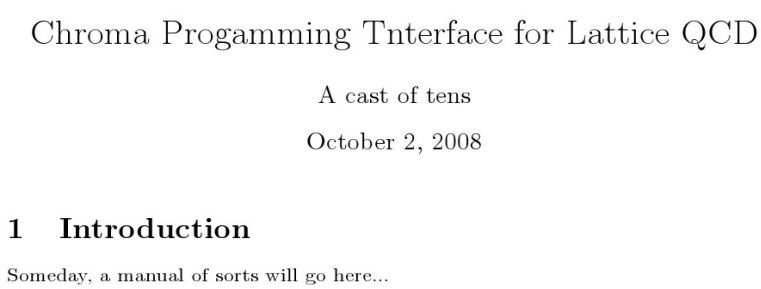
Computing correlation functions for lattice QCD using CHROMA

Andreas Jüttner

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Lattice practices 2008, DESY Zeuthen
08-10.10.2008

- CHROMA is huge and there is no such thing as a comprehensive documentation



Chroma Progamming Tnterface for Lattice QCD

A cast of tens

October 2, 2008

1 Introduction

Someday, a manual of sorts will go here...

- there are some notes in the doc/notes-directory
- if you want to find out how things work either
  - browse DOXYGEN
    (an up-to-date version can be found on the USLQCD web-page
    http://usqcd.jlab.org/usqcd-docs/chroma/)
  - or browse the directory tests/ for example xml-input
  - or use Huey-Wen Lin's compilation of notes
    http://www.jlab.org/~hwlin/Chroma_notes/chroma_note.html
  - or go through one of Balint Joo's many tutorials
    (e.g. http://www.jlab.org/~dgr/HackLatt07/Chroma_measurement.htm)

A basic measurement:

- read the gauge configuration
- make all kind of quark sources
- invert the on the sources to compute a quark propagator
- potentially save the propagator for later use
- sink-smear the propagator
- perform desired contractions on the quark propagators
- write results to disk

Plan:

- I will now discuss the basic xml-inupt to compute some of the observables which were discussed in the previous part
- the examples have partly been taken from the chroma/tests/chroma-tree
- the screen-shots have been taken from the firefox-xml-browser

Basic structure of the XML-input to CHROMA

```
- <chroma>
    <annotation> Hadron spectrum input </annotation>
  + <Param></Param>
  + <RNG></RNG>
  + <Cfg></Cfg>
  </chroma>
```

format for comments to chroma in input XML

Blocks specifying measurement parameters, the random number generator and the configurations on which the measuremts will be carried out

It is not a bad idea to use Firefox or similar to browse XML-files since it lets you expand and collapse the XML-tree – CHROMA-XML input can become quite complicated

Basic structure of the XML-input to CHROMA

```
– <chroma>
    <annotation> Hadron spectrum input </annotation>
  – <Param>
    – <InlineMeasurements>
        + <elem></elem>
        + <elem></elem>
        + <elem></elem>
        + <elem></elem>
        + <elem></elem>
        + <elem></elem>
        + <elem></elem>
        + <elem></elem>
       </InlineMeasurements>
       <nrow>4 4 4 8</nrow>
    </Param>
  + <RNG></RNG>
  + <Cfg></Cfg>
</chroma>
```

<InlineMeasurements> contains

- MAKE_SOURCE
- PROPAGATOR
- ERASE_NAMED_OBJECT (source)
- QIO_WRITE_NAMED_OBJECT (save propagator)
- SINK_SMEAR
- ERASE_NAMED_OBJECT (prop without sink smearing)
- HADRON_SPECTRUM
- ...

specifies the lattice geometry
default: (x,y,z,t) (actually j_decay sets the time-direction

- there are mandatory blocks (sink_smearing) which have to be there – leave them as trivial (point sink) if not desired – otherwise: crash
- CHROMA allows to deallocate memory space by erasing the “NAMED_OBJECT”
- I will now go through the <InlineMeasurement> blocks

Make a source – point source

```
- <elem>
    <Name>MAKE_SOURCE</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>6</version>
    - <Source>
        <version>2</version>
        <SourceType>POINT_SOURCE</SourceType>
        <j_decay>3</j_decay>
        <t_srce>0 0 0 0</t_srce>
      - <Displacement>
          <version>1</version>
          <DisplacementType>NONE</DisplacementType>
        </Displacement>
      </Source>
    </Param>
  - <NamedObject>
      <gauge_id>default_gauge_field</gauge_id>
      <source_id>pt_source_0</source_id>
    </NamedObject>
  - <File>
      <file_name>./pt_source_0</file_name>
      <file_volfmt>MULTIFILE</file_volfmt>
    </File>
  </elem>
```

Speciy source type here: e.g.
POINT_SOURCE
WALL_SOURCE
SHELL_SOURCE
BNDST_SOURCE
RAND_Z2_WALL_SOURCE
RAND_U1_WALL_SOURCE
the structure of the remainder of <Parameter> depends on the source type

decay direction and source time-slice

The source needs a name to refer to it at a later stage. In Chroma this is done via <NamedObject>

In some cases (e.g. light quark propagator) it makes sense to write the source to disk for later usage

Make a source – noise source

```
- <elem>
  - <annotation>
      Diluted complex Z(2) = Z(4) random source. Even sites.
    </annotation>
    <Name>MAKE_SOURCE_FERM</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>6</version>
    - <Source>
        <SourceType>RAND_DILUTE_ZN_SOURCE</SourceType>
        <version>1</version>
        <N>4</N>
        <j_decay>3</j_decay>
        <t_source>0</t_source>
      - <ran_seed>
        - <Seed>
            <elem>201</elem>
            <elem>213</elem>
            <elem>215</elem>
            <elem>217</elem>
          </Seed>
        </ran_seed>
        <spatial_mask_size>2 2 2</spatial_mask_size>
      - <spatial_mask>
          <elem>0 0 0</elem>
          <elem>1 1 0</elem>
          <elem>1 0 1</elem>
          <elem>0 1 1</elem>
        </spatial_mask>
        <color_mask>0 1 2</color_mask>
        <spin_mask>0 1 2 3</spin_mask>
      </Source>
    </Param>
  - <NamedObject>
      <gauge_id>default_gauge_field</gauge_id>
      <source_id>zN_source</source_id>
    </NamedObject>
  </elem>
```

Z(4) is $\frac{1}{\sqrt{2}}(\pm 1 \pm i1)$

seeds for RNG

very complicated way of diluting in the volume
(I think this one gives you the even sites)

color and spin dilution – no dilution in this example

Make a source – Gauss smeared

```
- <elem>
    <Name>MAKE_SOURCE</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>6</version>
    - <Source>
        <version>2</version>
        <SourceType>SHELL_SOURCE</SourceType>
        <j_decay>3</j_decay>
        <t_srce>0 0 0 0</t_srce>
      - <SmearingParam>
          <wvf_kind>GAUGE_INV_GAUSSIAN</wvf_kind>
          <wvf_param>2.0</wvf_param>
          <wvfIntPar>5</wvfIntPar>
          <no_smear_dir>3</no_smear_dir>
        </SmearingParam>
      </Source>
    </Param>
  - <NamedObject>
      <gauge_id>default_gauge_field</gauge_id>
      <source_id>sh_source_1</source_id>
    </NamedObject>
  </elem>
```

Generic name for wave function smearing

Wave function can be either of:
GAUSSIAN
EXPONENTIAL
GAUGE_INV_GAUSSIAN
WUPPERTAL
JACOBI

this corresponds to α and the number of iterations as discussed in the prev. lecture and also: 'Don't smear in the time direction' (which is 3 in chroma)

Make a propagator – from a point source

```
- <elem>
    <Name>PROPAGATOR</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>10</version>
      <quarkSpinType>FULL</quarkSpinType>
      <obsvP>false</obsvP>
      <numRetries>1</numRetries>
    - <FermionAction>
        <FermAct>CLOVER</FermAct>
        <Kappa>0.115</Kappa>
        <clovCoeff>1.17</clovCoeff>
      - <AnisoParam>
          <anisoP>false</anisoP>
        </AnisoParam>
      - <FermionBC>
          <FermBC>SIMPLE_FERMBC</FermBC>
          <boundary>1 1 1 -1</boundary>
        </FermionBC>
      </FermionAction>
    - <InvertParam>
        <invType>CG_INVERTER</invType>
        <RsdCG>1.0e-8</RsdCG>
        <MaxCG>1000</MaxCG>
      </InvertParam>
    </Param>
  - <NamedObject>
      <gauge_id>default_gauge_field</gauge_id>
      <source_id>sh_source_1</source_id>
      <prop_id>sh_prop_0</prop_id>
    </NamedObject>
  </elem>
```

hopping parameter, clover coefficient and anisotropy if desired

specify boundary conditions. Here periodic in space and anti-periodic in time. For twisted bcs try

```
- <FermionBC>
    <FermBC>TWISTED_FERMBC</FermBC>
    <boundary>1 1 1 -1</boundary>
    <phases_by_pi>0 0.15915494309189535 0</phases_by_pi>
    <phases_dir>0 1 2 </phases_dir>
  </FermionBC>
</FermionAction>
```

Inverter residual and maximum iteration count

Specify here on which source to invert

Sink smear the propagator

```
- <elem>
    <Name>SINK_SMEAR</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>5</version>
    - <Sink>
        <version>2</version>
        <SinkType>POINT_SINK</SinkType>
        <j_decay>3</j_decay>
      </Sink>
    </Param>
  - <NamedObject>
      <gauge_id>default_gauge_field</gauge_id>
      <prop_id>sh_prop_0</prop_id>
      <smeared_prop_id>sh_pt_sink_1</smeared_prop_id>
    </NamedObject>
  </elem>
```

Not much to say here – this applies the same smearing as for the source to the sink of a given propagator. In this example we stick to a simple point at the sink.

Creating 2pt-meson correlators

```
- <elem>
    <annotation> Compute the hadron spectrum. </annotation>
    <Name>HADRON_SPECTRUM</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>1</version>
      <MesonP>true</MesonP>
      <CurrentP>true</CurrentP>
      <BaryonP>true</BaryonP>
      <time_rev>false</time_rev>
      <mom2_max>3</mom2_max>
      <avg_equiv_mom>true</avg_equiv_mom>
    </Param>
  - <NamedObject>
      <gauge_id>default_gauge_field</gauge_id>
    - <sink_pairs>
      - <elem>
          <first_id>sh_pt_sink_1</first_id>
          <second_id>sh_pt_sink_1</second_id>
        </elem>
      </sink_pairs>
    </NamedObject>
    <xml_file>hadspec.dat.xml</xml_file>
  </elem>
```

MesonP: diagonal meson correlators
CurrentP: currents needed for O(a)-improvement
BaryonP: baryon 2pt functions
time_rev: use time-reversion symmetry
avg_equiv_mom: avg. over momentum channels

set maximum $\vec{p}^2$ to be induced

Here you can specify a whole series of propagator pairs on which to evaluate the correlators

Make a source – sequential source

```
- <elem>
    <annotation> a0-a0 seqsource </annotation>
    <Name>SEQSOURCE</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>2</version>
    - <SeqSource>
        <version>1</version>
        <SeqSourceType>a0-a0</SeqSourceType>
        <j_decay>3</j_decay>
        <t_sink>6</t_sink>
        <sink_mom>1 0 0</sink_mom>
      </SeqSource>
    </Param>
  - <PropSink>
      <version>5</version>
    - <Sink>
        <version>2</version>
        <SinkType>SHELL_SINK</SinkType>
        <j_decay>3</j_decay>
      - <SmearingParam>
          <wvf_kind>GAUGE_INV_GAUSSIAN</wvf_kind>
          <wvf_param>2.0</wvf_param>
          <wvfIntPar>5</wvfIntPar>
          <no_smear_dir>3</no_smear_dir>
        </SmearingParam>
      - <Displacement>
          <version>1</version>
          <DisplacementType>NONE</DisplacementType>
        </Displacement>
      - <LinkSmearing>
          <LinkSmearingType>APE_SMEAR</LinkSmearingType>
          <link_smear_fact>2.5</link_smear_fact>
          <link_smear_num>0</link_smear_num>
          <no_smear_dir>3</no_smear_dir>
        </LinkSmearing>
      </Sink>
    </PropSink>
  + <NamedObject></NamedObject>
  </elem>
```

will be expalined in a minute

specification of sequential source (time-direction is 3, the sink of the first prop is on time-slice 6 and the induced momentum is $\vec{p}=\frac{2\pi}{L}(1,0,0)$

In this example the sink of the initial propagator will be smeared with the iterated laplacian discussed in the previous lecture and the gauge links have been APE-smeared

a word on γ matrices in CHROMA

Each of the 16 γ-matrices can be represented as a product $\gamma_1^{n_1}\gamma_2^{n_2}\gamma_3^{n_3}\gamma_4^{n_4}$
The indices n_i are used to represent the numbers 0-15 by 4-bit numbers
The following table gives the gamma matrices corresponding to the integers 0-16 in chroma:

i	$n_4 n_3 n_2 n_1$	Γ_i	equivalent	sink insertion 3pt
0	0000	$\mathbb{1}$	$\mathbb{1}$	a0-a0
1	0001	γ_1	γ_1	a0-rho x 1
2	0010	γ_2	γ_2	a0-rho y 1
3	0011	$\gamma_1\gamma_2$	$\gamma_1\gamma_2$	a0-b1 z 1
4	0100	γ_3	γ_3	a0-rho z
5	0101	$\gamma_1\gamma_3$	$\gamma_1\gamma_3$	a0-b1 y 1
6	0110	$\gamma_2\gamma_3$	$\gamma_2\gamma_3$	a0-b1 x 1
7	0111	$\gamma_1\gamma_2\gamma_3$	$\gamma_5\gamma_4$	a0-pion 2
8	1000	γ_4	γ_4	a0-a0 2
9	1001	$\gamma_1\gamma_4$	$\gamma_1\gamma_4$	a0-rho x 2
10	1010	$\gamma_2\gamma_4$	$\gamma_2\gamma_4$	a0-rho y 2
11	1011	$\gamma_1\gamma_2\gamma_4$	$\gamma_3\gamma_5$	a0-a1 z 1
12	1100	$\gamma_3\gamma_4$	$\gamma_3\gamma_4$	a0-rho z 2
13	1101	$\gamma_1\gamma_3\gamma_4$	$\gamma_5\gamma_2$	a0-a1 y 1
14	1110	$\gamma_2\gamma_3\gamma_4$	$\gamma_1\gamma_5$	a0-a1 x 1
15	1111	γ_5	γ_5	a0-pion

XML-tag <SeqSourceType>
is one of these, so the insertion at the sink corresponds to these gamma matrices, left- and right-multiplied by gamma_5

Creating 3pt-meson correlators

```
- <elem>
    <annotation> BuildingBlock input file. </annotation>
    <Name>BUILDING_BLOCKS</Name>
    <Frequency>1</Frequency>
  - <Param>
      <version>5</version>
      <use_sink_offset>true</use_sink_offset>
      <mom2_max>3</mom2_max>
      <links_max>1</links_max>
      <canonical>false</canonical>
      <time_reverse>false</time_reverse>
      <translate>false</translate>
    - <FermState>
        <Name>SIMPLE_FERM_STATE</Name>
      - <FermionBC>
          <FermBC>SIMPLE_FERMBC</FermBC>
          <boundary>1 1 1 -1</boundary>
        </FermionBC>
      </FermState>
    </Param>
  - <BuildingBlocks>
      <OutFileName>./examplebb_v1.out</OutFileName>
      <GaugeId>default_gauge_field</GaugeId>
      <FrwdPropId>sh_prop_1</FrwdPropId>
    - <BkwdProps>
      - <elem>
          <BkwdPropId>sh_a0-a0_seqsource_prop_0</BkwdPropId>
          <BkwdPropG5Format>G5_B_G5</BkwdPropG5Format>
          <GammaInsertion>0</GammaInsertion>
          <Flavor>U</Flavor>
          <BBFileNamePattern>a0-a0_qz%c%1d_qy%c%1d_qx%c%1d.bb</BBFileNamePattern>
        </elem>
      </BkwdProps>
    </BuildingBlocks>
    <xml_file>bb.dat.xml</xml_file>
  </elem>
```

3pts are computed in “Building_Blocks”

Interesting bit here – how contraction is being carried out and which γ to insert at the external current insertion

The last bit

```
- <RNG>
  - <Seed>
      <elem>11</elem>
      <elem>11</elem>
      <elem>11</elem>
      <elem>0</elem>
    </Seed>
  </RNG>
- <Cfg>
    <cfg_type>UNIT</cfg_type>
    <cfg_file>test_purgaug.cfg1</cfg_file>
  </Cfg>
```

Seed for the random number generator

Specify the gauge configuration to be used here or use

- UNIT (free theory)
- WEAK_FIELD (close to free field, small fluct.)
- DISORDERED (random/hot)

Now it's your turn

- go to this http://www.hep.phys.soton.ac.uk/~juettner/lap08/index_v1.html URL and follow the steps explained here to study noise sources

If you can't get enough of CHROMA then: build on what you have learned yesterday in Carsten's first tutorial – 6.xml (http://people.physik.hu-berlin.de/~urbach/lap08/6.xml)

- continue where you stopped with Carsten's tutorial yesterday
- get yourself aquainted with the tests, in particular those in tests/chroma/hadron
- exercise - amend Carsten's input-XML (6.xml) starting from the example files in the chroma-tree:
  - create a sequential source from two sh_prop_1 propagators in order to compute the electromagnetic pion form factor
  - now generate a new quark sh_prop_th_1 which is the same as sh_prop_th_1 but with twisted boundary conditions (twist angle 0.5 into the x-direction)
  - contract the propagators to build a 3pt function using “building blocks”
  - run the program and start browsing the XML-output for the meson propagators and try to make sense of the produced data