Recent Progress

Senjie Zhu

2023.9.12

```cpp
#include "src/MFlow.cxx"

void TestMFlow() {
  // R__LOAD_LIBRARY(/data/Software/MPack/lib/libMF
  MFGF::ArrayFlowInit();

  MFGF::fArrayFlow[MFGV::kHadron].Fill(0.1,1.);
  MFGF::fArrayFlow[MFGV::kHadron].Print();
  MFGF::ArrayFlowReset(MFGV::kHadron);
  MFGF::fArrayFlow[MFGV::kHadron].Print();

  // MFGF::ArrayFlowFill(MFGV::kHadron, 0.1, 1.);
  // MFGF::ArrayFlowPrint(MFGV::kHadron);
  // MFGF::ArrayFlowReset(MFGV::kHadron);
  // MFGF::ArrayFlowPrint(MFGV::kHadron);
}
```

```
Processing TestMFlow.cpp...
MFlow::Print() :0x7f90405480c0
fQvectors.size() = 1
fQvectors[0].size() = 1
Q(0,0) = (1,0i)
MFlow::ReSet() :0x7f90405480c0
MFlow::Print() :0x7f90405480c0
fQvectors.size() = 1
fQvectors[0].size() = 1
Q(0,0) = (0,0i)
```

```cpp
void MFlowGlobaFunctions::ArrayFlowInit()
{
  for (int i = 0; i < MFGV::kNTypes; i++) {
    fArrayFlow[i] = MFlow();
  }
}

void MFlowGlobaFunctions::ArrayFlowReset(int kType)
{
  if (kType == MFGV::kNTypes) {
    for (int i = 0; i < MFGV::kNTypes; i++) {
      fArrayFlow[i].ReSet();
    }
  } else if (kType == MFGV::kJpsiRelevant) {
    for (int i = MFGV::kHadron + 1; i < MFGV::kNTypes; i++) {
      fArrayFlow[i].ReSet();
    }
  } else {
    fArrayFlow[kType].ReSet();
  }
}
```

```cpp
#include "MFlow.h"

void TestMFlow() {
  R__LOAD_LIBRARY(/data/Software/MPack/lib/libMFlow.so);
  MFGF::ArrayFlowInit();

  MFGF::fArrayFlow[MFGV::kHadron].Fill(0.1,1.);
  MFGF::fArrayFlow[MFGV::kHadron].Print();
  MFGF::ArrayFlowReset(MFGV::kHadron);
  MFGF::fArrayFlow[MFGV::kHadron].Print();

  // MFGF::ArrayFlowFill(MFGV::kHadron, 0.1, 1.);
  // MFGF::ArrayFlowPrint(MFGV::kHadron);
  // MFGF::ArrayFlowReset(MFGV::kHadron);
  // MFGF::ArrayFlowPrint(MFGV::kHadron);
}
```

```
sjzhu.test$ root -l TestMFlow.cpp
root [0]
Processing TestMFlow.cpp...
MFlow::Print() :0x7fd5e7fb80c0
fQvectors.size() = 1
fQvectors[0].size() = 1
Q(0,0) = (1,0i)
MFlow::ReSet() :0x7fd5d732a840
MFlow::Print() :0x7fd5e7fb80c0
fQvectors.size() = 1
fQvectors[0].size() = 1
Q(0,0) = (1,0i)
```

```cpp
#include "MFlow.h"

void TestMFlow() {
  R__LOAD_LIBRARY(/data/Software/MPack/lib/libMFlow.so);
  MFGF::ArrayFlowInit();

  // MFGF::fArrayFlow[MFGV::kHadron].Fill(0.1,1.);
  // MFGF::fArrayFlow[MFGV::kHadron].Print();
  // MFGF::ArrayFlowReset(MFGV::kHadron);
  // MFGF::fArrayFlow[MFGV::kHadron].Print();

  MFGF::ArrayFlowFill(MFGV::kHadron, 0.1, 1.);
  MFGF::ArrayFlowPrint(MFGV::kHadron);
  MFGF::ArrayFlowReset(MFGV::kHadron);
  MFGF::ArrayFlowPrint(MFGV::kHadron);
}
```

```
sjzhu:Test$ root -l TestMFlow.cpp
root [0]
Processing TestMFlow.cpp...
MFlow::Print() :0x7f2d150f9840
fQvectors.size() = 1
fQvectors[0].size() = 1
Q(0,0) = (1,0i)
MFlow::ReSet() :0x7f2d150f9840
MFlow::Print() :0x7f2d150f9840
fQvectors.size() = 1
fQvectors[0].size() = 1
Q(0,0) = (0,0i)
```

```cpp
void MFlowGlobaFunctions::ArrayFlowInit()
{
  for (int i = 0; i < MFGV::kNTypes; i++) {
    fArrayFlow[i] = MFlow();
  }
}

void MFlowGlobaFunctions::ArrayFlowReset(int kType)
{
  if (kType == MFGV::kNTypes) {
    for (int i = 0; i < MFGV::kNTypes; i++) {
      fArrayFlow[i].ReSet();
    }
  } else if (kType == MFGV::kJpsiRelevant) {
    for (int i = MFGV::kHadron + 1; i < MFGV::kNTypes; i++) {
      fArrayFlow[i].ReSet();
    }
  } else {
    fArrayFlow[kType].ReSet();
  }
}

void MFlowGlobaFunctions::ArrayFlowFill(int kType, double phi,
double weight)
{
  fArrayFlow[kType].Fill(phi, weight);
}
```