README

> Luchang Jin

Contraction functions

See `Qlattice/docs/contraction.md` and `Qlattice/docs/file-format.md`.

Notation

Gamma matrix

$$\gamma_\mu^{\text{va}} = \begin{cases} \gamma_\mu & 0 \leqslant \mu < 4 \\ \gamma_\mu \gamma_5 & 4 \leqslant \mu < 8 \end{cases} \tag{1}$$

$$\Gamma_{a+2b+4c+8d} = \gamma_x^a \gamma_y^b \gamma_z^c \gamma_t^d \tag{2}$$

$$\gamma_5 = \Gamma_{15} = \gamma_x \gamma_y \gamma_z \gamma_t \tag{3}$$

$$\theta_\mu = \begin{cases} 1 & 0 \leqslant \mu < 4 \\ -1 & 4 \leqslant \mu < 8 \end{cases} \tag{4}$$

Propagator

Wall source propagator, with Coulomb gauge fixing:

$$S(\vec{x}, t_{\text{snk}}; t_{\text{src}}) = \sum_{\vec{y}} S(\vec{x}, t_{\text{snk}}; \vec{y}, t_{\text{src}}) \tag{5}$$

$$S(t_{\text{snk}}; \vec{y}, t_{\text{src}}) = \sum_{\vec{x}} S(\vec{x}, t_{\text{snk}}; \vec{y}, t_{\text{src}}) \tag{6}$$

$$S(t_{\text{snk}}; t_{\text{src}}) = \sum_{\vec{x}, \vec{y}} S(\vec{x}, t_{\text{snk}}; \vec{y}, t_{\text{src}}) \tag{7}$$

`compute-two-point-func.h`

```
ssprintf("analysis/lat-two-point/%s/results=%d", job_tag.c_str(), traj)
```

$$\mathrm{ld}[0 \leq t_{\mathrm{sep}} < T][0 \leq \mathrm{op}_{\mathrm{src}} < 16][0 \leq \mathrm{op}_{\mathrm{snk}} < 16] \tag{8}$$

```
ssprintf("/two-point-%d-%d.lat", type1, type2)
```

$$\mathrm{ld}[t_{\mathrm{sep}}][\mathrm{op}_{\mathrm{src}}][\mathrm{op}_{\mathrm{snk}}] = \mathrm{Tr}\Big(\big(\sum_{\vec{x}} S_1(\vec{x}, t_{\mathrm{snk}}; t_{\mathrm{src}})\Gamma_{\mathrm{op}_{\mathrm{src}}} \gamma_5 S_2(\vec{x}, t_{\mathrm{snk}}; t_{\mathrm{src}})^\dagger \gamma_5\big)\Gamma_{\mathrm{op}_{\mathrm{snk}}}\Big) \tag{9}$$

```
ssprintf("/two-point-wall-snk-sparse-corrected-%d-%d.lat", type1, type2)
```

$$\begin{aligned}
\mathrm{ld}[t_{\mathrm{sep}}][\mathrm{op}_{\mathrm{src}}][\mathrm{op}_{\mathrm{snk}}] &= \mathrm{Tr}\Big(\big(\sum_{\vec{x}} S_1(\vec{x}, t_{\mathrm{snk}}; t_{\mathrm{src}})\Gamma_{\mathrm{op}_{\mathrm{src}}} \sum_{\vec{y}} S_2(t_{\mathrm{src}}; \vec{y}, t_{\mathrm{snk}})\big)\Gamma_{\mathrm{op}_{\mathrm{snk}}}\Big) && (10)\\
&= \mathrm{Tr}\Big(\big(\sum_{\vec{x}} S_1(\vec{x}, t_{\mathrm{snk}}; t_{\mathrm{src}})\Gamma_{\mathrm{op}_{\mathrm{src}}} \gamma_5 \sum_{\vec{y}} S_2(\vec{y}, t_{\mathrm{snk}}; t_{\mathrm{src}})^\dagger \gamma_5\big)\Gamma_{\mathrm{op}_{\mathrm{snk}}}\Big) && (11)
\end{aligned}$$

`compute-three-point-func.h`

```
ssprintf("analysis/lat-three-point/%s/results=%d", job_tag.c_str(), traj)
```

$$\mathrm{ld}[0 \leq t_{\mathrm{sep}} < T][0 \leq t_{\mathrm{op}} < T][0 \leq \mathrm{op} < 16] \tag{12}$$

```
ssprintf("/three-point-%d-%d-%d.lat", type1, type2, type3)
```

$$\mathrm{ld}[t_{\mathrm{sep}}][t_{\mathrm{op}}][\mathrm{op}] = \mathrm{Tr}\sum_{\vec{x}}\Big(S_1(t_{\mathrm{op}}, \vec{x}; t_{\mathrm{src}})\gamma_5 S_3(t_{\mathrm{src}}; t_{\mathrm{snk}})\gamma_5\big(\gamma_5 S_2(t_{\mathrm{snk}}; t_{\mathrm{op}}, \vec{x})^\dagger \gamma_5\big)\Big)\Gamma_{\mathrm{op}} \tag{13}$$

`compute-psel-fsel-distribution.h`

```
ssprintf("analysis/field-psel-fsel-distribution/%s/results=%d", job_tag.c_str(),
traj)
```

Data format: `FieldM<Complex, 1>` with `write_field_double`.

```
ssprintf("/pos.field")
```

The expectation value is:

$$H(x - y) = 1 \tag{14}$$

The data is created by summing over all selected points for x and all point source locations for y, and then properly normalize the data.

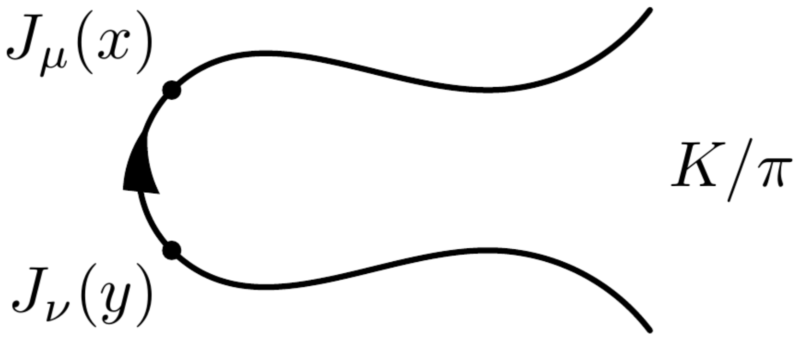
```
ssprintf("/neg.field")
```

Same as the `pos.field` but with x and y reversed.

```
ssprintf("/avg.field")
```

The average of the above two data sets.

`compute-meson-vv.h`



```
ssprintf("analysis/field-meson-vv/%s/results=%d", job_tag.c_str(), traj)
```

Data format: `FieldM<Complex, 8 * 8>` with `write_field_float_from_double`.

Use y as the point source location in calculation.

```
ssprintf("/decay-%d-%d-%d.field", type1, type2, type3)
```

$$H_{\text{decay-1-2-3}}(x-y)[8\mu+\nu] = \mathrm{Tr}[S_3(x;y)\gamma_\nu^{\text{va}} S_2(y;t_{\text{src}})\gamma_5 S_1(t_{\text{src}};x)\gamma_\mu^{\text{va}}] \tag{15}$$

where:

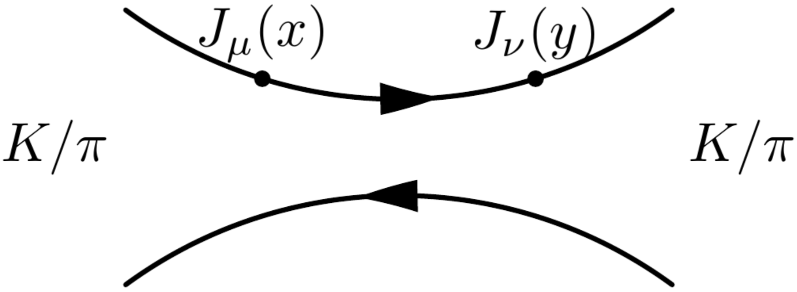
$$t_{\text{src}} = \min(x_t, y_t) - t_{\text{sep}} \tag{16}$$

$$t_{\text{snk}} = \max(x_t, y_t) + t_{\text{sep}} \tag{17}$$

and for t_{sep}:

```
inline int tsep_op_wall_src(const std::string& job_tag)
// parameter
{
  if (job_tag == "24D" or job_tag == "32D" or job_tag == "24DH") {
    return 8;
  } else if (job_tag == "32Dfine") {
    return 10;
  } else if (job_tag == "48I") {
    return 12;
  } else if (job_tag == "64I") {
    return 18;
  } else {
    qassert(false);
  }
  return 8;
}
```

`compute-meson-vv-meson.h`



```
ssprintf("analysis/field-meson-vv-meson/%s/results=%d", job_tag.c_str(), traj)
```

Data format: `FieldM<Complex, 8 * 8>` with `write_field_float_from_double`.

Use y as the point source location in calculation.

```
ssprintf("/forward-%d-%d-%d-%d.field", type1, type2, type3, type4)
```

$$H_{\text{forward}}(x-y)[8\mu+\nu] = \text{Tr}[S_1(t_{\text{snk}};x)\gamma_\mu^{\text{va}} S_4(x;y)\gamma_\nu^{\text{va}} S_2(y;t_{\text{src}})\gamma_5 S_3(t_{\text{src}};t_{\text{snk}})\gamma_5] \quad (18)$$

where:

$$t_{\text{src}} = \min(x_t, y_t) - t_{\text{sep}} \quad (19)$$
$$t_{\text{snk}} = \max(x_t, y_t) + t_{\text{sep}} \quad (20)$$

and for t_{sep}:

```
inline int tsep_op_wall_src(const std::string& job_tag)
// parameter
{
  if (job_tag == "24D" or job_tag == "32D" or job_tag == "24DH") {
    return 8;
  } else if (job_tag == "32Dfine") {
    return 10;
  } else if (job_tag == "48I") {
    return 12;
  } else if (job_tag == "64I") {
    return 18;
  } else {
    qassert(false);
  }
  return 8;
}
```

`compute-meson-snk-src.h`

```
ssprintf("analysis/lat-meson-snk-src/%s/results=%d", job_tag.c_str(), traj);
```

$$\text{ld}[0 \le t_{\text{snk}} < T][0 \le t_{\text{src}} < T] \quad (21)$$

```
ssprintf("/meson-snk-src-%d-%d.lat", type1, type2);
```

$$\text{ld-1-2}[t_{\text{snk}}][t_{\text{src}}] = \text{Tr}[S_1(t_{\text{snk}};t_{\text{src}})\gamma_5 S_2(t_{\text{src}};t_{\text{snk}})\gamma_5] \quad (22)$$

`compute-chvp.h`

```
ssprintf("analysis/field-chvp/%s/results=%d", job_tag.c_str(), traj);
```

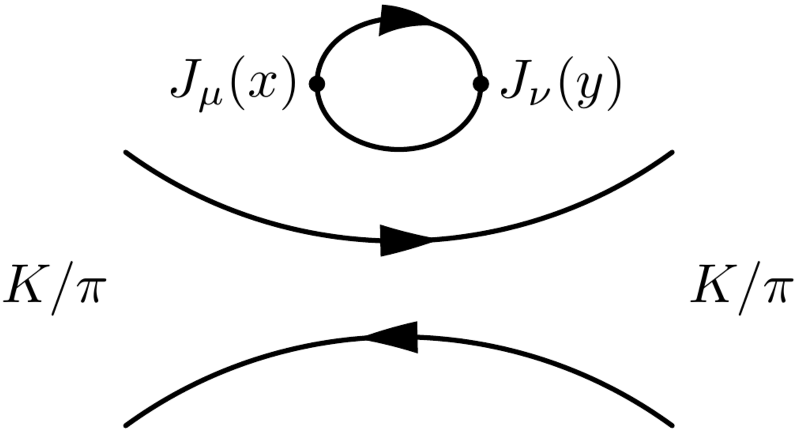
Data format: `FieldM<Complex, 8 * 8>` with `write_field_float_from_double`.

Use y as the point source location in calculation.

```
ssprintf("/chvp-%d-%d.field", type1, type2);
```

$$H_{\text{chvp-1-2}}(x-y)[8\mu+\nu] = \mathrm{Tr}[S_1(x;y)\gamma^{\text{va}}_{\nu} S_2(y;x)\gamma^{\text{va}}_{\mu}] \tag{23}$$

`compute-meson-chvp.h`



```
ssprintf("analysis/lat-meson-snk-src-shift-weight/%s/results=%d",
job_tag.c_str(), traj);
```

$$\mathrm{ld}[0 \leq t_{\text{snk}} < T][0 \leq t_{\text{src}} < T] \tag{24}$$

```
ssprintf("/meson-snk-src-%d-%d-%d-%d.field", type1, type2, type3, type4);
```

$$\text{ld-1-2}[t_{\text{snk}}][t_{\text{src}}] = \mathrm{Tr}[S_1(t_{\text{snk}};t_{\text{src}})\gamma_5 S_2(t_{\text{src}};t_{\text{snk}})\gamma_5] \tag{25}$$

Weighted properly for different time slice according to number of point source propagator available.

```
ssprintf("analysis/field-meson-chvp/%s/results=%d", job_tag.c_str(), traj);
```

Data format: `FieldM<Complex, 8 * 8>` with `write_field_float_from_double`.

Use y as the point source location in calculation.

```
ssprintf("/mchvp-%d-%d-%d-%d.field", type1, type2, type3, type4);
```

$$H_{\text{1-2-3-4}}(x-y)[8\mu+\nu] \mathrel{+}= \mathrm{Tr}[S_1(t_{\text{snk}};t_{\text{src}})\gamma_5 S_2(t_{\text{src}};t_{\text{snk}})\gamma_5]\mathrm{Tr}[S_3(x;y)\gamma^{\text{va}}_{\nu} S_4(y;x)\gamma^{\text{va}}_{\mu}] \tag{26}$$

where:

$$t_{\text{src}} = \min(x_t, y_t) - t_{\text{sep}} \tag{27}$$
$$t_{\text{snk}} = \max(x_t, y_t) + t_{\text{sep}} \tag{28}$$

and for t_{sep}:

```
inline int tsep_op_wall_src(const std::string& job_tag)
// parameter
{
  if (job_tag == "24D" or job_tag == "32D" or job_tag == "24DH") {
    return 8;
  } else if (job_tag == "32Dfine") {
    return 10;
  } else if (job_tag == "48I") {
    return 12;
  } else if (job_tag == "64I") {
    return 18;
  } else {
    qassert(false);
  }
  return 8;
}
```