

package	function
CheckArgs	checkList(@_);
GaugeFix	fixed_exist(\$conf, \$hyp_smear, \$gauge_type, \$save_cfg_head); print_body(\$conf, \$hyp_smear, \$gauge_type, \$save_cfg, \$save_cfg_head);
SourcePropagator	print_body_WallSource_CGProp(\$mom, \$t_src, \$gauge_id, \$quark_mass, \$clover, \$Residual, \$StoutLinkSmearing); print_body_PointSource_CGProp(\$ix, \$iy, \$iz, \$t_src, \$ns, \$nt, \$smear_size, \$smear_iter, \$smear_mom, \$gauge_id, \$quark_mass, \$clover, \$Residual, \$StoutLinkSmearing); print_body_PointSource_MGProp(\$ix, \$iy, \$iz, \$t_src, \$ns, \$nt, \$smear_size, \$smear_iter, \$smear_mom, \$gauge_id, \$quark_mass, \$clover, \$Residual, \$StoutLinkSmearing, \$mg_layout);
SequentialPropagator	print_body_NUCL_CGProp_PointSink(\$n_src, \$t_seq, \$mom_seq, \$bwd_quark_mass, \$prop_spec1_id, \$prop_spec2_id, \$gauge_id, \$nt, \$clover, \$Residual, \$StoutLinkSmearing); print_body_NUCL_CGProp_SmearedSink(\$n_src, \$t_seq, \$mom_seq, \$bwd_quark_mass, \$prop_spec1_id, \$prop_spec2_id, \$gauge_id, \$nt, \$clover, \$Residual, \$smear_size, \$smear_iter, \$smear_mom, \$StoutLinkSmearing); print_body_NUCL_MGProp_SmearedSink(\$n_src, \$t_seq, \$mom_seq, \$bwd_quark_mass, \$prop_spec1_id, \$prop_spec2_id, \$gauge_id, \$nt, \$clover, \$Residual, \$smear_size, \$smear_iter, \$smear_mom, \$StoutLinkSmearing, \$mg_layout); print_body_PION_CGProp_PointSink(\$n_src, \$t_seq, \$mom_seq, \$bwd_quark_mass, \$prop_spec_id, \$gauge_id, \$nt, \$clover, \$Residual, \$StoutLinkSmearing);
PropAdd	print_body_AddProps(\$mom, \$it0, \$it_cut, \$n_src, \$nt, \$direction);
SinkSmear	PointSink(\$prop_id, \$smear_prop_id, \$gauge_id); SmearedSink(\$prop_id, \$smear_prop_id, \$gauge_id, \$smear_size, \$smear_iter, \$smear_mom);
EraseObject	EraseVariable(\$nameobj); EraseList(@nameobj_list);
StapleWilsonLink	print_body(\$Lmax, \$bmax, \$bdir, \$gauge_id);
HadronSpectrum	print_baryon(\$conf, \$had_mom_x, \$had_mom_y, \$had_mom_z, \$had_key, \$gauge_id, \$curr_prop_id, \$spec1_prop_id, \$spec2_prop_id, \$prefix); print_meson(\$conf, \$had_mom_x, \$had_mom_y, \$had_mom_z, \$had_key, \$gauge_id, \$prop1_id, \$prop2_id, \$prefix);
Nonlocal2pt	print_body(\$prop_mom, \$anti_prop_mom, \$prop_id, \$anti_prop_id, \$gauge_id, \$staple_link_id, \$Lmax, \$bmax, \$Zmax, \$conf, \$it0, \$prefix);
Nonlocal3pt	print_body(\$gamma_id, \$mom_src, \$hadron_key, \$FrwdProp_id, \$BkwdProp_id, \$gauge_id, \$staple_link_id, \$Lmax, \$bmax, \$Zmax, \$bdir, \$conf, \$t_seq, \$prefix);
NonlocalQuarkME	print_body(\$gamma_id, \$mom_forward_prop, \$mom_backward_prop, \$forward_prop_id, \$backward_prop_id, \$gauge_id, \$staple_link_id, \$Lmax, \$bmax, \$Zmax, \$conf, \$t_seq, \$prefix);
EndOfPerl	print_body(\$ns, \$nt, \$conf, \$hyp_smear, \$gauge_type, \$save_cfg_head, \$conf_type, \$conf_path);

- **CheckArgs::checkList(@_):**

Purpose: Perl will not raise an Error when readin a parameter without assignment, just set it as Null.

This function will check whether the paramter list contains one without assignment or not.

If exist, this function will raise an error.

Example: Add the following sentence at the beginning of a new defined function:

```
CheckArgs::checkList(@_);
```

- **GaugeFix::fixed_exist(\$conf, \$hyp_smear, \$gauge_type, \$save_cfg_head):**

Params: \$hyp_smear: 0 (do hyp smear), 1 (not do hyp smear);

\$gauge_type: 0 (not do gauge fixing), -1 (Landau gauge), 3 (Coulomb gauge);

Return: (\$fixed_exist, \$conf_path_gfix) -> **list**

\$fixed_exist: saved gauge fixed configurations exist or not

\$conf_path_gfix: path of saved gauge fixed configurations

Example:

```
my($fixed_exist, $conf_path_gfix)=GaugeFix::fixed_exist($conf, $hyp_smear, $gauge_type, $save_cfg_head);
```

- **GaugeFix:: print_body(\$conf, \$hyp_smear, \$gauge_type, \$save_cfg, \$save_cfg_head):**

Params: \$hyp_smear: 0 (do hyp smear), 1 (not do hyp smear);

\$gauge_type: 0 (not do gauge fixing), -1 (Landau gauge), 3 (Coulomb gauge);

\$save_cfg: 0 (not save gauge fixed configurations), 1 (save);

Return: \$gfixed_id

Example:

```
$gauge_id=GaugeFix::print_body($conf, $hyp_smear, $gauge_type, $save_cfg, $save_cfg_head);
```

print the xml-text and return the \$gauge_id called by following sections.

- **SourcePropagator:: print_body_WallSource_CGProp(\$mom, \$t_src, \$gauge_id, \$quark_mass, \$clover, \$Residual, \$StoutLinkSmearing):**

Params: \$mom: momentum of the source;

\$t_src: source position t;

\$StoutLinkSmearing: 0 for not do stout link smearing, 1 for do;

Return: \$prop_id;

Example:

```
$prop_id= SourcePropagator:: print_body_WallSource_CGProp($mom, $t_src, $gauge_id, $quark_mass, $clover, $Residual);
```

print the xml-text and return the \$prop_id called by following sections.

- **SourcePropagator:: print_body_PointSource_CGProp(\$ix, \$iy, \$iz, \$t_src, \$ns, \$nt, \$smear_size, \$smear_iter, \$smear_mom, \$gauge_id, \$quark_mass, \$clover, \$Residual, \$StoutLinkSmearing):**

Params: (\$ix, \$iy, \$iz, \$t_src): source position;

\$smear_size, \$smear_iter, \$smear_mom: source smearing parameters;

Return: \$prop_id;

- **SequentialPropagator::print_body_NUCL_CGProp_PointSink(\$n_src, \$t_seq, \$mom_seq, \$bwd_quark_mass, \$prop_spec1_id, \$prop_spec2_id, \$gauge_id, \$nt, \$clover, \$Residual, \$StoutLinkSmearing):**

Purpose: make the sequential source of nucleon with type "NUCL_ISO_UNPOL_NONREL" (can change to others), read in 2 spectator propagators to make sequential source, and then make the sequential (backward) propagator, different with the "SourcePropagator" section, the \$n_src sequential propagators from different sources have been cutted and add together;

Params: \$mom_seq: momentum of the sequential source;
 \$bwd_quark_mass: quark mass of the sequential propagator;
 \$prop_spec1(2)_id: prop_id of 2 spectator propagators;

Return: id of the added (and no sink smear) sequential propagator;

- **SequentialPropagator::print_body_NUCL_CGProp_SmearedSink(\$n_src, \$t_seq, \$mom_seq, \$bwd_quark_mass, \$prop_spec1_id, \$prop_spec2_id, \$gauge_id, \$nt, \$clover, \$Residual, \$smear_size, \$smear_iter, \$smear_mom, \$StoutLinkSmearing):**

Purpose: Similar with the last one, but do the sink smearing for the sequential source.

Return: id of the added (and "MOM_GAUSSIAN" type smeared sink) sequential propagator;

- **SequentialPropagator:: print_body_PION_CGProp_PointSink(\$n_src, \$t_seq, \$mom_seq, \$bwd_quark_mass, \$prop_spec_id, \$gauge_id, \$nt, \$clover, \$Residual, \$StoutLinkSmearing):**

Purpose: make the sequential source of pion with type "MesonSeqSrc_G15", the difference with the nucleon one is only read in 1 spectator propagator;

- **PropAdd::print_body_AddProps (\$mom, \$it0, \$it_cut, \$n_src, \$nt, \$direction):**

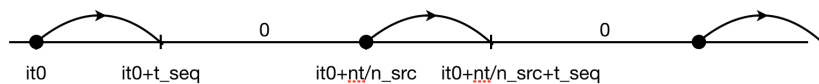
Purpose: cut the propagators from different sources and add them together. Note that only the propagators from "SourcePropagator" need to cut and add, the sequential ones have been already handled, no need this section;

Params: \$prop_head: momentum of the propagators;
 \$it_cut: the cut range of each propagator, if \$it_cut=-1, the range is from the source to next one;
 \$direction: the direction of the cutted propagator, should be "Forward" or "Backward";

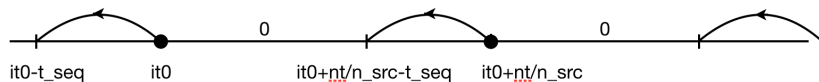
Return: the cutted and added propagator;

Example:

Forward:



Backward:



- **SinkSmear::PointSink(\$prop_id, \$smear_prop_id, \$gauge_id):**
Purpose: read in the propagator and gauge_id, do the sink smearing and return the smeared prop_id;S
- **EraseObject:: EraseVariable(\$nameobj):**
Purpose: erase the object \$nameobj;
- **EraseObject:: EraseList(@nameobj_list):**
Purpose: erase the list of objects @nameobj_list;
- **StapleWilsonLink:: print_body(\$Lmax, \$bmax, \$bdir, \$gauge_id):**
Purpose: make a series of staple shaped Wilson link with il*ib, this part will make a object of *multild<LatticeColorMatrix>* with shape=Lmax*bmax;
Params: see “Nonlocal3pt” section;
Return: id of the staple Wilson link;
- **HadronSpectrum:: print_baryon(\$conf, \$had_mom_x, \$had_mom_y, \$had_mom_z, \$had_key, \$gauge_id, \$curr_prop_id, \$spec1_prop_id, \$spec2_prop_id, \$prefix):**
Params: (\$had_mom_x, \$had_mom_y, \$had_mom_z): 3 momentum of the baryon;
\$had_key: 121050 -> for the $(u^T C \gamma_5 d)u$ type baryons, the first u and d correspond to \$spec1_prop_id and \$spec2_prop_id, the u outside corresponds to the \$curr_prop_id;
123050 -> for the $(u^T C \gamma_5 d)s$ type, also can used for $\Lambda_c(udc)$, $\Xi_c(usc)$ etc., which contains 3 different flavor quarks, by employing corresponding quark ids;
\$curr_prop_id: note that this quark correspond to the current quark in 3pt;
\$prefix: the save path of 2pt data;
- **HadronSpectrum:: print_meson(\$conf, \$had_mom_x, \$had_mom_y, \$had_mom_z, \$had_key, \$gauge_id, \$prop1_id, \$ prop2_id, \$prefix):**
Params: \$had_key: 100111515 -> pion, eta_s, eta_c (with different prop_id), the “11” means two quark have same flavor, and “1515” means the Dirac structures in 2pt are both Gamma(15);
100131515 -> Kaon; 100440404 -> j/\psi;
- **Nonlocal3pt:: print_body(\$gamma_id, \$mom_src, \$hadron_key, \$FrwdProp_id, \$BkwdProp_id, \$gauge_id, \$staple_link_id, \$Lmax, \$bmax, \$Zmax, \$bdir, \$conf, \$t_seq, \$prefix);**
Purpose: do the contraction of tmdpdf, with the nonlocal current like $O = \bar{\psi}(b)\Gamma U(L+b, L+z)\psi(z)$, shift the anti_prop b, and shift the prop z.
Params: \$gamma_id: 0~15 for each of 16 different Dirac structures, -1 for calculate all 16 structures;
\$mom_src: only used for the filename;
\$hadron_key: only used for the filename;
\$FrwdProp_id: id of forward propagator;
\$BkwdProp_id: id of backward propagator;
\$bdir: direction for the b shift, 1 for the forward direction, -1 for the backward direction, and 0 for calcualte both directions at same time (this has not yet);

◆ Test time:

Use 1 node with 4 gpu cards on 419, a12m310 (24*24*24*64) for test, do at least 3 times tests and take the average time.

- 4-strange quark propagators: (38.0s-29.5s-29.5s-29.5s)
- Shift U link 100 times (with U): 0.78s [U_1 = U[Linkz]*shift(U_1, FORWARD, Linkz);]
- Shift U link 100 times (without U): 0.65s [U_1 = shift(U_1, FORWARD, Linkz);]
- Shift propagator 100 times (with U): 12.6s [prop_1 = U[Linkz] * shift(prop_1, FORWARD, Linkz);]
- Shift propagator 100 times (without U): 11.3s [prop_1 = shift(prop_1, FORWARD, Linkz);]
- Trace 100 times: 17.2s [corr[i] = trace(adj(prop_2) * Gamma(8) * prop_1 * Gamma(15));]
- Writing data 100 times: 4.59s