$ and larger and thus the solution can be found faster.

```
#Resetting params
#MD integrator
#Choose the integrator parameters
params['INTGRTOR']='LCM_STS_LEAPFROG'
params['INTGRTOR_STEPS']='64'
params['INTGRTOR_TAU']='1'

#HMC updates
params['WARMUPS']='1'
params['PROD_UPDATES']='1'
params['NUPDATES']='1'

#Lattice and gauge field
params['VOL']=' 4 4 4 16'
params['CFG']='WEAK_FIELD'



#Run 1 Preconditioned
params['Monomials']={}
params['Monomials']['TWO_FLAVOR_EOPREC_OO']={'ACTION':'WILSON','CLOV_COEFF':'None',
                                        'MASS':'-0.04','ID':'HMC::light_oo'}

defs.setMonomialIds(params)
defs.display_action(params['Monomials'])
outdir='/home/run_PREC_0'

%mkdir $outdir
ini,out,log,stdout=defs.writeInputFile(params,outdir) #update input file
!echo $ini
! mpiexec --allow-run-as-root --oversubscribe -np 1 /home/software/chroma/install/chroma/bin/hmc -geom 1 1 1 1 -i $ini -o $

!grep -r "HMC: total time" $stdout #
```

```
**************************************************
               My chosen action is:
**************************************************
```

$$\mathcal{S}_{\mathrm{QCD}} = \int \sum_{f=1}^{2} \bar{\psi}_f (i\gamma^\mu \partial_\mu - m_f)\psi_f$$

```
mkdir: cannot create directory ‘/home/run_PREC_0’: File exists
/home/run_PREC_0/hmc.ini.xml
HMC: total time = 16.223388 secs
```

```
#Run 2 Unpreconditioned
params['Monomials']={}
params['Monomials']['TWO_FLAVOR_UNPREC']={'ACTION':'WILSON','CLOV_COEFF':'None',
                                        'MASS':'-0.04','ID':'HMC::light'}

defs.setMonomialIds(params)
defs.display_action(params['Monomials'])
outdir='/home/run_PREC_1'

%mkdir $outdir
ini,out,log,stdout=defs.writeInputFile(params,outdir) #update input file
!echo $ini
! mpiexec --allow-run-as-root --oversubscribe -np 1 /home/software/chroma/install/chroma/bin/hmc -geom 1 1 1 1 -i $ini -o $

!grep -r "HMC: total time" $stdout #
```

```
**************************************************
               My chosen action is:
**************************************************
```

$$\mathcal{S}_{\mathrm{QCD}} = \int \sum_{f=1}^{2} \bar{\psi}_f (i\gamma^\mu \partial_\mu - m_f)\psi_f$$

```
/home/run_PREC_1/hmc.ini.xml
HMC: total time = 22.570882 secs
```

Exploring more calculations

Below are the cells with all the parameters that you can modify to perform new simulations.

Changes to try:

1. MD trajectory length: TAU
2. Fermion masses: MASS
3. Gauge coupling: BETA
4. MD steps: NSTEPS
5. Gauge field type

```
#Monomials
#Gauge Monomials
#Fermion Monomials
params={}
params['Monomials']={}

params['Monomials']['GAUGE_MONOMIAL']={'ACTION':'WILSON_GAUGEACT',
                                       'BETA':'3.4','ID':'HMC::gauge'}
params['Monomials']['ONE_FLAVOR_EOPREC']={'ACTION':'CLOVER',
                                          'MASS':'-0.04','CLOV_COEFF':'1.2','ID':'HMC::strange'}
params['Monomials']['TWO_FLAVOR_EOPREC']={'ACTION':'CLOVER','MASS':'-0.02',
                                          'CLOV_COEFF':'1.2','ID':'HMC::light'}

#MD integrator
#Choose the integrator parameters
params['INTGRTOR']='LCM_STS_LEAPFROG'
params['INTGRTOR_STEPS']='64'
params['INTGRTOR_TAU']='1'

#HMC updates
params['WARMUPS']='1'
params['PROD_UPDATES']='1'
params['NUPDATES']='1'

#Lattice and gauge field
params['VOL']=' 4 4 4 8'
params['CFG']='WEAK_FIELD'  #DISORDERED,UNIT

defs.setMonomialIds(params)
defs.display_action(params['Monomials'])
```

```
    ************************************************
                  My chosen action is:
    ************************************************
```

$$\mathcal{S}_{\rm QCD} = \int \tfrac{1}{4}\mathrm{Tr}\, F_{\mu\nu}F^{\mu\nu} + \int \bar{\psi}_f(i\gamma^\mu\partial_\mu - m_f)\psi_f + \int \sum_{f=1}^{2} \bar{\psi}_f(i\gamma^\mu\partial_\mu - m_f)\psi_f$$

```
outdir='/home/run_0'
%mkdir $outdir
ini,out,log,stdout=defs.writeInputFile(params,outdir) #update input file
!echo "Input file is {ini}"
! mpiexec --allow-run-as-root --oversubscribe -np 1 /home/software/chroma/install/chroma/bin/hmc -geom 1 1 1 1 -i $ini -o $ou
```

```
mkdir: cannot create directory ‘/home/run_0’: File exists
Input file is /home/run_0/hmc.ini.xml
```

References

For those who would like to dive deeper into the computional aspects here are some references.

LQCD hackathon by Balint Joo

https://archive.int.washington.edu/PROGRAMS/12-2c/week3/joo_01.pdf https://archive.int.washington.edu/PROGRAMS/12-2c/week3/joo_02.pdf https://archive.int.washington.edu/PROGRAMS/12-2c/week3/joo_03.pdf

https://archive.int.washington.edu/PROGRAMS/12-2c/week3/joo_04.pdf https://archive.int.washington.edu/PROGRAMS/12-2c/week3/joo_05.pdf

Prof. Dr. Christoph Lehner lectures

This lectures have jupyter notebooks using Grid Python Toolkit https://github.com/lehner/gpt.

https://homepages.uni-regensburg.de/~lec17310/teaching/wise2324/lqft.html

Chroma build scripts

https://github.com/henrymonge/ChromaBuildScripts